Using the E21 Workbench Setup Screen

The E21 Workbench Setup Screen is where users actually design their individual workbench programs. Here we will discuss in detail the steps involved in creating an E21 Workbench, and also each function available within the setup screen. The following information assumes that all SQL Datasets necessary for the workbench have already been created.

For more detailed information on building Workbench SQL Datasets, see Using The E21 Workbench SQL Builder.

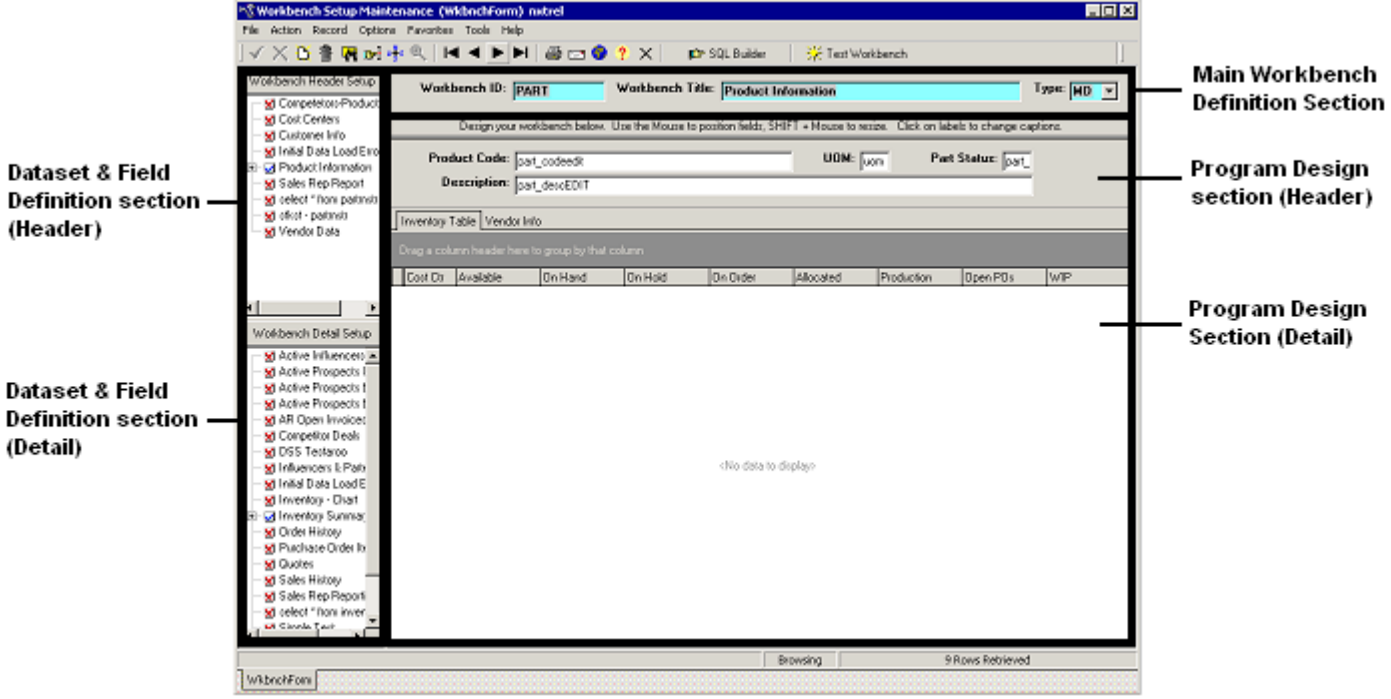
Understanding the layout and terminology

When you first enter the E21 Workbench Setup Screen, you will notice that although it generally looks like most other E21 programs, there are a few major differences. The Workbench Setup Screen is divided into three distinct sections.

1) The Main Workbench Definition section - This where you will define the Workbench ID, the Workbench Title, and the type of Workbench you are creating.

2) The Dataset & Field Definition section (Header & Detail) - This is where you will choose which SQL queries will be used to create the dataset(s) for both the header and (when applicable) detail section(s) of your workbench, and also which fields will be visible on your workbench form. It is also within this section that you will have the ability to alter the properties for each of the fields you add to your workbench.

3) The Workbench Program Designer (Header & Detail) - This is where you design your actual workbench form (IE, what the actual workbench form will look like at runtime). When you add fields to the workbench, objects will be drawn here. These fields will display as Labels & Edit Boxes in the header, and grid columns or EntDSS dimensions & measures in the detail.



Creating a new workbench

Each workbench screen will have a unique Workbench ID, along with a Title and Type. To create a new E21 Workbench, enter the Workbench Setup Screen and click the Add button from the standard E21 toolbar. This will clear all 3 sections of the workbench and put you in insert mode. While you are in insert mode, the Field Definition & Program Design sections will be hidden and only the "Main Workbench Definition Section" will be enabled to avoid confusion.

Enter the following three pieces of information:

Workbench ID: This is the unique identifier of your workbench. It can be up to 10 alphanumeric characters long. This code will be used internally by the workbenches, but it will also be the "Program Name" used by the Enterprise21 menu and security processes. For this reason it must not only be unique to other Workbench ID's, but it should also be unique from any existing program names in E21.

Workbench Title: This is the title that will be displayed when you run your workbench.

Type: This is the type of workbench you are creating. Valid entries include:

- HO – Header Only
- MD – Master Detail
- DO – Detail Only (Not available in older releases)

Once you have entered these 3 pieces of information, save your workbench by clicking the accept button in the standard E21 toolbar. All 3 section of the workbench for will be redisplayed upon saving your new workbench. You are now ready to add your workbench dataset information by using the Workbench Dataset & Field Definition section.

Note: Not all setup information below is relevant for all workbench types.

Header Section Setup

Choosing your Header SQL Dataset and fields

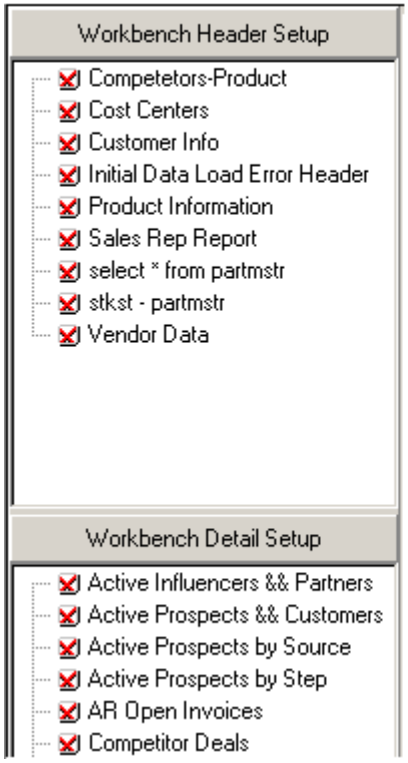
Once you have created your initial workbench information and saved it, you will see a list of every applicable SQL dataset available to you. SQL Datasets are created in the Workbench SQL Query Builder, and which datasets you have access to will be dependant on which type of workbench you are creating. Datasets that were identified as "HG" will be displayed in the Workbench Header Setup section. Datasets that were identified as either "DC" or "DG" will be displayed in the Workbench Detail Setup section.

For more detailed information on building Workbench SQL Datasets, see Using The E21 Workbench SQL Builder.

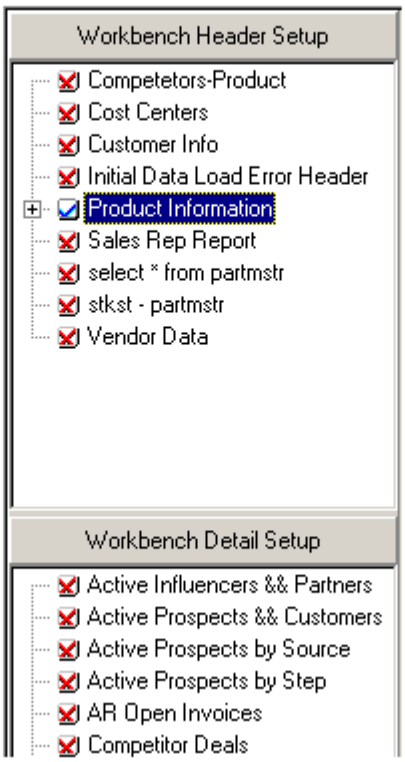
Each workbench can have only one Header dataset, but can have an unlimited number of Detail datasets. You should always complete the header portion of your workbench (when applicable) before beginning the detail section.

In the Workbench Header Setup section, identify the main dataset that you wish to use for this workbench and click the "red X" checkbox next to its name. It will switch to a blue checkmark with a + (plus) sign letting you know that it is now the active header dataset and that there are fields associated with that dataset you may now choose from. Click the plus sign to expand the treeview to see those fields. Once you see the fields available to you, click on the red X's next to each of those fields that you wish to place on your workbench. Once again, the Red X's will turn into blue checkmarks letting you know which fields have been added.

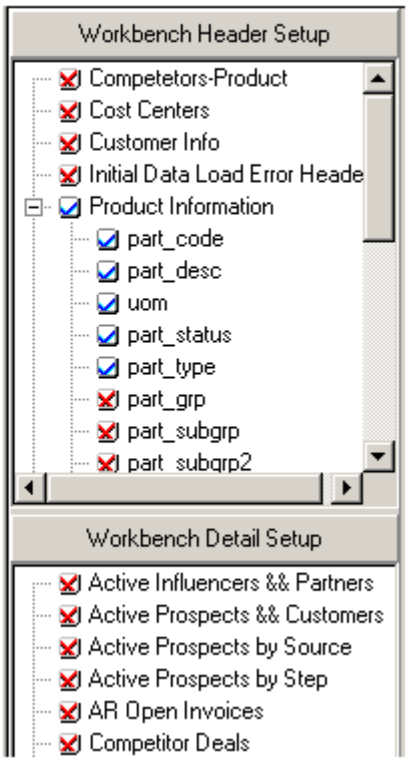
Step 1: Browse your valid Datasets



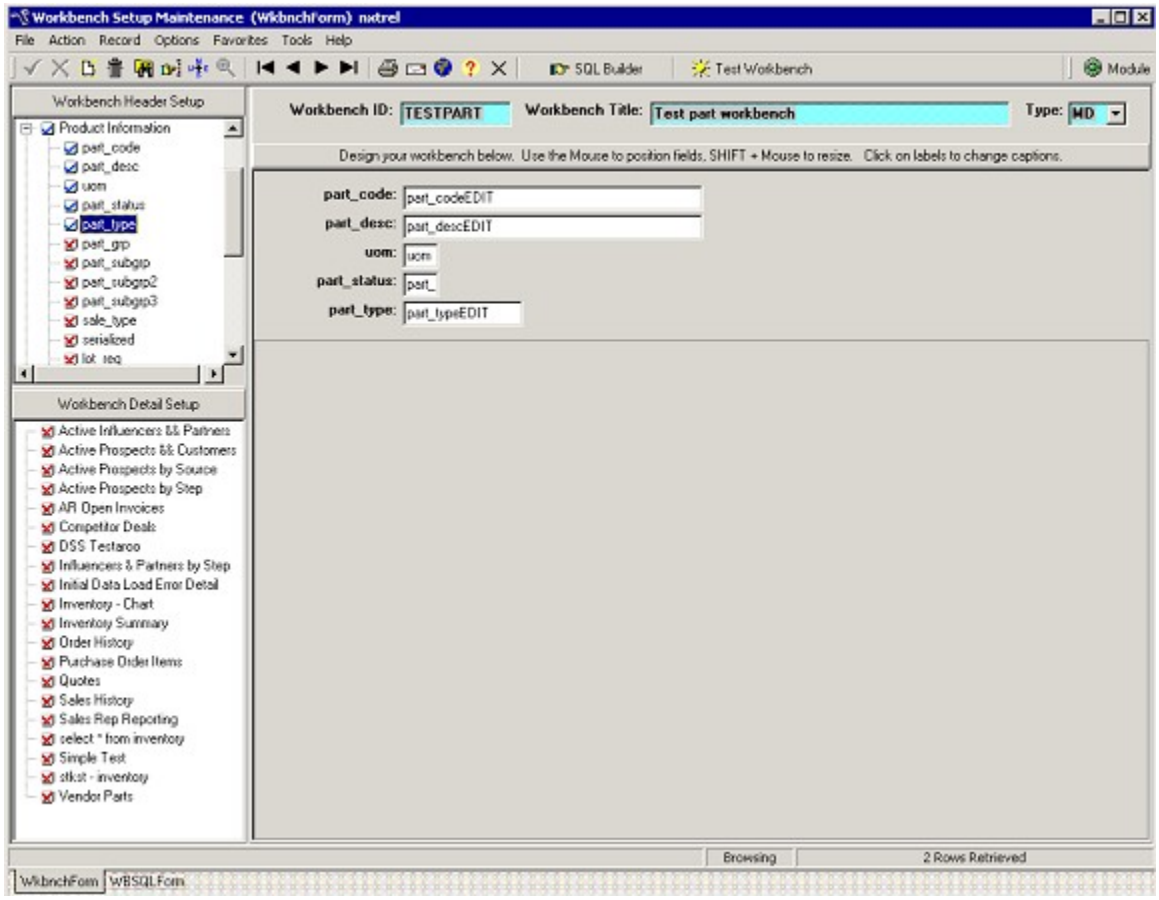
Step 2: Select the appropriate dataset for your workbench



Step 3: Select the fields you wish to display on your workbench



All header fields with blue checkmarks will have components placed into the header section of the Workbench Program Designer. (See Image below)

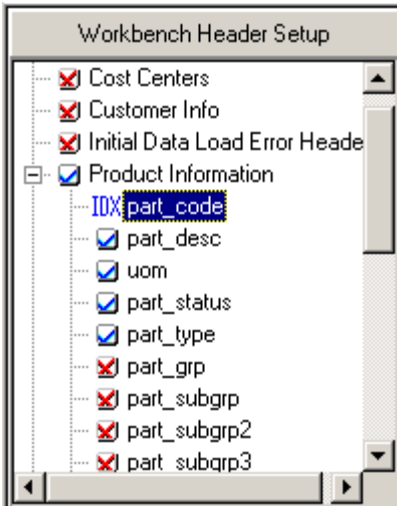


Identifying the join field(s) of the Header Dataset (Master-Detail Only)

When creating Master Detail workbenches (Workbench Type=MD) you must identify which field or fields from the datasets will join the header and detail records. For example, if you were creating a product workbench, the field which joined the header to the details records would typically be your product code. If you were creating an orders workbench, the fields that would join your header & detail records would be the order number & release number, etc.

To set the header join index, identify the field you wish to set as the index and make sure that it is turned on (Blue checkmark). Right-click on the field to open its popup menu. Choose the option "Mark Selected Field As Index." The blue checkmark icon next to the field should change to a blue "IDX" icon (See image below). This tells you that the field has been successfully identified as a header index field. Repeat this step for each field that makes up the index.

Note: Header index fields must be active (blue check mark) on the header portion of the form to be selected as an index. This differs from the detail sections that typically will NOT have the index fields displaying on the form.



Arranging the header fields and labels in the Workbench Program Design Section

Once you have added all of your fields to the workbench header section, you will likely want to arrange the components on your form so that they appear in the locations & format you want. This might include moving fields around on the form, changing the tab order of your fields, resizing some fields, and changing the display labels.

At the top of the Workbench Program Designer you will notice a message with the instructions on how to move / resize fields:

Repositioning Header Fields
To move Edit Boxes & Labels around the header portion of the workbench program designer, click on the corresponding Edit Box of the field and while holding the mouse button down, drag it to the desired location on the screen. When you have the field in the correct position, release the mouse. The corresponding label will be moved as well.

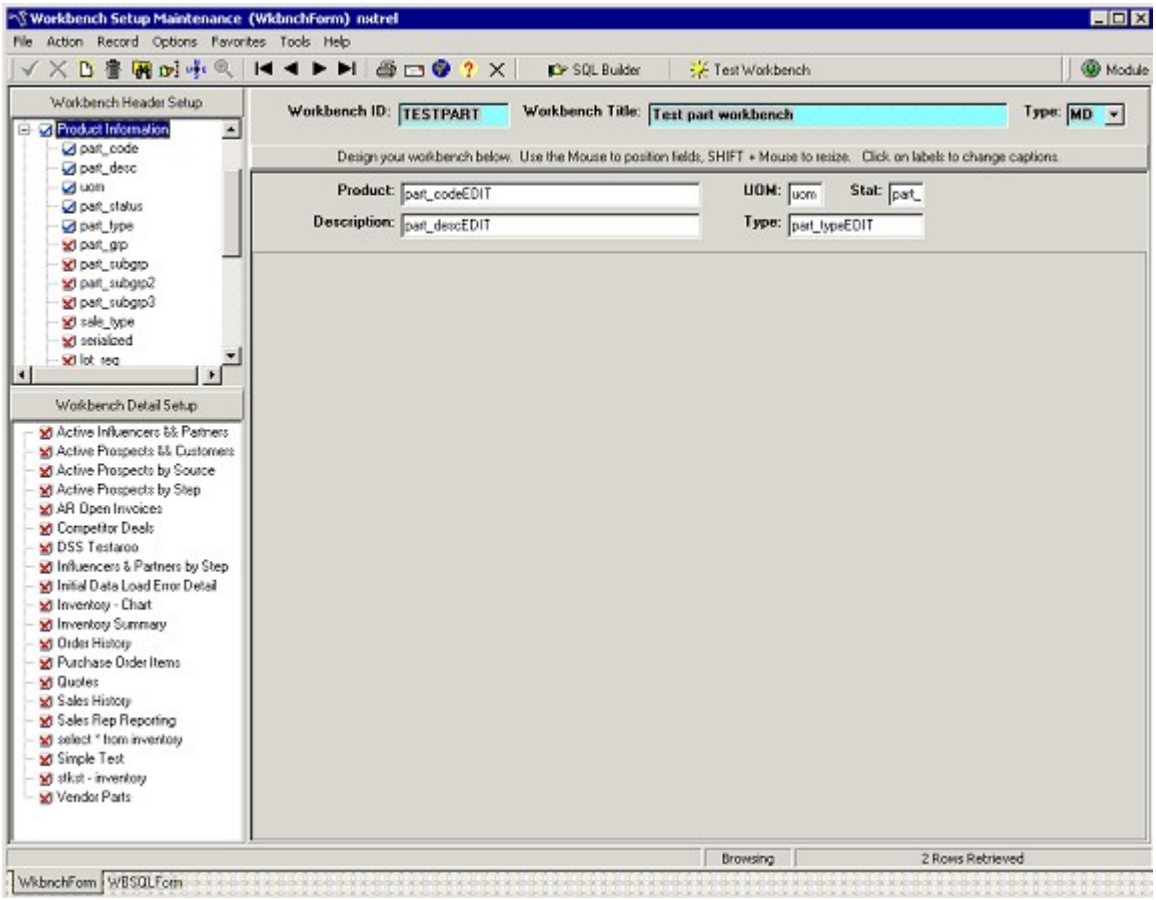
Resizing Header Fields
When you first add a field to the workbench, the default Edit Box width will be directly related to the field definition from the dB. In other words, fields that are defined as char(3) in the dB will have smaller Edit Boxes created than fields that are defined as char(30). If you wish to change the default Edit Box width, hold the SHIFT key on your keyboard while at the same time clicking on the Edit Box. Now, while holding both the shift key and mouse button down, drag the mouse to your left or right. The edit box should resize larger or smaller accordingly. Release the mouse when the Edit Box is the desired width.

Changing Label Captions
When you first add fields to the header portion of the Workbench Program Designer, the label caption will default to the associated field name. To change this, simply click on the label. The label will switch to edit mode where you can type the new caption for your label. After changing a caption, you MUST exit the label captions edit (either by clicking someplace else or hitting the tab key), otherwise your changes will not be saved.

Note: Header label objects are created with object names [field_name]LBL. If this object is specified in the E21 Multi Language Screen Layout Maintenance program, any labels associated with the workbench will be overridden by the multi-language settings at runtime.

Changing the Header Field Tab Order

See "Arranging / fine tuning the header fields, label captions & tab order by altering the raw data." (Not available in older releases)



Arranging / fine tuning the header fields, label captions & tab order by altering the raw data

In addition to using your mouse to drag and drop fields around the form, you may also access the field position information as raw data. This is especially useful if you are trying to line up fields to the exact Left & Top locations, and also is the only method for altering the tab order of the header fields (The tab order defaults to the sequence in which the header fields were added to the workbench). Altering the raw data may also be a slightly faster method of organizing the forms for experienced users.

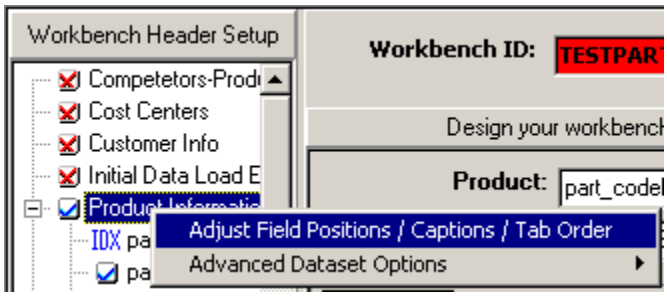
Note – Only unlocked workbenches can be modified, so this option will be disabled if the record is locked.

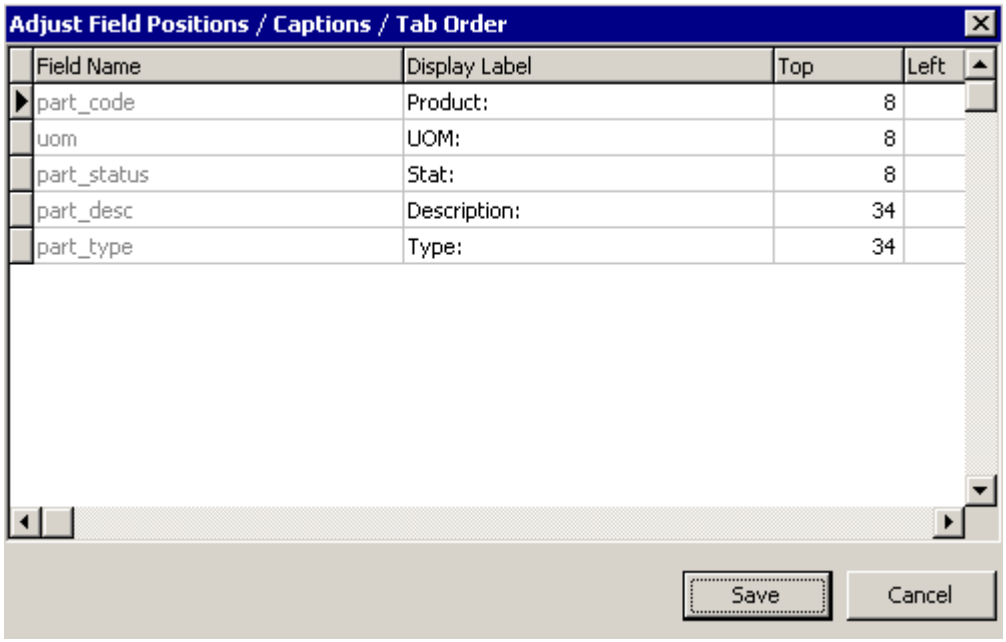
To access the field position data directly:

1) From within the Workbench Header Setup treeview, right click on either the appropriate header SQL Dataset, or any of the selected fields on the screen to open the selected item's popup menu.
2) Choose Adjust Field Positions / Captions / Tab Order

3) Modify the raw data (Display Label, Top, Left, or Width, or Tab Order) as desired. (Altering the Tab Order is not available in older releases)
4) Click Save or Cancel.

Your fields will redisplay using the modified raw data.

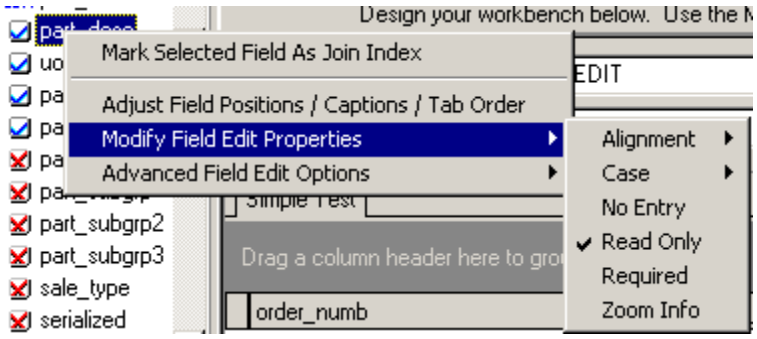




Modifying the runtime Properties of the Header Field Edits

Each selected field from the header dataset has a corresponding Edit Box on the Workbench Program Designer. Each of these Edit Boxes have associated properties (Alignment, Case, etc) which tell the workbench program how they should perform at runtime. To modify these properties:

1) Right Click on the specific field you wish to modify to open the selected item's popup menu.
2) Choose "Modify Field Edit Properties." If this menu option is not enabled, you do not have a valid field selected or the workbench is locked.



From the Modify Column Properties menu option, you may choose from the following 6 selections:

Alignment
This property determines if the data in the Edit Box will be right, left or center justified. The default is that all text fields will left-justify and all numeric fields will right-justify.

Case
This property determines if the data in the Edit Box will be formatted as upper case, lower case, or mixed case. The default is that all text will display in mixed case, but that the workbench screens will use the system E21 case settings. Setting this property to upper or lower will override the E21 system settings.

Noentry
This property determines if the data in the Edit Box will be enabled in query mode. All fields not on the "Main Table" of the Workbench SQL Dataset will automatically be set to Noentry at runtime.

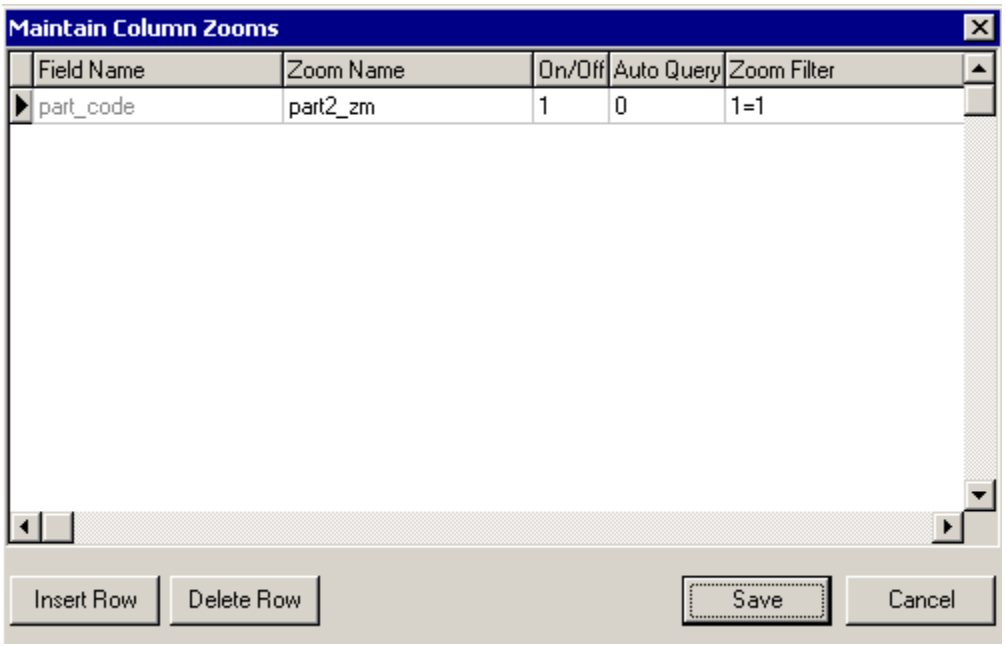
Note: The "Main Table" of an SQL is established in the Workbench SQL Builder. For more detailed information on building Workbench SQL Datasets, see Using The E21 Workbench SQL Builder.

Readonly
This property determines if the data in the Edit Box will be set to ReadOnly in Insert and Edit modes (Workbench Table Maintenance screens). Only TGI system developers can create Workbench Table Maintenance screens. All workbenches created by customers will be Query Only.

Required
This property determines if the data in the Edit Box will be set to Required in Insert and Edit modes (Workbench Table Maintenance screens). Only TGI system developers can create Workbench Table Maintenance screens. All workbenches created by customers will be Query Only.

Zoom Info
Choosing this option will open a "Maintain Column Zooms" window which will allow users to add E21 style zooms to individual workbench fields. The zooms should already be created in the E21 Zoom Maintenance screen. To add a new zoom, click the Insert Row button (there can only be one zoom per field). For each zoom, you must fill in the Zoom Name, set the On/Off value (1=On, 0=Off), choose whether you want the Zoom to auto Query (1=Auto Query, 0=DO Not Auto Query), and finally, enter a Zoom Filter if so desired. Click Save or Cancel when you are done to return to the Workbench Setup Screen.

Note: You can only activate zooms when a field is editable. Therefore, on workbenches, you will only have access to zooms when you are in QBE mode, or on Workbench Table Maintenance screens.



Detail Section Setup

Choosing your Detail SQL Dataset and fields

The process for choosing detail dataset(s) and fields to be placed on the workbench grids is very similar to the process for adding datasets and fields to the header section. As such, this section of the documentation will focus on the detail specific functionality of the workbench setup screen. For an overview of the basics of adding datasets and fields to the workbenches, please review "Choosing your Header SQL Dataset and fields" before continuing.

Once you have created your initial workbench information and saved it, you will see a list of all detail SQL dataset available to you in the Workbench Detail Setup section. SQL Datasets are created in the Workbench SQL Query Builder, and which datasets you have access to will be dependant on which type of workbench you are creating. Datasets that were identified as "HG" will be displayed in the Workbench Header Setup section. Datasets that were identified as either "DC" or "DG" will be displayed in the Workbench Detail Setup section.

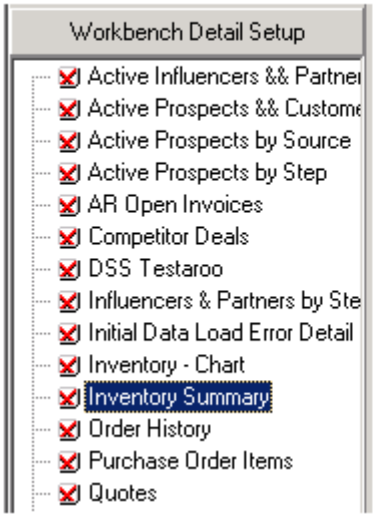
For more detailed information on building Workbench SQL Datasets, see Using The E21 Workbench SQL Builder.

Each workbench can have only one Header dataset, but can have an unlimited number of Detail datasets. For Master-Detail type workbenches, you should always complete the header portion of your workbench (including assigning the join fields of the header dataset) before beginning the detail section.

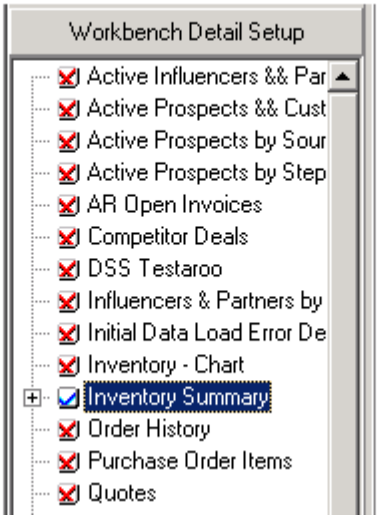
In the Workbench Detail Setup section, identify a dataset that you wish to use for this workbench and click the "red X" checkbox next to its name. It will switch to a blue checkmark with a + (plus) sign letting you know that it is now an active detail dataset. Additionally, you will also see a new detail grid tab appear on the detail portion of your workbench form. Now click the plus sign of the dataset to expand the treeview to all available fields. Once you see the fields available to you, click on the red X's next to each of those fields that you wish to place on your workbench. Once again, the Red X's will turn into blue checkmarks letting you know which fields have been added.

Note: On Master-Detail type workbenches, the detail datasets you choose MUST have corresponding fields to join the detail dataset to the header (detail-only workbenches have no restrictions). For more information on joining the datasets on Master-Detail workbenches, see Identifying the join field(s) of the Header Dataset (Master-Detail Only) and Identifying the join field(s) of the Detail Dataset (Master-Detail Only).

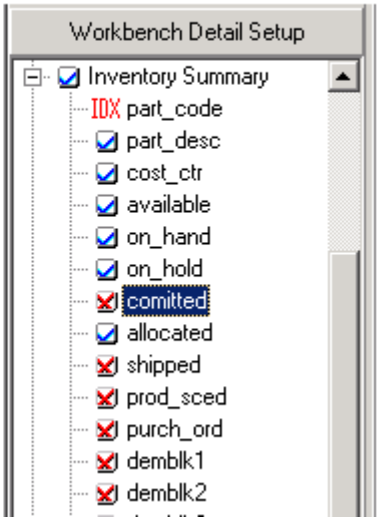
Step 1: Browse your valid Datasets



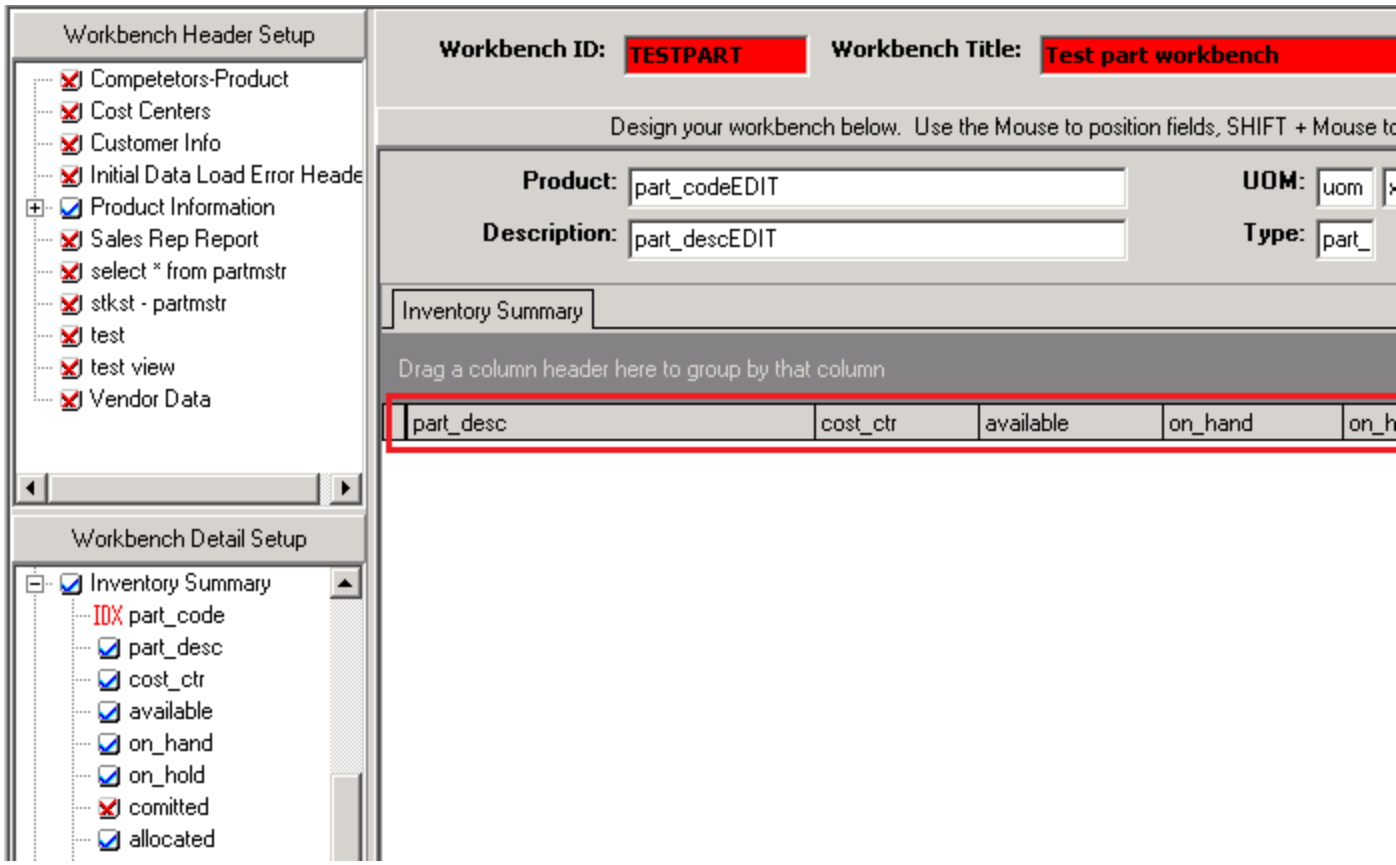
Step 2: Select the appropriate dataset for your workbench



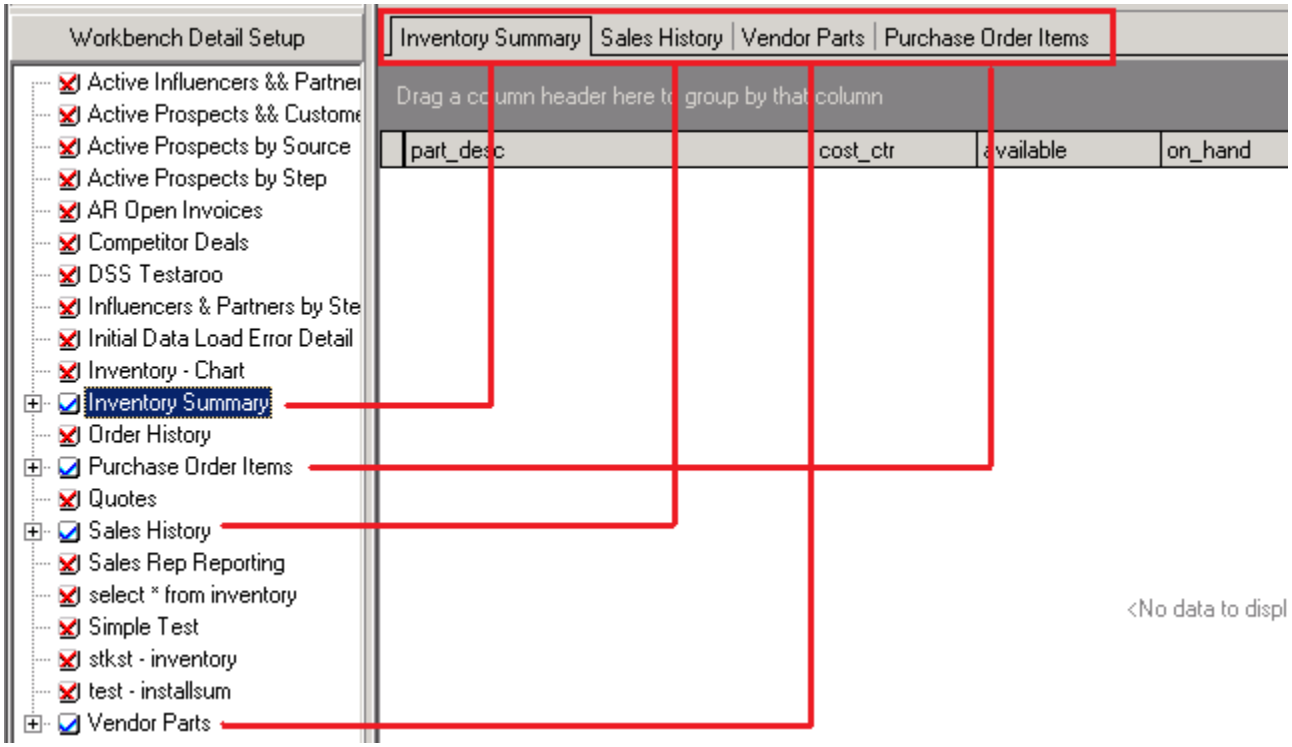
Step 3: Select the fields you wish to display on your workbench



All detail fields with blue checkmarks will have columns placed into the appropriate tab of the Workbench Program Designer. (See Image below)



Furthermore, a separate tab will be created for each detail dataset you turn on. You may have an unlimited number of tabs on the detail section of your workbench.



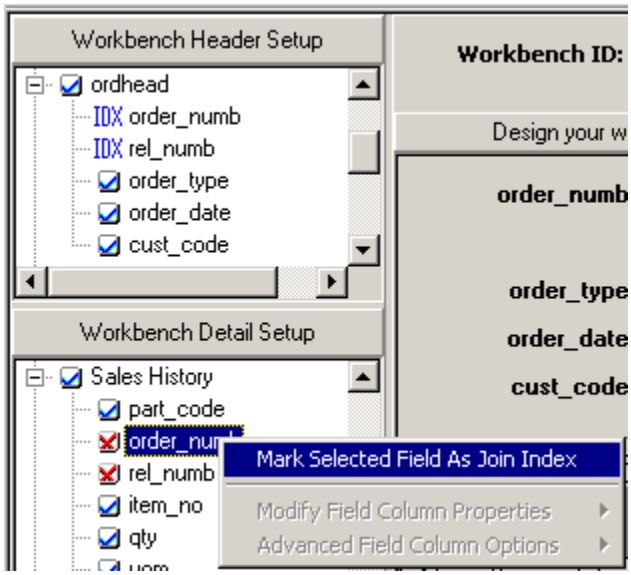
Identifying the join field(s) of the Detail Datasets (Master-Detail Only)

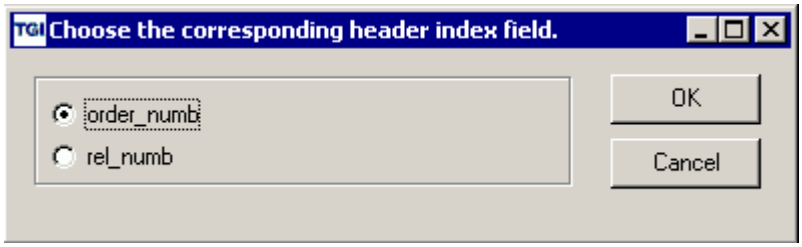
When creating Master Detail workbenches (Workbench Type=MD) you must identify which field or fields from the datasets will join the header and detail records. For example, if you were creating a product workbench, the field which joined the header to the details records would typically be your product code. If you were creating an orders workbench, the fields that would join your header & detail records would be the order number & release number, etc.

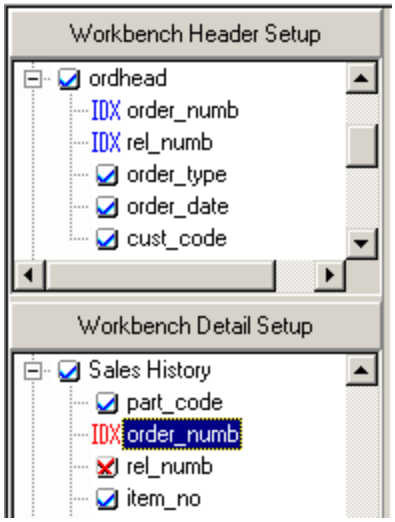
Note – You MUST assign the header join index field(s) before you will be able to assign the detail join index fields. For more information on assigning the header join field(s), see "Identifying the join field(s) of the Header Dataset."

To set the detail join index field(s), identify the field(s) you wish to set as the index and make sure that they are turned OFF (red X). This differs from the header index fields which you always want turned on.

Right-click on the field to open its popup menu. Choose the option "Mark Selected Field as Join Index." A dialog window will open asking you to "Choose the corresponding header Index Field." There will be a separate radio button for each field you have assigned as an index from the header dataset. Select the appropriate field and click OK. The Red X icon next to the field should change to a red "IDX" icon (See image below). This tells you that the field has been successfully identified as a detail index field. Repeat this step for each field that makes up the index.





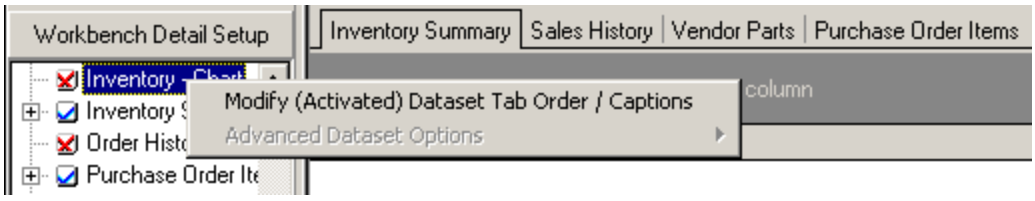


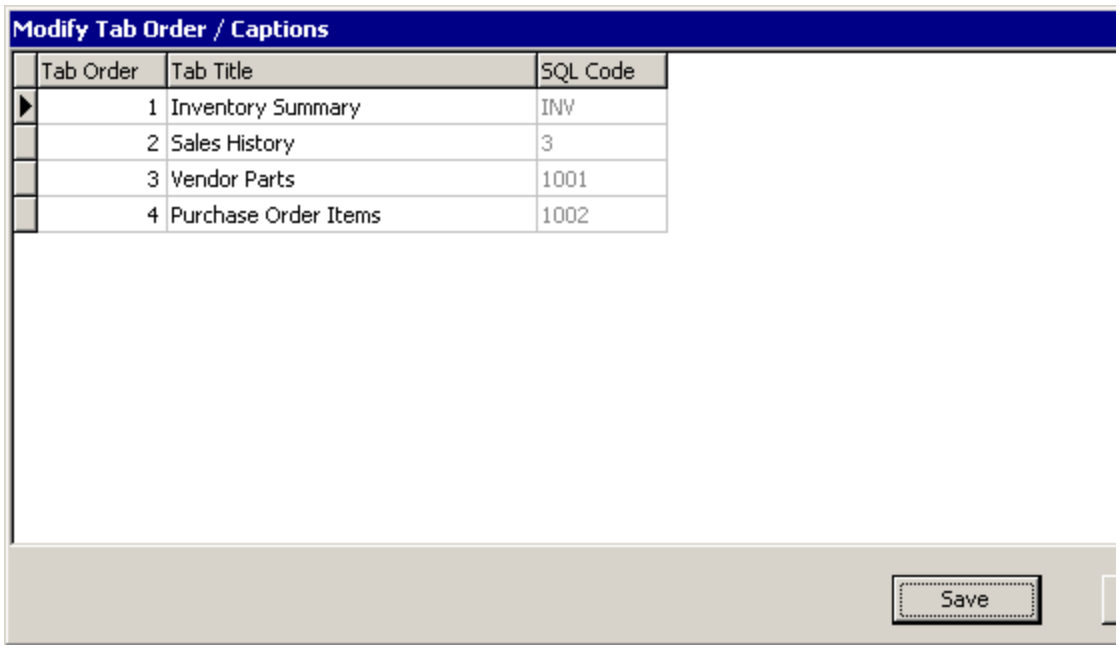
Arranging the detail tab order and changing their default captions

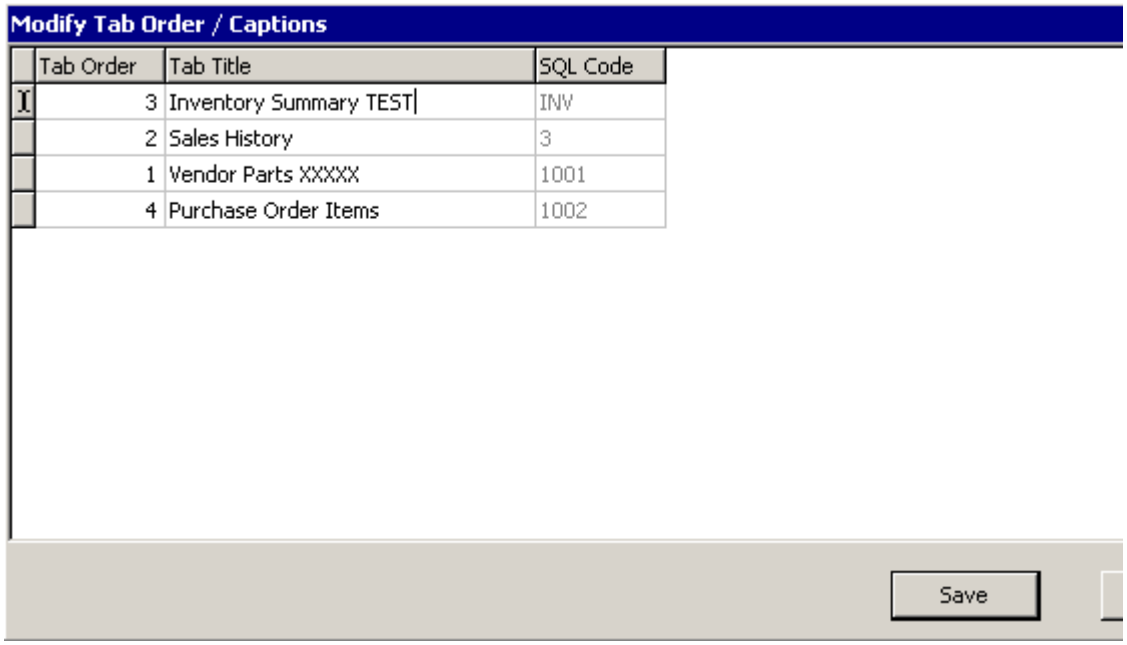
When you add detail dataset(s) to your workbench, the detail tabs added to the workbench grid will be created in the order they are added, and they will display the SQL Dataset description as the tab caption. You have the ability to override these default settings.

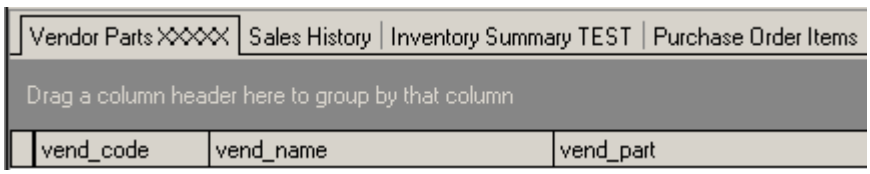
1) Right-click ANY dataset from the Workbench Detail Setup to open the Dataset popup menus.
2) Choose the “Modify (Activated) Dataset Tab Order / Caption” menu option.
3) The “Modify Tab Order / Captions” dialog will appear.
4) Change the “Tab Order” or Tab Title” fields as desired.
5) Click the Save button to save your changes.

Your detail grid will now reflect your changes.




Modify Tab Order / Captions
Tab Order
Tab Title
SQL Code
1
Inventory Summary
INV
2
Sales History
3
3
Vendor Parts
1001
4
Purchase Order Items
1002
Save


Modify Tab Order / Captions
Tab Order
Tab Title
SQL Code
3
Inventory Summary TEST
INV
2
Sales History
3
1
Vendor Parts XXXXX
1001
4
Purchase Order Items
1002
Save


Vendor Parts XXXXX
Sales History
Inventory Summary TEST
Purchase Order Items
Drag a column header here to group by that column
vend_code
vend_name
vend_part

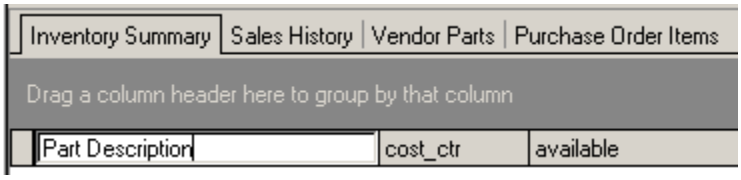
Arranging the detail field column width, location & captions within the individual detail tabs

When you add field columns to your detail tabs, the columns will be created in the order they are added, with a width corresponding to their database field definition, and they will display the columns heading caption of their corresponding database fields. You have the ability to override these default settings.

To change the column width or location
Use your mouse to drag and drop the individual field columns on the detail workbench tab. Keep in mind that each user has the ability to override their default width and location at run-time, so what you design may not be what the users actually see.

To change the field column heading captions
Double-click on the field column heading you wish to modify. An edit box will open allowing you to type a new column heading.

Note: Detail columns objects are created with object names [field_name]V[n]Col (where n equals the numeric value of the tab on the grid). If this object is specified in the E21 Multi Language Screen Layout Maintenance program, any labels associated with the workbench will be overridden by the multi-language settings at runtime.

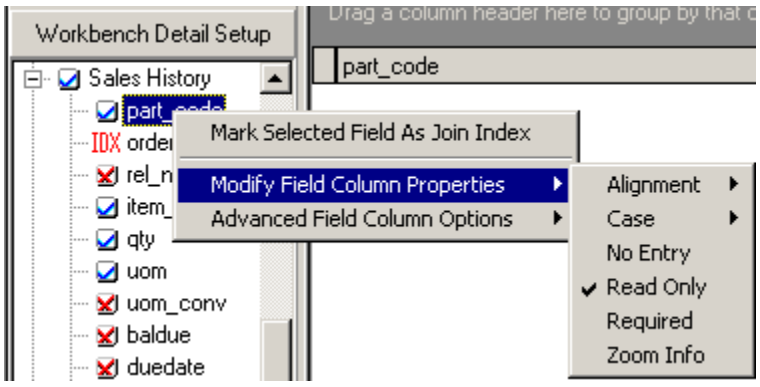


Modifying the runtime Properties of the Detail Field Columns

Each selected field from the detail dataset has a corresponding column on the Workbench Detail Grid View. Each of these columns has associated properties (Alignment, Case, etc) which tell the workbench program how they should perform at runtime. To modify these properties:

1) Right Click on the specific field you wish to modify to open the selected item's popup menu.
2) Choose "Modify Field Column Properties."

If this menu option is not enabled, you do not have a valid field selected or the workbench is locked.



From the Modify Field Column Properties menu option, you may choose from the following 6 selections: Alignment, Case, Noentry, Readonly, Required & Zoom Info. Each of these options behaves exactly the same as on the header. For that reason, we will not go into further detail on these options here.

For more information, see "Modifying the Runtime Edit Box Properties of the Header Dataset fields."

Advanced Options

Once you have setup your workbench, there may be a need to alter the default functionality your workbench screens. Only technically savvy users should attempt to alter the advanced options. These options differ depending on whether you are altering the Dataset or the individual Fields on the dataset, so be sure to differentiate between the "Advanced Dataset Options" and the "Advanced Field Edit Options."

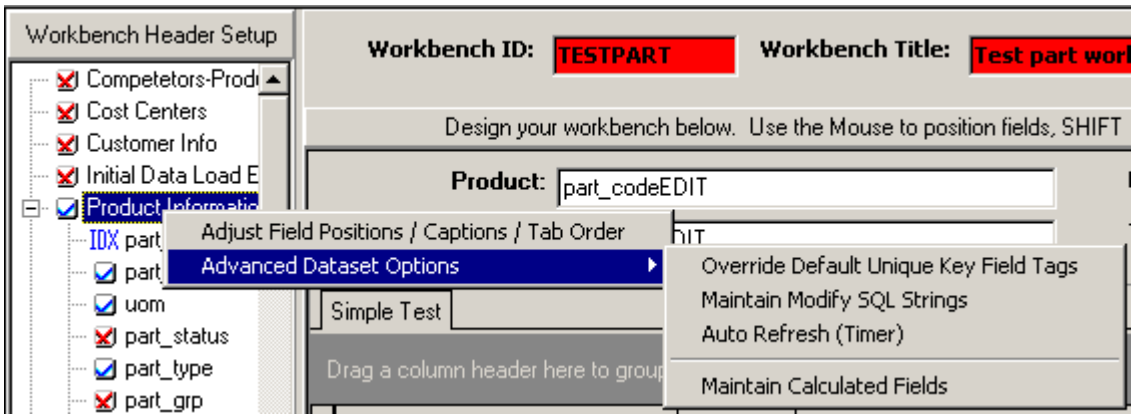
Note: Advanced options work exactly the same for both the header and detail except where specified.

Advanced Dataset Options

To access the Advanced Dataset Options:

1) From within the appropriate Workbench Setup treeview, right-click on the appropriate dataset to open the selected items popup menu.
2) Choose Advanced Dataset Options

Note – The dataset should be active and the workbench cannot be locked or this option will not be enabled and all of the submenu items will not be accessible.



Override Default Unique Key Tags

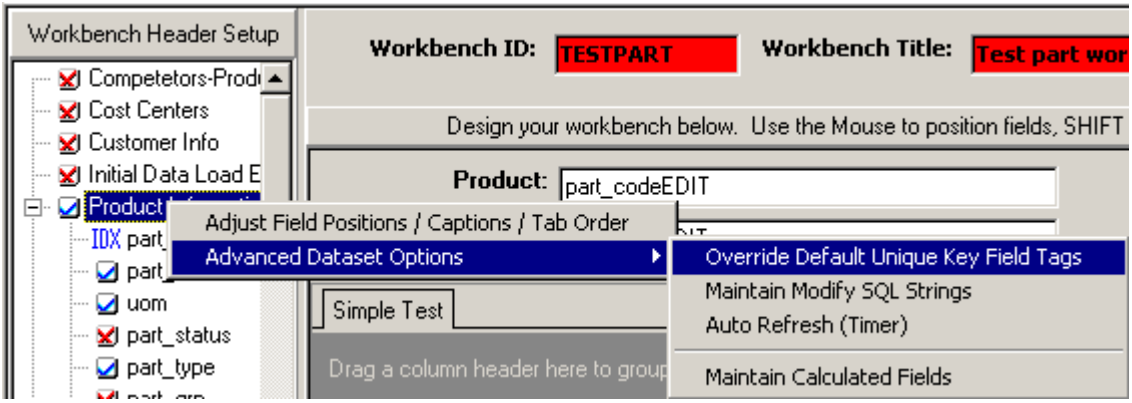
"Unique Key Field Tags" are used by E21 programs to uniquely identify each record in a dataset. On Workbench Table Maintenance screens (TGI Only) this information is necessary so that users cannot enter duplicate data or modify the unique key of existing data. For all screens (including query-only workbenches), we need this information if we are going to be calling other programs from the E21 navigation system (See E21 documentation for setting up navigation items).

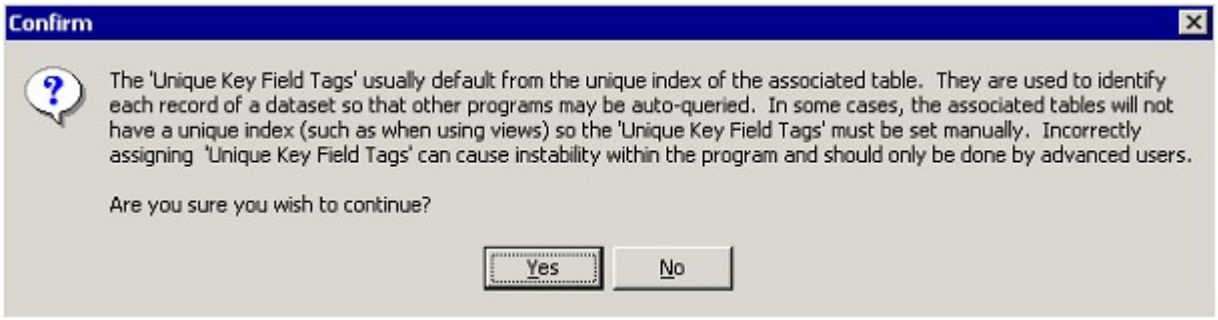
By default, the workbench knows how to uniquely identify each record in a dataset by reading the unique key off of the associated table from the database. This is not possible however, when the "Main Table" from your SQL is a View. When this is the case, you will need to assign the Unique Key Field Tags manually.

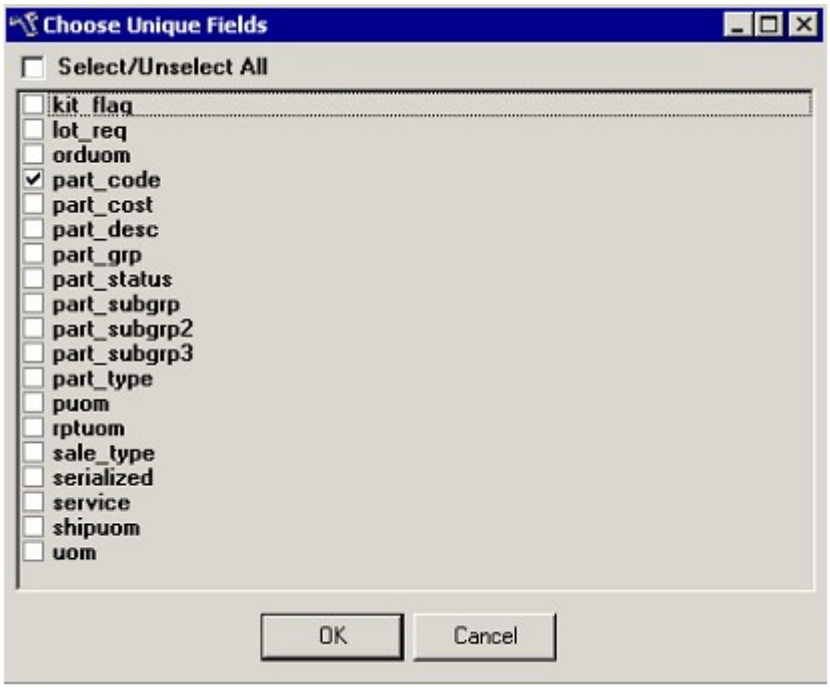
Note: The “Main Table” of an SQL is established in the Workbench SQL Builder. For more detailed information on building Workbench SQL Datasets, see Using The E21 Workbench SQL Builder.

To assign the “Unique Key Field Tags” manually:

1) From the Advanced Dataset Options, choose the “Override Default Unique Key Field Tags” submenu.
2) You will get a warning message stating that overriding the unique key tags can lead to instability in the program. Click ‘Yes’ to continue.
3) Finally, you will be shown a list of all fields on the dataset. Choose the fields that uniquely identify each record on the dataset and when you are finished, click OK to save your work.







Maintain Modify SQL Strings

Workbenches allow you to take a standard SQL from the SQL Builder and modify it locally. This will allow you to do things like filter a workbench SQL by cost center, by user, etc. You can use existing E21 functions to modify your SQL such as "get_val_loc('ccmstr', 'cost_ctr')", or you can enter any valid SQL modifiers such as "(part_status = 'A')". To do so:

1) From the Advanced Dataset Options, choose the "Maintain Modify SQL Strings" submenu.
2) The Modify SQL Strings Maintenance Dialog will open.
3) Enter as many Modifiers as you wish per SQL. Click Insert to add new filters. Click Save when you are done.

The workbench program will parse your original SQL from the SQL Builder and place the filters you enter into the SQL to be used in the workbench Query components. As an example, here is how a string entered in the Modify SQL Strings dialog will effect an SQL.

Modify SQL String: (part_status = 'A')

Example 1. Original SQL from SQL Builder:

Select * from partmstr

Becomes:

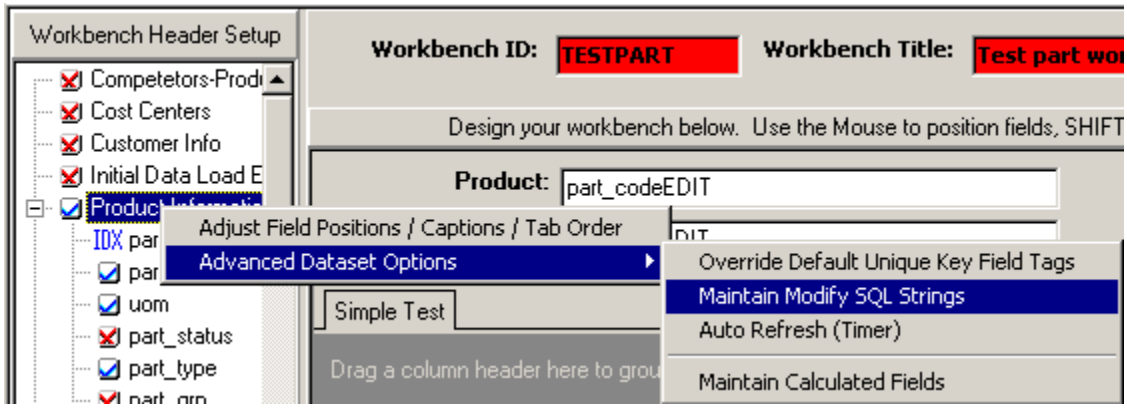
Select * from partmstr where (part_status = 'A')

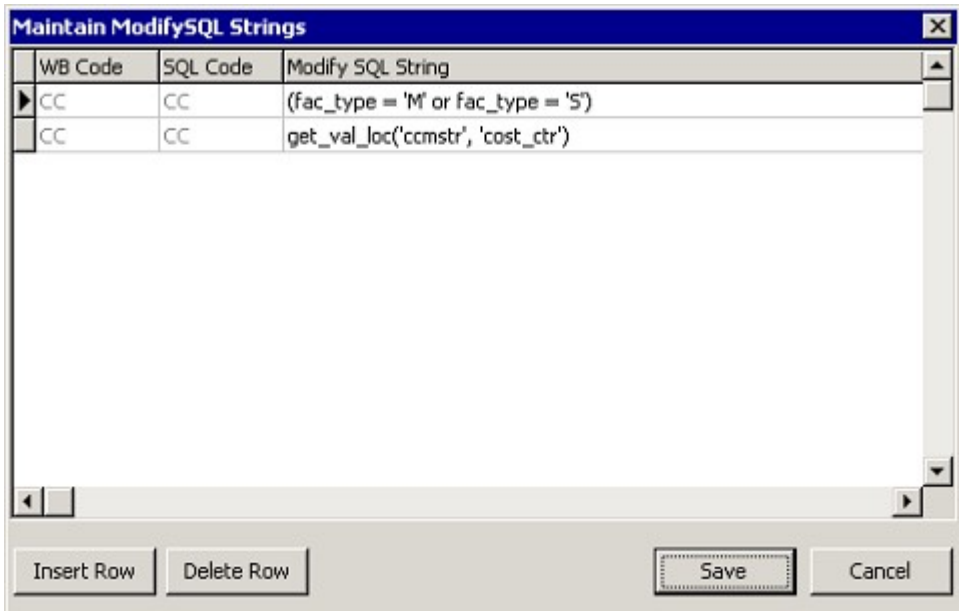
Example 2. Original SQL from SQL Builder:

Select * from partmstr where part_type = 'X'
Order by part_code

Becomes:

Select * from partmstr where part_type = 'X' and (part_status = 'A')
Order by part_code





Note: Only experienced TGI programmers who understand how E21 functions such as "get_val_loc" affect the dataset should use these options.

Note: The syntax entered in the Modify SQL Strings dialogs are not verified to be correct and can cause your workbenches to become unstable or return unexpected results if not entered correctly. Based on the complexity of the original SQL, in some cases, you will not be able to modify the SQL as desired from within the workbench itself and must alter the SQL from within the Workbench SQL Builder. For more detailed information on building Workbench SQL Datasets, see Using The E21 Workbench SQL Builder.

Auto Refresh (Timers)

TGI Workbenches allow you to create screens which automatically refresh themselves on desired intervals so that you can design screens which are always up to date.

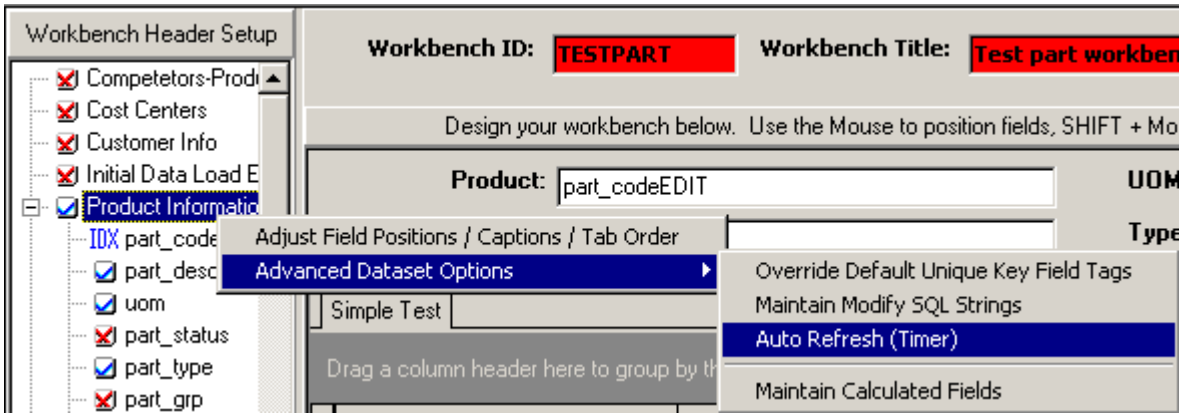
Note: This option is only available in E21 versions 7.1 and higher

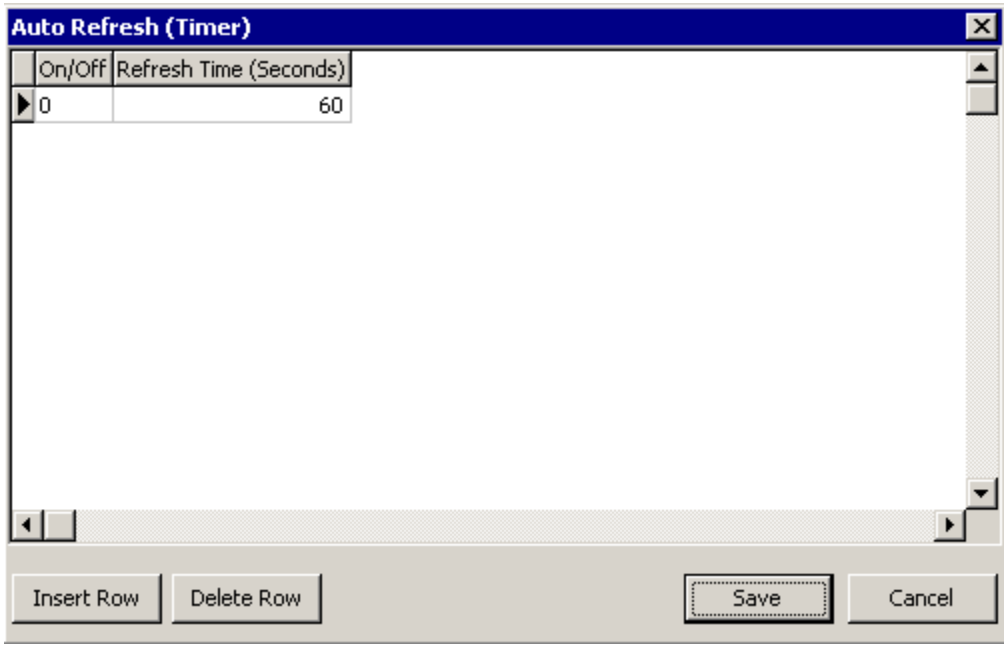
To add Auto Refresh capabilities:

1) From the Advanced Dataset Options, choose the "Auto Refresh (Timer)" submenu.
2) The Auto Refresh (Timer) dialog window will open.
3) Click "Insert Row" to add a new timer, & enter the interval upon which you want this dataset to refresh itself. Intervals times are in seconds (10=10 seconds, 300=5 minutes, etc).
4) Click Save when you are done.

Note: You should only add 1 Timer per dataset.

Note: For Header-Detail screens, the active detail in the workbench will also refresh each time the header dataset is refreshed. The other tabs of the detail will only be refreshed once their tabs are clicked on to activate them.





Maintain Calculated Fields

Calculated fields are fields that are added manually to your workbench because they not on the actual dataset. They are used primarily as secondary lookup fields from foreign tables. For example, let's say you wanted to display the description for a product, but the only field on your dataset was the product code. You could create a calculated field that would be added to your dataset manually. You could then place this field on your form just like any other field.

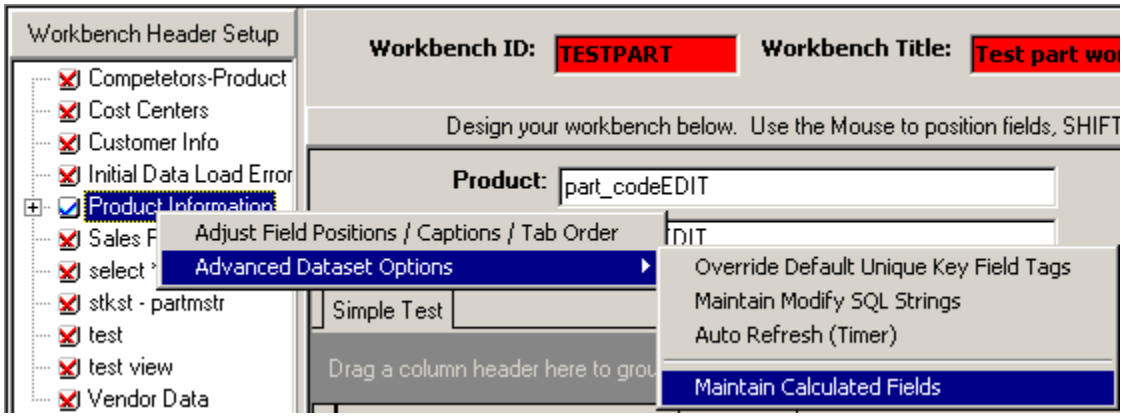
Note: Calculated fields by themselves don't perform any logic. They should be used in conjunction with "Field Lookups." For more information on creating Field Lookups, see "Advanced Field Edit Options."

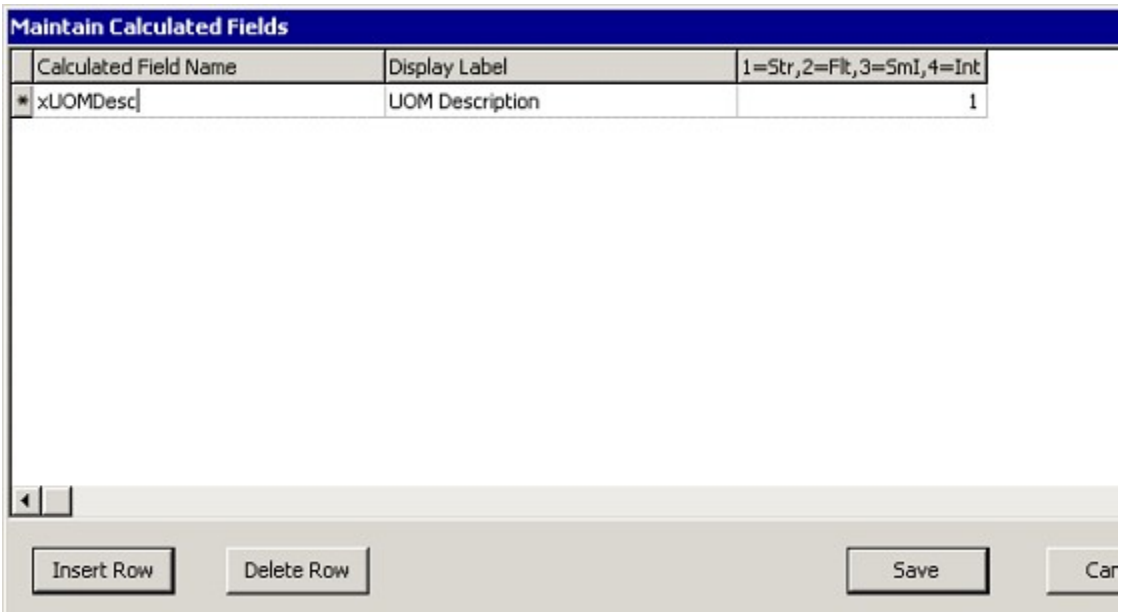
To add calculated fields to your dataset:

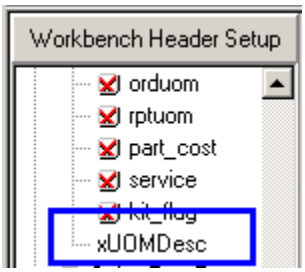
1) From the Advanced Dataset Options menu, choose the "Maintain Calculated Fields" submenu.
2) The Maintain Calculated Fields dialog window will open.
3) Click the Insert Row button to add a new calculated field.
4) For each calculated field you wish to add, you must enter a Calculated Field Name (must be a valid field name for your database), a Display Label, and the field datatype (1=String, 2=Float, 3=SmallInt, 4=Integer).
5) Click Save.

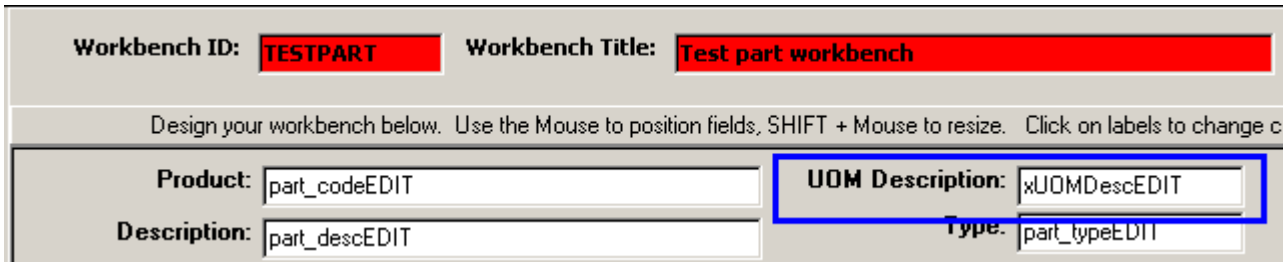
Your new calculated field will appear in both your Dataset treeview and also on your workbench form. You may then modify your calculated field (move it, resize it, change the caption, etc) as you would any other field on your dataset.

Note: Calculated fields do not have checkmarks next to them on the dataset treeview. To remove calculated fields from your workbench, return to the Maintain Calculated Fields dialog, highlight the field you wish to remove, and click the "Delete Row" button.









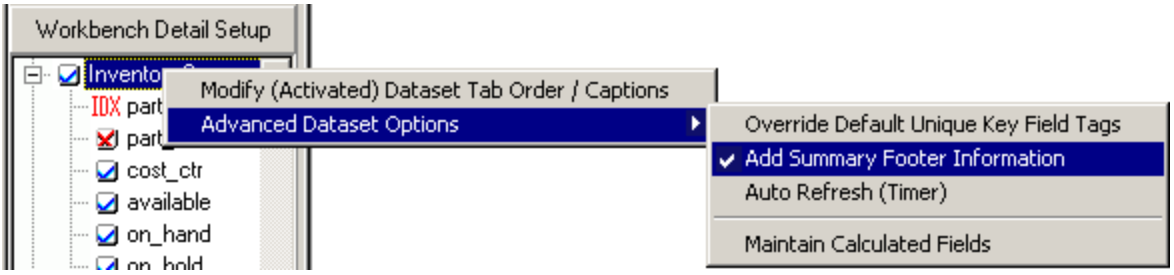
Add Summary Footer Information (Detail Datasets Only):

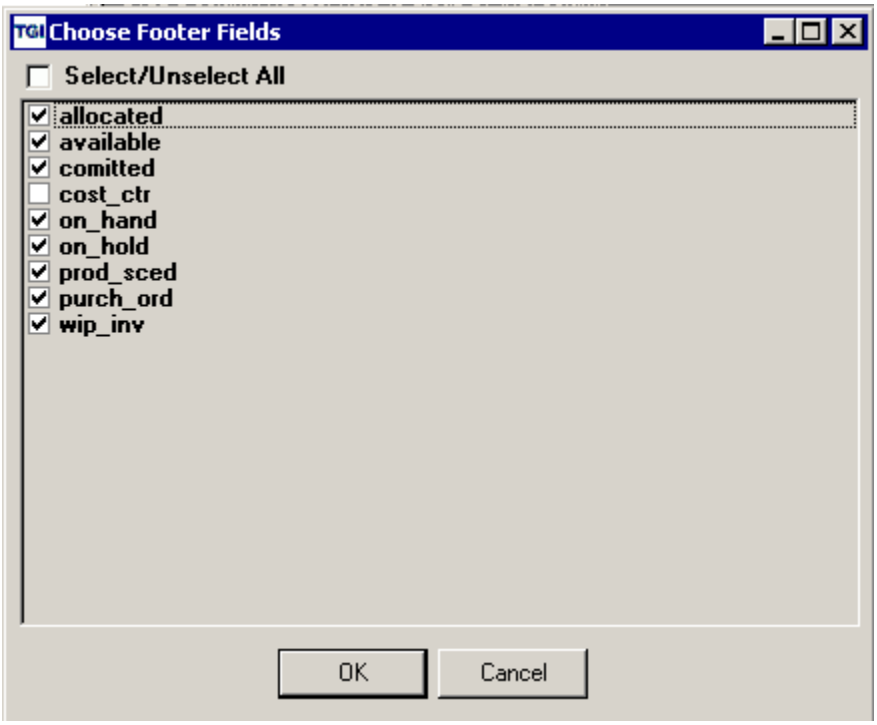
Create a summary footer row for any numeric field displayed on a detail grid.

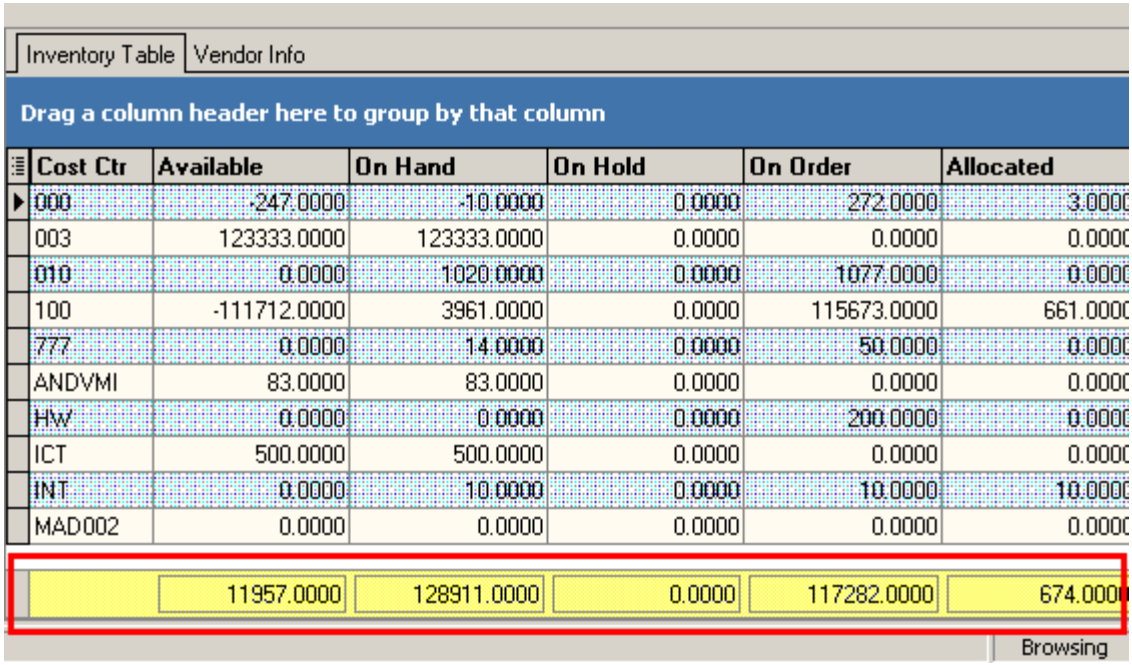
To turn on Summary Footer Information:

1) From the Advanced Dataset Options menu, choose the “Add Summary Footer Information” submenu.
2) The Choose Footer Fields dialog window will open.
3) Select any **numeric** fields you wish to add to the footer row. The dialog will allow you to select non-numeric fields, however doing so will cause your data to concatenate and will likely cause undesired / unexpected results.
4) Click the OK button

Note - You will not see the footer information from within the workbench setup screen, however the footer information will display as defined at runtime.







Advanced Field Edit / Column Options

To access the Advanced Field Edit / Column Options:

From within the appropriate Workbench Setup treeview, right-click on the appropriate dataset field to open the selected items popup menu.

For header datasets, choose “Advanced Field Edit Options.” For detail datasets, choose “Advanced Field Column Options.”

Note – The dataset should be active and the workbench cannot be locked or this option will not be enabled.

Field Validation

Field Validation is necessary when you are in Insert or Edit mode (Workbench Table Maintenance screens) and you are entering field data that is validated against secondary tables in the database. For example, if you were entering product information and you wanted to enter a UOM code, the system would validate that the UOM entered was a legitimate value. Only TGI system developers can create Workbench Table Maintenance screens.

Note: All workbenches created by customers will be Query-Only and therefore this option will not be enabled.

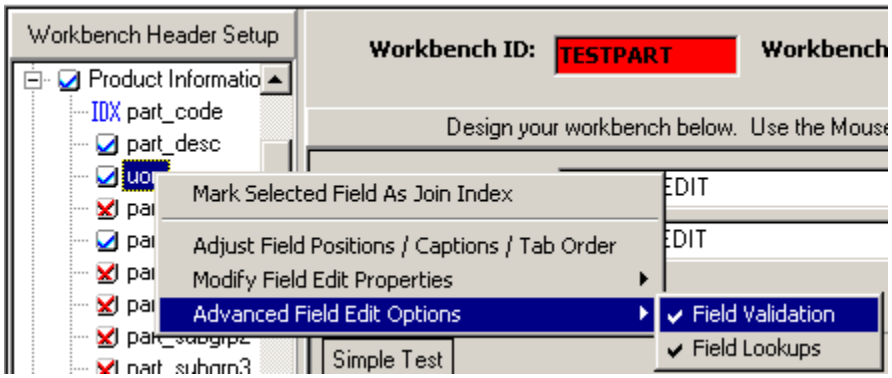
To add Field Validations to a Field Edit / Column:

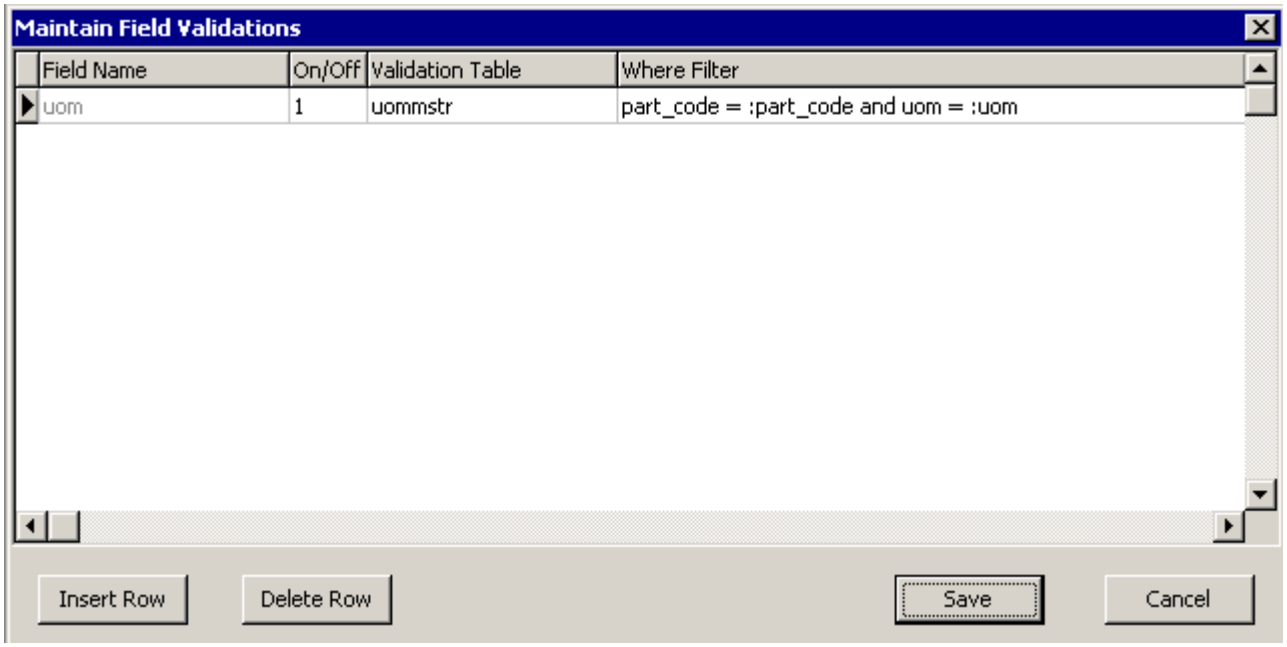
1) From the Advanced Field Edit / Column Options menu, choose the “Field Validation” submenu.
2) The Maintain Field Validations dialog window will open.
3) Click the Insert Row button to add a new Validation. Typically you will enter only 1 validation record per field.
4) For each calculated field you wish to add, you must enter the following 4 pieces of information:
   - Field Name: The key field name from the secondary table.
   - On/Off: 1=On, 0=Off.

- Validation Table: The secondary table name.
- Where Filter: The where section of the SQL that will be built to verify your data. Syntax should be as follows: [<secondary_key_field_name> = :<local_dataset_field>]. See below examples.

5) Click Save.

Where Filter Example 1: part_type = :part_type
Where Filter Example 2: part_type = :part_type and (part_type = 'F' or part_type = 'R')





Field Lookups

Field lookups are secondary SQL statements that grab additional information from the database that is not included in your original SQL dataset.

Note: Calculated fields need a corresponding dataset field **not on the original dataset**. They should be used in conjunction with "Dataset Calculated Fields." For more information on creating Calculated Fields, see "Advanced Dataset Options."

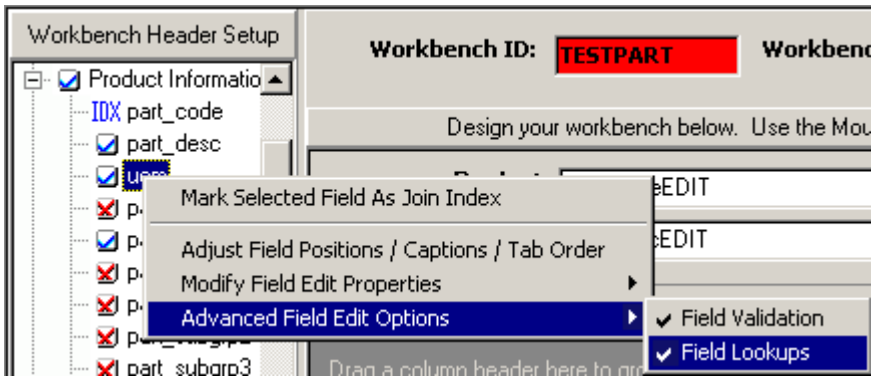
To add Field Lookups to a Field Edit / Column:

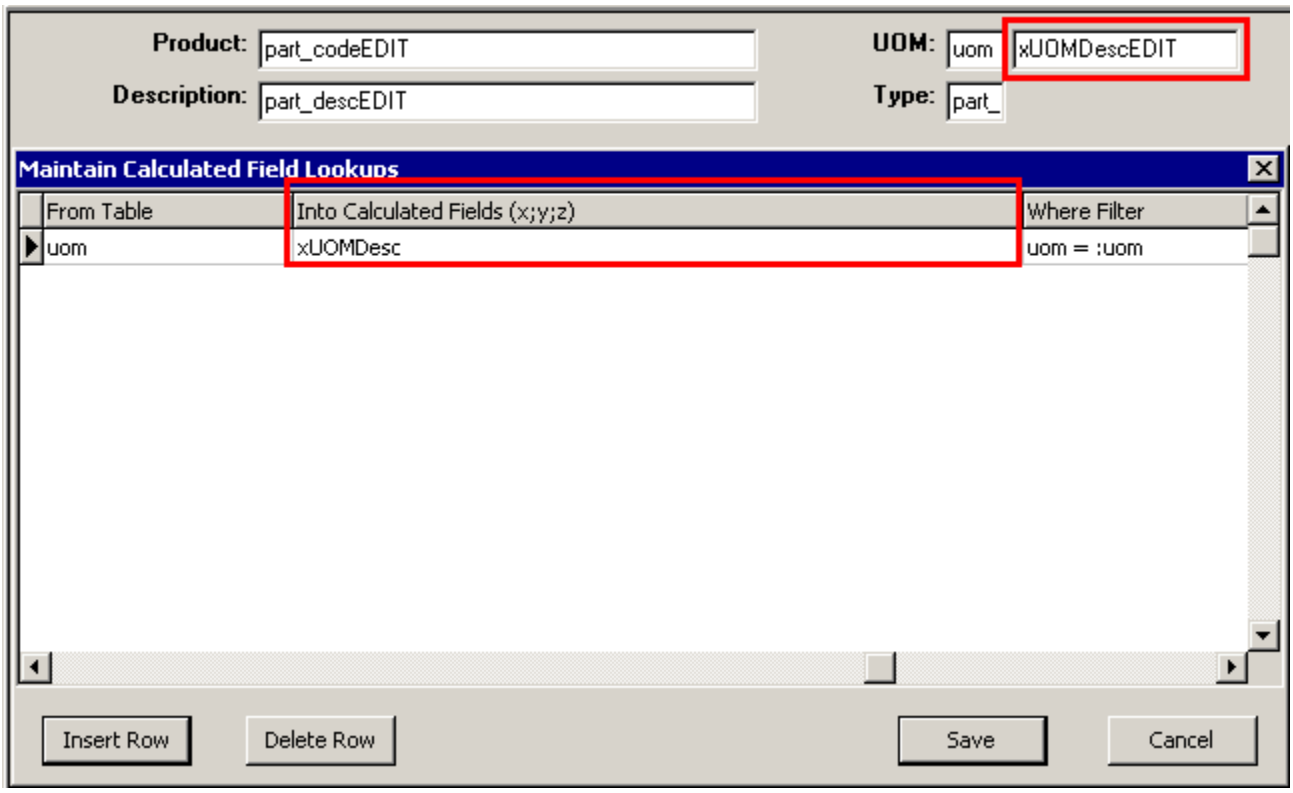
1) Create your Calculated Fields before adding Field Lookups.
2) Determine which field from the dataset will be the key for your calculated field.
3) From the Advanced Field Edit Options menu, choose the "Field Lookups" submenu.
4) The Maintain Calculated Field Lookups dialog window will open.
5) Click the Insert Row button to add a new Lookup. You should enter only 1 validation record per dataset field, however, you can select multiple calculated fields to lookup per record.
6) For each calculated field you wish to add, you must enter the following 6 pieces of information (Not all 6 pieces of info fit on the dialog window at the same time, so be sure to use your window's horizontal scroll bar to add everything properly before saving):
   - Field Name: The key field name from the secondary table. This should be the join to your selected dataset field.
   - On/Off: 1=On, 0=Off.
   - Lookup Fields: The fields you will be selecting from the secondary table to place into your calculated field. Separate fields by commas if you need to select multiple fields.
   - From Table: The secondary table name.
   - Into Calculated Fields: The name(s) of the calculated field you created previously. This information should correspond to the "Lookup Fields" above. Separate your calculated fields by semi-colon if you need to select multiple fields.
   - Where Filter: The where section of the SQL that will be built to verify your data. Syntax should be as follows: [<secondary_key_field_name> = :<local_field>]. See below examples.
7) Click Save.

Lookup Fields Example: vend_name, address1, city + ', ' state + ' ' + zip

Into Calculated Fields Example: xVendName; xAddress; xCSZ

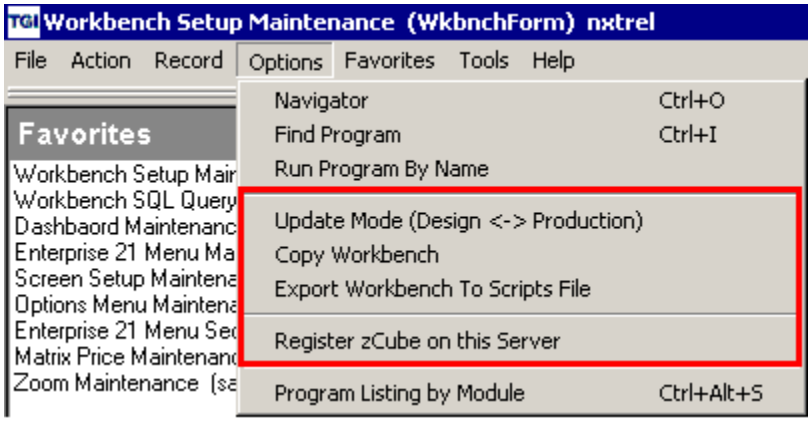
Where Filter Example 1: vend_code = :vend_code
Where Filter Example 2: vend_code = :vend_code and vend_status = 'A'





Additional Workbench Menu Options

In addition to the standard E21 menu options available to all programs, there are additional options specific to designing / creating / maintaining E21 Workbenches. These options are access thru the main form "Options" menu.



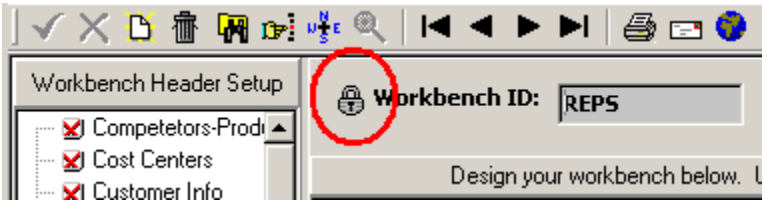
Update Mode (Design <-> Production)
Workbenches have two distinct modes: "Design Mode" and "Production Mode." This menu option allows you to toggle between the two modes.

When you create a new workbench it starts off in Design Mode. All this means is that you can make changes to the workbench, and although the system may popup messages asking you to verify certain

changes, it will basically allow you to do anything you want. When a workbench is in "Production Mode" it locks the workbench from any further changes and you cannot edit it in any way. You should not put a workbench into Production Mode until you have completed all changes, it has been thoroughly tested, and you are ready to release the workbench to your users.

Note – Altering a workbench in anyway will cause user defined screen settings (such as custom column sizing) to be cleared the next time a user enters a workbench screen. As such this can be annoying to end-users. Only alter workbenches that are release to end-users when necessary.

When a workbench is in Production Mode, you will see a small Lock Icon display next to the "Workbench ID" label in the Main Workbench Definition Section. If you do wish to alter a locked workbench, you much first convert it back to Design Mode.



Copy Workbench
Just as it sounds, this option allows you to copy existing Production workbenches. This option is not available for workbenches in Design Mode.

Export Workbench To Scripts File
This option allows you to move workbenches from one dB to another. Unlike standard E21 forms that are compiled into the E21 executable, workbench form data is all stored in your database. The benefits of this design are numerous, however, as such, workbench forms are only initially available in the dB in which they were created.

This option unloads all pertinent data into an SQL script file that can then be run to load workbenches across numerous dB's.

Register ZCube on this Server
If you are using Enterprise DSS style charts in any of your workbenches then you will need to register the ZCube.DLL library file on your server. This option makes registering the library as easy as clicking the menu option.

For this option to execute properly, the ZCube.DLL file must reside in the same directory as your E21 executable.

Miscellaneous Workbench Setup Screen Notes and Features

Soft SQL Filtering

In addition to the Maintain Modify SQL Strings functionality which builds hard filters into your workbench program (Same filter for all users), the workbenches also have the ability to setup soft-filters (Different filters for different users).

To use soft filtering, you will need to access 2 separate programs within E21. These programs (which may or may not be included in your menu system) are:

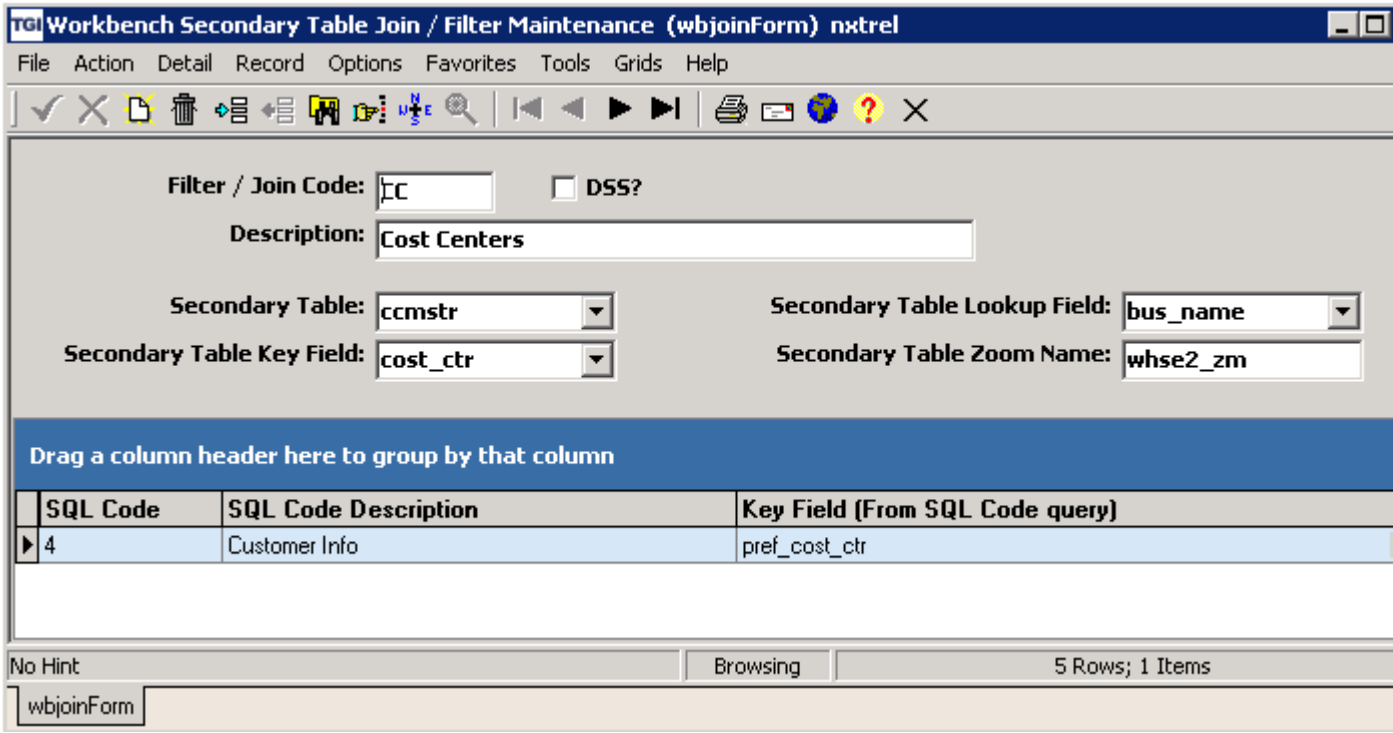
Workbench Filter / Join Maintenance (WbjoinForm)
Workbench Filter Assignment (WbfusrForm)

NOTE – Even if the programs are not a part of your E21 menu system, you should be able to access them via the "Options -> Run Program By Name" option of your main E21 menu toolbar. You may also set these two programs up as "Hot Buttons" for your Workbench Setup screen.

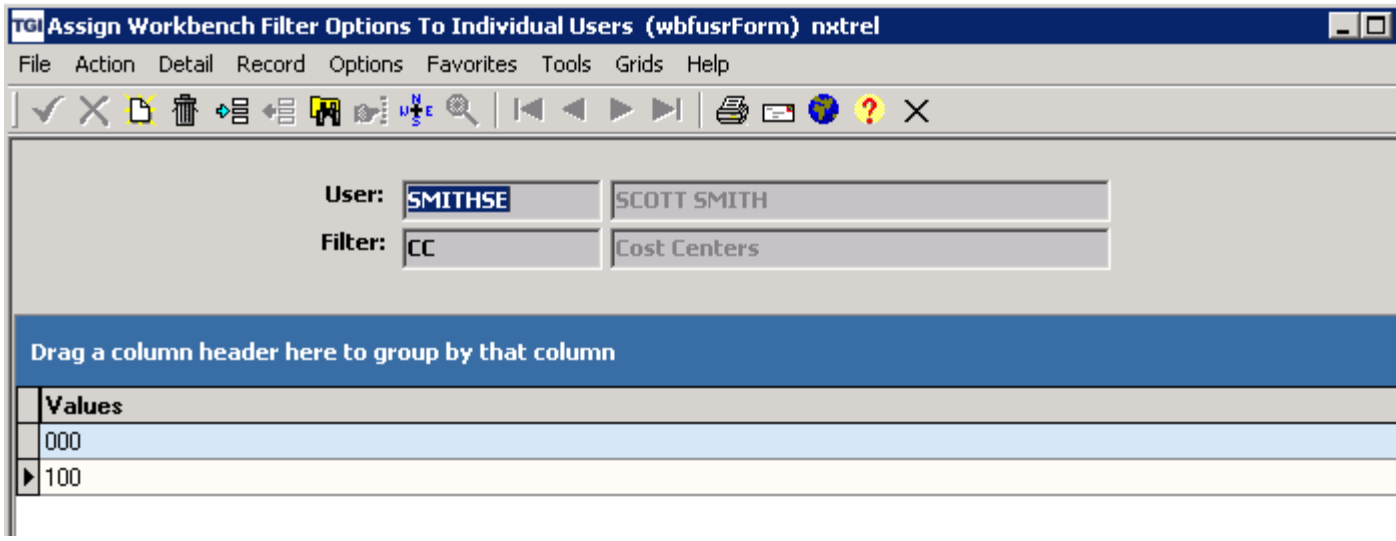
In the "Workbench Filter / Join Maintenance" program you establish rules, and then attach those rules to your previously created SQL codes. In the "Workbench Filter Assignment" program you then assign which values are accessible for each rule by user. When a user enters a workbench that has a soft filter, they will only see the data that they have been setup to see.

Lets say you have created an SQL in the Workbench SQL Builder which selects all sales commission information from your system. You then create a workbench that uses this SQL. By default, all users who use this workbench will see all commission information for all users in the system. By using a soft filter, you can have the workbench only display information that is relevant/allowed for that particular user. A sales rep might only see their own info, whereas a sales manager might see all of the info for all of the reps in their region, both from within the same workbench.

In the example below, E21 user "SMITHSE" will only have access to records where "pref_cost_ctr" = [000 or 100] for any workbench which uses SQL Code "4". In the header portion of the WbjoinForm you simple create your Filter code and description. The secondary field information in the WbjoinForm program is not used by workbenches. They are only there so that users can validate and zoom values available in WbfusrForm. In the detail portion of the form enter any SQL Codes that you wish to add a soft filter for, along with the key field for this filter.



In the WbfusrForm, now enter the E21 user and the filter code in the header, and all values (for this particular filter) that the user will have access to in their workbench:

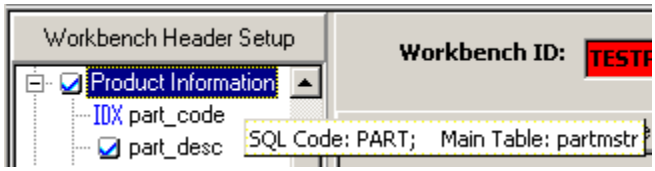


That's it! When user "SMITHSE" opens any workbench which uses SQL Code "4", that dataset will only return records for Cost Ctr's 000 and 100.

Miscellaneous

Holding your mouse over any of the items (dataset or field nodes) on the Header or Detail Setup sections will show the SQL Code that that item comes from along with the Main Table for that SQL Code.

This is especially useful when you are not familiar with the design of the datasets. Using this information you can quickly and easily lookup the SQL Codes from the SQL Builder to see where the information comes from and how it is derived.



Workbench Tables Names and Descriptions

Workbenchh
This is the top-level workbench table used by all workbenches. It contains all workbench definition information such as workbench code, type, description, etc., along with the some additional fields such as SQL Code and maintenance permissions (allow add, update, remove) for header datasets. Detail-Only workbenches do not use these additional fields.

Workbenchd
This table contains all the necessary information to create the field components for the header portion of your workbench. This includes things like field names, position, captions, edit properties, etc. One record will exist for each field that has been turned on (blue checkmark) in the header setup section of your workbench. Detail-Only workbenches do not use this table.

Workbenchdeth
This table contains the dataset-level information for all dataset(s) used by the detail potion of your workbench. It contain the SQL Codes, tab orders, tab titles, etc., and also contains the maintenance permissions (allow add, update, remove) for detail datasets. One record will exist for each dataset that has been turned on (blue checkmark) in the detail setup section of your workbench. Header-Only workbenches do not use this table.

Workbenchdetd
This table contains the column-level information for all fields (for all datasets) used by the detail potion of your workbench. This includes things like field names, column labels, column properties, visibility, etc. Unlike the corresponding table for header fields, one record will exist for each field on your dataset, regardless of whether it is turned on in the workbench. Header-Only workbenches do not use this table.

Note: If your corresponding select statement selects all fields (*) from a table, the workbench runners will compare the number of records on workbenchded to the number of fields on your dataset. If the number of fields on the dataset is greater than the number of records on workbenchdetd, the system assumes new fields have been added to the corresponding table and workbenchdetd will be updated automatically.

Workbenchsqlh
This is the top-level table used to store the SQL information as defined in the Workbench SQL Builder. It contains the SQL Codes, Description, SQL Type, main table name, etc.

Wokbenchsqld
This is the table used to store the actual sql text as defined in the Workbench SQL Builder.

Wbfieldfoot
This table stores the information necessary to create Summary Footer Information capabilities within your workbench. Header-Only workbenches do not use this table.

Wbfieldtags
This table stores which fields have been manually assigned as the unique keys for a given dataset in the "Override Default Unique Key Tags" procedure.

Wbfilterh
This table is used by the header section of WbfusrForm to assign soft filter joins to individual users within your workbench.

Wbfilterd
This table is used by the detail section of WbfusrForm to assign valid soft filter values to individual users within your workbench.

Wbjointabh
This table is used by the header section of WbjoinForm to create soft filters to be used within your workbench.

Wbjointabh
This table is used by the detail section of WbjoinForm to assign soft filters to SQL Codes to be used within your workbench.

Wblookups
This table stores the information necessary to create field lookups for manually added calculated fields within your workbench.

Wbmodifysql
This table stores the information necessary to maintain Modify SQL Strings within your workbench.

Wbtimers
This table stores the information necessary to create Auto Refresh capabilities (Timers) within your workbench.

Wbvalidate
This table stores the information necessary to create Field Validation within your workbench.

Wbzooms
This table stores the information necessary to create Field Zooms within your workbench.

Using the E21 Workbench SQL Builder

The E21 Workbench SQL Builder is a visual SQL query builder that allows users to build, parse & analyze SQL queries visually, with minimal knowledge of SQL (Structured Query Language). You will need a basic knowledge of SQL concepts, but the E21 Workbench SQL Builder will help you to write correct SQL code, hiding technical details. Its is a true two-way query builder, so for users who DO have more SQL expertise, you may combine visual query building with direct SQL query text editing for a "best-of-all-worlds" experience.

The E21 Workbench SQL Builder is where users will create the SQL's that will be transformed into Datasets available for use in the E21 Workbench Setup Screen, and eventually in your individual workbench programs.

Here we will discuss in detail the steps involved in creating E21 Workbench SQL's / Datasets, and also each function available within the program.

Note: Having knowledge of the underlying table structures of your database is a must in creating usable, functional Datasets for use in workbenches. For this reason, it is recommended that only qualified users have access to the E21 Workbench SQL Builder program.

Understanding the layout and terminology

Terminology

What is SQL
SQL, or Structured Query Language, is a database computer language designed for the retrieval and management of data in relational database management systems.

What is an SQL Query
An SQL Query is simply a text string written in SQL which is used by your relational database management system to extract data from your dB. The E21 workbench SQL builder helps users build these SQL Queries so that they can create the Datasets necessary in the E21 Workbench screens.

What is a Dataset
A Dataset is an object that contains all of the table and field information, along with any actual data which may have been retrieved from an SQL Query from the corresponding database;

Screen Layout

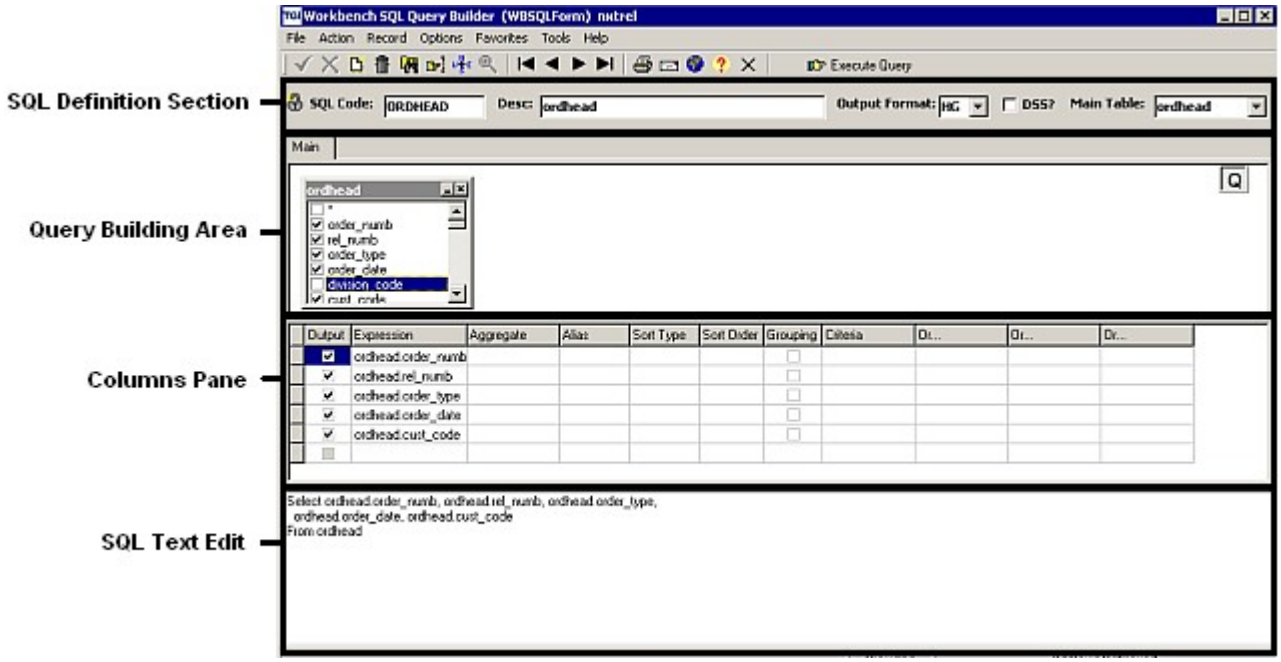
When you first enter the E21 Workbench SQL Builder, you will notice that although it generally looks like most other E21 programs, there are a few major differences. The Workbench Setup Screen is divided into four distinct sections.

1) The SQL Definition section - This where you will define the SQL Code, the SQL Description, and the other top-level dataset-specific information used by the workbenches.

2) The Query Building Area - This is where, using drag and drop along with other methods, you will add your database objects (tables, views, etc), join multiple objects together where applicable, select which fields will be available, etc.

3) The Columns Pane - This is where you add fields and expressions to your query, along with filter criteria and conditional statements.

4) SQL Text Editor – For users with in-depth SQL knowledge, you may use this section instead of the Query Building Area and Columns Pane to build / maintain your Datasets by entering the SQL statements manually.



Creating a new SQL

Each workbench screen will have a unique SQL Code, along with a Description, Output Format, DSS Flag and Main Table. To create a new E21 SQL, enter the Workbench SQL Builder Screen and click the Add button from the standard E21 toolbar. This will clear all sections of the SQL Builder and put you in insert mode. You must build your SQL Definition before building your SQL Query.

Enter the following five pieces of information:

SQL Code
This is the unique identifier of your SQL Query. It can be up to 10 alphanumeric characters long. This code will be used internally by the workbenches to differentiate all Datasets.

SQL Description
This is the description that will be given to your SQL codes. The description is what is displayed in the Workbench Setup Screen for each available dataset and is also used for default workbench form titles and tab captions.

Output Format
Although this information is not needed to build an SQL, the resulting Datasets that are build by the workbench screens differ based on how they will be used. Therefor it is necessary when you are defining the SQL Codes to define their output type. There are three valid output types:

- HG – Header Grid – These queries will only be available for the header sections of E21 workbenches.
- DG – Detail Grid - These queries will only be available for the detail sections of E21 workbenches and will only be used to return data in the standard E21 grids.
- DC – Detail Chart - These queries will only be available for the detail sections of E21 workbenches and will only be used to return data as EntDSS charts (Note that EntDSS grids are NOT available at this time in E21 workbenches)

DSS Flag
The database used by the Query Builder to select the valid objects available will be the same database used by your current instance of E21. In other words, if you open E21 in production, you will have access to all production tables and views, whereas if you open E21 in test, you will have access to all test tables and views. The only exception to this rule is if you turn on the DSS flag. In this case, the dB objects available will be from the associated DSS database as defined in dsssetup.entdss_odbc.

Main Table
Enter the main table name used by this dataset. When you create a new SQL, a default object for this table will be created in your Query Building Area, and your SQL query will default to "select * from <main table>".

Why enter a Main Table?

Prior to E21 workbenches being created, each dataset on our standard E21 forms needed to have a "Main Table" defined programmatically for some of our default form functionality to work. This functionality includes, but is not limited to QBE functionality, Field Help, Passing parameters to navigation options, etc. Although this field is not actually used in the query you are building in the SQL Builder, the workbench that the resulting dataset resides on will use it for many functions.

This is not an issue for single table queries as you can simply put the only table in the query. However, for multi-table joins this may be an issue for a variety of reasons:

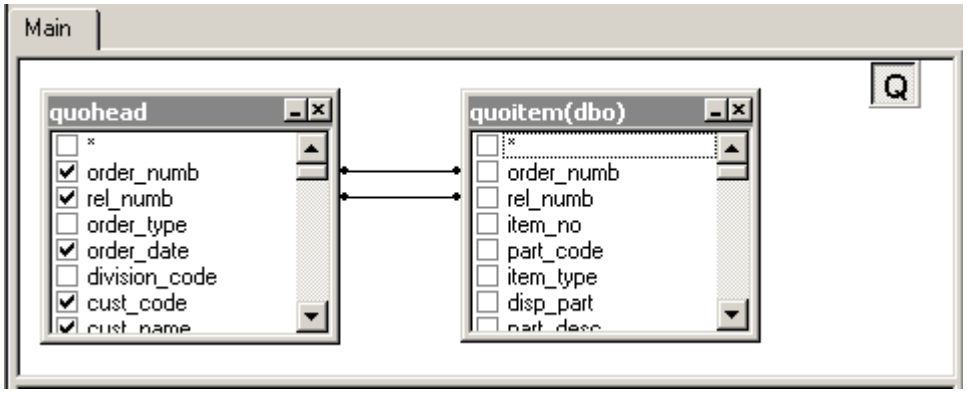
- The fields in the workbench you select as index fields must come from the Main Table;
- Only fields from the Main Table will allow QBE (others will be noentry in query mode);
- Field Help will not work for fields not on the Main Table
- More!

For this reason, unless you are simply joining table to do simple things like secondary name lookups where none of the above functionality is necessary, we strongly suggest using database Views for multi-table Datasets. The E21 Workbench setup screen easily allows you to convert the SQL's you have built to database views (and back), so doing this conversion is a very quick and simple solution to the problems listed above. For more information of converting your multi-table join into a database View, see the "Convert SQL to View" section of this manual.

Once you have entered these 5 pieces of information, save your SQL by clicking the accept button in the standard E21 toolbar. All sections of the SQL Builder will be enabled upon saving your new workbench. You are now ready to build your SQL Query by using the Query Building, Column Pane & Text Edit sections.

Query Building Area

The Query Building Area is where, using drag and drop along with other methods, you will add your database objects (tables, views, etc), join multiple objects together where applicable, select which fields will be available, etc.



Adding Database Objects to your Query

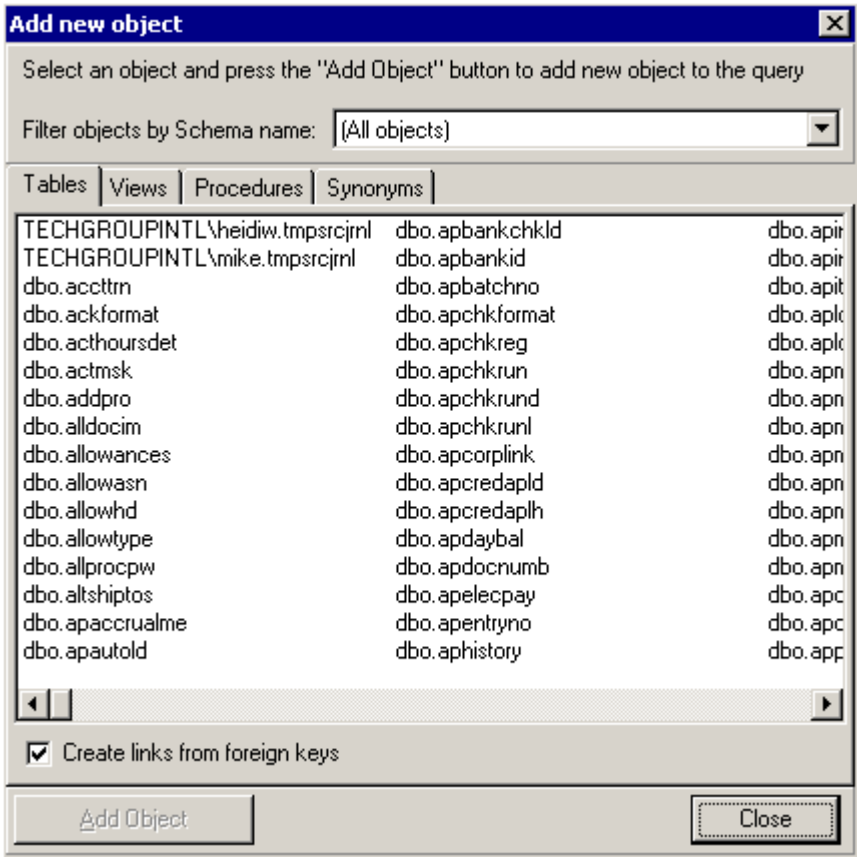
Although you can always change any part of your query by altering the actual SQL in the SQL Text Editor, this section will focus on building / maintaining your SQL via the methods available in the Query Building Area.

Note – Altering any part of your SQL will immeditely put your standard E21 form into "Edit" mode. To cancel any changes, simply click the Cancel button on your standard E21 toolbar and all info will revert to its original values. Click the "Accept" button.

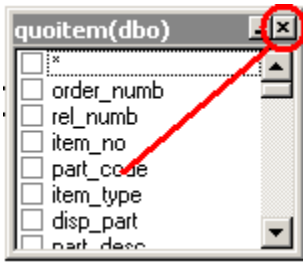
Upon adding a new SQL, the system will create a default object based on your Main Table entry.

To add additional objects, right-click anywhere within the Query Building Area and select the "Add Object" item from the context popup menu. The Add New Object window will appear. The Add New Object window allows you to add as many objects as you wish at once. The objects are grouped by type within four tabs: Tables, Views, Procedures (Functions) and Synonyms. You may select one or several objects by holding the Ctrl key and then press the Add Object button to add these objects to the query. You may repeat this operation several times. After you finish adding objects, press the Close button to hide this window.

For those database servers that have schemas or allow selection of objects from different databases, you may filter objects by database or schema name by selecting the necessary schema or database from the combo box at the top of the window, however, depending on what you are trying to do, cross database queries may cause instability within your final workbench program

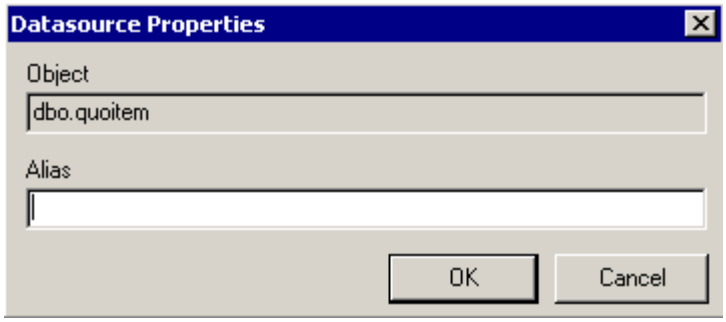


To delete database objects from your query, simply click the cancel icon at the top right of each object. Deleting a database object will automatically remove all reference to this object from your SQL, including joins, relevant field information, etc.



Setting Object Alias Names

To set alias for an object or derived table in the query, you may right click at the object and select the Properties item from context popup menu or you may double-click at the object header. The Datasource Properties dialog will appear.

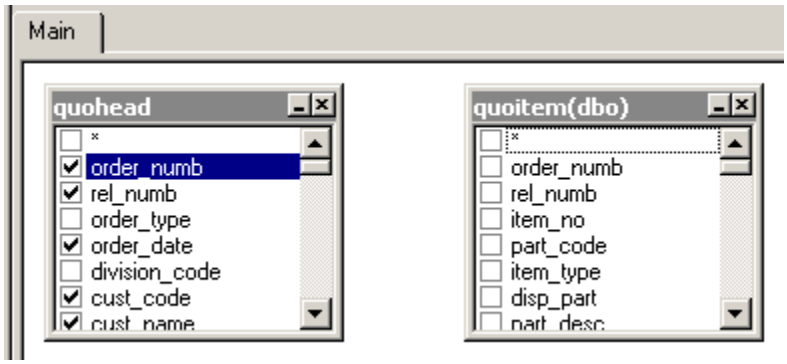


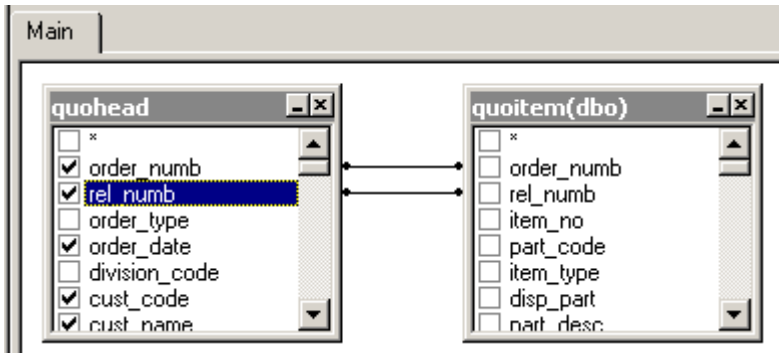
The Datasource Properties dialog may contain other server-specific datasource options, but the Alias property is the same for all database servers.

Adding and removing joins

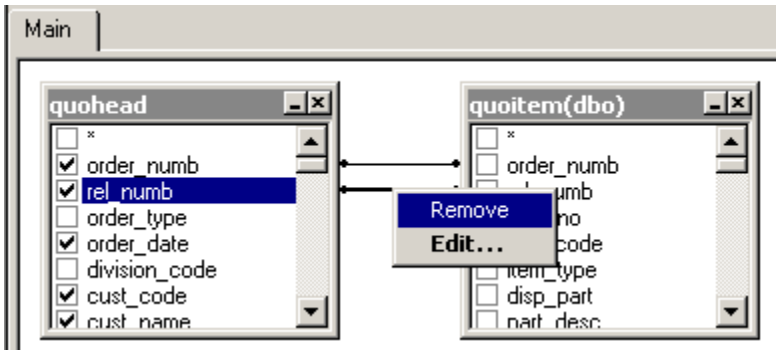
When two objects referenced with a foreign key relationship are added to the query, they become joined automatically with INNER JOIN. For those servers that have no support of JOIN clause, The E21 Workbecnh Setup Screen adds condition to WHERE part of the query.

To create a link between two objects (i.e. join them) manually, you should select the field by which you want to link the object with another and drag it to the corresponding field of the other object. After you finish dragging, a line connecting the linked fields will appear.





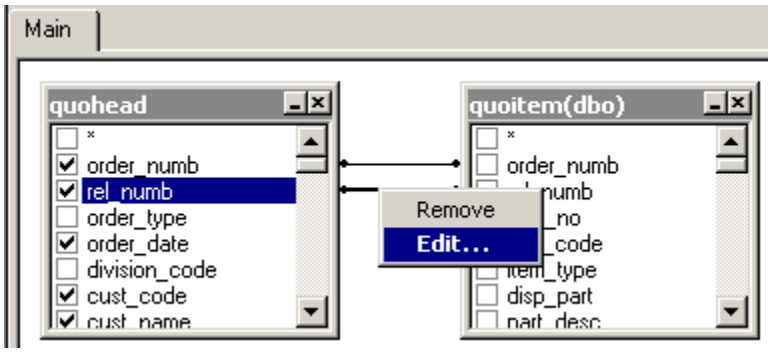
To remove a link between objects, right-click the link line and select the **Remove** item from the context popup menu or simply press **Delete** key.

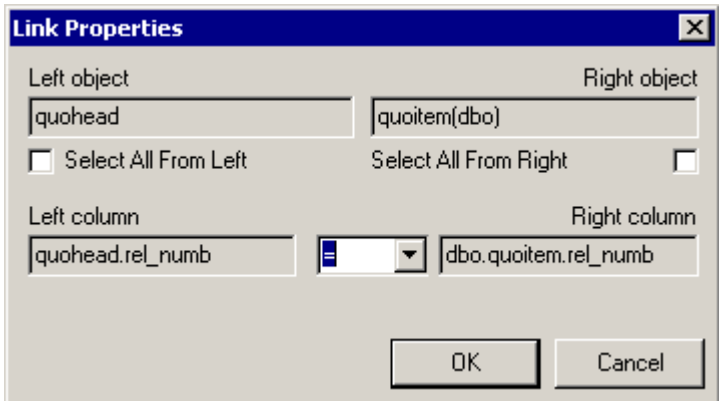


Changing the join type

The join type created by default is INNER JOIN, i.e. only matching records of both tables will be included in the resulting dataset. To change joint type, you may right click the link and choose the join type from "Change type" sub-menu of context popup menu.

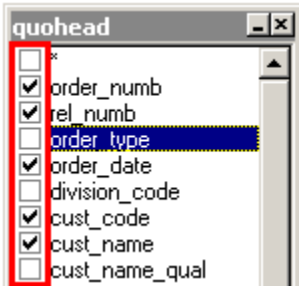
To define join type and other link properties you may right click the link and select the Properties item from the context popup menu or double-click it to open the Link Properties dialog.





Adding and removing fields and expressions

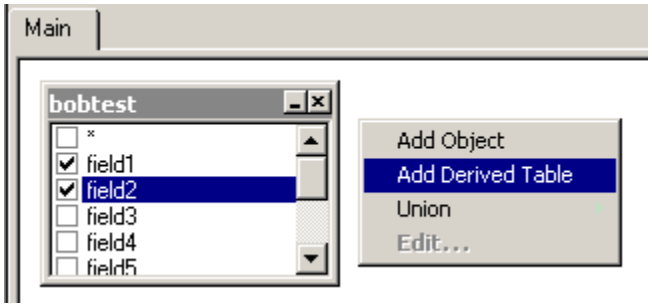
To add a field to the list of query output fields you may check the box at the left of a field name in the datasource field list at the Query Building Area. To include all the fields of the specific object you may check the box at the left of the asterisk item in the datasource field list. Also you may drag fields from the Query Building Area to the Columns pane to get the same result.



Working with Derived Tables

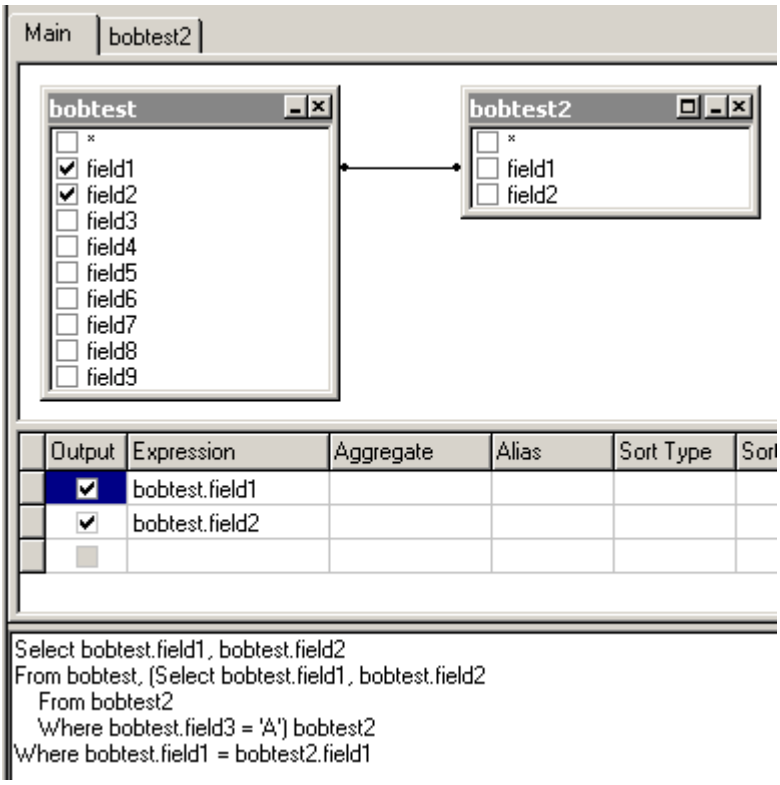
In very basic terms, a derived table is a sub-query calculated on the fly from a select statement and used as a datasource for the main query. In concept this is similar to creating a temporary table and then using the temporary table in your query, but the approach is much simpler, because it can all be done in one step. They can be tremendously useful in certain situations.

To add a derived table, you should right click on the Query Building Area and select the Add Derived Table item from context popup menu.



A new object representing the newly created derived table will be added to the query building area, and the corresponding tab will be created for it. This tab allows you to build it visually in the same way as the main query. Another way to switch to the corresponding derived table tab is to right click at the caption of an object representing the derived table and select the "Switch to derived table" item from context popup menu.

You may set an alias for derived table the same way as for ordinary database object.
You may always get back to the main query and switch to any sub-query or derived table using tabs above the Query Building Area.



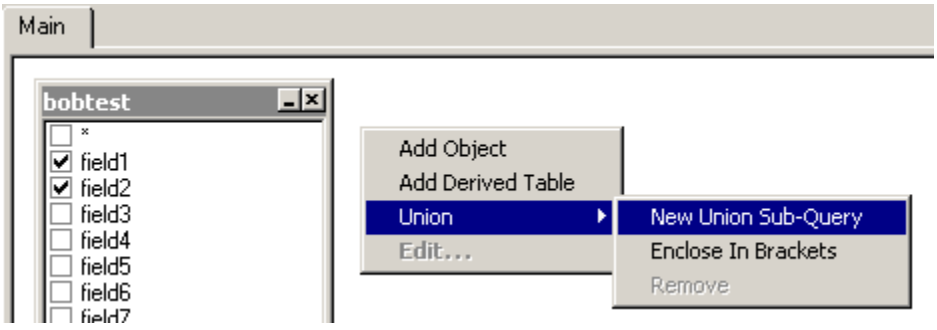
Working with Unions

Union sub-queries are managed within the Union Panel in the top-right corner of the Query Building Area. Initially there's only one union sub-query labeled with the "Q" letter. All required operations are performed by means of context popup menus.

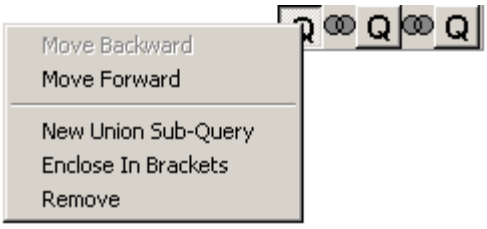
Union sub-queries can be grouped with other sub-queries and joined with different operators (UNION, UNION ALL, EXCEPT, INTERSECT).

To add a new union sub-query, right-click in the Query Building Area and select the New Union sub-query menu item.

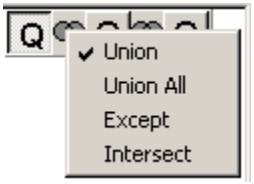
To enclose the sub-query in brackets, select the Enclose in Brackets menu item.



Right click on the union sub-query box (labeled with the "Q") to open its content popup menu. From there you may move the sub-query results within the query, add a new sub-union, enclose the union in brackets, remove the subquery or bracket, etc.



To change the union joining operator, right click on joining operator box to open its content popup menu. select the necessary operator in the list of supported operators in context popup menu.



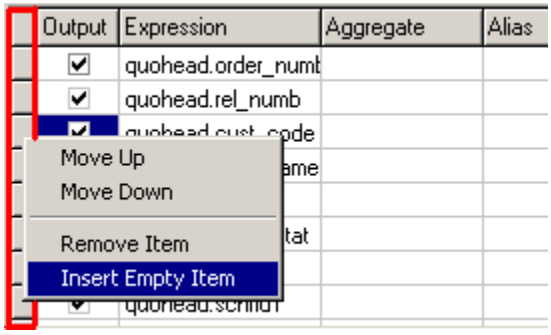
Columns Pane

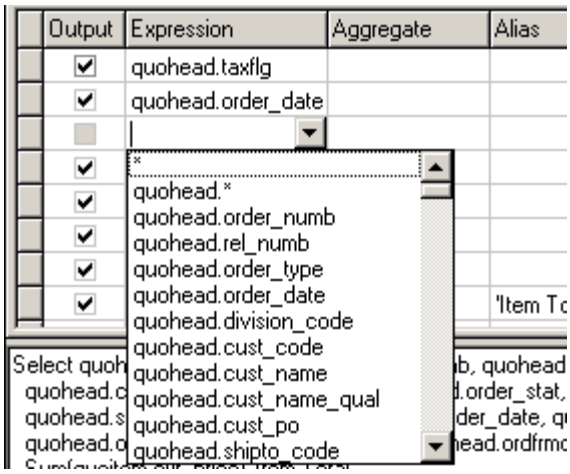
This is where you add fields and expressions to your query, along with filter criteria and conditional statements.

Adding, removing & re-ordering fields and expressions

In the previous section we learned how to add and remove fields and expressions using the Query Building Area. Another way to add a field is to select a field name from the drop-down list of the Expression column in the Columns Pane. And of course you may type any valid expression in the Expression column in the Columns Pane.

To insert an empty line to the Columns pane, right-click on the left hand side of any existing row to open the context popup menu and choose the “Insert Empty Item” option.





You may also remove and re-order lines at the Columns Pane using the context popup menu.

Note: Fields will display in your resulting Dataset in the same order they appear in your Column Pane.

Working with expressions in the Columns Pane

The Output column determines presence of expression in the SELECT list of the query.

The Alias column allows you to set aliases for your expressions. Aliases become headings of columns in result datasets.

Note: Never set an alias for an expression that you will use as the join field in a header detail workbench. This may cause unexpected results. Also, setting an alias may make it harder for users of the Workbench Setup Screen to know what field is actually being represented. We recommend avoiding setting aliases in your workbench datasets and instead alter your label and column headings in the Workbench Setup Screen itself.

Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria for
☑	quohead.order_numb					☑	For groups
☑	quohead.rel_numb					☑	For groups
☑	quohead.cust_code					☑	For groups
☑	quohead.cust_name					☑	For groups
☑	quohead.rep1		Sales Rep	Ascending	1	☑	For groups
☑	quohead.order_stat		Status	Ascending	2	☑	For groups
☑	quohead.numfld1			Descending	3	☑	For groups
☑	quohead.schfld1					☑	For groups

The Asterisk Item

Even if you don't select any fields from query datasources, an asterisk item will be added to the select list of result query ("Select * From ..."). This is made because a SELECT query without any columns will produce an error for most of the database servers, thus it's useless for end-users.

By selecting any fields or by adding any output expressions to the query, an asterisk item will be removed from result query (you may add it manually at any time). The asterisk items placed first at the datasource

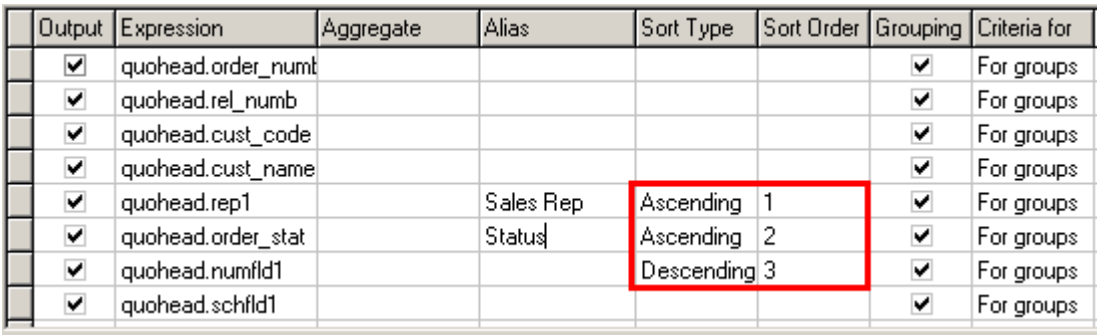
field's list are intended to add all fields from specific datasource, not from all of the datasources ("Select datasource.* From ..."). These asterisk items are not affect the rest of the datasource field check boxes, but act independently. If you want to check all the datasource field check boxes, not the asterisk item, use appropriate items of the datasource context popup menu.

Note: “select * from …” create datasets which contain every field in the associated database table when the query is executed. However, when detail workbenches (Type = DO or MD) are created, the columns added to that workbench are the fields on the database table that exists at that specific time. If subsequently fields are added (or even worse removed!) from your database table, this may cause instability. The detail workbench screens themselves will attempt to correct the difference if they detect that the number of columns on the workbench and the fields on the dataset table do not match, however, this solution is not bulletproof. For this reason, we recomend always defining the specific fields you would like on your dataset vs using the “Select * from ...” method.

Sorting

To define sorting of result dataset, you may use the Sort Type and Sort Order columns of the Columns Pane. The Sort Type column allows you to specify the way the fields will be sorted - in the Ascending or Descending order. The Sort Order column allows you to setup the order in which fields will be sorted, if more than one field will be sorted.

To disable sorting by some field you should clear the Sort Type column for this field.



Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria for
☑	quohead.order_numb					☑	For groups
☑	quohead.rel_numb					☑	For groups
☑	quohead.cust_code					☑	For groups
☑	quohead.cust_name					☑	For groups
☑	quohead.rep1		Sales Rep	Ascending	1	☑	For groups
☑	quohead.order_stat		Status	Ascending	2	☑	For groups
☑	quohead.numfld1			Descending	3	☑	For groups
☑	quohead.schfld1					☑	For groups

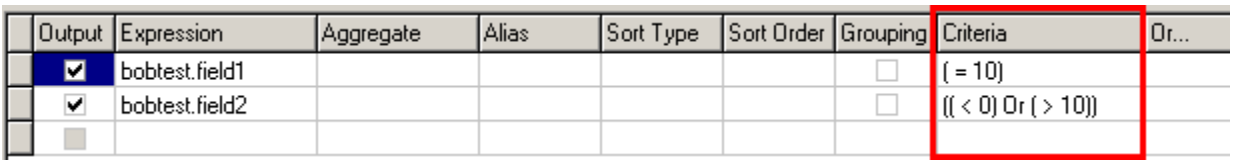
Filter Criteria

To define criteria, you may use the Criteria and all of the Or columns of the Columns Pane.
In these cells you should write conditions omitting the expression itself.

For example, to get the following criteria in your query:

WHERE (Field1=10) AND ((Field2 < 0) OR (Field2 > 10))

Your criteria should be as follows:



Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria	Or...
☑	bobtest.field1					☐	(= 10)	
☑	bobtest.field2					☐	((< 0) Or (> 10))	

Some expressions may be of Boolean type, for example the EXISTS clause. In this case you should type "= True" in Criteria column of such expressions or "= False" if you want to place NOT operator before the expression.

Hint: The easiest way to learn how to add criteria in the Column Pane is to is to add it manually to the SQL Text Editor and see how it will be parsed and represented visually.

Grouping

To build a query with grouping, you mark expressions for grouping with the Grouping checkbox.

A query with grouping must have only grouping or aggregate expressions in the SELECT list. Thus, The E21 SQL Query Builder allows to set the Output checkbox for grouping and aggregate expressions. If you try to set this checkbox for a column without Grouping or Aggregate function set, a Grouping checkbox will be set automatically to maintain validity of result SQL query.

When Columns Pane contains columns marked with the Grouping checkbox, a new column called "Criteria for" appears in the grid. This column specifies appliance of criteria to expression groups or to their values.

For example, you have a column "Field1" with Aggregate function "Avg" in your query and you type the "> 10" in Criteria column. Having the "for groups" value set in the Criteria for column, the result query will contain only groups with average quantity greater than 10, and your query will have the "Avg(Quantity) > 10" condition in HAVING clause.

In otherwords, the following settings:

	Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria for	Criteria
	☐	bobtest.field2					☑	For groups	
	☐	bobtest.field1	Avg				☐	For groups	> 10

Create this SQL:

*Select **
From bobtest
Group By bobtest.field2
Having Avg(bobtest.field1) > 10

Having the "for values" value set in the Criteria for column, the result query will calculate the Average aggregate function only for records with Quantity value greater than 10, and your query will have the "Quantity > 10" condition in WHERE clause.

	Output	Expression	Aggregate	Alias	Sort Type	Sort Order	Grouping	Criteria for	Criteria
	☐	bobtest.field2					☑	For groups	
	☐	Avg(bobtest.field1)					☐	For values	> 10

*Select **
From bobtest
Where Avg(bobtest.field1) > 10
Group By bobtest.field2

Hint: The easiest way to learn how to add groupings in the Column Pane is to is to add it manually to the SQL Text Editor and see how it will be parsed and represented visually.

Sub-Queries

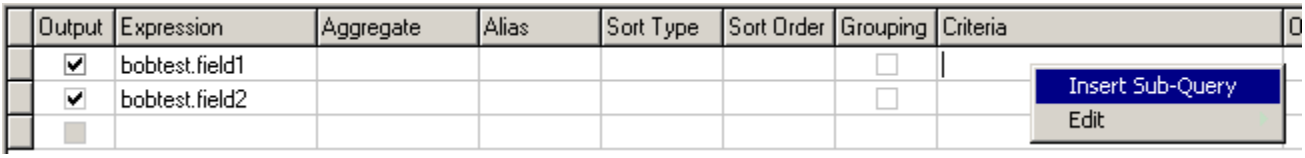
Note: This chapter describes working with Sub-queries used in criteria expressions to limit result dataset (in WHERE, HAVING and SELECT clauses). To find out how to work with sub-queries used as datasources for the main query (i.e. in the FROM and WITH clauses), please refer to the section on Derived Tables.

You may add a sub-query as part of the expression or condition in the Columns Pane for a specific field. To add a sub-query, you can either just type your subquery into the Criteria column, or you can use the visual controls to build the sub-query for you.

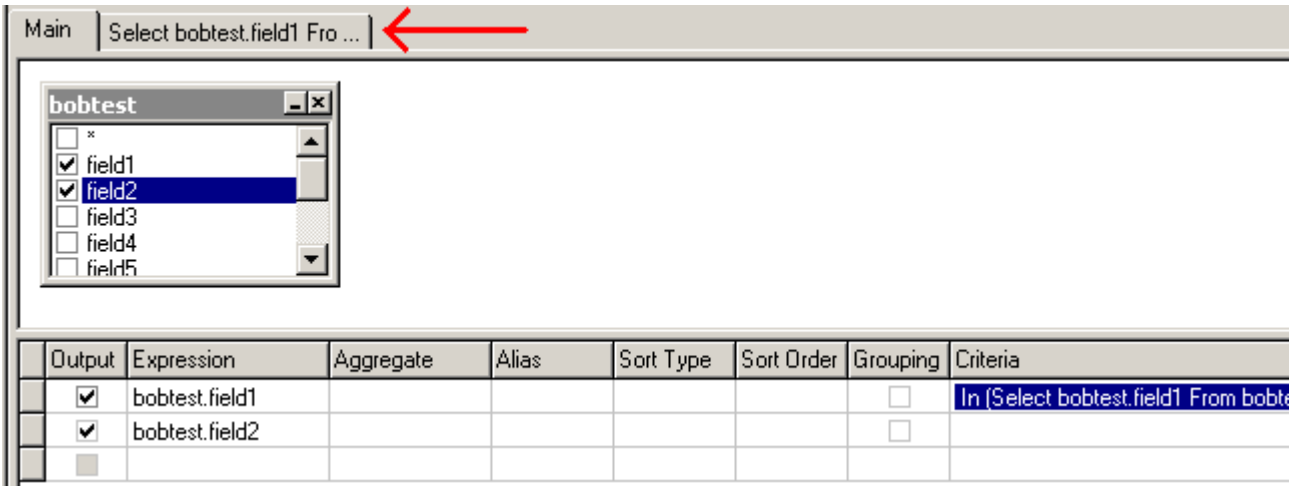
To add the subquery yourself, just type your sub-query into your Criteria field for the field:

In (select field1 from bobtest2 where field2 = 'TEST')

To use the object controls, first make sure the cursor is in the Criteria column of the Columns Pane. Then right click in the Criteria column for a new sub-query and select the Insert Sub-Query item from context popup menu.



To build a newly added sub-query visually, confirm editing by pressing the Enter key. A new tab is created above the Query Building Area. By clicking on the tab you'll be switched to a sub-query tab where you can build it visually in the same way you built the main query. If the cell contains more than one sub-query, a drop-down list will appear to select the necessary sub-query.



SQL Text Editor

For users with in-depth SQL knowledge, you may use the SQL Text Editor instead of the Query Building Area and Columns Pane to build / maintain your Datasets by entering the SQL statements manually.

Simply type in your SQL (or copy / paste it from other sources) and when you exit the SQL Text Editor by either clicking on any other area in the E21 Workbench SQL Builder, it will perform the following functions:

1) Put the record into Edit mode.
2) Validate the SQL. If it is invalid you will get an error message. Depending on the error, Invalid SQL's may break the Objects in the other SQL Builder components, so unless you are very certain you know how to fix the SQL manually, we suggest canceling the changes when the SQL is invalid.
3) Draw / refresh the proper objects & values in the other SQL Builder components.
4) Reformat (and possibly change) the SQL text to fit acceptable E21 Workbench standards

After entering your SQL, you can choose either Accept or Cancel from the standard E21 toolbar to save or cancel your changes.

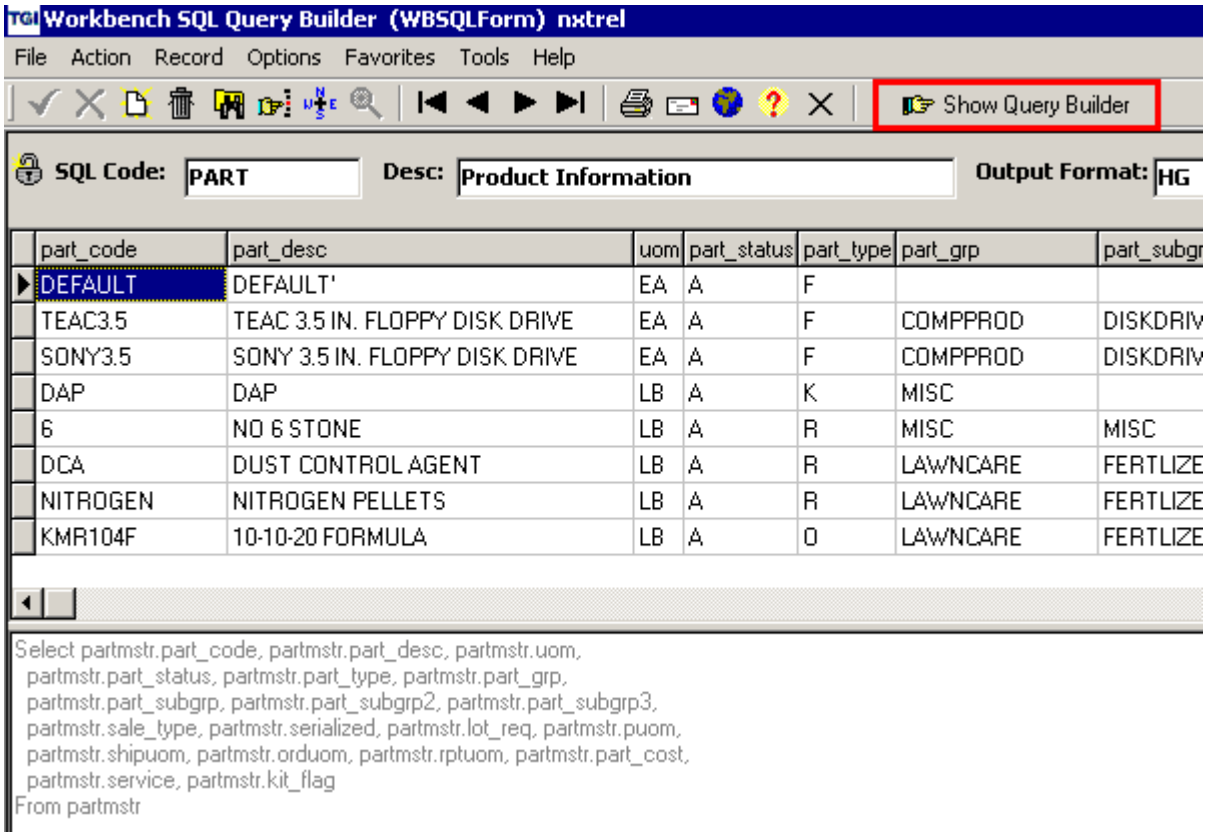
Additional Workbench SQL Builder Toolbar & Menu Options

In addition to the standard E21 menu options available to all programs, there are additional options have been added to make working with and maintaining your Workbench SQL Queries easier. These options are accessed via the standard E21 toolbar, and also via the main form "Options" menu.

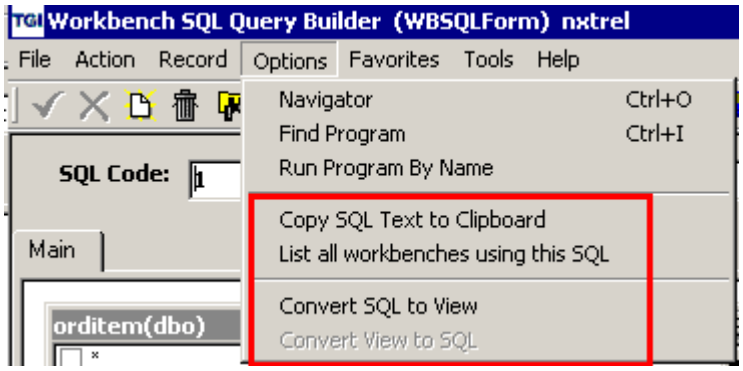
Execute Query Button

From the standard E21 toolbar, you can click the "Execute Query" button for any SQL to see the resulting Dataset.

When viewing the resulting dataset, the Query Building Area and Columns Pane will be hidden and the SQL Text Editor will be disabled. To return to the standard SQL Builder, click the "Show Query Builder" button, which will display in place of the "Execute Query" button while the dataset is visible.



The following options are all accessible via the Workbench SQL Query Builder program Options menu:



Copy SQL Text to Clipboard

Once an SQL is being used by a production workbench, the SQL is locked and the SQL Text Editor is disabled (which disables the ability to cut / paste the text). This option copies the SQL Text to your windows clipboard so you may paste it to other SQL programs / text editors.

List All Workbenches Using This SQL

Just like it sounds, this option displays a list of all workbenches using the currently displayed SQL.

Convert SQL to View

Most issues users have with the E21 workbenches stem from the inherent problem that the Base E21 inheritance forms have dealing with multi-table selects. 99% of the time, the fix for these problems usually revolves around converting these multi-table SQL's into views, and then in turn using the View in the SQL Builder. To save time and so that non-programmers can do this easily as well, we have automated this process in the workbench SQL builder.

To convert an SQL into a view within the E21 Workbench SQL Builder:

1) Select the Options -> Convert SQL to View menu option
2) The system will do some basic validation to see if your SQL is valid for a view. If not, it will tell you to make changes.
3) You will be prompted to enter a valid view name (alphanumeric, can't start with a number, etc).
4) The system will give a warning dialog that it is going to build the view and replace your SQL with a select from your new View instead. Click Yes to continue.
5) Finally, the system will replace your "Main Table" with the new view and also rebuild the SQL appropriately. You can then add any order by statements back to the View sql

Convert View to SQL

When most users create database Views, they dont end up keeping their scripts, and if they ever need to update the view they need to get a DBA to extract the SQL from the database, or start from scratch. To save time and so that non-programmers can do this easily as well, we have added the ability to convert Views back to their original SQL's for easy updating.

Note: Typically the queries should then be converted BACK to views once you are done making changes to the SQL for the same reason they were converted to view in the first place.

To convert a database View back to its original SQL within the E21 Workbench SQL Builder:

1) Select the Options -> Convert View to SQL menu option.
2) Verify that you understand that the SQL will be replaced with the original SQL statement and any changes you make to the Views SQL (adding an Order By) will be lost.
3) Select the table you will use as the "Main Table" for the E21 inheritance forms
4) That's it. Re-add any Order By's if necessary.