

**DAOA**

By Ron Stark

Advantzware, Inc.

Dynamic Advantzware Open Analytics: Lookup Developer’s Guide

Table of Contents

[Dynamic Subject Lookup with Pre-Populated Parameters 3](#_Toc39153670)

[ND3 Dynamic Parameter Builder 4](#_Toc39153671)

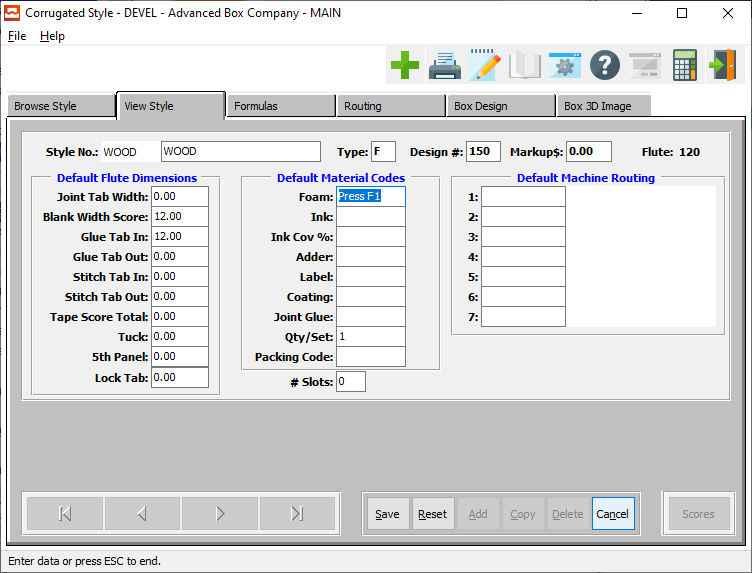
[ND2 Dynamic Subject Builder 7](#_Toc39153672)

[EB3 Corrugated Styles 24](#_Toc39153673)

[Dynamic Subject Lookup within a Help Trigger 29](#_Toc39153674)

[Dynamic Subject Lookup via applhelp.p 31](#_Toc39153675)

Dynamic Subject Lookup with Pre-Populated Parameters This guide will provide instruction on how to build Industry Type Dynamic Parameter, which will be used in creating a Dynamic Lookup for use in EB3 Corrugated Styles when it’s Industry Type is Foam.



The lookup has a query of:

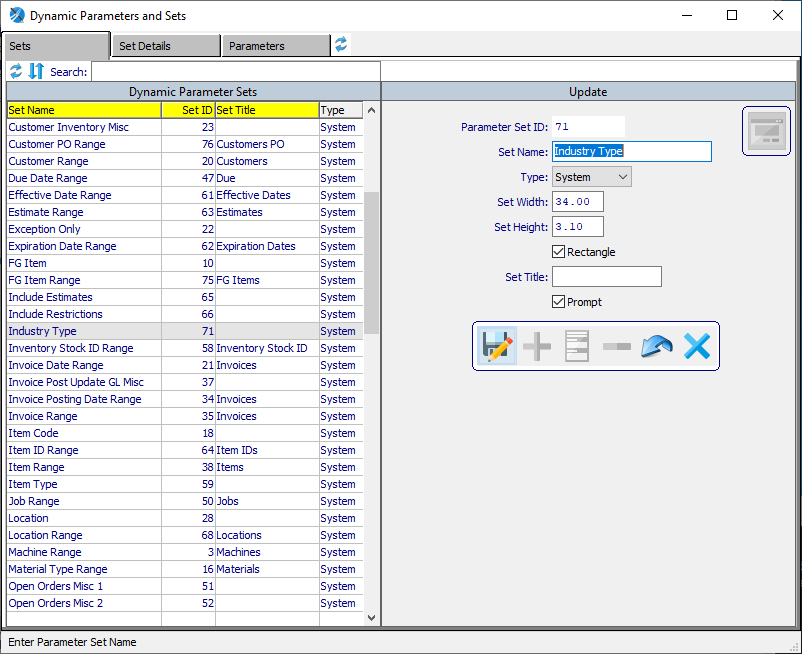
FOR EACH item NO-LOCK WHERE item.company EQ [[company]] AND (item.industry EQ [[IndustryType]] OR [[IndustryType]] EQ “All”) AND item.mat-type GE “1” AND item.mat-type LE “4”.

[[ … ]] bracketed fields indicates a parameter value, in this case the query will need Parameters for Company and Industry Type. The big challenge is populating these parameters with values without presenting a parameter screen to the end user, but also building the lookup with a parameter screen attached to the lookup subject query outlined above.

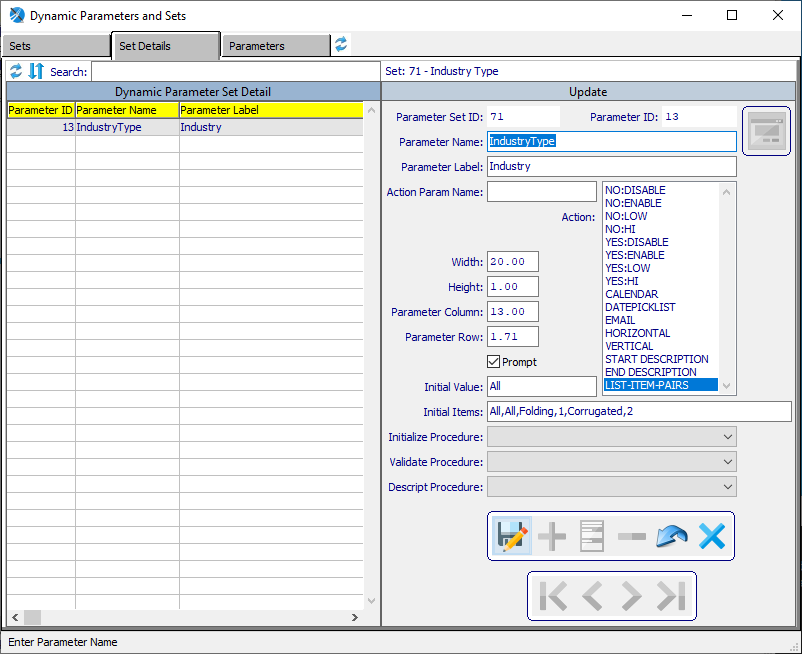
Here are the outlined steps needed to build this Dynamic Lookup.

1. Create and build the needed parameters, in this case Company and Industry Type.
2. Create Dynamic Subject Lookup.
3. Associate step 1 parameters to the Dynamic Subject Lookup.
4. Select query table.
5. Set WHERE statement using the above outlined query.
6. Add Columns which will render in the lookup browser grid.
7. Set the initial values of the parameters based on the selected record in EB3.
8. Code the help trigger in EB3 to run the Dynamic Subject Lookup.

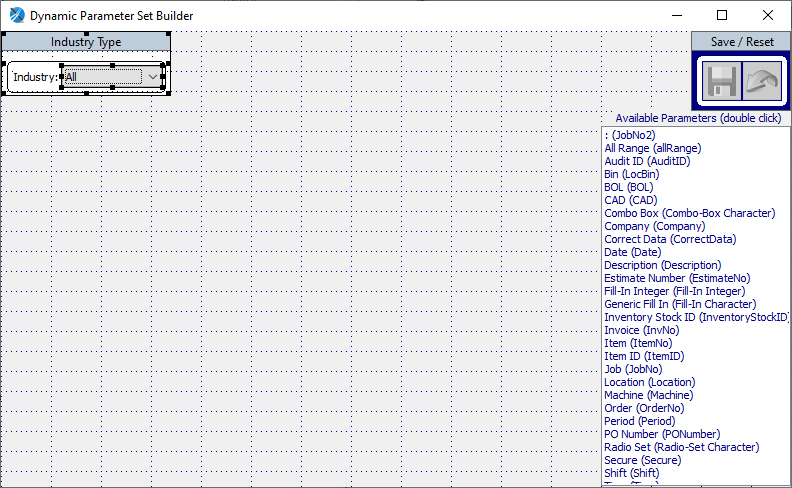
ND3 Dynamic Parameter Builder is the module used to build parameters and group one or more parameters to create a parameter set.



1. Create a new Parameter Set by clicking the Plus icon.
2. Enter a Set Name and ensure Type is set to System, which causes the Parameter Set ID to auto generate a value below 5000. If set to User, the ID will be 5000+. This allows client users to create their own parameters which won’t be overwritten during an upgrade.
3. Save by clicking the Pencil/Floppy Disk icon.
4. Click the Set Details tab.

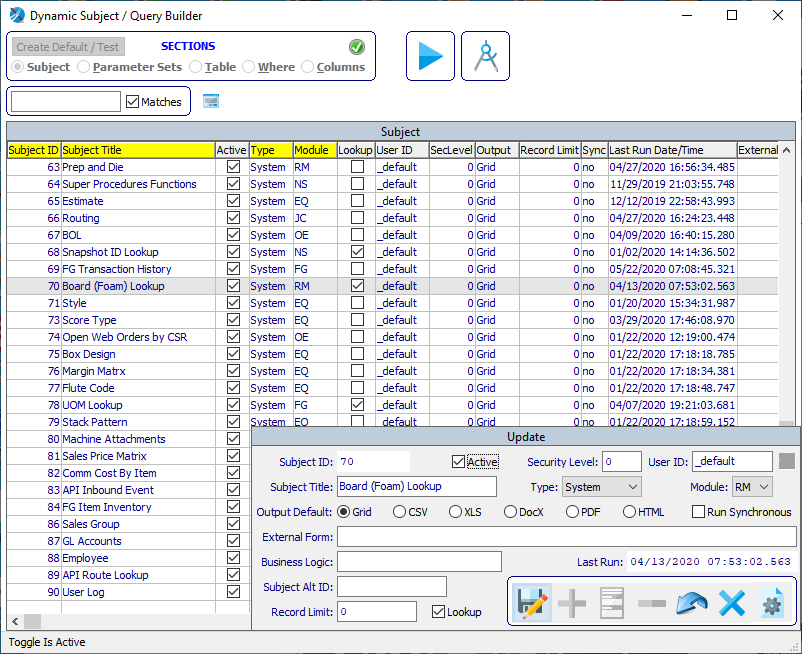


1. A Company parameter already exists, so no need to create, but no Industry Type parameter exists.
2. Create a new Parameter Set Detail by clicking the Plus icon.
3. Enter a Parameter Name (IndustryType), Label Industry), make it a LIST-ITEM-PAIRS, Initial Value (All) and Initial Items (All,All,Folding,1,Corrugated,2).
4. Save by clicking the Pencil/Floppy Disk icon.
5. The Parameter Set Builder will auto render and allow changes to size of the parameter object and it frame size.

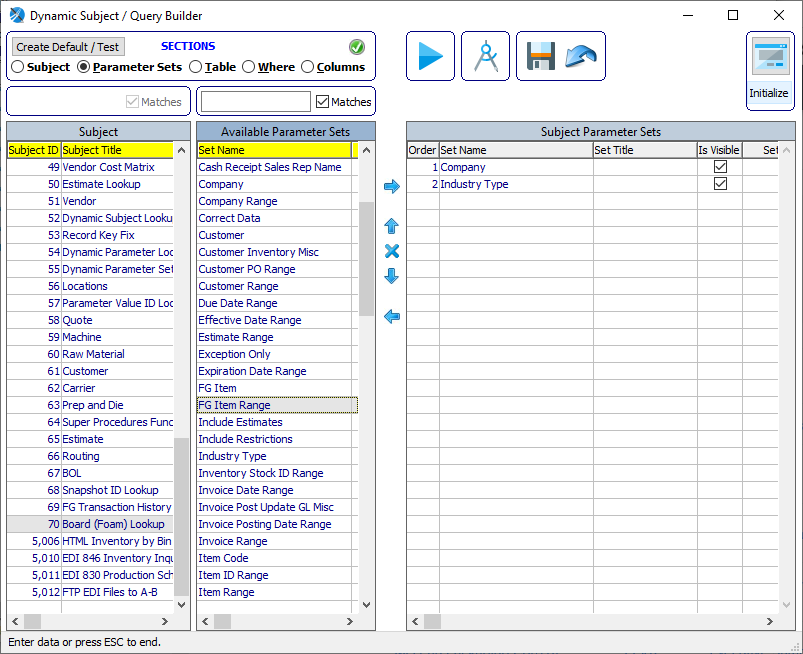


Once all needed parameters and parameter sets are built, run NS2 to build the Dynamic Subject Lookup.

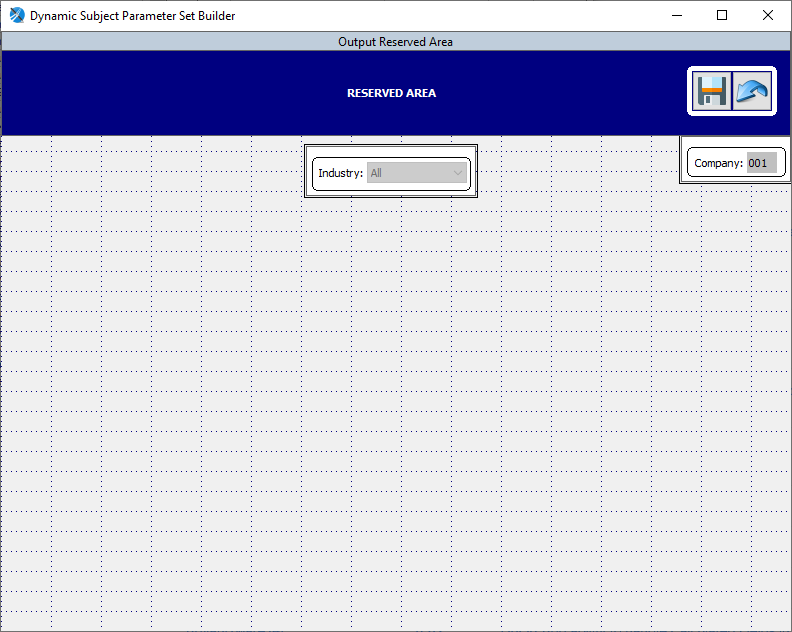
ND2 Dynamic Subject Builder is the module used to build Dynamic Subjects.



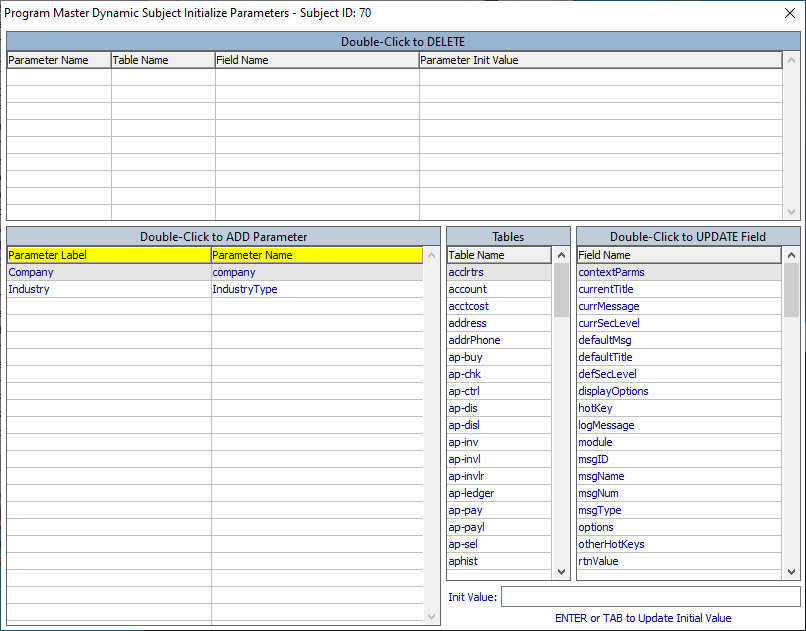
1. Create a subject by clicking the Plus icon. Generally the Security Level should be set to 0 (zero) so it is accessible to all users.
2. Enter a Subject Title (Board (Form) Lookup (note: the keywork Lookup must appear in the Subject Title for the Dynamic framework to recognize it as a Lookup – future enhancements will contain a schema change to indicate a subject is a lookup).
3. Select Type as System so when a Save is performed it will auto generate a Subject ID below 5000, if set to User, the Subject ID will be 5000+. This allows client users to create their own subjects which won’t be overwritten during an upgrade.
4. Select the appropriate Module value.
5. Click to checkoff this Subject is set as a Lookup.
6. Save by clicking the Pencil/Floppy Disk icon.
7. Close the Subject viewer.
8. Select section Parameter Sets.

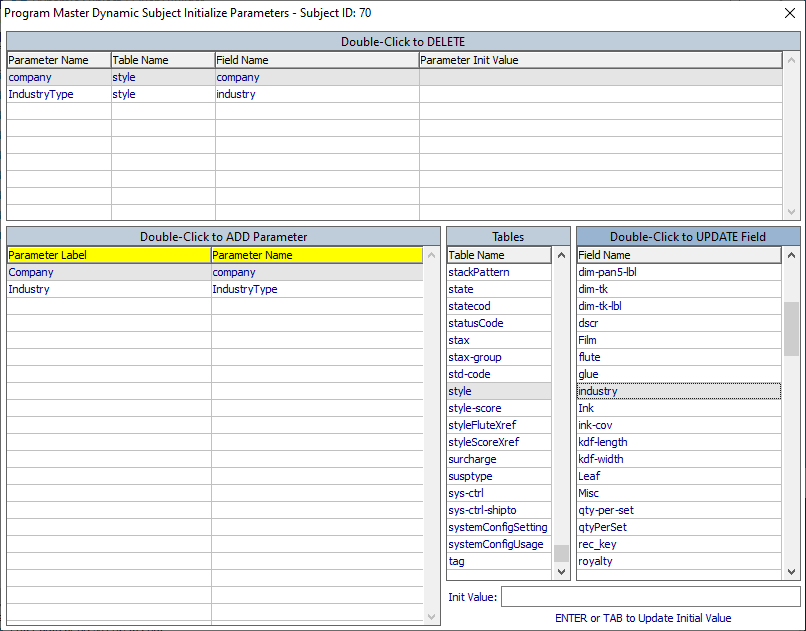


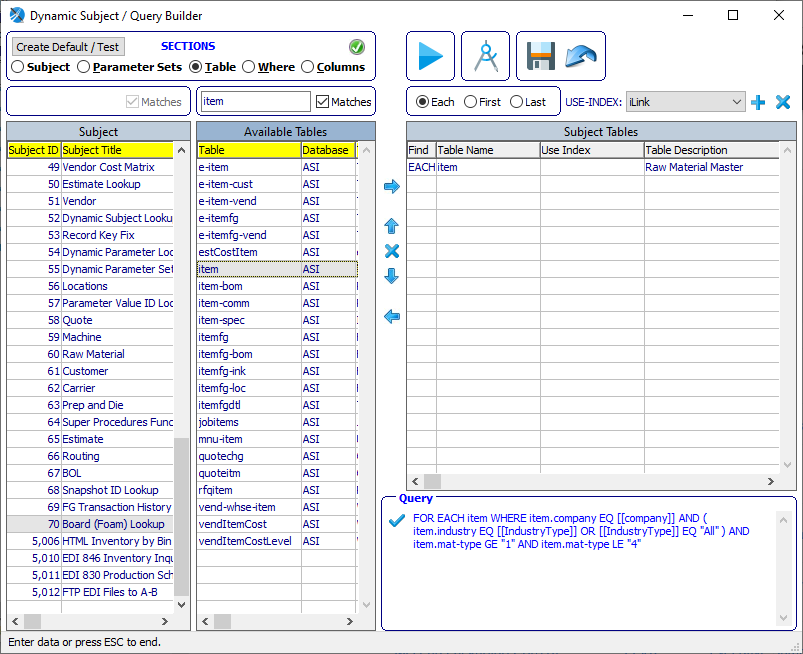
1. Locate Set Name “Company” either by highlighting it and click the right pointing arrow or simply double click “Company” and it will auto add itself to Subject Parameter Sets.
2. Locate Industry Type and add it also.
3. When any additions and/or deletes occur, the Save and Reset icons will enable.
4. Save by clicking the floppy disk icon.
5. Click the Parameter Set Designer in the upper right corner, just above the work Initialize.



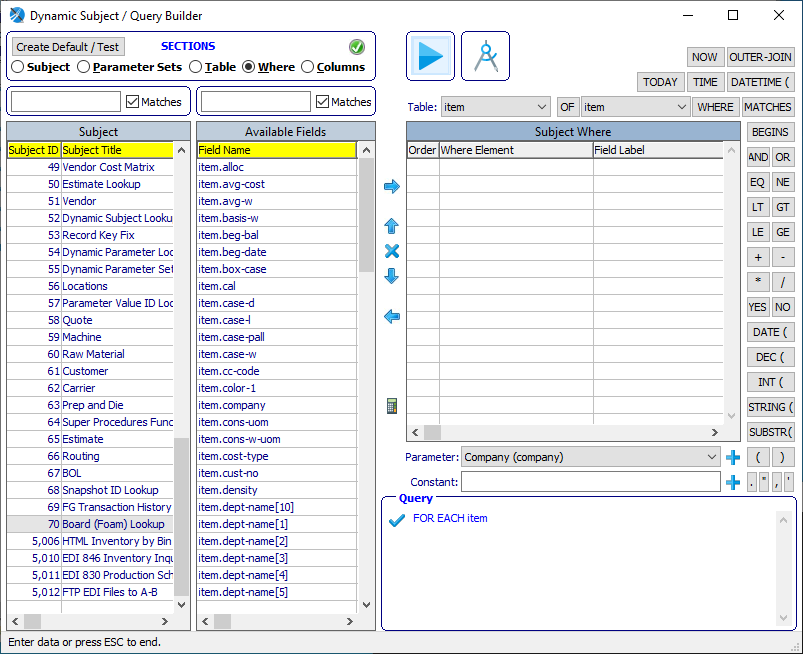
1. Drag and Drop the Parameter Sets to the desired screen location.
2. The Company parameter is none interactive (meaning it doesn’t enable, it initializes to the current Company value). Following standard screen positioning of existing parameter sets, is the place it in the upper right corner just below the RESERVED AREA.
3. Save by clicking the floppy disk.
4. Click on Initialize which is directly below the Parameter Set Builder icon.



1. The major challenge using a Dynamic Subject Lookup that needs to pre-populate the parameter values used in the query, is linking the parameters with the selected record, in this case within EB3 the lookup with need the Company and Industry Type values from the style table.
2. Locate and double-click the parameter Company.
3. Locate the table style within the Tables browser which then displays the style table fields to the right.
4. Locate and double-click the field company.
5. Locate and double-click the parameter Industry.
6. Locate the table style within the Tables browser which then displays the style table fields to the right.
7. Locate and double-click the field industry.
8. 
9. Closing this module performs an auto Save.
10. Select section Table.

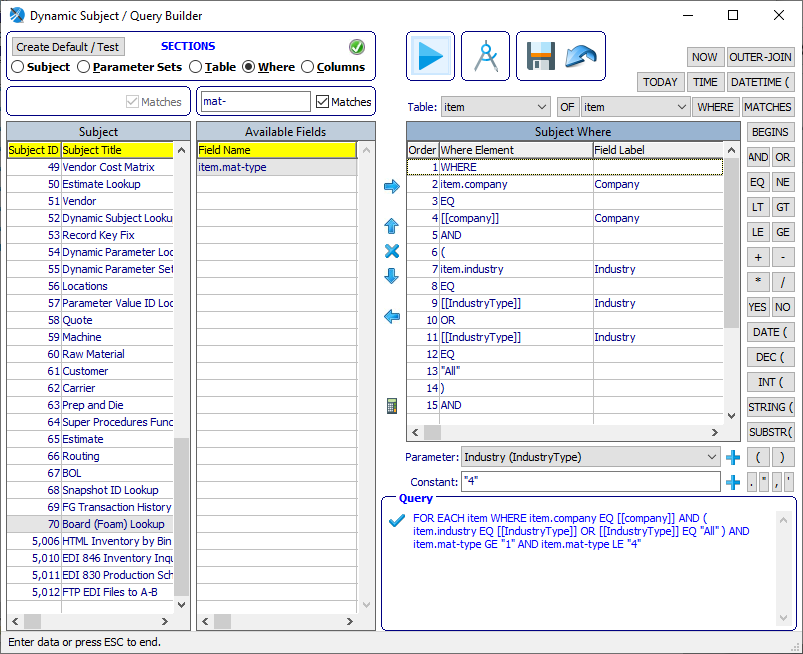


1. Locate table “item” and add it to Subject Tables by using the right arrow or double clicking.
2. Save by clicking the floppy disk.
3. Select section Where.

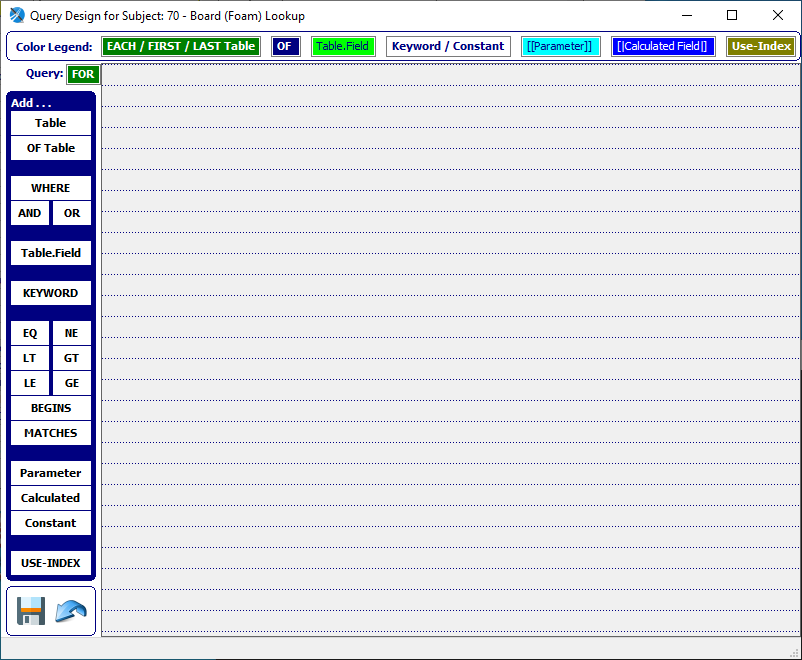


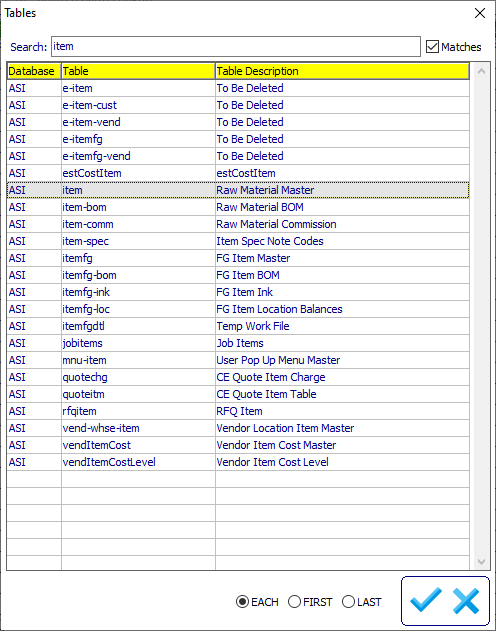
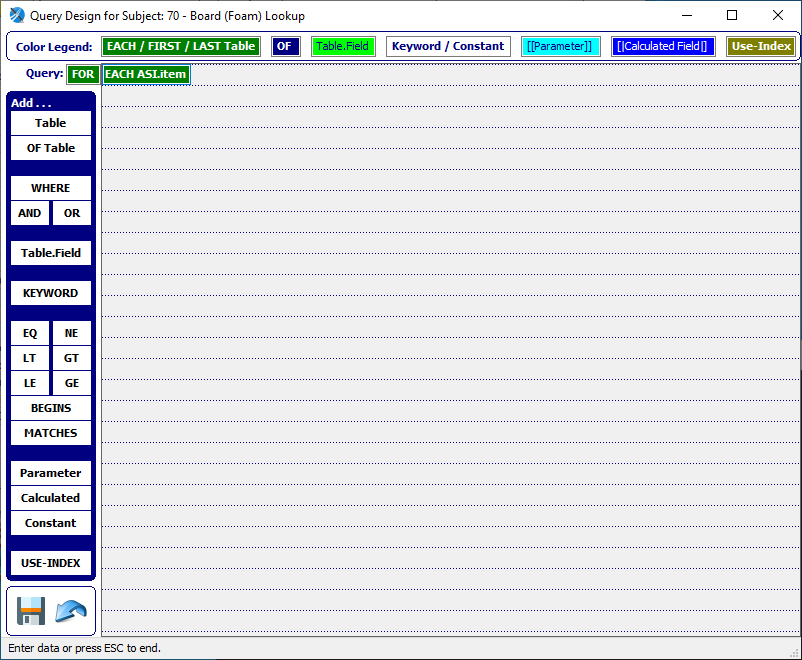
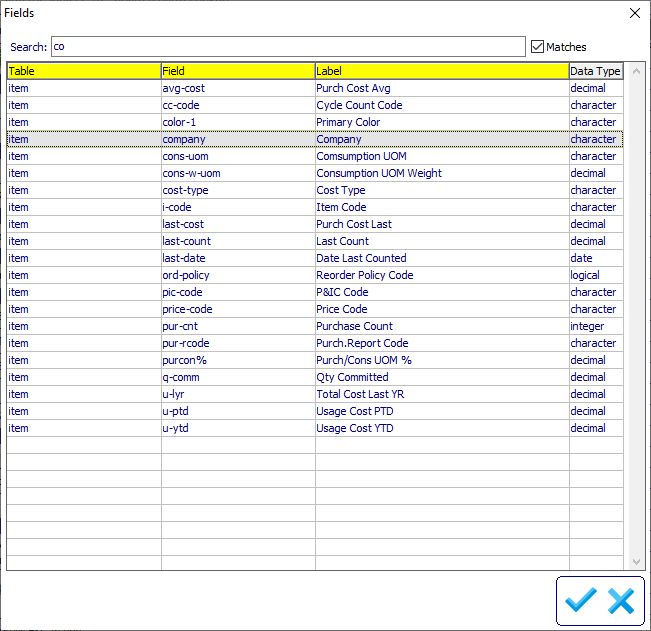
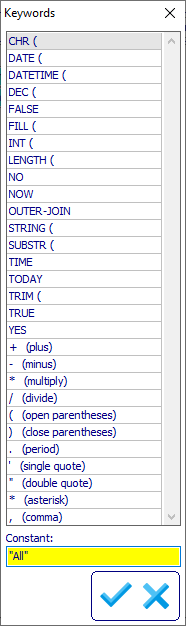
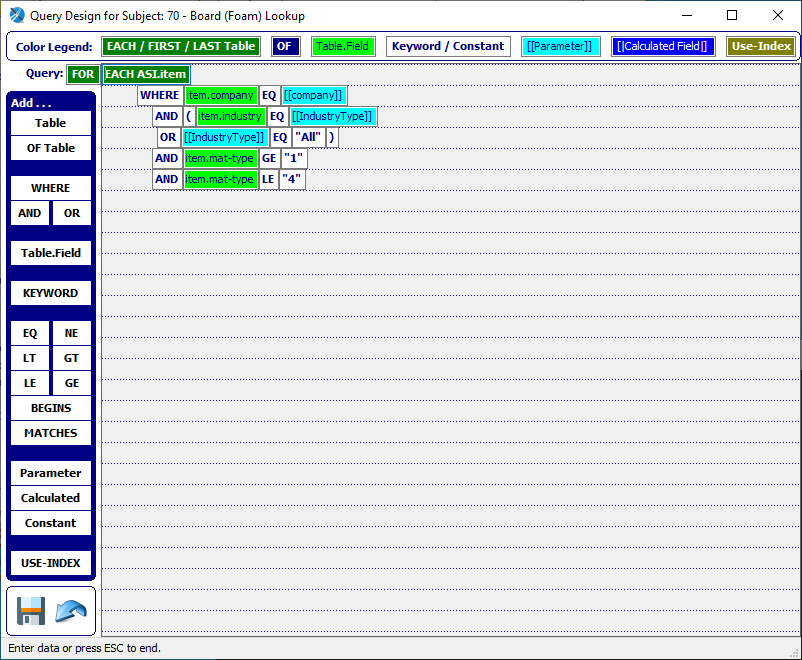
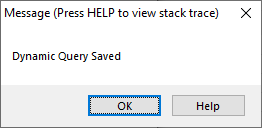
Next step is to build the WHERE phrase. WHERE item.company EQ [[company]] AND (item.industry EQ [[IndustryType]] OR [[IndustryType]] EQ “All”) AND item.mat-type GE “1” AND item.mat-type LE “4”.

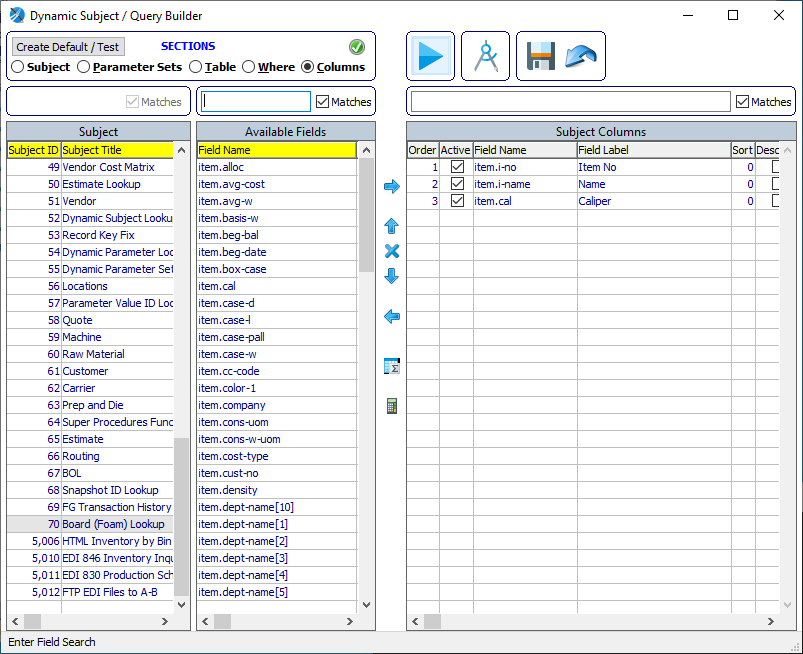
1. Click [WHERE] button to add to the Subject Where.
2. Add field item.company from Available Fields by highlighting and clicking right arrow or double-click the field.
3. Click [EQ] button.
4. Select Parameter “Company (company)” and click the plus icon to the right of the Parameter combo-box.
5. Click [AND] button.
6. Click [(] open parenthesis button.
7. Select field item.industry to add.
8. Click [EQ] button.
9. Select Parameter “Industry (Industry Type)” and click parameter plus icon to add.
10. Click [OR] button.
11. Select Parameter “Industry (Industry Type)” and click parameter plus icon to add.
12. Click [EQ] button.
13. Enter “All” in the Constant fill-in field and click Constant plus icon to the right of Constant.
14. Click [)] close parenthesis button.
15. Click [AND] button.
16. Add field item.mat-type to add.
17. Click [GE] button.
18. Enter “1” as a Constant and add.
19. Click [AND] button.
20. Add field item.mat-type to add.
21. Click [LE] button
22. Enter “4” as a Constant and add.
23. Save by click the floppy disk icon.

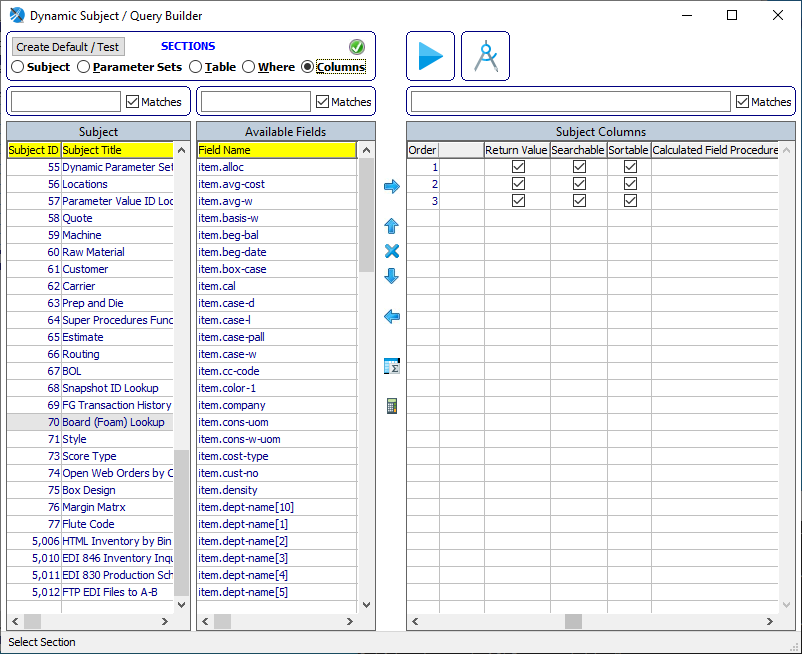


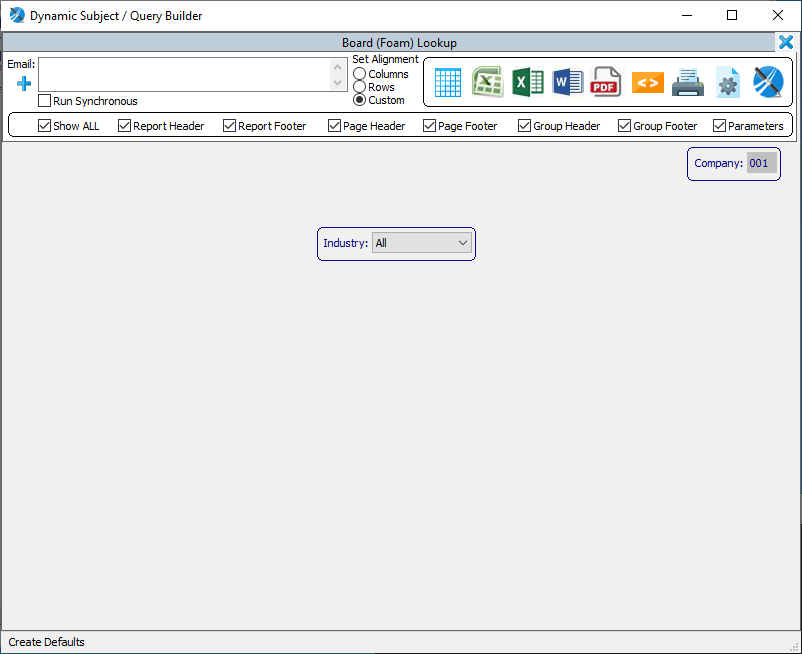
Another method to create the query is to use the Query Design by clicking the Architect Compass icon.



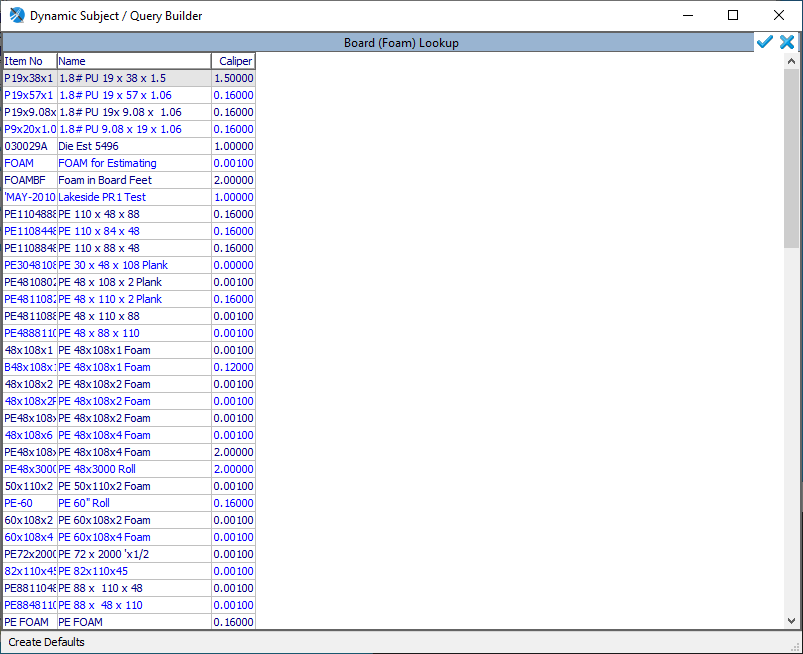
1. Click [Table] button and position the mouse cursor over the design grid, click the mouse and locate table “item”.
2. 
3. Click the checkmark icon to select. It will auto position itself on the design grid.
4. 
5. Click [WHERE] button, position mouse over the design grid and click the mouse. It will auto position itself on the design grid.
6. Click [Table.Field] button, position mouse over the design grid and click the mouse. Locate the field item.company.
7. 
8. Click the checkmark icon to select. It will auto position itself on the design grid.
9. Click [EQ] button, position over grid, mouse click to add.
10. Click [Parameter] button, locate Company parameter, click checkmark icon to add.
11. Click [AND] button, position, mouse click.
12. Click [KEYWORD] button, position, mouse click, select ( open parenthesis, click checkmark icon to add.
13. Click [Table.Field] button, position, mouse click, locate field item.industry, click checkmark to add.
14. Click [EQ] button, position, mouse click.
15. Click [Parameter] button, position, mouse click, locate parameter IndustryType, click checkmark to add.
16. Click [OR] button, position, mouse click.
17. Click [Parameter] button, position, mouse click, locate parameter IndustryType, click checkmark to add.
18. Click [EQ] button, position, mouse click.
19. Click [Constant] button, position, mouse click, enter “All” into yellow highlighted area.
20. 
21. Click checkmark to add.
22. Click [KEYWORD] button, position, mouse click, select ) close parenthesis, click checkmark icon to add.
23. Click [AND] button, position, mouse click.
24. Click [Table.Field] button, position, mouse click, locate field item.mat-type, click checkmark to add.
25. Click [GE] button, position, mouse click.
26. Click [Constant] button, position, mouse click, enter “1” into yellow highlighted area, click checkmark to add.
27. Click [AND] button, position, mouse click.
28. Click [Table.Field] button, position, mouse click, locate field item.mat-type, click checkmark to add.
29. Click [LE] button, position, mouse click.
30. Click [Constant] button, position, mouse click, enter “4” into yellow highlighted area, click checkmark to add.
31. 
32. Save by clicking floppy disk icon.
33. 
34. Close the Query Design window.
35. Select section Columns.



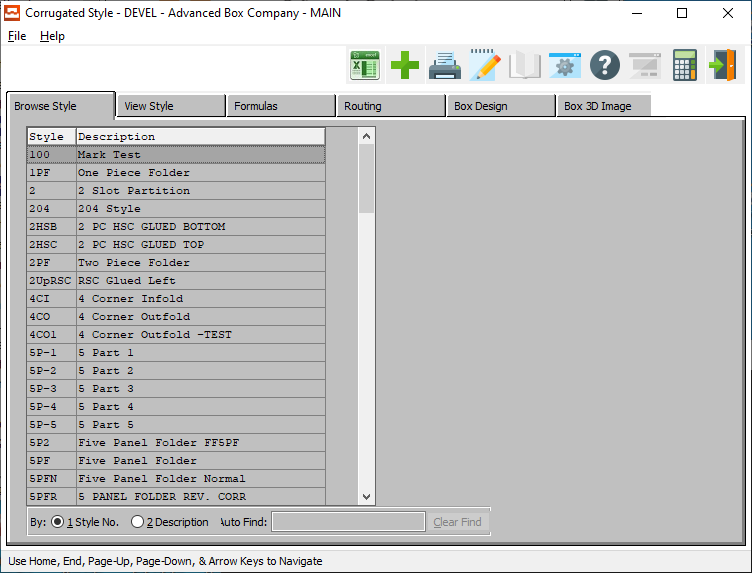
1. Add field item.i-no from Available Fields by highlighting and clicking right arrow or double-click the field.
2. Add field item.i-name.
3. Add field item.cal.
4. 
5. Checkoff Return Value if a column’s value is to be returned to the calling program.
6. Checkoff Searchable if a column is Searchable within the Lookup module.
7. Checkoff Sortable if a column is Sortable within the Lookup module.
8. Save by clicking the floppy icon.
9. Click [Create Default / Test] button.



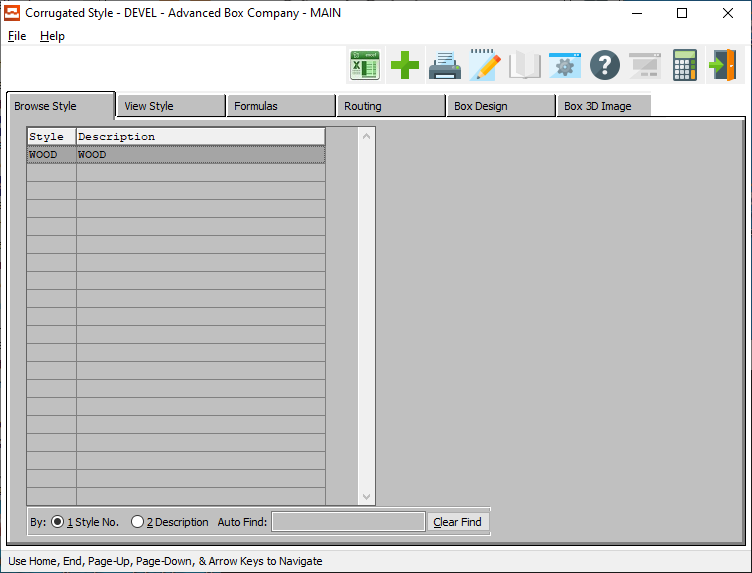
Click the Grid icon which generates the Dynamic Subject Lookup results and also creates the \_default DynParamValue record which is the basis for users running custom versions of the Dynamic Subject and also the ability to schedule them in the Task Monitor.



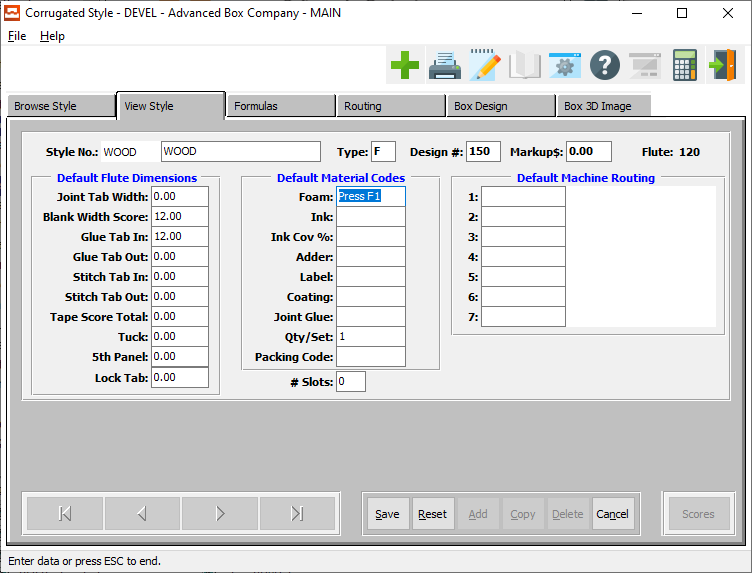
EB3 Corrugated Styles is the module which the above Dynamic Subject Lookup will be attached.



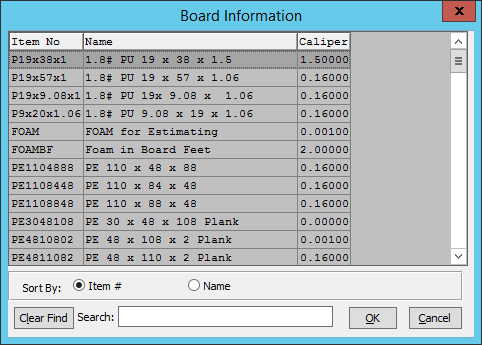
Filter on “Wood” to locate a Style that has a Style Type of F for Foam.



Double click the highlighted record or click the View Style tab and click the [Update] button.



Navigate to the field Foam and Press F1. This will render the currently associated lookup.



Note: currently EB3 Viewer (viewers/stylec.w) fails to populate the Foam field with the selected lookup value.

Because the original design of this viewer, a HELP TRIGGER was attached to the FRAME, so we cannot rely of attaching a Dynamic Subject Lookup via using applhelp.p (this is discussed as an option at the end of this document), the next phase is to modify “viewers/stylec.w” to attach the Dynamic Subject Lookup #70 Board (Foam) Lookup replacing the above lookup.

Replace the following code:

IF AVAIL style AND style.type:screen-value = "f" THEN /\* foam \*/

RUN windows/l-boardf.w (style.company,lv-ind,ls-cur-val,OUTPUT char-val).

With the following code:

IF AVAILABLE style AND style.type:SCREEN-VALUE EQ "f" THEN DO: /\* foam \*/

RUN AOA/dynLookupSetParam.p (70, ROWID(style), OUTPUT char-val).

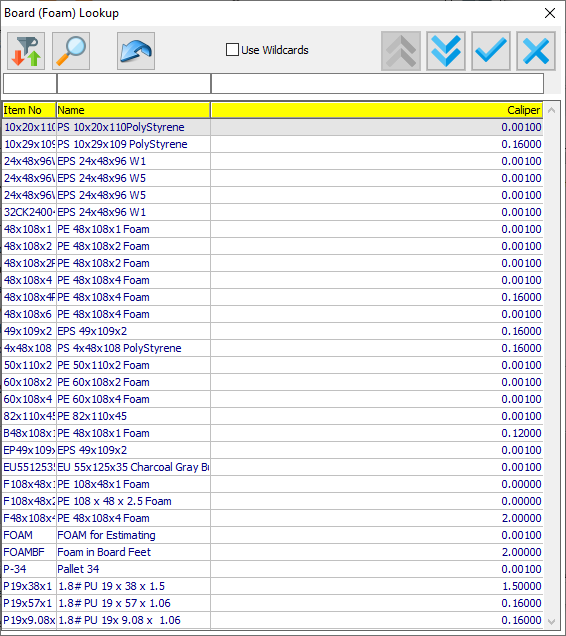
style.material[1]:SCREEN-VALUE = DYNAMIC-FUNCTION("sfDynLookupValue", "i-no", char-val).

APPLY "ENTRY":U TO style.material[1].

END. /\* if foam \*/

A few items of note here that need discussing. First, AOA/dynLookupSetParam.p needs two input parameters and an output parameter. First INPUT parameter is the Subject ID, which in this case is 70. Second INPUT parameter is the ROWID of the style record. The reason for this is within the dynamic framework a quick query is created to obtain the style record using the ROWID value, thus providing the values needed to pre-populate the parameters used in the Dynamic Subject Lookup query. The framework logic needs to know the table and the ROWID to perform this functionality. The OUTPUT parameter is a pairs list pipe delimited string of all the fields displayed in the lookup. In our example here it would contain “i-no|<value>|i-name|<value>|cal|<value”. To extract the desired value for EB3 utilize the DYNAMIC-FUNCTION(“sfDynLookupValue”, “<field>”, <pairs-list>).

Once this code change is in place, run EB3 again, filter on WOOD to locate a record with Style Type F for Foam. Perform an [UPDATE], navigate to the Foam field and Press F1 to render the Dynamic Subject Lookup.



Dynamic Subject Lookup within a Help Trigger

Listed below is a current HELP trigger for utilizing JKTech’s system/openLookup.p module. Note how as a developer you would need to know in what position within the cFoundValue pipe delimited variable the desired fields to extract exist. This would require the developer to open system/openLookup.p and search the CASE statement for “job-no” and make note of the fields being returned.

ON HELP OF ls-jobno IN FRAME F-Main

DO:

DEFINE VARIABLE cFieldsValue AS CHARACTER NO-UNDO.

DEFINE VARIABLE cFoundValue AS CHARACTER NO-UNDO.

DEFINE VARIABLE recFoundRecID AS RECID NO-UNDO.

RUN system/openlookup.p (

INPUT ipcCompany,

INPUT "job-no", /\* Lookup ID \*/

INPUT 0, /\* Subject ID \*/

INPUT "", /\* User ID \*/

INPUT 0, /\* Param Value ID \*/

OUTPUT cFieldsValue,

OUTPUT cFoundValue,

OUTPUT recFoundRecID

).

IF cFoundValue NE "" THEN DO:

SELF:SCREEN-VALUE = cFoundValue.

APPLY "LEAVE":U TO SELF.

ASSIGN

cb-jobno2:SCREEN-VALUE = IF NUM-ENTRIES(cFieldsValue,"|") GE 6 AND

INDEX(cb-jobno2:LIST-ITEMS, STRING(INTEGER(ENTRY(6,cFieldsValue,"|")),"99")) GT 0 THEN

ENTRY(6,cFieldsValue,"|")

ELSE

ENTRY(1,cb-jobno2:LIST-ITEMS)

cb-formno:SCREEN-VALUE = IF NUM-ENTRIES(cFieldsValue,"|") GE 8 AND

INDEX(cb-formno:LIST-ITEMS, STRING(INTEGER(ENTRY(8,cFieldsValue,"|")),"99")) GT 0 THEN

ENTRY(8,cFieldsValue,"|")

ELSE

ENTRY(1,cb-formno:LIST-ITEMS)

cb-blankno:SCREEN-VALUE = IF NUM-ENTRIES(cFieldsValue,"|") GE 10 AND

INDEX(cb-blankno:LIST-ITEMS, STRING(INTEGER(ENTRY(10,cFieldsValue,"|")),"99")) GT 0 THEN

ENTRY(10,cFieldsValue,"|")

ELSE

ENTRY(1,cb-blankno:LIST-ITEMS)

NO-ERROR.

END.

END.

With a Dynamic Subject Lookup, things are a lot more simplified. The code changes from above are as follows:

ON HELP OF ls-jobno IN FRAME F-Main

DO:

DEFINE VARIABLE cFieldsValue AS CHARACTER NO-UNDO.

DEFINE VARIABLE cFoundValue AS CHARACTER NO-UNDO.

DEFINE VARIABLE recFoundRecID AS RECID NO-UNDO.

RUN system/openLookup.p (

INPUT ipcCompany,

INPUT "", /\* Lookup ID \*/

INPUT 3, /\* Subject ID \*/

INPUT "", /\* User ID \*/

INPUT 0, /\* Param Value ID \*/

OUTPUT cFieldsValue,

OUTPUT cFoundValue,

OUTPUT recFoundRecID

).

IF cFoundValue NE "" THEN DO:

SELF:SCREEN-VALUE = cFoundValue.

APPLY "LEAVE":U TO SELF.

ASSIGN

cb-jobno2:SCREEN-VALUE = DYNAMIC-FUNCTION("sfDynLookupValue", "job-no2", cFieldsValue)

cb-formno:SCREEN-VALUE = DYNAMIC-FUNCTION("sfDynLookupValue", "frm", cFieldsValue)

cb-blankno:SCREEN-VALUE = DYNAMIC-FUNCTION("sfDynLookupValue", "blank-no", cFieldsValue)

.

END.

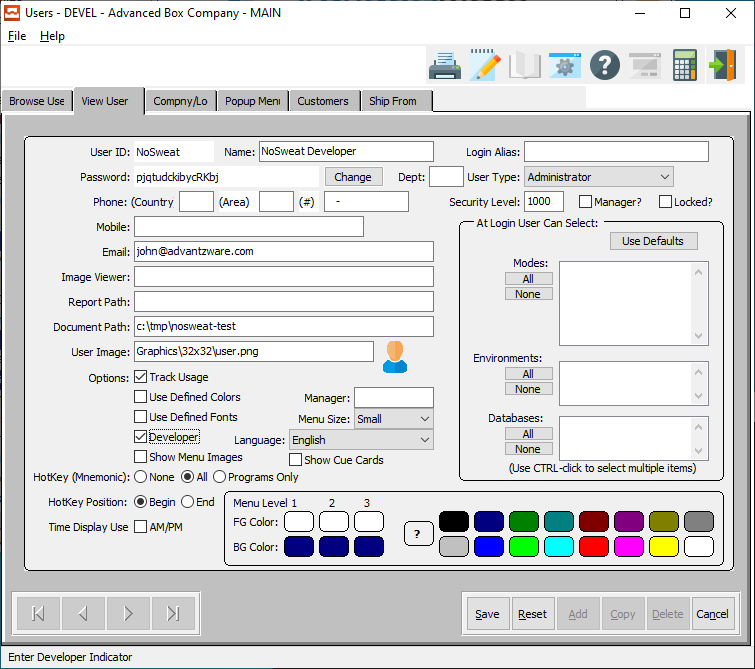
END.

Note how there is no need to know the pipe delimited position of the desired values to extract when needing more than one value from a Dynamic Subject Lookup.

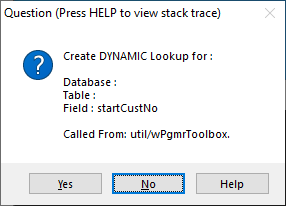
Dynamic Subject Lookup via applhelp.p

When no HELP TRIGGER for the Frame and/or the individual screen object, whether it be a database field or a locally defined object, a Dynamic Subject Lookup can be dynamically added via applhelp.p. Applhelp.p is a memory resident program incorporated with the Progress Run Time framework. When a HELP TRIGGER is invoked, such as F1 or F3, Progress logic will first look for a localized HELP TRIGGER. Finding none, the logic will next attempt to locate applhelp.p. If it does not exist, it simply ignores the HELP TRIGGER event. Applhelp.p does exist with the ASI framework and will be invoked when no local HELP TRIGGER exists.

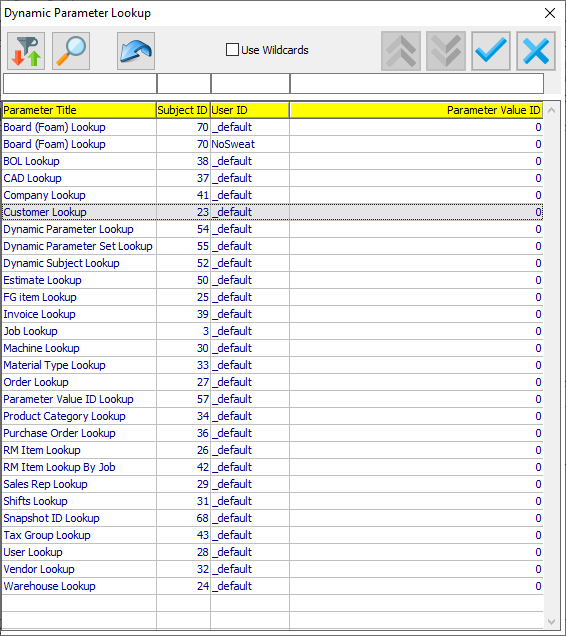
The process will check if the USERID(“ASI”) against the NU3 Users table. If the Developer attribute is set to YES, applhelp.p will prompt the developer with a series of Questions allowing for the association of a Lookup to the screen object from which the HELP TRIGGER event was invoked.



Navigate to a screen object which needs a Dynamic Subject Lookup and Press F1 and the following prompt will render.

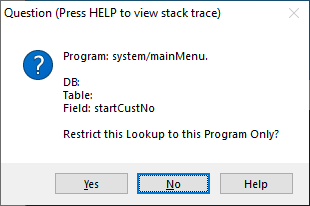


When Create DYNAMIC Lookup is YES:



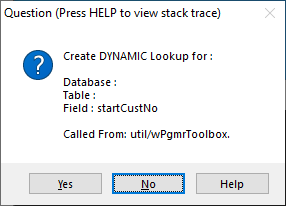
Select the desired DYNAMIC Lookup.

After selecting a DYNAMIC Lookup, select if the DYNAMIC Lookup is to be associated with only this module of the screen object.

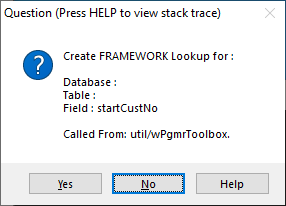


When Restrict this Lookup to this Program Only is YES: will associate the DYNAMIC Lookup to this screen object and only within the module with which the screen object appears.

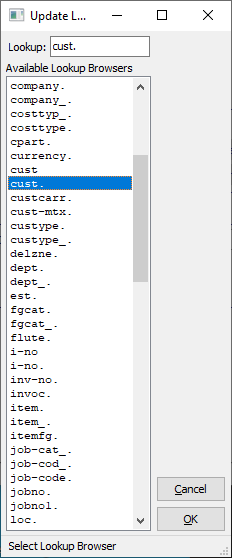
When Restrict this Lookup to this Program Only is NO: will associate the DYNAMIC Lookup to any similar named screen object regardless of which module is appears.



When Create DYNAMIC Lookup is NO:

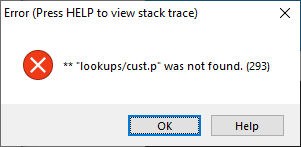


When Create FRAMEWORK Lookup is YES:

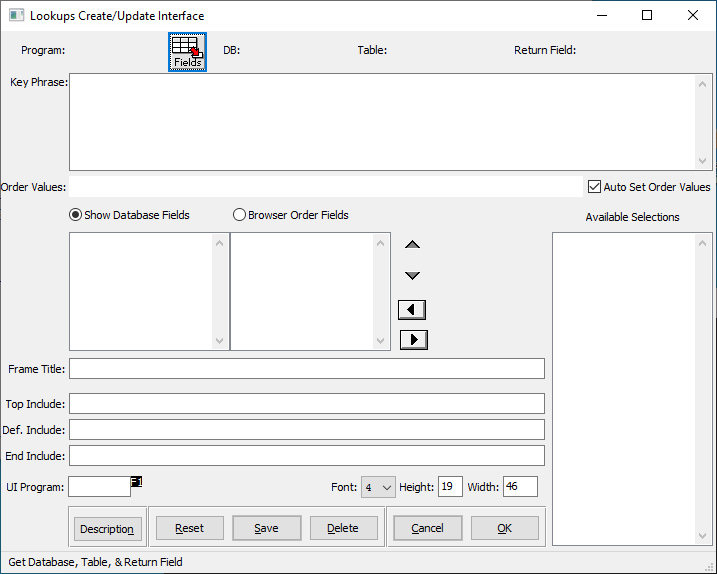


Select the desired FRAMEWORK Lookup.

Because the source code for lookup/cust.p is not part of the PROPATH, the following error will occur.

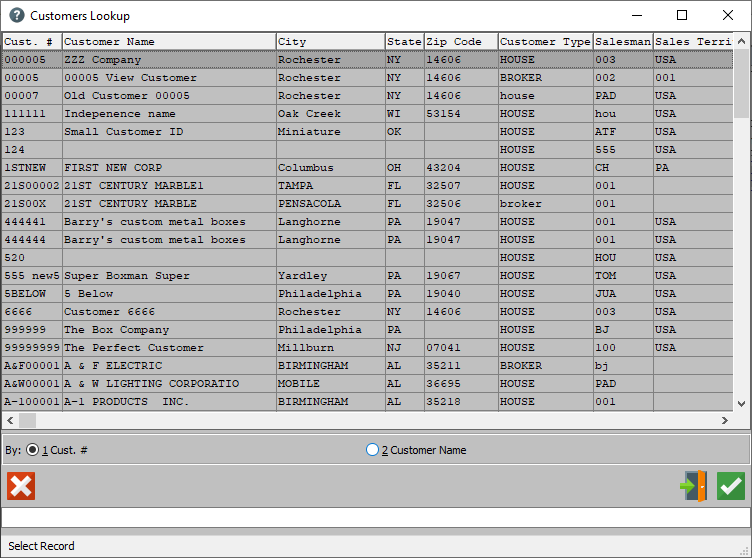


Click OK and the following module will render the Create/Update FRAMEWORK Lookup:

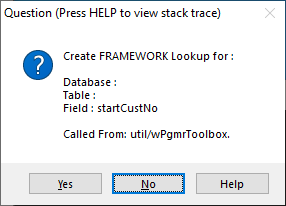


This module allows a developer to create the p-code for FRAMEWORK Lookup. However, because of our current folder structure coupled with our PROPATH settings, this is not viable without further training from Ron Stark. For purposes of this document, this phase is not documented.

Simply click CANCEL, and the r-code version will render.

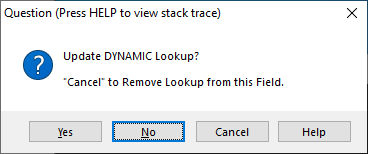


Select a record and/or close the Lookup and this FRAMEWORK Lookup is now associated with the screen object.



When Create FRAMEWORK Lookup is NO: Exit, no further prompts, no Lookup is associated with the screen object.

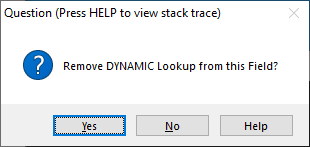
If a DYNAMIC Lookup association already exists, developers will be presented with the following prompt:



When Update DYNAMIC Lookup is YES: will render the DYNAMIC Parameter Lookup module shown above.

When Update DYNAMIC Lookup is NO: will simply run the Lookup.

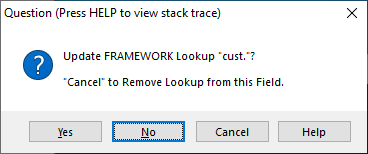
When Update DYNAMIC Lookup is CANCEL:



When Remove DYNAMIC Lookup is YES: will remove the DYNAMIC Lookup from the screen object.

When Remove DYNAMIC Lookup is NO: will merely exit with no changes.

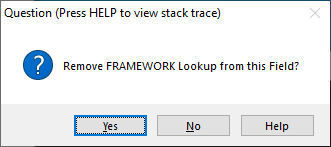
If a FRAMEWORK Lookup association already exists, developers will be presented with the following prompt:



When Update FRAMEWORK Lookup is YES: will render the Create/Update FRAMEWORK Lookup module shown above.

When Update FRAMEWORK Lookup is NO: will simply run the Lookup.

When Update FRAMEWORK Lookup is CANCEL:



When Remove FRAMEWORK Lookup is YES: will remove the FRAMEWORK Lookup from the screen object.

When Remove FRAMEWORK Lookup is NO: will merely exit with no changes.