

Document #: 1206

Last Revised Date: 12/14/98

Last Revised by: Rawle Gibson

How To Use \$MENU\$ Pattern Matching With A Query Menu

Action Request System Version 3.2.1

Title:

How To Use \$MENU\$ Pattern Matching With A Query Menu

Category:

Use knowledge-input category menu

Keyword(s)

query, menu, pattern, match, \$MENU\$

Problem: (Short Description)

How do I use the \$MENU\$ pattern match on a Query Menu?

Problem: (Long Description)

I have a Query Menu attached to a character field that uses the \$MENU\$ Pattern matching and I am receiving an ARERR 306 'Value does not fall within the limits specified for the field' whenever I select a value from the Menu.

OS Version:

All

Reference:

RxPress Volume 5, Number 3, 1997

Solution Overview:

\$MENU\$ AND DYNAMIC MENUS

Character menus are an important part of many AR System solutions. The ability to specify a pattern using the keyword \$MENU\$ allows you to perform an integrity check to ensure that the data entered matches one of the values allowed. However, when the menu in question is a query or SQL style menu that references a field on the current screen, that is, a dynamic menu, the system does not support the \$MENU\$ pattern. The following procedure focuses on a method to use filters to build the equivalent of a \$MENU\$ pattern check for a field with a dynamic menu.

** The Goal **

First, we need to take a look at what we are attempting to accomplish. Simply stated, we want to ensure that the value specified is one of the legal values for that field as defined by a query or SQL statement. The one twist on the menu is that it uses data from the current screen to find the set of values allowed.

** The Strategy **

Looking at things from a high vantage point, we are going to perform the same operation as the character menu to see if the value being assigned is one of the values allowed. If not, we are going to return an error message. If it is legal, we will allow the operation to occur.

** The Implementation **

To accomplish our goal, we are going to use a set of three filters. Each of these filters will be described in detail: Validating a Value Against a Dynamic Menu:

- 1. Check the value and mark it as being checked*
- 2. Return an error if there is no matching value*
- 3. Cleanup marking that check occurred*

All of the filters will be tied to the target schema and will be defined to execute on Submit and Modify. The execution order of the filters needs to be set such that they fire in the sequence described in this article (for example: 1, 2, 3 or 500, 501, 502). For discussion purposes, we are loading a field called 'Part' using a menu that is querying a 'Parts-List' schema selecting the 'Name' field in the menu.

** Check the Value and Mark it as Being Checked **

The qualification for this filter will be 'TR.Part' != \$NULL\$. This will cause the filter to fire whenever the value is being set. It contains a single Set Fields action that pulls data from the same schema and server as the character menu. The qualification is the same as the qualification for the character menu with one additional clause: AND 'Name' = \$Part\$ This qualification indicates to select only the one row from the set of matches that is the current value being assigned. For AR System 3.2.1, the no match handling setting should be to set fields to \$NULL\$ and the multiple entries match should be to use the first. (For 2.x, there are no settings, but the system defaults to the equivalent of the choices above.) One field should be assigned a value, 'Part', the field we are checking. It should be assigned as follows: \$Name\$ + "@" The "@" character is really any character that is not present in your menu. It is being added as a marker that the value has been validated. At the end of this step, the field will have either the value being assigned with an "@" appended or just the value "@".

** Return an Error if There Is No Matching Value **

To see if a value was matched or not, you use a second filter. This filter has a qualification of 'TR.Part' = "@". From the first operation, the value of Part will be either the value you entered with "@" added or just the marker "@". It also contains a single action, but in this case, the action is a Message action that returns an Error message and stops the Modify or Submit operation. The qualification will not match if there is no assignment to the field or if the value is valid. At the end of this step, we would have returned an error if there is one and caused the operation to stop. If the value is legal, no action has occurred and we have the value with a trailing "@" character.

** Cleanup Marking That Check Occurred **

Finally, the third filter in the list will remove the tag character that we added to the value in the first step. The qualification for this filter is the same as the first, 'TR.Part' != \$NULL\$. We want this filter to fire every time the first one fires. It contains a single Set Fields action that works from the Current Transaction. The operation is to remove the extra character from the end of the field. This is accomplished by assigning the following value to the field: LEFT(\$Part\$, LENGTH(\$Part\$) - 1) In other words, take the leftmost characters for one byte less than the full length of the value and assign them to the field. This effectively strips the final character, which leaves you with your original value. At the end of this step, we are back where we started with the same value in the field. However, we have validated the field and know that the contents meet our integrity constraints.

** What if it is an SQL Style Menu? **

The same strategy is used for SQL style menus. The only change is to specify the SQL command from the character menu with an extra clause, as in the query in Step 1. The other two steps are identical.

Conclusion:

This strategy will allow you to add integrity constraints to your character menus and to enforce a \$MENU\$ pattern capability on dynamic menus. Using this ability with query style menus or with SQL style menus allows you to close the gap that exists in validating the contents of a field against an associated menu



5924 Stoneridge Dr.
Pleasanton, CA
94588 USA
510.469.4000 tel
510.469.4001 fax
support@remedy.com

Action Request System and AR System are trademarks of Remedy Corporation.
All other trademarks or service marks mentioned in this document are
the properties of their respective companies and owners.