Data Selectors: User Manual

SHUTTLE ENTERPRISE SERVER

What is a Data Selector?

Before we even start looking at what it is, let us impress on you that it is one of the most useful tools given to you, the User, in your SHUTTLE system. When you do Data Queries and / or Reports by yourself, you will be dealing with Files (Data Sources) for which some Data Selectors will already be defined. However, you may well wish to construct some of your own when there is no existing Data Selector with the exact Selection Rules that you wish to apply, and that is why it is well worth the effort to huff and puff through this Manual.

Considering that most people can relate to a Customer List, we will use it for our examples. Typically, in an Application, the details of the Customer Records would be stored in a File (some people prefer to call it a 'table'). Then, the Application may offer a variety of Data Input, Update and Enquiry Screens that operate on this collection of Customer Information, as well as Reports.

When you wish to update the Address for a particular Customer, you will need to know something about the Customer that distinguishes it from other Records, else you would have to look through the entire File to find the correct Record. One way that Applications are typically designed would be to use the Customer Code as the primary retrieval Key to find a given Customer Record. In other words, if we number our Customers 1, 2, 3 … then we could ask the system to retrieve the Customer Record numbered 553, and it would show us that Record. We could say that the system applies the stated Key as the method to 'select the data', hence our preferred term 'Data Selectors'.

It is also possible that we may not know the Customer Code when we are looking for a particular Customer Record, but we may know the Customer's Surname, and perhaps the Address. So we would really like our system to be able to retrieve or 'select' data on more than just one attribute. We could follow on by explaining that each Customer Record in our Application comprises multiple Fields or Attributes, e.g. Customer Code, Surname, Initials, Address, etc., and that we would like to be able to select 1 or more Customer records on the basis of specifying 'criteria' that should be true before for a Record before it will be selected. For example, out of our Customer list of +- 5 000, we may wish to produce a Report that includes only those Customers where the Surname starts with 'A', where the Customer lives in a state called 'Georgia', and where the Customer owes us less than $1000.

DATA SELECTORS may be understood to be the Rules that the system will apply to select Data, i.e. in compliance with what the User wants.

OK, it's not going to be very practical to work with 5000 + Customers in this Manual, so let's construct a shorter list to work with, and one that will make it easy to explain the concepts. Bearing in mind that different Files may have many more 'attributes' than we will use in our example, the principles remain the same.

CustCode	Surname	Initials	State	Amount Owing
A001	Albrecht	J.T.	Georgia	$995
A002	Aloh	J.	Georgia	$1250
A003	Ahlstrom	F.K.S.	California	$0
B001	Brown	B.C.	California	$0

C001	Callitz	G.	California	$870
D001	Donkin	D.E.	New York	$555

So how would we typically ‘select data’ from these records?

Example 1: Select Customers with Surname starting with “a”
Result: Retrieves A001, A002, A003.
Explanation: These are the only Records where the Surname starts with “a” – our example is NOT case sensitive!

Example 2: Select Customers with Surname starting with “a” or where Key includes “00”
Result: Retrieves A001, A002, A003, B001, C001, D001.
Explanation: All the Records have “00”

Example 3: Select Customers with Surname starting with “a” and where Key includes “00” or where Initials include “G”
Result: Retrieves A001, A002, A003.
Explanation: Only these Records satisfies the 1st Condition, and “and” is applied.

Example 4: Select Customers with Surname starting with “a” or where Key includes “00” and where Initials include “G”
Result: ?
Explanation: We could tell you the result, but rather we wish to make the point that when you start mixing ‘and’ and ‘or’, it might get confusing as to what you are trying to tell the system, and it’s after all only a computer, and cannot read your mind. The bottom line is that Users get frustrated because the system does not deliver what they are trying to tell it to do.

So, in example 4, we realize that although it may be reasonably easy for Programmers and Technical people to do the ‘computer speak’, other Users may find it confusing.

That is why we designed the SHUTTLE Data Selectors with a ‘nesting’ concept, allowing you to string Data Selectors along in a queue of up to 9, and where each Data Selector operates on the results set of the former. So, if you need to explain 3 rules, you can make each ‘rule’ simple and easy to understand, and include only a single rule per Data Selector.

Hint: Although Complex Selection Statements do pretty much the same behind the scenes, they are not easy for Users to construct. If you have sufficient confidence, you may also include multiple rules on one SHUTTLE Data Selector. However, for Users, we recommend that you include no more than 2 or 3 Rules per Data Selector, separated by either ‘and’ or ‘or’. That way it is easy to understand.

Now let’s have a look at how this works in practice. In our example # 4 above, if what we really meant was : Get me the Records where the Surname starts with “A”, or if it does not start with “a” but the Key includes “00” and at the same time the Initials include a “G”, then grab that Record, then we could do it like this –

Data Selector 2 with Rule = Grab all Records where Surname starts with “a” OR where the Key includes “00”

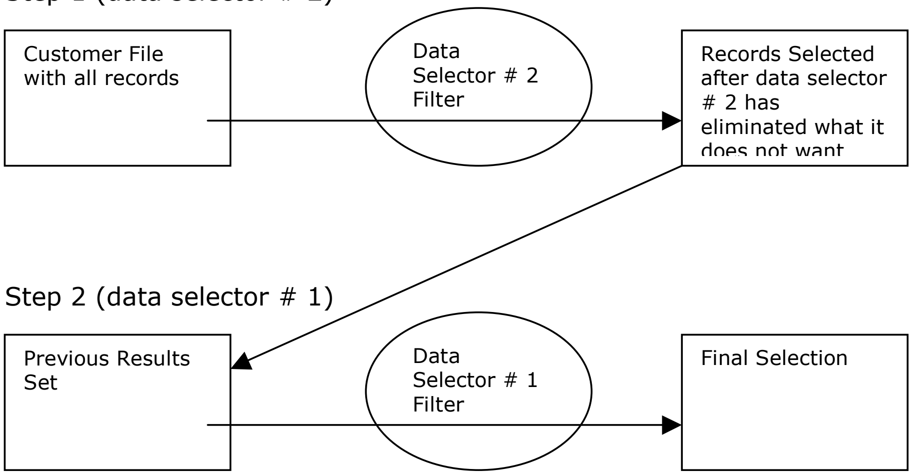
Data Selector 1 with Rule = Grab all Records where Surname starts with “a” or where Initials include “G”

Now why did we list Data Selector 2 before Data Selector 1? Simply because the system will execute the 'deepest' Selector first, and work it's way to the top. It may also be understood as 'elimination' of all the Records we do not want, which will leave us with those that we do want. Thus, Data Selector 1 will indicate that it 'follows on' Data Selector 2. Therefore, Data Selector 2 is executed 1st, and then Data Selector 1 operates on the results of what Data Selector 2 achieved. *(Hint: You can obviously number the Keys e.g. "1" and "2" the other way around. We just did this deliberately to make a point. In fact, the Keys do not have to be numbers at all.)*

In our example, Data Selector 2 will select all the Records, because we use the "OR" operator, and because all the Keys include "00". Data Selector 1 will then select A001, A002 and A003 because in these cases the Surname starts with "a", and it will also select C001, because in that case the initials include "G". Once more we have used the "OR" operator.

We may represent this visually as follows –

Step 1 (data selector # 2)



Customer File with all records
Data Selector # 2 Filter
Records Selected after data selector # 2 has eliminated what it does not want
Step 2 (data selector # 1)
Previous Results Set
Data Selector # 1 Filter
Final Selection

So what if we wanted to apply the same rules, but add 1 more rule, e.g. to select only Customers from California?

Data Selector 3 with Rule: Grab all Records where State = "California"

Data Selector 2 with Rule: Grab all Records where Surname starts with "a" OR where the Key includes "00"

Data Selector 1 with Rule: Grab all Records where Surname starts with "a" or where Initials include "G"

We would specify that '1' follows on '2' which follows on '3'. Therefore, '3' will be executed by the system 1st, then '2', then '1'.

And we would end up with: A003 C001

So far so good. We now realize that we can apply simple rules of elimination, and have one Data Selector after the other operate on the results remaining from the former, until we have our final remaining Records, i.e. the ones we are attempting to select. But there is more smoke along the way! We cannot simply speak “English” to the computer, because it won’t allow us to. However, we can get pretty close.

We specify our Rules as shown below :-

With attribute = “[a]”
With attribute # “b]” and with attribute > “12”
With attribute < “99” or with attribute # “x^x”

Our primary syntax is to use “with”, “and”, “or”, and to apply Operators (= # < > >= <=), while ‘attribute’ refers to some Field Name for the File on which we are selecting. What is a Field Name? It is a name that we define for each piece of information in each Customer record (or whatever other File we may be working with). In SHUTTLE, we use a Dictionary Manager for defining these Field Names, and it is not something that Users are normally allowed to do, since the Programmers develop the Application. However, as we shall see later, these Field Names are offered to us as a Lookup List when we define a Data Selector. Once we have stated the Data Source (File) upon which the Data Selector must operate, the system will know which Field Names are available to us for use in our Rules of Selection (or elimination?).

Field Names are usually kept short, ideally as descriptive as possible, and do not contain spaces. Here are a few examples of what the Programmers may do with Field Names :-

Surname	surn, sur, famname
State	state
Description	desc, descr

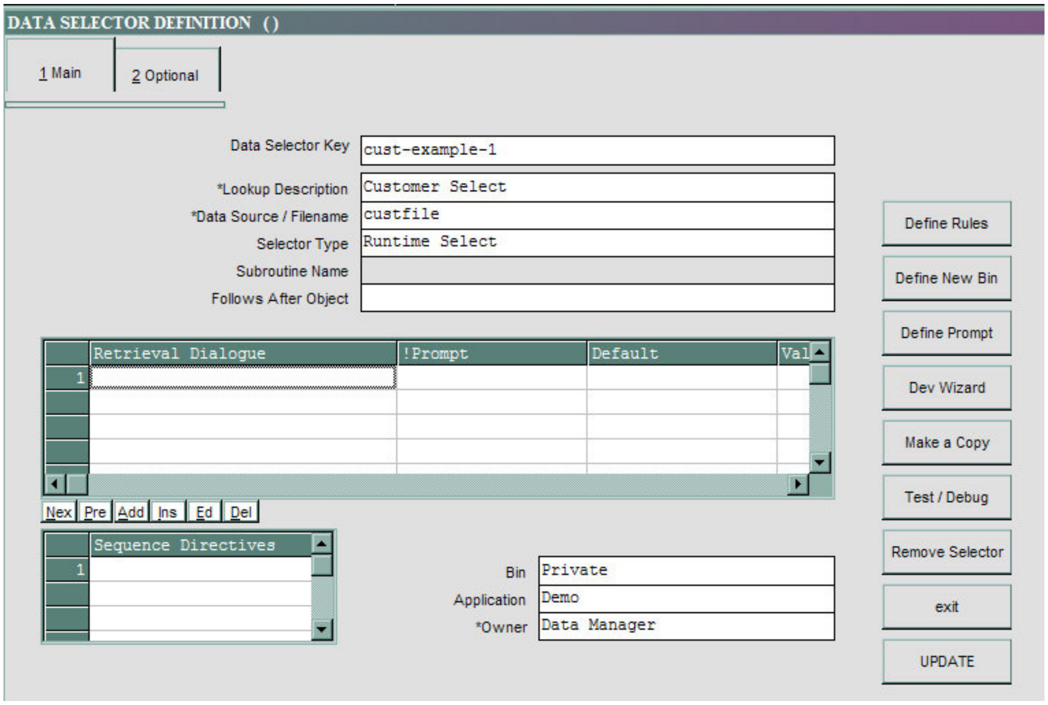
The next point to note is how we specify Values to compare against. We put such values in quotes (“”), and there are 3 wildcard characters available to us: [ ] ^

[means match characters to the right, ignore anything up to ‘[‘
] means match characters to the left, ignore anything after ‘]’
^ means accept any character in this position
[] means all values, because we ignore to both left and right

therefore, we understand it as follows –

Is “abc” = “a^c”	True
Is “abc” # “a^c”	False
Is “ Abrahams” = “[bra]”	True
Is “Abrahams” = “[]”	True
Is “Abrahams” = “braham]”	False

And now it is time to look at a Data Selector definition!



DATA SELECTOR DEFINITION ()

Data Selector Key	cust-example-1
*Lookup Description	Customer Select
*Data Source / Filename	custfile
Selector Type	Runtime Select
Subroutine Name	
Follows After Object	

	Retrieval Dialogue	!Prompt	Default	Val
1				

	Sequence Directives
1	

Bin	Private
Application	Demo
*Owner	Data Manager

In the picture above, we note the following :-

a) All the Fields shown have on-line help (click the Help Icon or choose F1), which will give you more detail than we will explain here.
b) The 'Data Selector Key' is of your own design, and for private Data Selectors, it is not a bad idea to include your UserCode in the Key, to differentiate it from other Selectors.
c) 'Lookup Description' needs to be meaningful. When you attempt to retrieve or choose an existing Data Selector, this is the information upon which you base your choice.
d) 'Data Source' is the File upon which the Selector will operate. There is of course a Lookup available on this Field.
e) 'Selector Type' is by default set to 'Runtime Select', and for most Users this will suffice. Advanced Users may check the Help on this Field to consider other options.
f) 'Subroutine Name' will never be used with "Runtime Select", only with Selectors where type = Subroutine, and you need to have programming skills to make one of those.
g) 'Follows After Object' is where you will state which other Data Selector this one will rely on, if any. In other words, when we discussed the concept of nesting before, we had Selectors '1' and '2', and in that particular example we would specify at this Field the name of the other Data Selector that must be executed before this one.
h) The 'Retrieval Dialogue' grid deals with the actual Rules, and we will discuss this in more detail in a moment.
i) The 'Sequence Directives' grid is optional, and here you may specify 1 or more Field Names to sort by. If you do, you specify them in the order you wish them to be sorted, 1 Field per line, like this:

    By famname
    By state

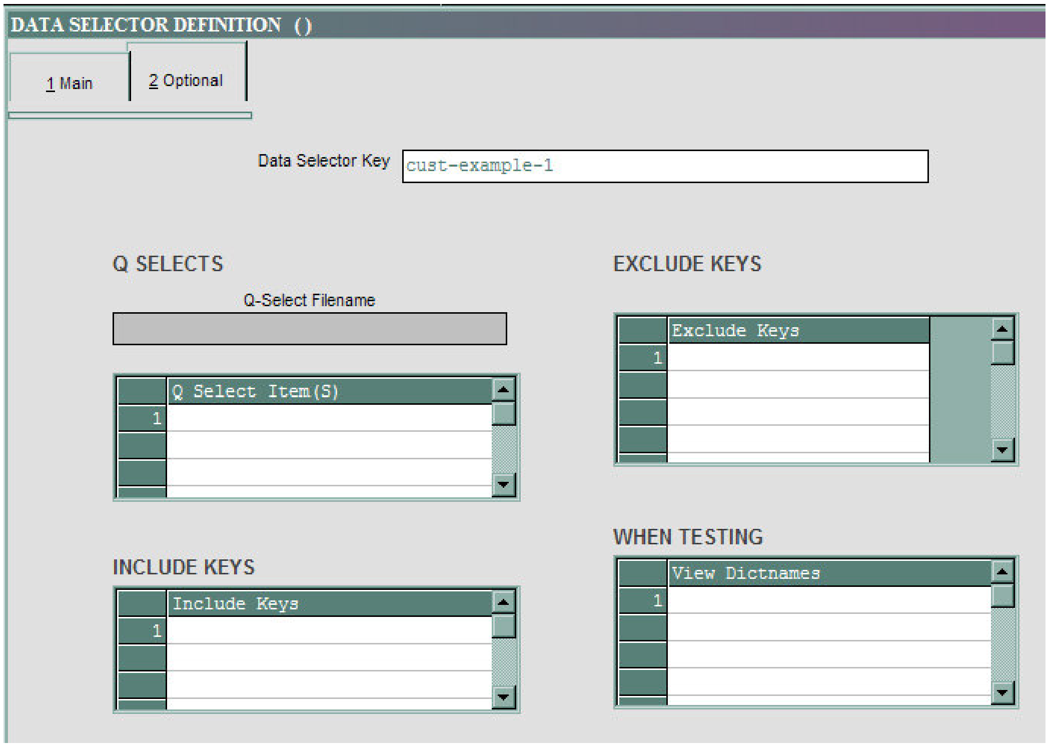
j) 'Bin' should always be 'Private' for your personal Data Selectors.
k) At 'Application', for your personal Data Selectors, it is best to use 'uutil', which will display: User Services

l) The ‘Owner’ Field is there to protect you against others being able to change your Data Selectors. When your own UserCode is displayed in this Field, only you and nobody else may change the Data Selector. If you really wish others to be able to change your Data Selector, you may state ‘public’ here, but there should be no need, since other Users may anyway ask the system to make a ‘copy’ for themselves, which they will then ‘own’ and may change, while leaving your Selector intact.

m) The Function Buttons :

- i) Define Rules (You will use this to create entries in the ‘Retrieval Dialogue’ grid)
- ii) Define New Bin (You will probably not have access to this, it is mostly for Developers and your ‘private’ Bin will already exist)
- iii) Define Prompt (You don’t need it, since the ‘Define Rules’ wizard will do it for you automatically; DEFINE PROMPT is used to make your own prompts for when you manually edit the grid, i.e. without using the DEFINE RULES wizard)
- iv) Dev Wizard (links you to the Business Objects Wizard, but Users generally do not have any options here)
- v) Make a Copy (is what you will use when retrieving somebody else’s Data Selector, which you quite like, and wish to make your own copy of, so you can use it and perhaps refine it further)
- vi) Test / Debug (is your option to TEST your Data Selector before putting it to real use; you will be able to see whether it is actually doing what you thought you told it to do)
- vii) Remove Selector (when you no longer want to use it ever again)
- viii) Exit (Quit without Saving)
- ix) Update (Save current version of the Selector)

There is another tab called ‘Optional’ to this Form. What does it contain?



Data Selector Key: cust-example-1

Q SELECTS

Q-Select Filename:

	Q Select Item(S)
1	

EXCLUDE KEYS

	Exclude Keys
1	

INCLUDE KEYS

	Include Keys
1	

WHEN TESTING

	View Dictnames
1	

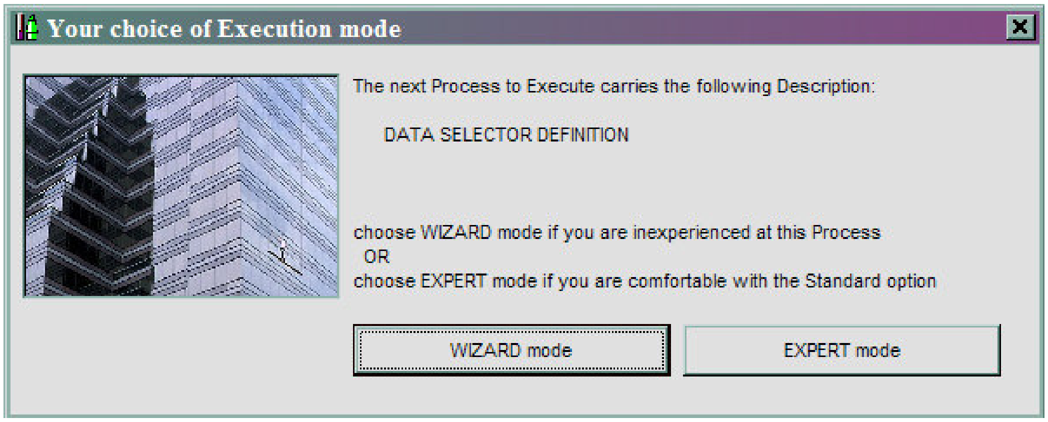
WHEN TESTING allows you to list some Field Names to display when you test / debug your Data Selector. If nothing is specified here, you only see the Record Keys that are selected.

Q SELECTS are advanced options, and not covered in this Manual.

INCLUDE KEYS may be used when you do not wish to specify Rules for Data Selection. Instead you specify which specific Record Keys to retrieve. For example, instead of saying 'with famname = "a]"', you change the Selector Type from 'Runtime Select' to 'k' for Keys Select, and then the system knows it should simply retrieve the Record Keys that you specify here.

EXCLUDE KEYS are useful for final massaging after your Selection Rules are applied. If your rules work fine, but the Logic required to eliminate a certain few Keys which you do not want to select is too complicated, you just list those Keys here, and then they will not be included in the final result. Note that this facility is only checked on the final Data Selector when you use nesting, i.e. the last Data Selector to execute.

After this 'mouthful' of explanations, we may tell you that SHUTTLE makes it easier for you. When you enter the DATA SELECTOR option, you will see this –



Your choice of Execution mode

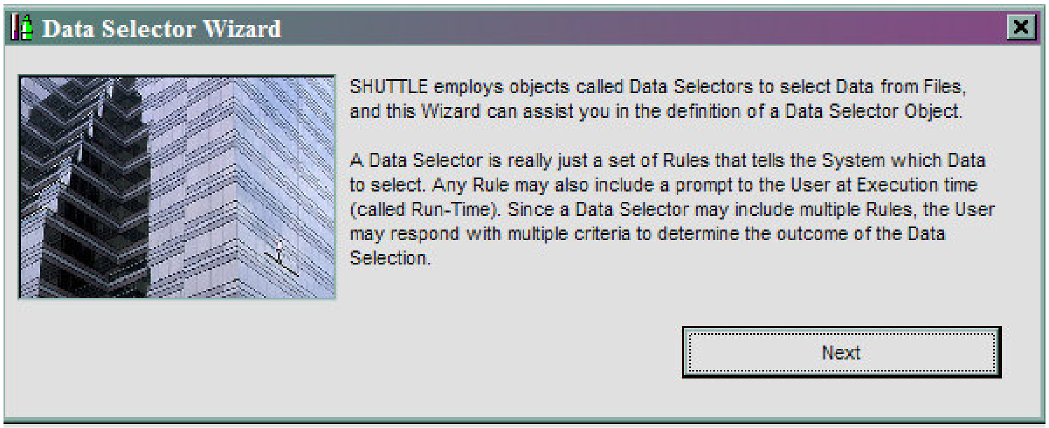
The next Process to Execute carries the following Description:

DATA SELECTOR DEFINITION

choose WIZARD mode if you are inexperienced at this Process
OR
choose EXPERT mode if you are comfortable with the Standard option

When you become quite 'expert' with Data Selectors, you will start choosing EXPERT mode, but until then, you are better off with WIZARD mode. Let's go through the steps of creating a Data Selector with this Wizard :-

There is a short introduction –

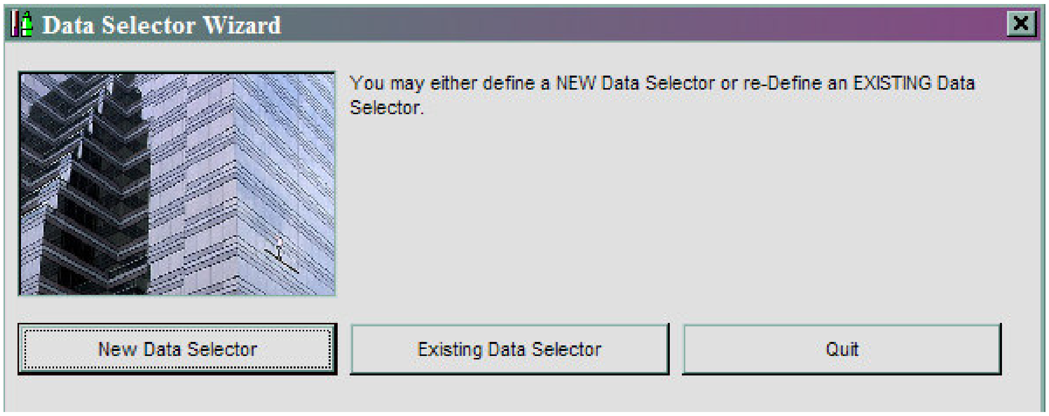


Data Selector Wizard

SHUTTLE employs objects called Data Selectors to select Data from Files, and this Wizard can assist you in the definition of a Data Selector Object.

A Data Selector is really just a set of Rules that tells the System which Data to select. Any Rule may also include a prompt to the User at Execution time (called Run-Time). Since a Data Selector may include multiple Rules, the User may respond with multiple criteria to determine the outcome of the Data Selection.

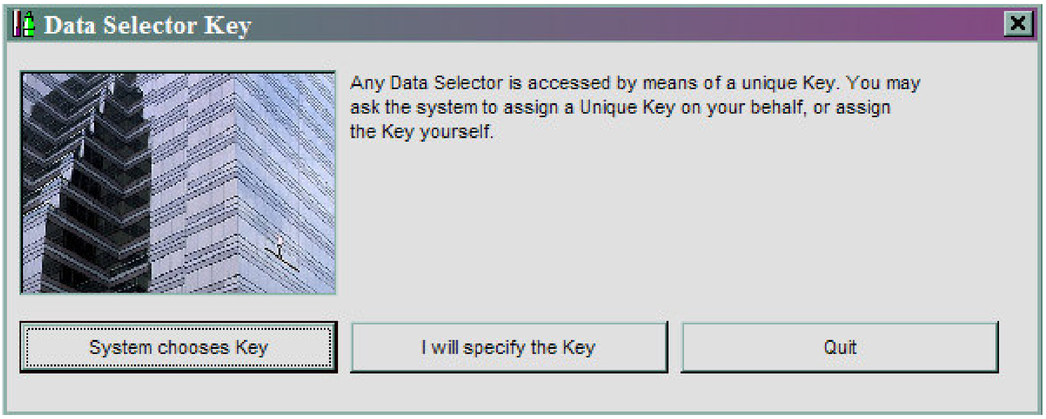
You may choose between a NEW Data Selector and doing more work on one that already exists –



Data Selector Wizard

You may either define a NEW Data Selector or re-Define an EXISTING Data Selector.

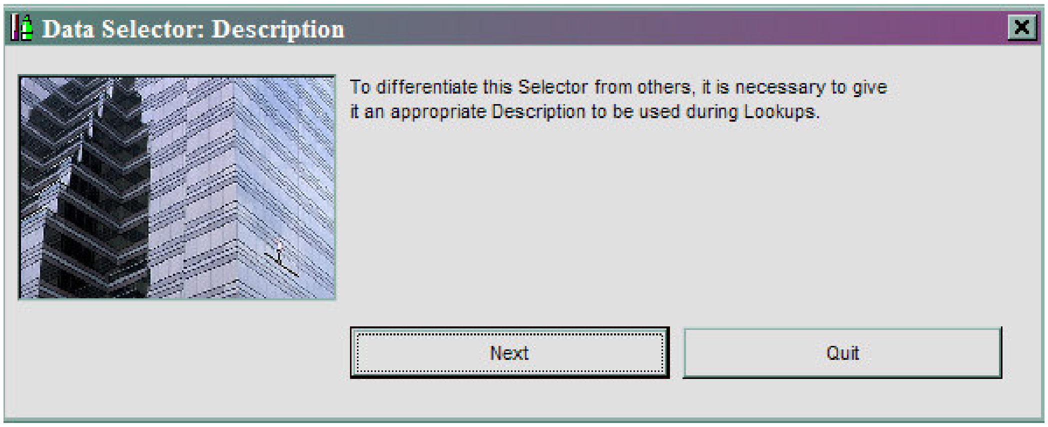
If you choose EXISTING, the Wizard will take you through the steps to retrieve the desired Data Selector, which you may then edit. In this example, we will choose NEW –



Data Selector Key

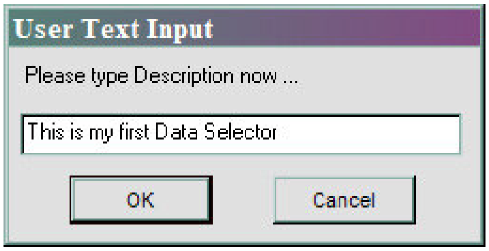
Any Data Selector is accessed by means of a unique Key. You may ask the system to assign a Unique Key on your behalf, or assign the Key yourself.

We have already discussed the KEY higher up. You may choose whether you want to specify the Key yourself, or simply let the system create a unique Key for you.


Data Selector: Description

To differentiate this Selector from others, it is necessary to give it an appropriate Description to be used during Lookups.

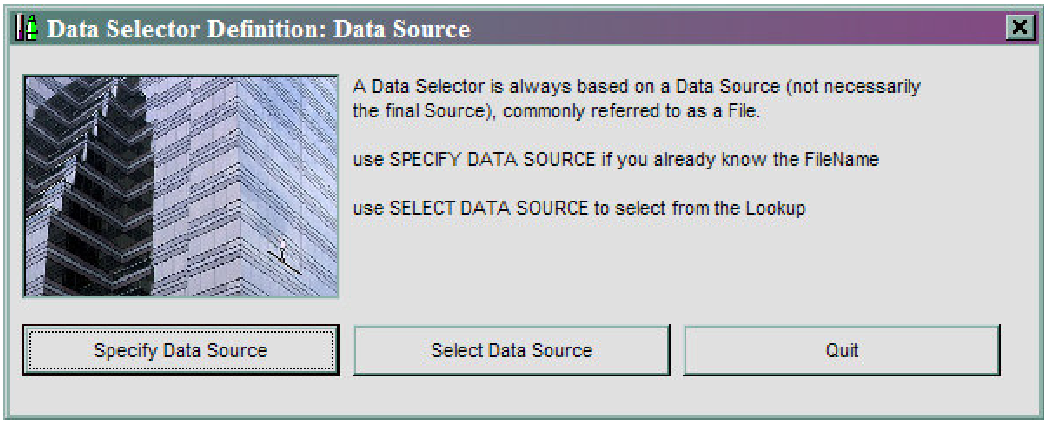
This is followed by the all important Description, which will appear with Lookups on your existing Data Selectors. You can change this Description at any time afterwards.


User Text Input

Please type Description now ...

This is my first Data Selector

And then the system wants to know which Data File we will be using. If you already know the name of the File, you may specify it directly, or else the system will take you through the steps to find the File Name.

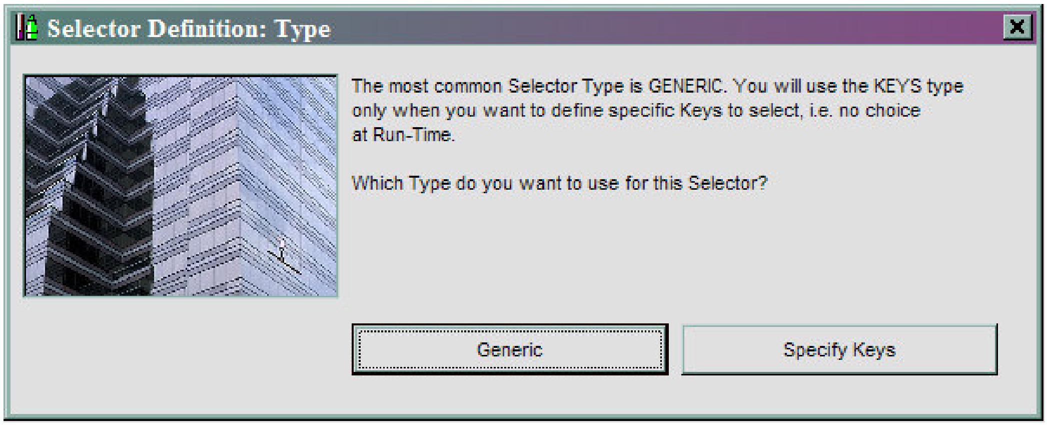

Data Selector Definition: Data Source

A Data Selector is always based on a Data Source (not necessarily the final Source), commonly referred to as a File.

use SPECIFY DATA SOURCE if you already know the FileName

use SELECT DATA SOURCE to select from the Lookup

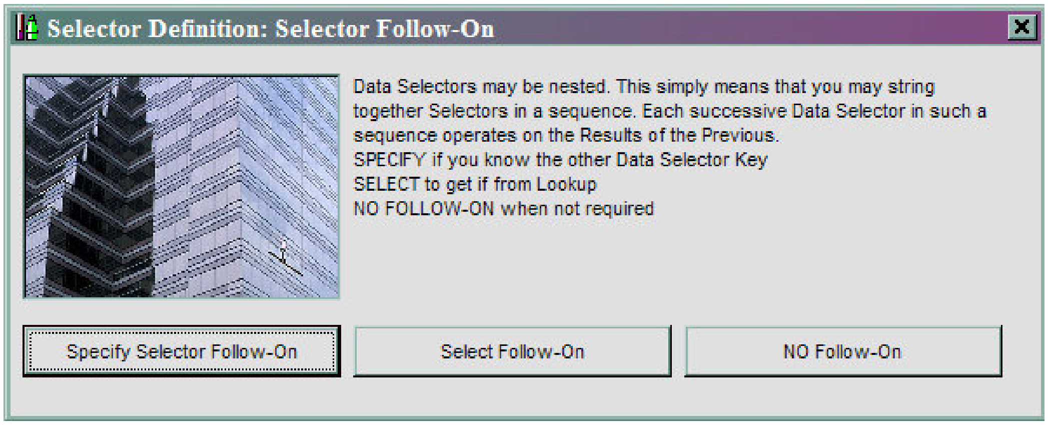
The next choice is the TYPE of Data Selector this will be. We have already explained that you will mostly use the ‘Runtime Select’, so you should choose GENERIC, except when you wish to do a ‘Keys’ type. (Hint: The advanced Q Select and Subroutine options are not offered by the Wizard. People who use those options do not use the WIZARD mode.)



The most common Selector Type is GENERIC. You will use the KEYS type only when you want to define specific Keys to select, i.e. no choice at Run-Time.

Which Type do you want to use for this Selector?

And then we have an opportunity to specify FOLLOW ON (the nesting capability that we explained before) –

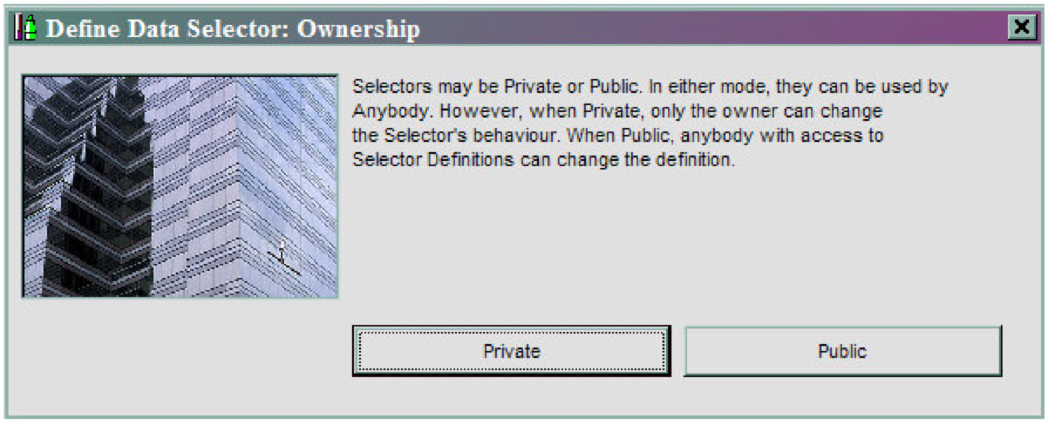

Selector Definition: Selector Follow-On
Specify Selector Follow-On
Select Follow-On
NO Follow-On

Data Selectors may be nested. This simply means that you may string together Selectors in a sequence. Each successive Data Selector in such a sequence operates on the Results of the Previous.
SPECIFY if you know the other Data Selector Key
SELECT to get if from Lookup
NO FOLLOW-ON when not required

If this Data Selector follows on another, you can either specify the Key of the other, or ask the system to help you find it;

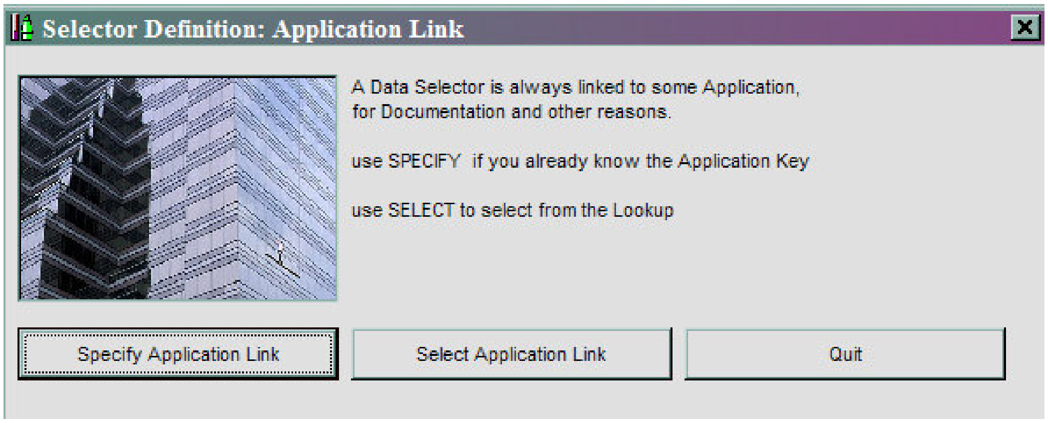
Hint: If you make any 'bad' choices during this entire sequence of steps, simply continue until you are offered the Data Selector Screen. At that stage you can change anything.

Now choose between PRIVATE and PUBLIC –



Selectors may be Private or Public. In either mode, they can be used by Anybody. However, when Private, only the owner can change the Selector's behaviour. When Public, anybody with access to Selector Definitions can change the definition.

And then select the Application it is associated with (we have recommended USER SERVICES for private Selectors, but you may choose a more appropriate Application Association if you wish) –



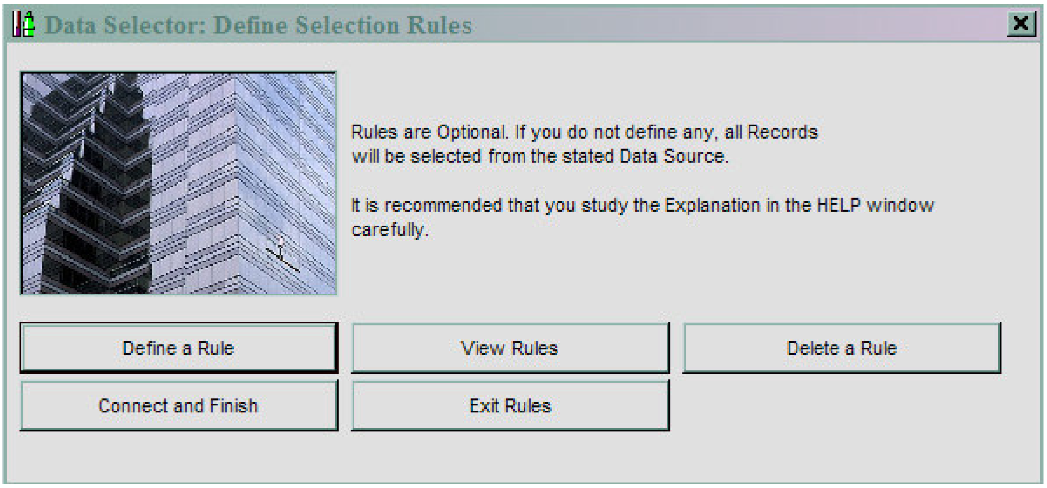
Selector Definition: Application Link

A Data Selector is always linked to some Application, for Documentation and other reasons.

use SPECIFY if you already know the Application Key

use SELECT to select from the Lookup

And finally, we get to the RULES !



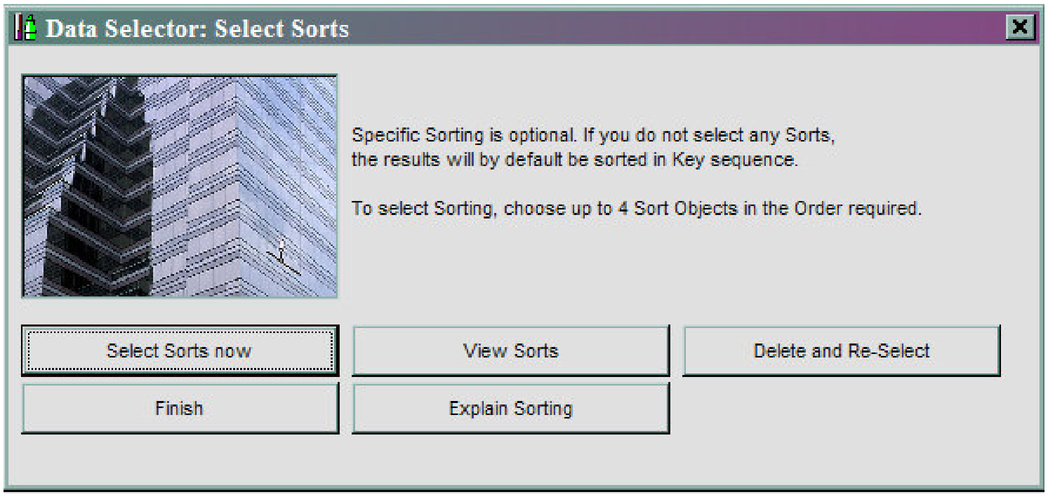
Data Selector: Define Selection Rules

Rules are Optional. If you do not define any, all Records will be selected from the stated Data Source.

It is recommended that you study the Explanation in the HELP window carefully.

We will not explain these steps now, since they are the same as when we choose DEFINE RULES from the Data Selector Screen, and we will look at that in a moment.

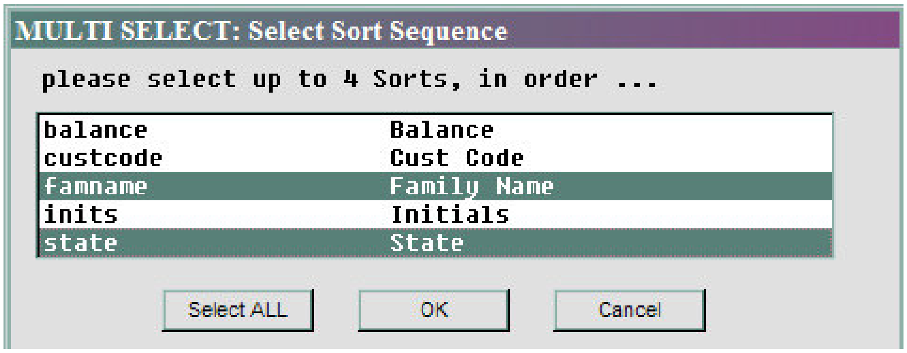
In the meantime, after RULES, you can optionally specify some SORTing.



Specific Sorting is optional. If you do not select any Sorts, the results will by default be sorted in Key sequence.

To select Sorting, choose up to 4 Sort Objects in the Order required.

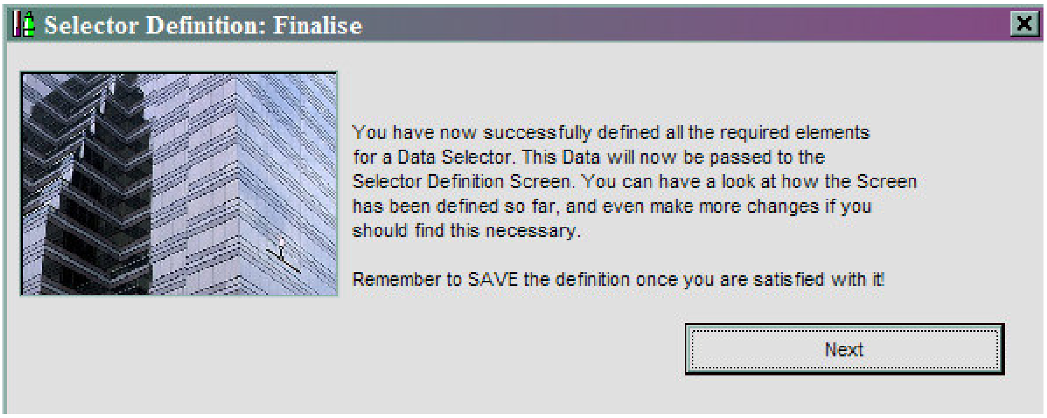
If we choose ‘Select Sorts now’, we get offered a list to choose from, and we may choose 1 or more, in the required order of Sorting …



please select up to 4 Sorts, in order ...

balance	Balance
custcode	Cust Code
famname	Family Name
inits	Initials
state	State

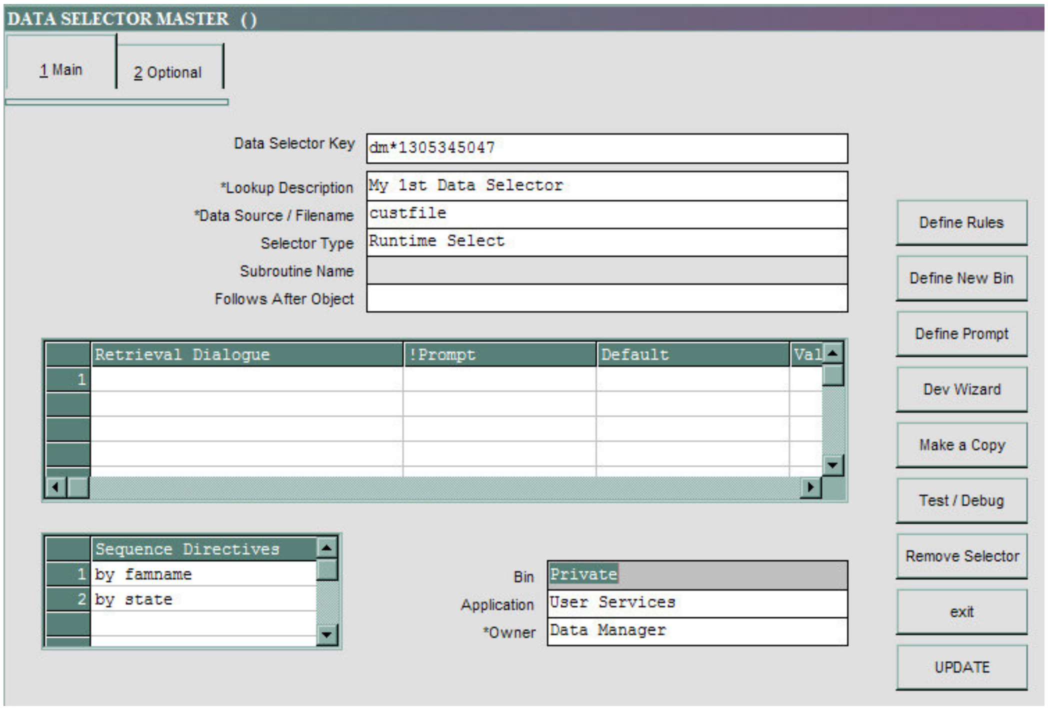
And then the system informs us that we have put together all the required elements for a Data Selector, after which we enter the same Screen which we would have entered if we chose EXPERT mode up front (although in that case the Screen would have been blank, whereas now we have almost everything already specified).



You have now successfully defined all the required elements for a Data Selector. This Data will now be passed to the Selector Definition Screen. You can have a look at how the Screen has been defined so far, and even make more changes if you should find this necessary.

Remember to SAVE the definition once you are satisfied with it!

What’s on our Screen?



Data Selector Key: dm*1305345047
*Lookup Description: My 1st Data Selector
*Data Source / Filename: custfile
Selector Type: Runtime Select
Subroutine Name:
Follows After Object:

	Retrieval Dialogue	!Prompt	Default	Val
1				

	Sequence Directives
1	by famname
2	by state

Bin: Private
Application: User Services
*Owner: Data Manager

There are no Rules specified, because we chose not to. But we can still do them with DEFINE RULES … However, we need to explain another concept 1st. There are ‘fixed’ Rules and ‘open’ Rules. A Fixed Rule is something that the Data Selector will simply apply, whereas an Open Rule will result in the User being prompted for a Response.

For example:

With famname = “a]”

> Is a Fixed Rule; the system will select Records where ‘famname’ starts “a”

With famname = “?”

> Is an Open Rule; the system will request the User to indicate the Value. Therefore, the same Data Selector can be used to select different records every time it is used, one time you may reply “a]” to get the A’s, next you may reply “b]” to get the B’s.

There are 4 Columns on the ‘Retrieval Dialogue’ grid :-

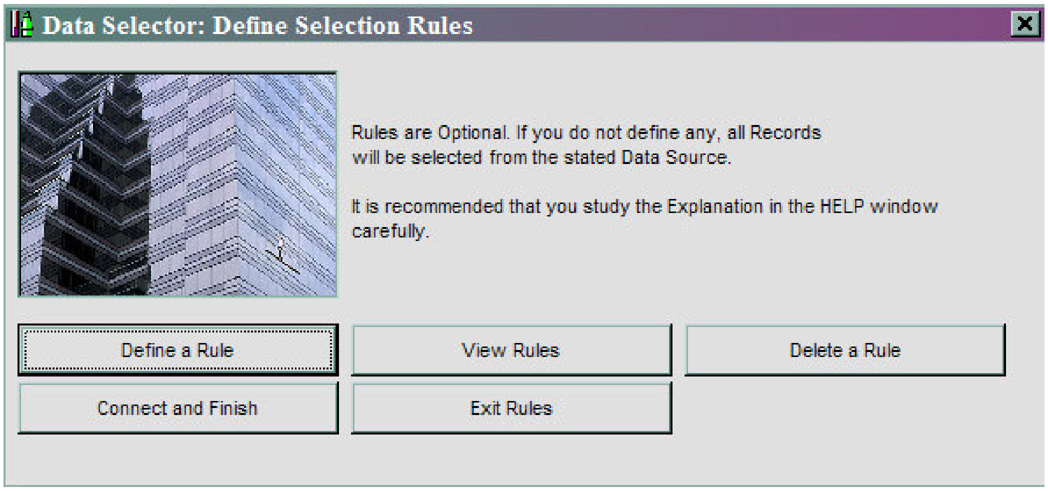
Retrieval Dialogue, Prompt, Default, Validation

For Fixed Rules, only the 1st Column applies, for Open Rules we may use all the Columns, although you probably will not need ‘Validation’, which is an advanced option. The Prompt Column is used to specify a Key on the MESSAGES File, which contain the Prompt to use to display to the User. It is no good to prompt the User ‘famname = ?’, because it is not meaningful enough. Instead, we define a Prompt which the User will understand, and may be something like: Pattern to look for in Family Name ?

And the Default is what we provide for simple acceptance, i.e. if the User does not wish to be more specific. More often than not, this Default is specified as ‘[]’, i.e. to select all Records.

Hint: When there are multiple Open Rules specified on nested Data Selectors, the system will analyze these and offer them together on a single Prompt Screen for the User, before executing the 1st Data Selector.

As we will see, the DEFINE RULES wizard does most of these things for us transparently … Let’s choose DEFINE RULES and see what happens.

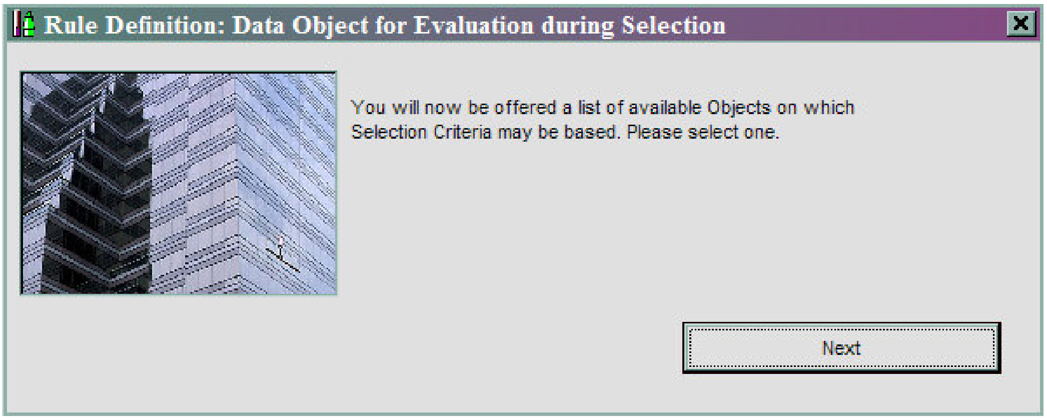


Data Selector: Define Selection Rules

Rules are Optional. If you do not define any, all Records will be selected from the stated Data Source.

It is recommended that you study the Explanation in the HELP window carefully.

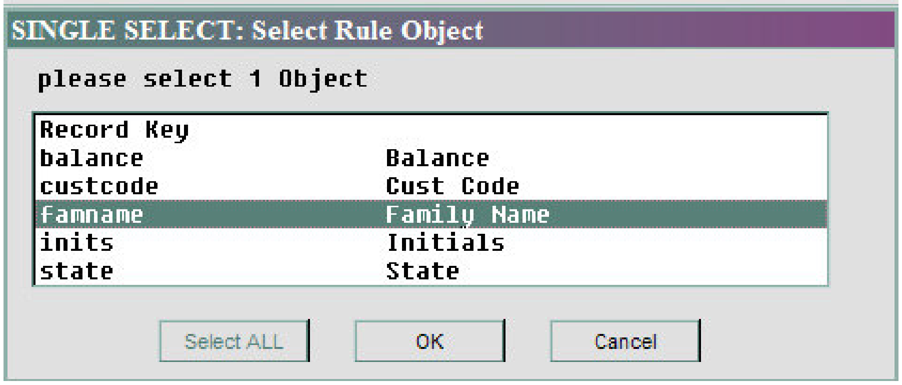
Let’s choose DEFINE a RULE –



Rule Definition: Data Object for Evaluation during Selection

You will now be offered a list of available Objects on which Selection Criteria may be based. Please select one.

The above means that we will now choose from the available Field Names on our Data Source (File) on which to base the Rule we are busy defining …

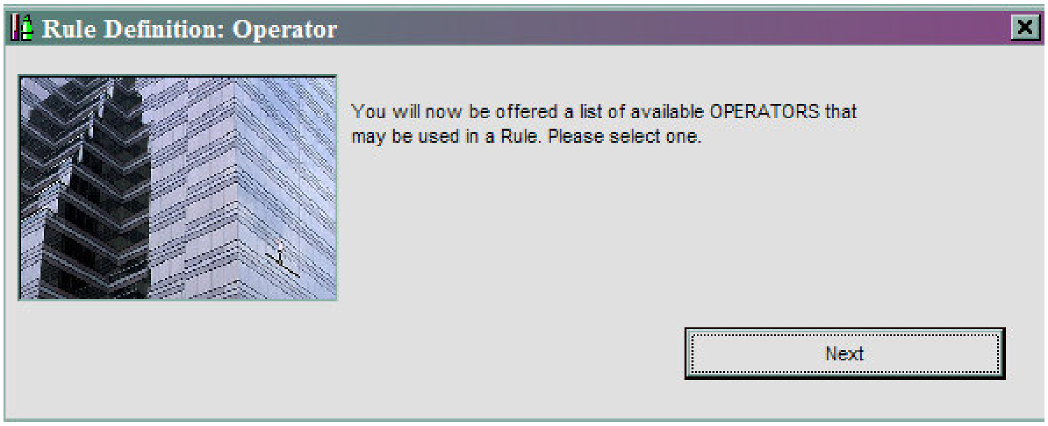


SINGLE SELECT: Select Rule Object

please select 1 Object

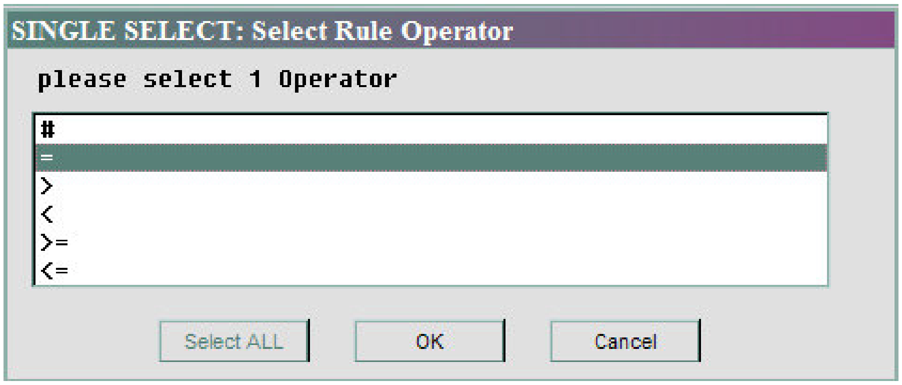
Record Key	
balance	Balance
custcode	Cust Code
famname	Family Name
inits	Initials
state	State

We choose FAMILY NAME –


Rule Definition: Operator

You will now be offered a list of available OPERATORS that may be used in a Rule. Please select one.

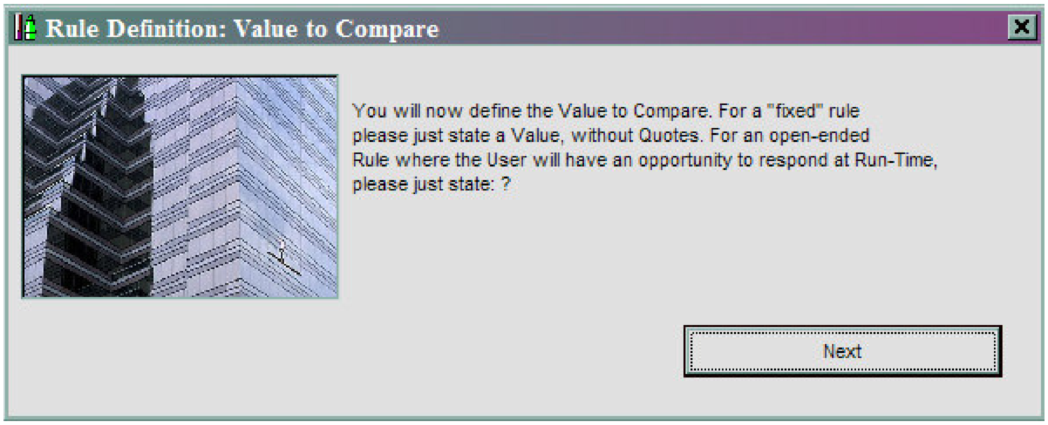
The next step will be to select an Operator –


SINGLE SELECT: Select Rule Operator

please select 1 Operator

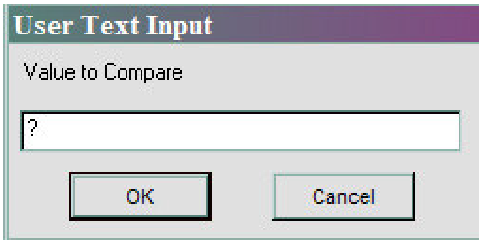
```
#
=
>
<
>=
<=
```

We choose 'equal to' –


Rule Definition: Value to Compare

You will now define the Value to Compare. For a "fixed" rule please just state a Value, without Quotes. For an open-ended Rule where the User will have an opportunity to respond at Run-Time, please just state: ?

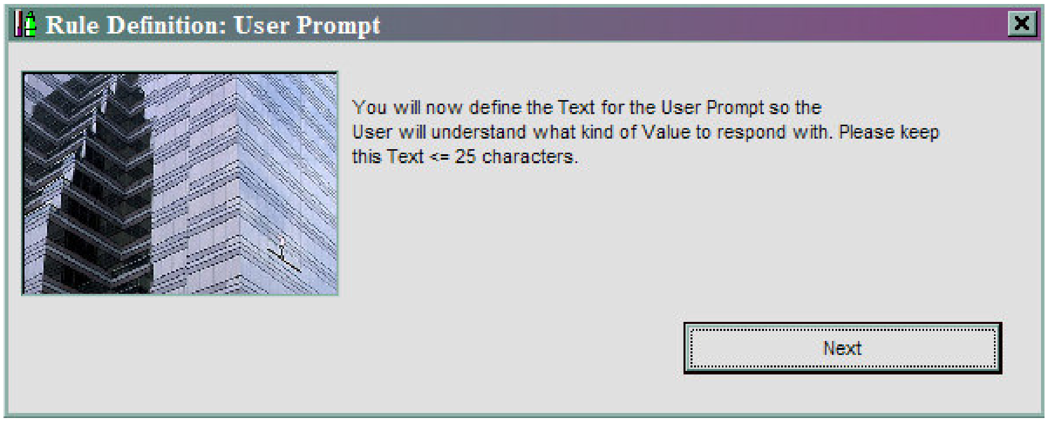
Choose whether this is an OPEN or FIXED Rule. For FIXED, we can supply the FIXED value, e.g. "a]", whereas for OPEN, we state '?', and this will then lead to the Prompt to be defined.



Value to Compare

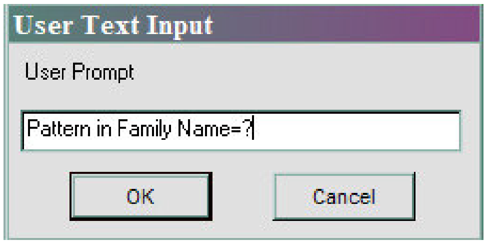
?

We choose an OPEN Rule –



You will now define the Text for the User Prompt so the User will understand what kind of Value to respond with. Please keep this Text <= 25 characters.

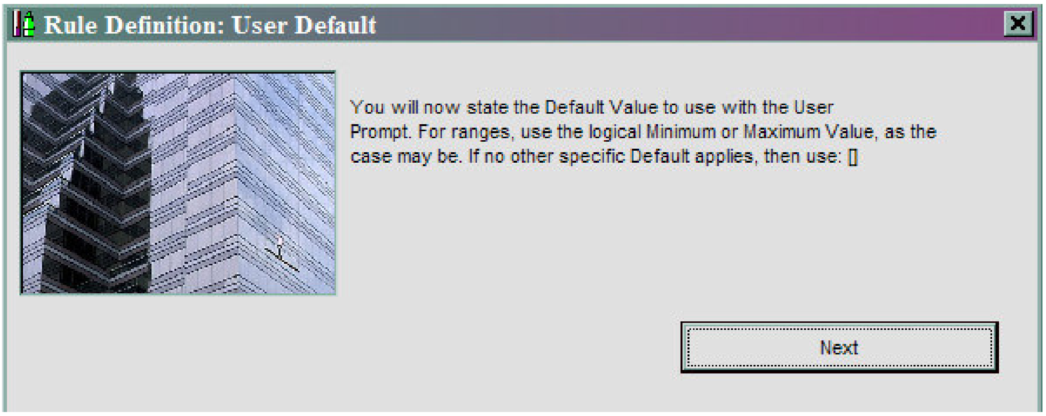
The TEXT that will be displayed to the User should be kept <= 25 Characters …



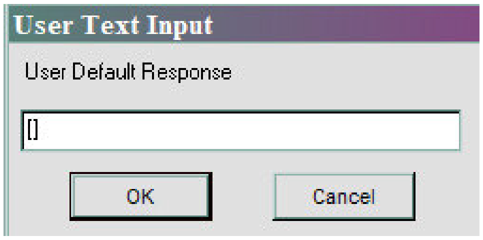
User Prompt

Pattern in Family Name=?

And we still need to provide the Default Value, in case the User does not wish to specify anything …



You will now state the Default Value to use with the User Prompt. For ranges, use the logical Minimum or Maximum Value, as the case may be. If no other specific Default applies, then use: []

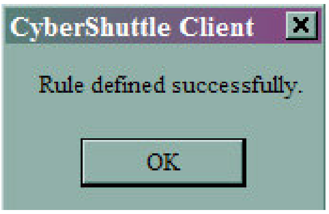


User Text Input

User Default Response

[]

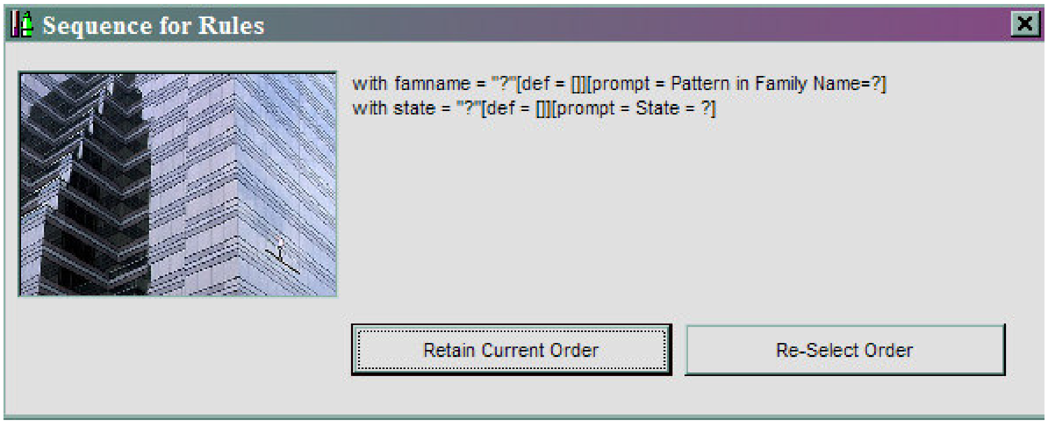
And we've done it …



CyberShuttle Client

Rule defined successfully.

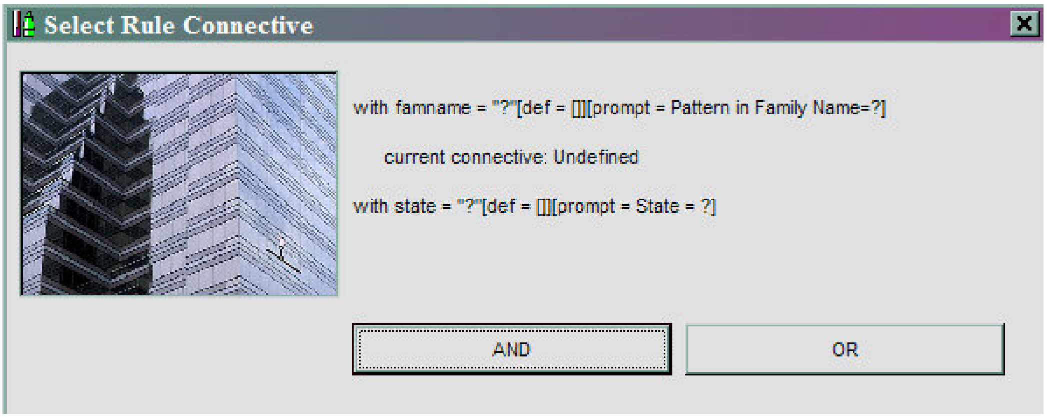
Now we can define another Rule, View the Rules, Delete a Rule, or CONNECT and FINISH, the latter being the final step we should always do before we are done. This step will prompt us where to use OR and where to use AND …



Sequence for Rules

with famname = "?"[def = []][prompt = Pattern in Family Name=?]
with state = "?"[def = []][prompt = State = ?]

When we do CONNECT and FINISH, the system displays our Rules, and we may re-order the sequence if necessary (note that we have defined another Rule, following the same steps as with the 1st) …



Select Rule Connective

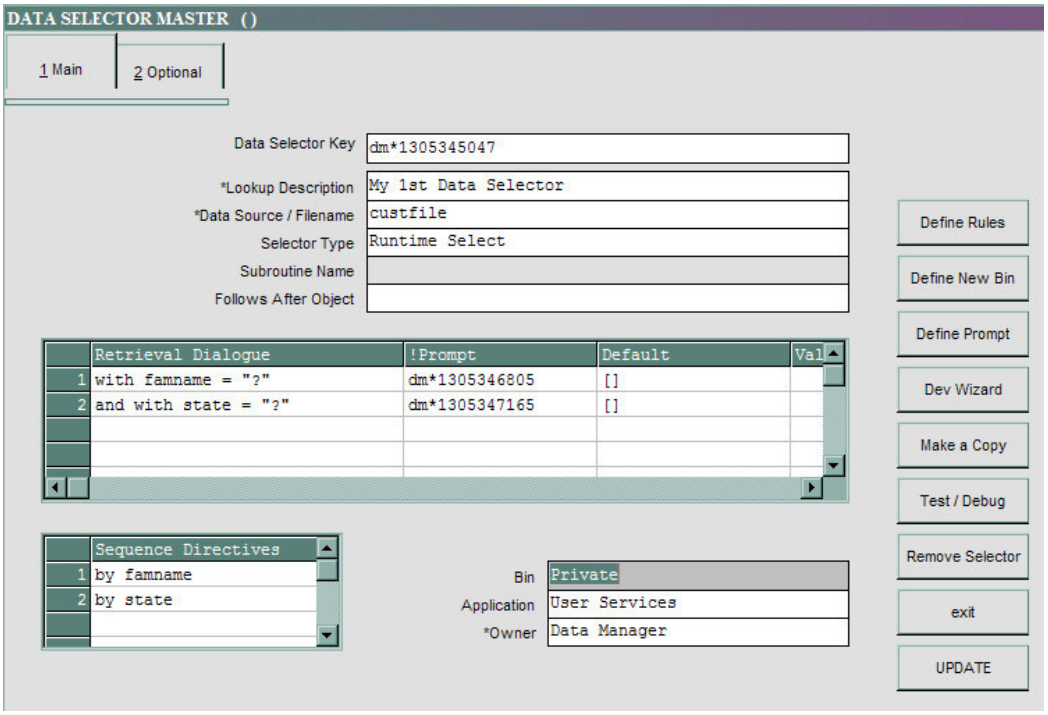
with famname = "?"[def = []][prompt = Pattern in Family Name=?]

current connective: Undefined

with state = "?"[def = []][prompt = State = ?]

We have to choose whether to use OR or AND (we choose AND in this case)

And now the latest version of our Data Selector looks like this …


DATA SELECTOR MASTER ()

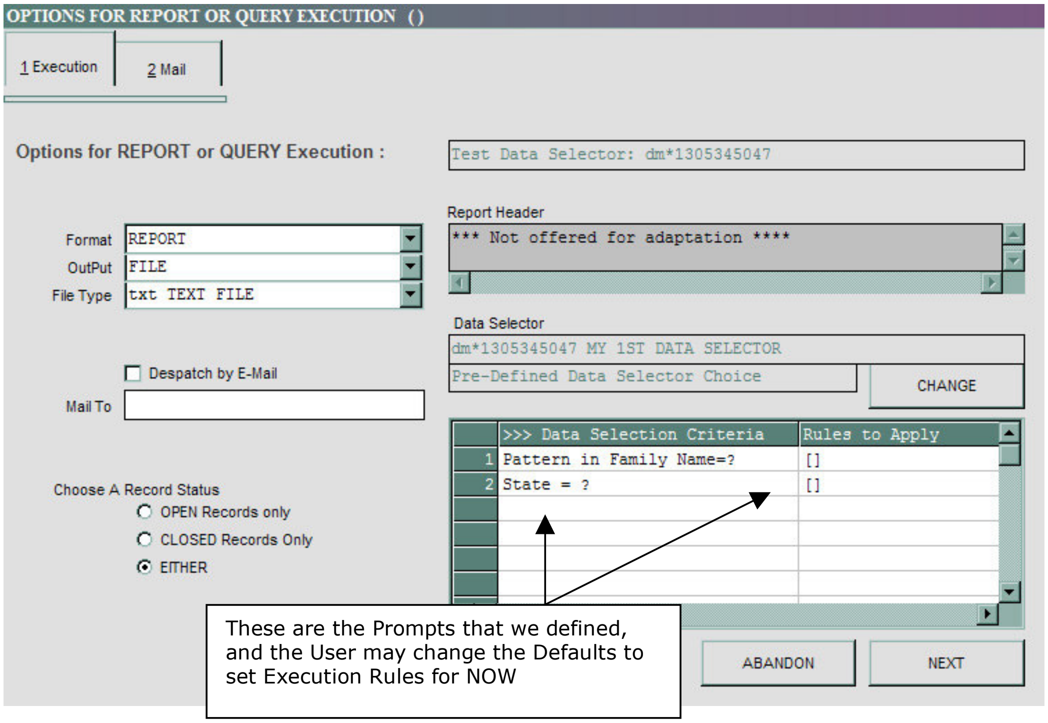
Data Selector Key	dm*1305345047
*Lookup Description	My 1st Data Selector
*Data Source / Filename	custfile
Selector Type	Runtime Select
Subroutine Name	
Follows After Object	

	Retrieval Dialogue	!Prompt	Default	Val
1	with famname = "?"	dm*1305346805	[]	
2	and with state = "?"	dm*1305347165	[]	

	Sequence Directives
1	by famname
2	by state

Bin	Private
Application	User Services
*Owner	Data Manager

So let’s choose TEST / DEBUG and see what happens …


OPTIONS FOR REPORT OR QUERY EXECUTION ()

Options for REPORT or QUERY Execution : Test Data Selector: dm*1305345047

Format	REPORT
OutPut	FILE
File Type	txt TEXT FILE

□ Despatch by E-Mail

Mail To

Report Header: *** Not offered for adaptation ****

Data Selector: dm*1305345047 MY 1ST DATA SELECTOR

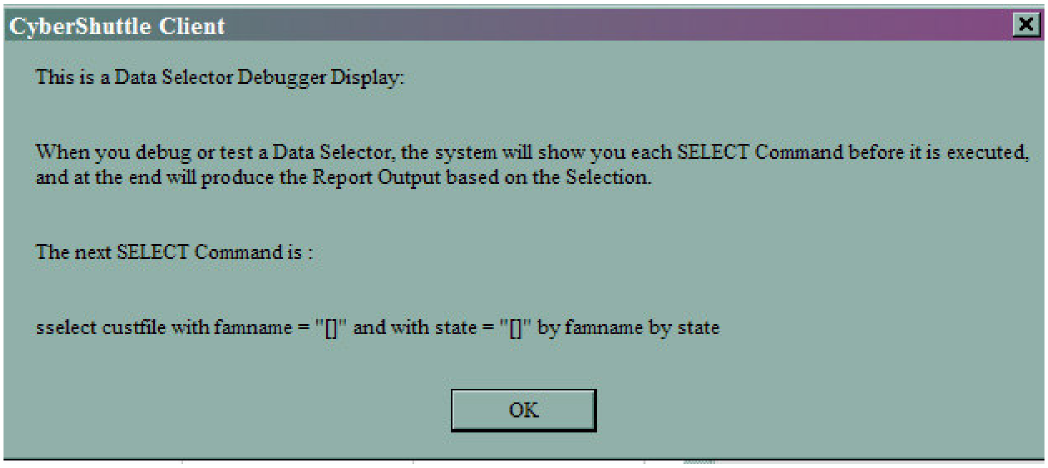
Pre-Defined Data Selector Choice

Choose A Record Status
○ OPEN Records only
○ CLOSED Records Only
● EITHER

	>>> Data Selection Criteria	Rules to Apply
1	Pattern in Family Name=?	[]
2	State = ?	[]

These are the Prompts that we defined, and the User may change the Defaults to set Execution Rules for NOW

After prompting whether the Data Selector may be SAVEd 1st, the system turns the Data Selector into a Report, and offers the Options. We choose NEXT …



This is a Data Selector Debugger Display:

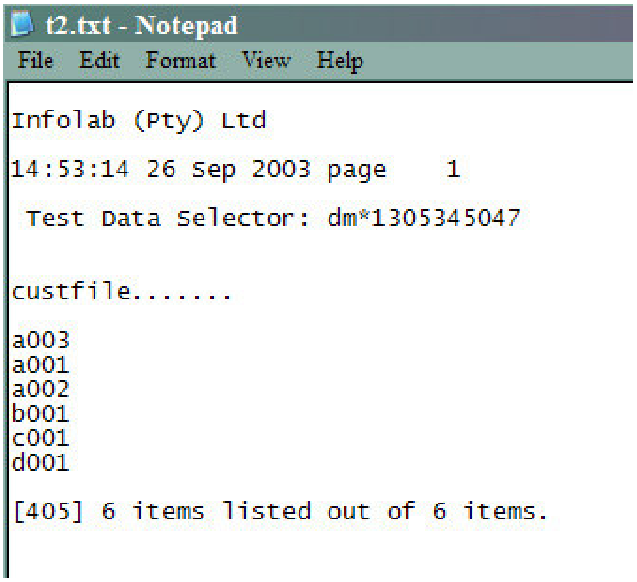
When you debug or test a Data Selector, the system will show you each SELECT Command before it is executed, and at the end will produce the Report Output based on the Selection.

The next SELECT Command is :

```
sselect custfile with famname = "[]" and with state = "[]" by famname by state
```

Because we are 'debugging', the system shows us each Command it executes (for nested Data Selectors, there will be more than 1 such Command);

And then it produces the Results. In this example, because our Criteria are "[]", which are Wildcards, all the Records are selected.



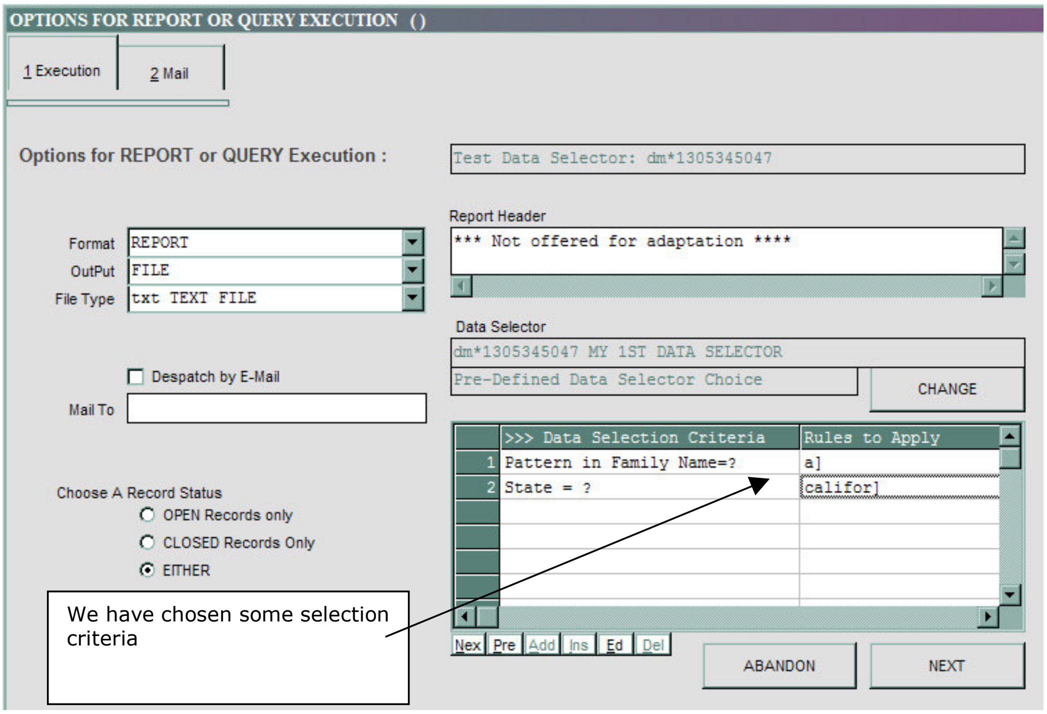
```
Infolab (Pty) Ltd
14:53:14 26 Sep 2003 page    1
 Test Data Selector: dm*1305345047

custfile.......

a003
a001
a002
b001
c001
d001

[405] 6 items listed out of 6 items.
```

Now we will re-run, and change the Criteria …



OPTIONS FOR REPORT OR QUERY EXECUTION ( )

Options for REPORT or QUERY Execution :

Test Data Selector: dm*1305345047

Format: REPORT

OutPut: FILE

File Type: txt TEXT FILE

Report Header: *** Not offered for adaptation ****

Data Selector: dm*1305345047 MY 1ST DATA SELECTOR

Pre-Defined Data Selector Choice

☐ Despatch by E-Mail

Mail To:

Choose A Record Status

○ OPEN Records only

○ CLOSED Records Only

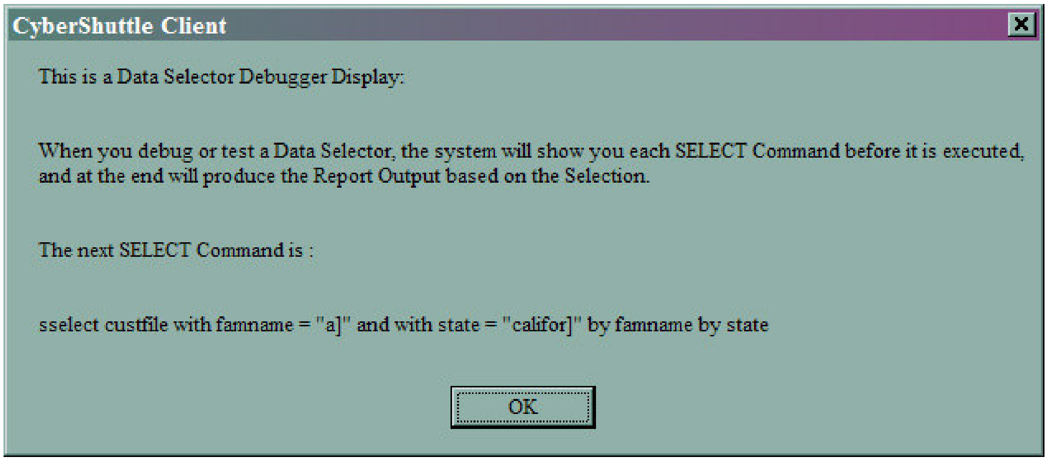
● EITHER

	>>> Data Selection Criteria	Rules to Apply
1	Pattern in Family Name=?	a]
2	State = ?	califor]

We have chosen some selection criteria

Now we are selecting only Records where Family Name starts with "a" and where the State starts with "califor".

And here are the results –



CyberShuttle Client

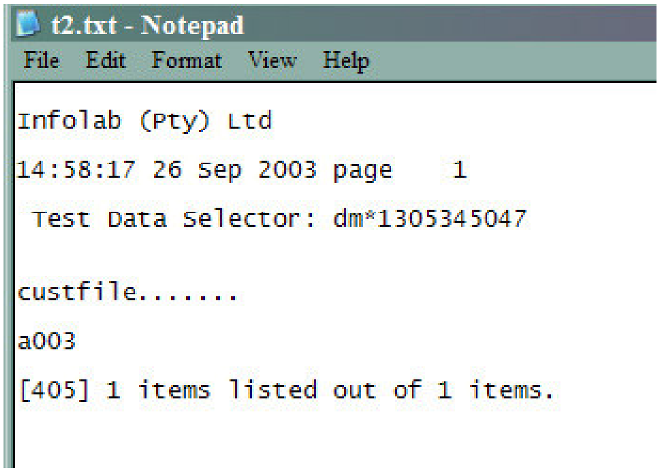
This is a Data Selector Debugger Display:

When you debug or test a Data Selector, the system will show you each SELECT Command before it is executed, and at the end will produce the Report Output based on the Selection.

The next SELECT Command is :

sselect custfile with famname = "a]" and with state = "califor]" by famname by state

Only 1 Record qualifies …



```
Infolab (Pty) Ltd
14:58:17 26 Sep 2003 page    1
 Test Data Selector: dm*1305345047

custfile.......
a003
[405] 1 items listed out of 1 items.
```

Hint: Compare with the Data we used (in the beginning of this Manual).

Once you can do Data Selectors, you can use them with your Data Queries and Reports. When you create your own Data Selectors, remember that the golden rules are –

1) If you are selecting Data based on 1 condition only, it is a simple condition, and stated on 1 line on 1 Data Selector. (You can also have no Conditions, in which case all Records will be selected.)
2) If you are selecting on multiple conditions (2 or more) and only 1 condition needs to be satisfied, you will use "OR" and can safely state all the conditions on 1 Data Selector, 1 condition per line.
3) If you are selecting on multiple conditions (2 or more) and ALL conditions must be satisfied, you will use "AND" and can safely state all the conditions on 1 Data Selector, 1 condition per line.
4) If you are selecting on multiple conditions (2 or more) and it fits neither 2 nor 3 above, then you may need to split the conditions between nested Data Selectors.
   a) Choose the rule that will eliminate the most unwanted Records to execute first.
   b) Restrict the conditions on any one Data Selector to either "OR" or "AND".
   c) Remember that each Data Selector, when executing, operates on the results of the previous Data Selector.