AXIOM User and System Data Handling in Integral Enterprise Viewer (IEV)

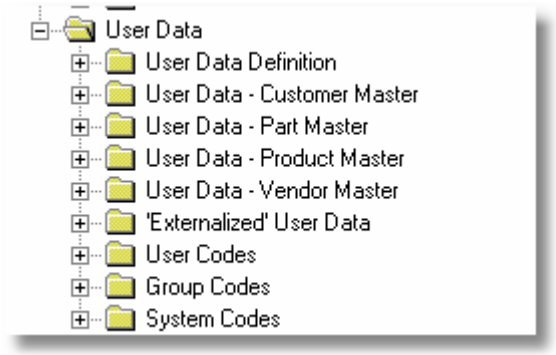
General

In order to assist user in creating and using inquiries in IEV that access User Data and System Data, a series of functions and inquiries have been developed.

The functions are T-SQL User Defined Functions that allow an inquiry developer to access and use user data as if they were regular fields in the database. The inquiries assist with finding and browsing the user defined field definitions, and with a click of the mouse, allows the inquiry developer to paste the exact syntax of the function into a field definition in IEV.

Inquiries

There are now several inquiries that can be used to find user data, as seen below:



As you can see, there is a general “User Data Definition” inquiry, which is the place to start when you need to include a User Data field in an inquiry. This inquiry allows you to browse and construct the needed function with a right-click of the mouse (more on this in the “example” section of this document). This inquiry is based on a T-SQL Stored Procedure that parses out the user data definitions table so that each user defined field gets returns a record of its own.

There are also inquiries for User Codes, Group Codes, and System Codes, which also allow you to browse this data, and construct the needed function with a right-click. Multiple scripts are supplied so that returning different elements such as Abbreviation, Short Description, Long Description, etc. are a snap.

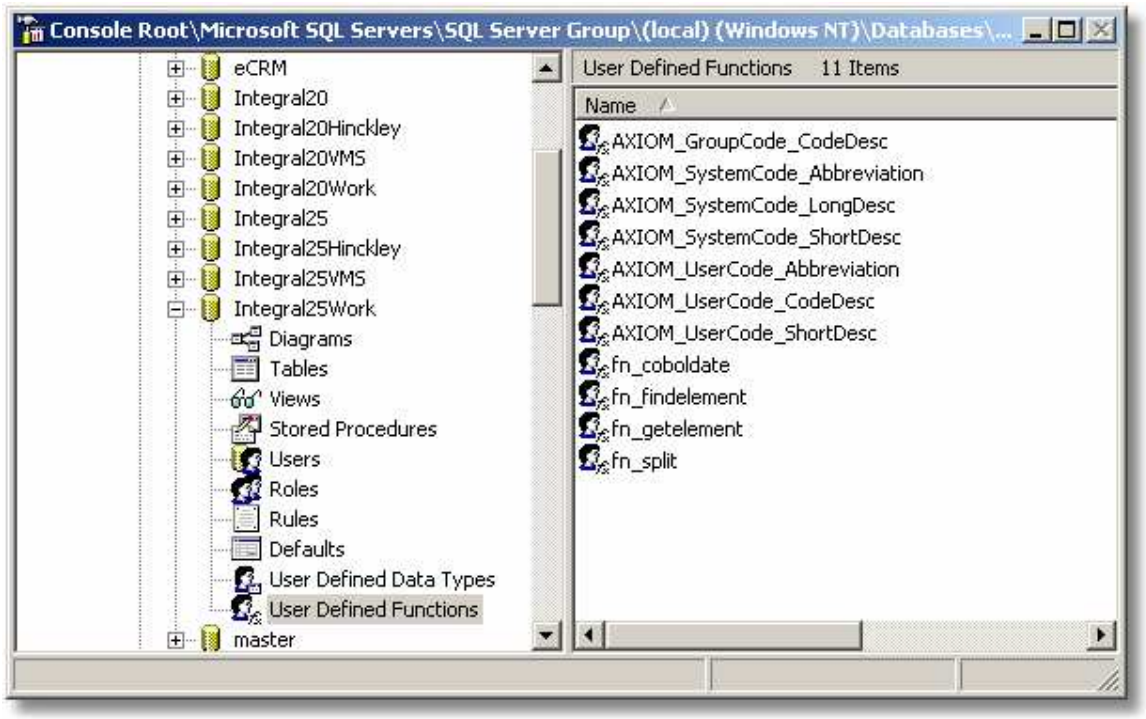
Also included are inquiries that allow you to browse the actual user data (not just definitions) for key areas of the system. At the time of this writing, there are a limited number of these – but more can be easily developed. These allow you to see the entire user data values for these areas – but can also be used to verify

that you have the right user field ID. Drill-downs to these inquiries are available from the primary “Definition” inquiry as well.

Functions

Several T-SQL functions have been developed to make returning user data in IEV a snap. Also, since they are T-SQL functions, they can behave exactly the same as “normal” database fields do – you can filter and sort on these as if the data being returned is native to the table in the database!

Here are the function definitions viewed from SQL Enterprise Manager:



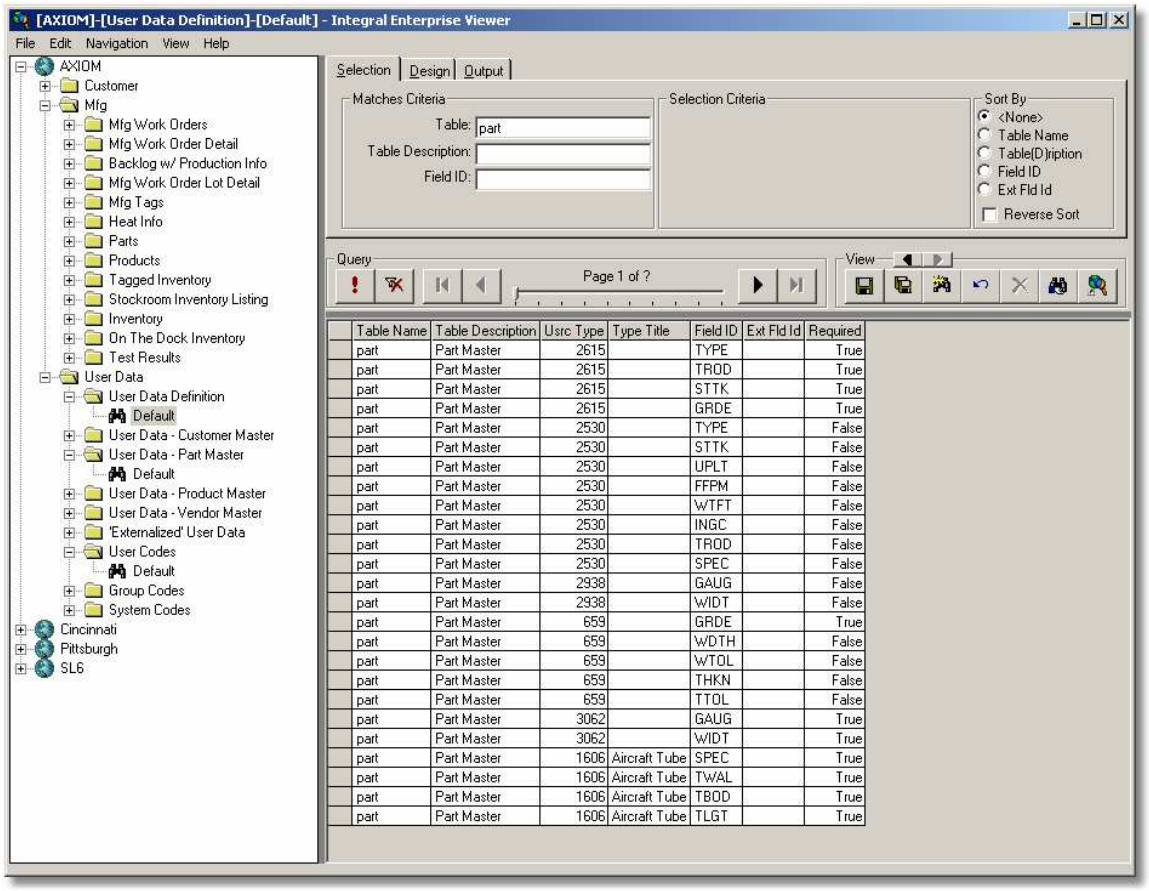
Each of these functions can be used to return data automatically from AXIOM, as seen in the examples below.

User Data Example

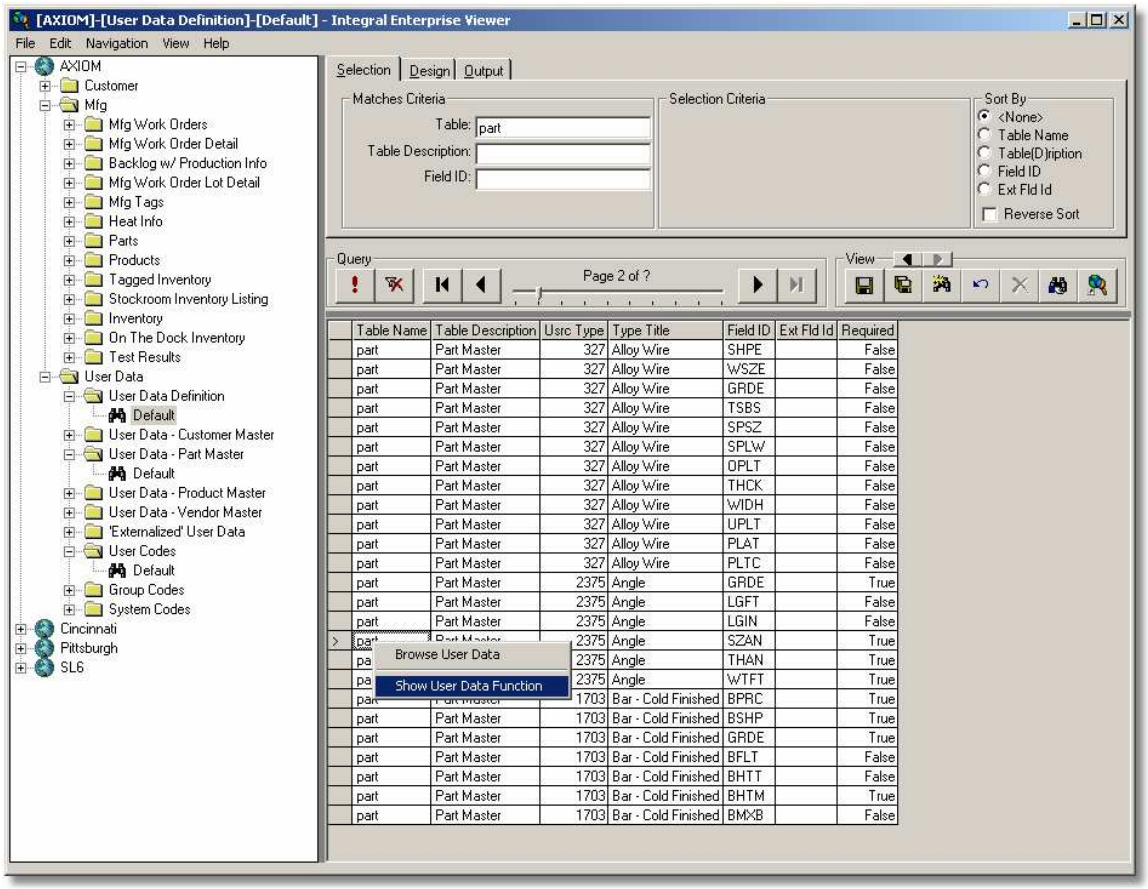
Let's say that I have a user data field in AXIOM for the Part Master table. It's under the "Angle" data types, and it's the Size field.

I can easily tell from AXIOM that the Field ID for this field is "SZAN".

So first, I bring up the "User Data Definition" inquiry and type in "part" in the table name field in order to filter the results:



Next, I browse through the records until I see the field definition I want, select that row, and then right-click and choose "Show User Data Function".



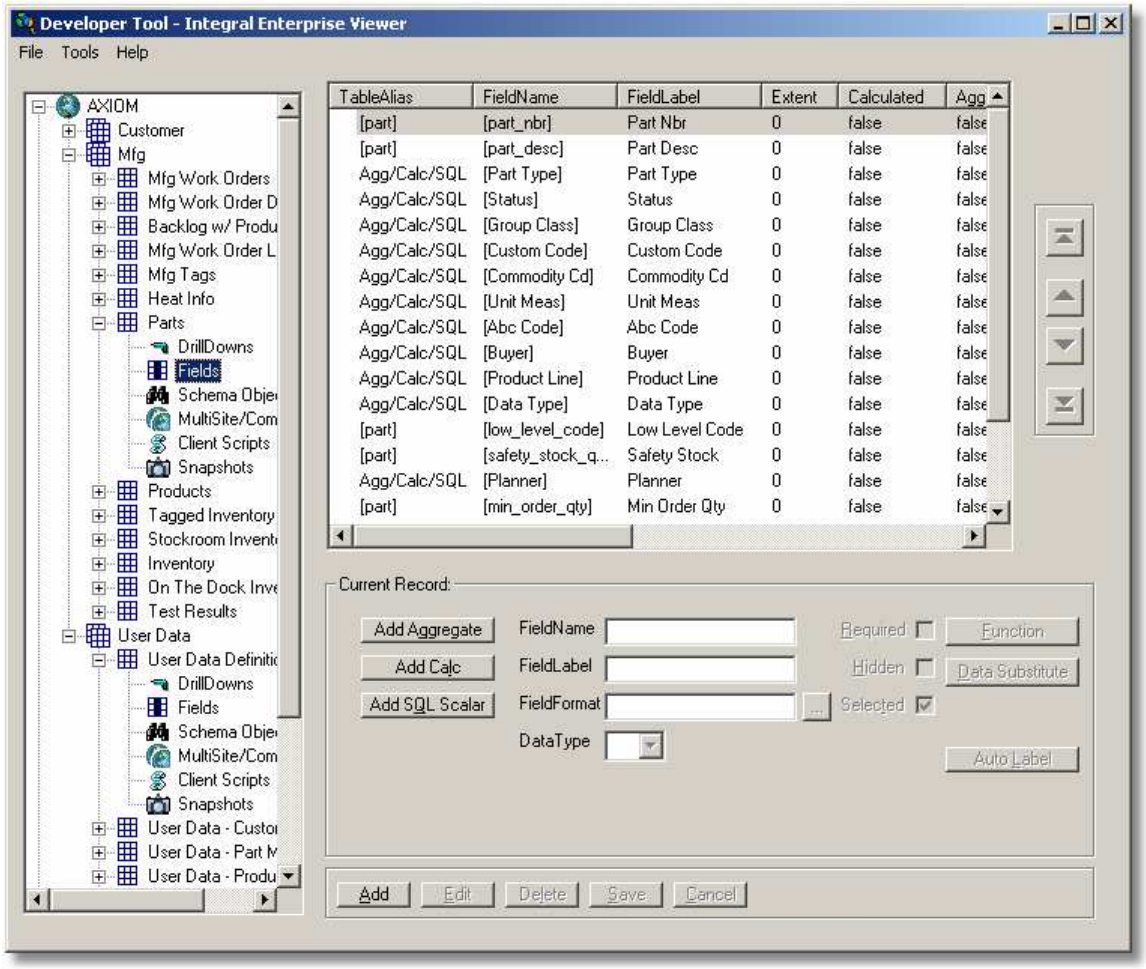
The following message box appears:

As you can see, it spells out the function name as well as the parameters needed for the User Data function. In addition, it automatically places the function in the Windows clipboard for you to be pasted into a field definition.

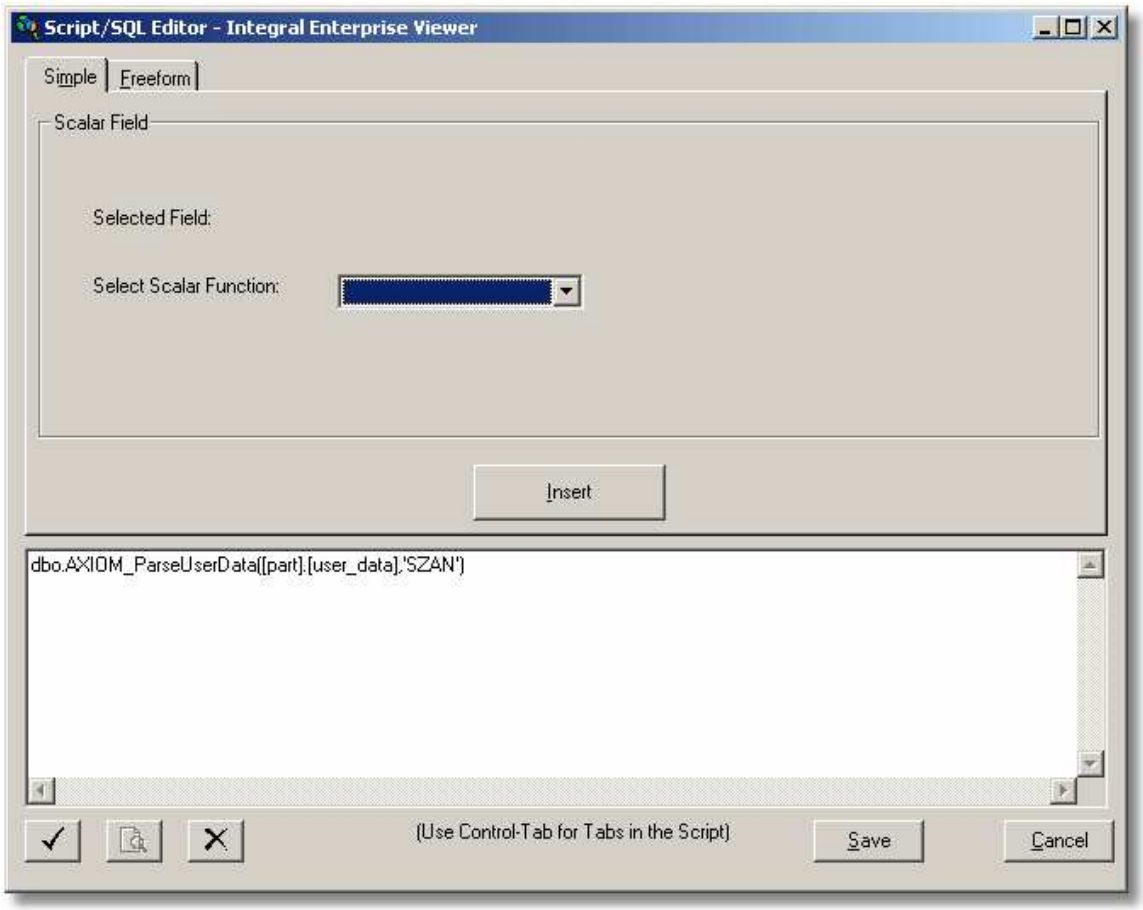
I could also choose to drill-down to the “User Data – Part Master” inquiry to see the data for this field, and thus verify that it’s the one I want.

So now that I have my function, I can bring up any inquiry that has the [part] table listed as one of the tables in its join. I’m going to use the existing “Parts” inquiry.

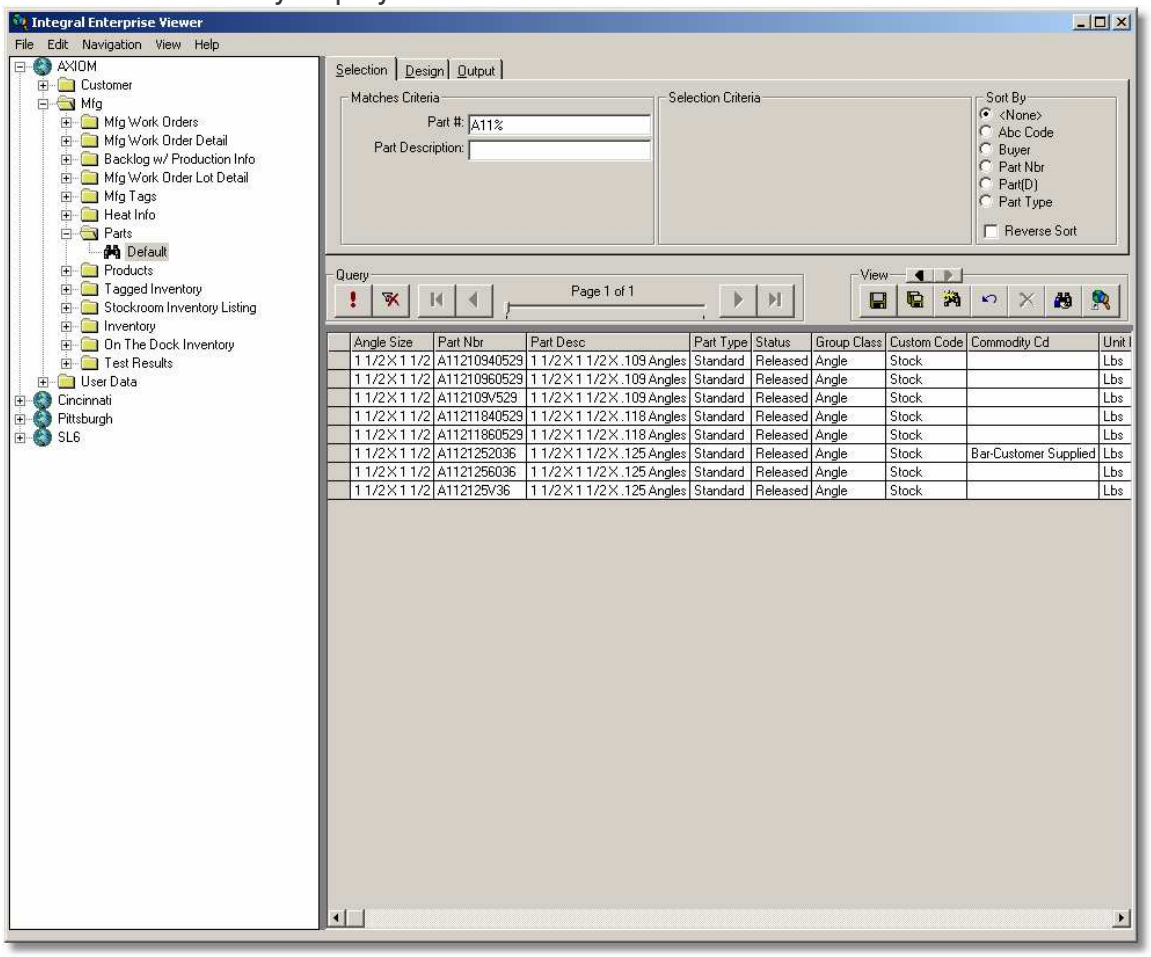
First, I bring up the IEV Developer Tool and open up the Parts inquiry, and select the “Fields” collection:



Next, I simply press the “Add SQL Scalar” button, give the new field a name and a label, and choose the correct data type (in this case “chr” for character data). Next, I press the “Function” button, which gives me a builder to where I can paste in my User Data function:


Script/SQL Editor - Integral Enterprise Viewer
Simple
Freeform
Scalar Field
Selected Field:
Select Scalar Function:
Insert
dbo.AXIOM_ParseUserData([part].[user_data],'SZAN')
(Use Control-Tab for Tabs in the Script)
Save
Cancel

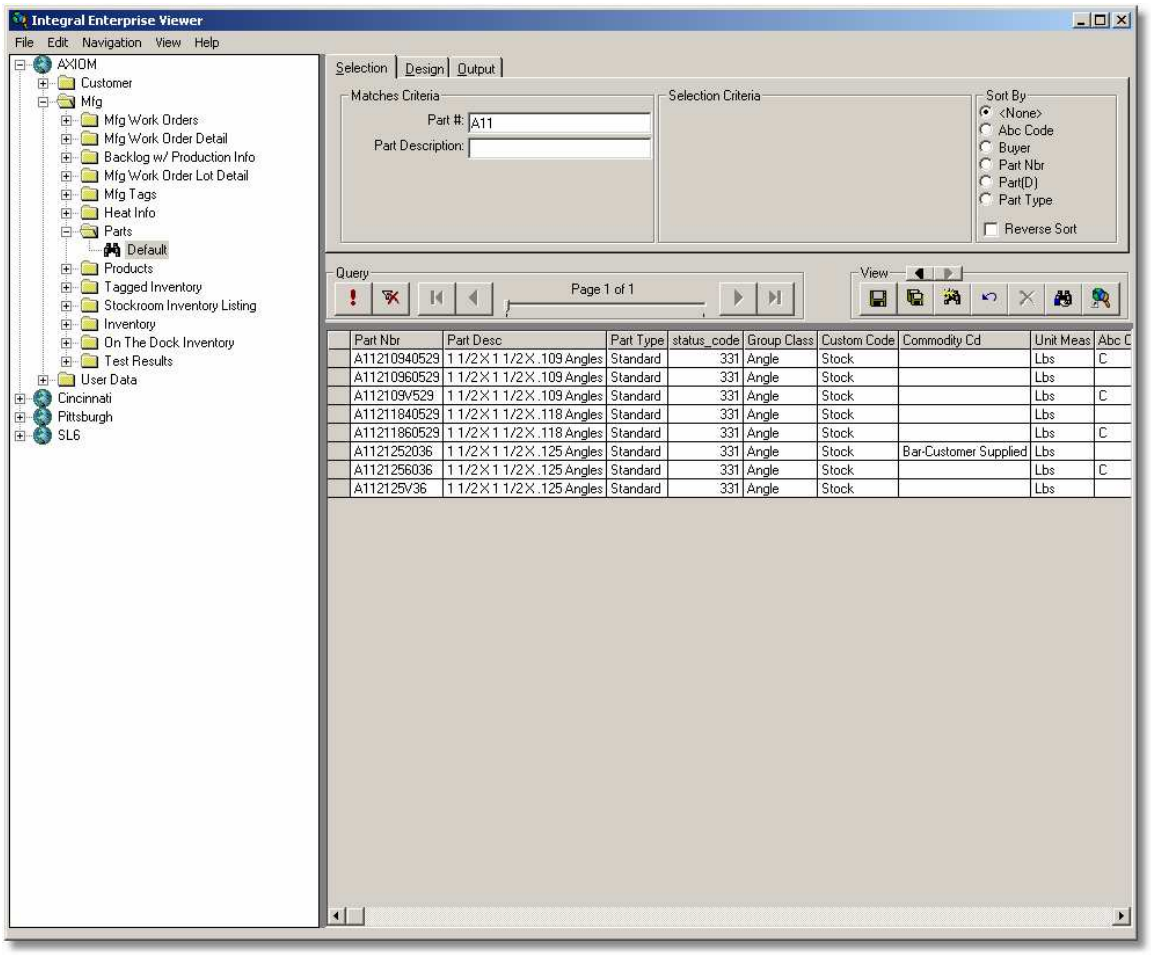
Next, I press Save, and Save again to save the entire field definition – and that's it. Now I can test my inquiry:



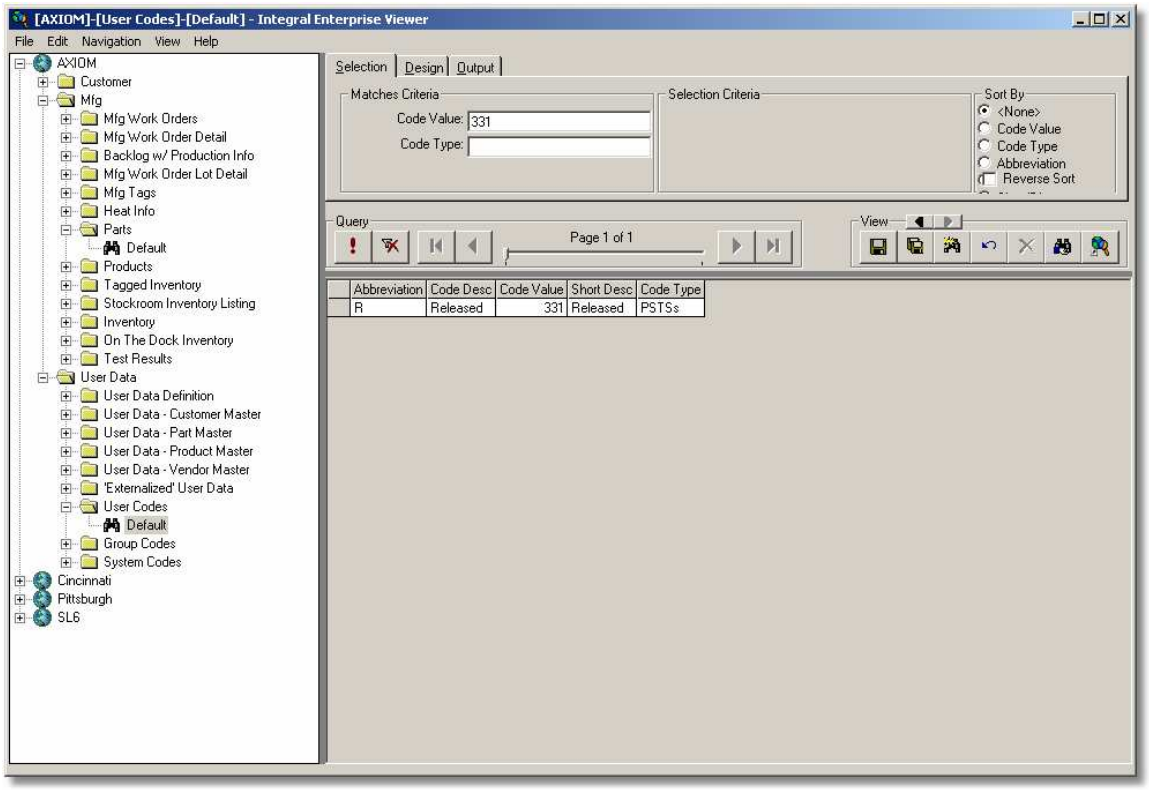
As you can see, the data appears. At this point I can use this new field in my Sort By or as a filter field.

User Codes Example

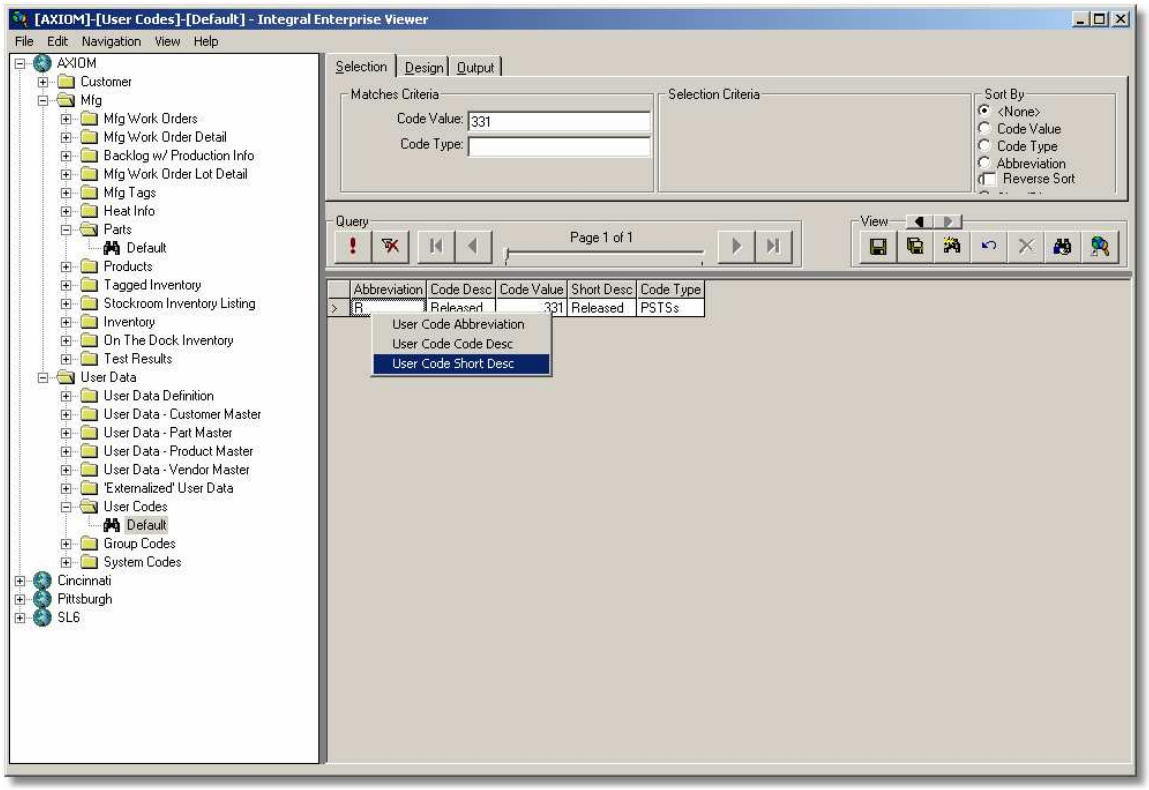
Now let's say that I want to add Status information to the Parts inquiry. I notice that when I add the "status_code" field from this table, that it contains a numeric value – which is a sure sign it's a user or system code definition.



First, I note the status_code – in this case "331". Next, I launch and run the "User Codes" inquiry, and type in 331 in the "Code Value" field:



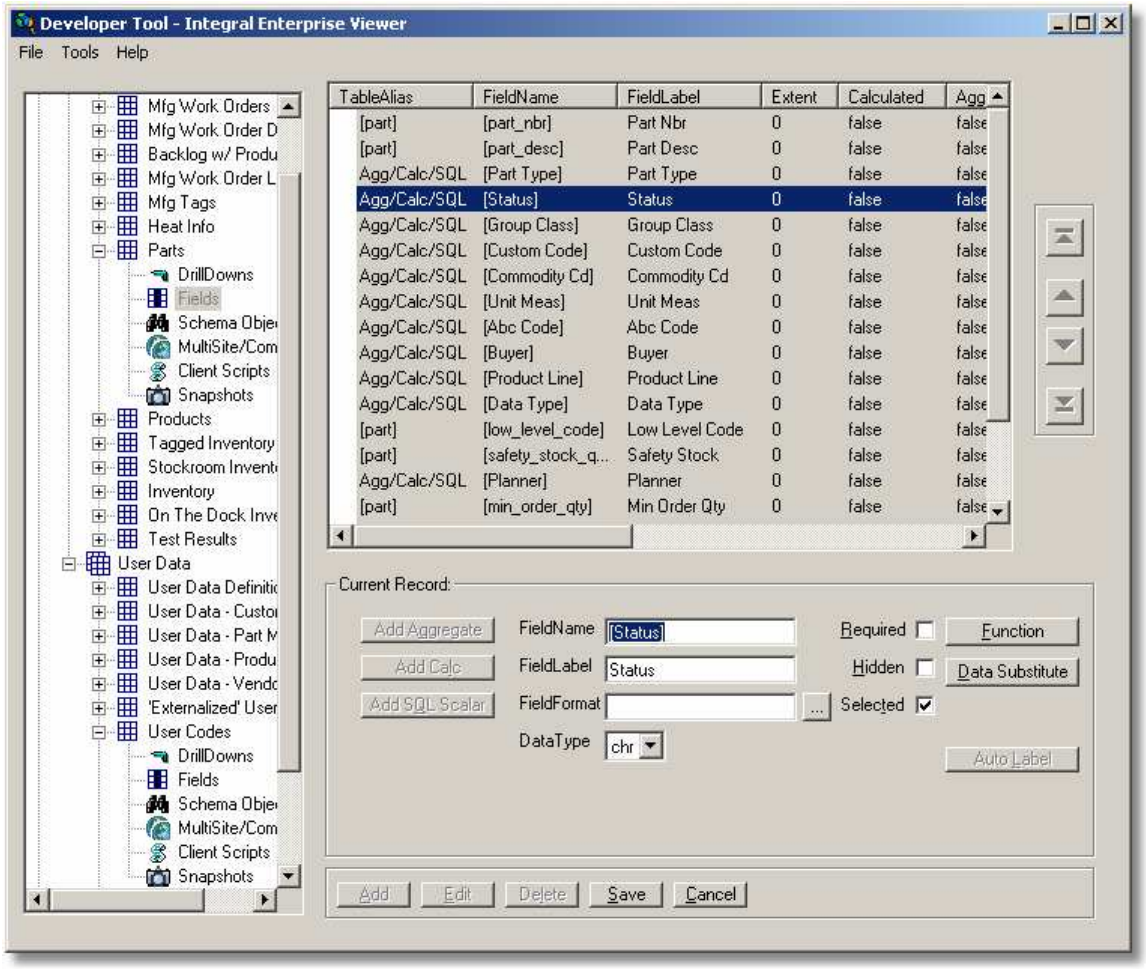
This result is logical. So I simply right-click on the row, and click on the script option “User Code Short Desc”.



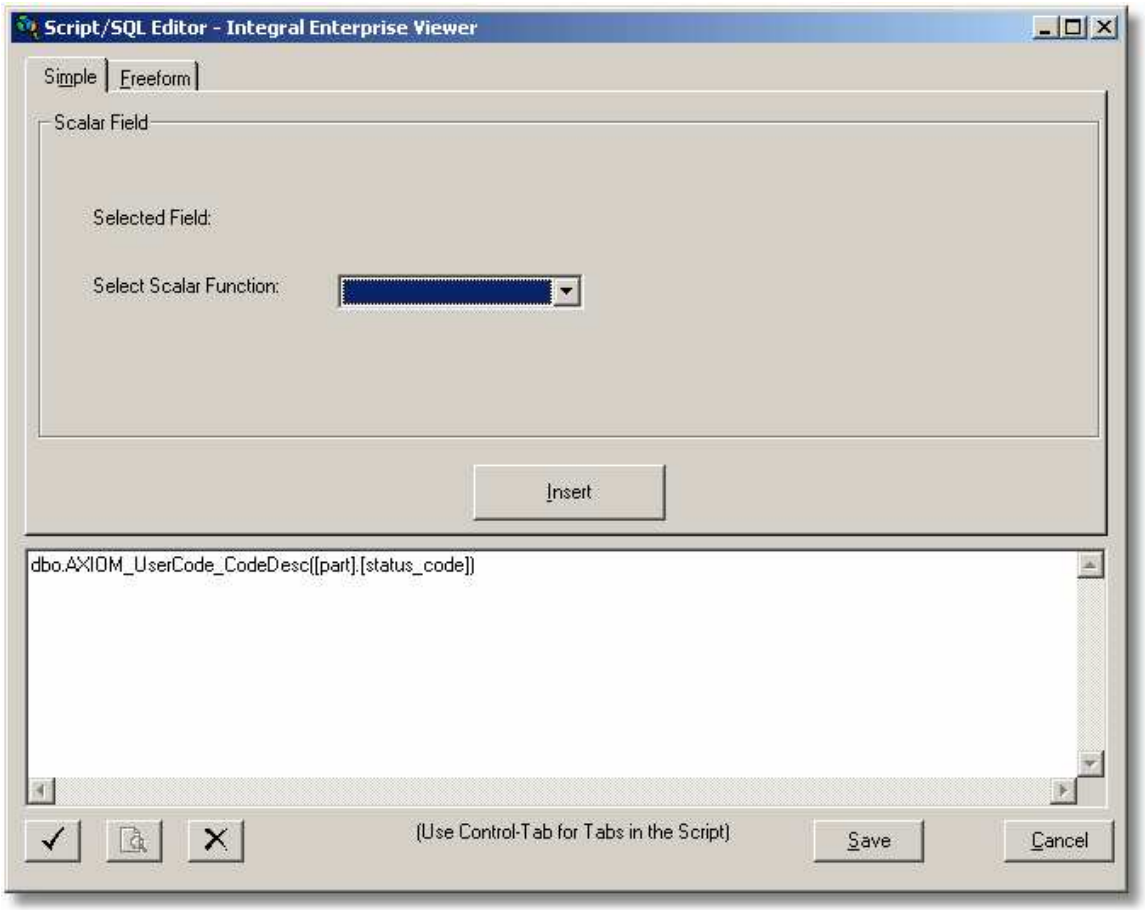
This makes a message box appear:

Which again gives me the function that I need, and automatically copies it to the clipboard. Notice, however, that in this case I will need to replace "[table].[code_value_field]" with the actual table and field names where the code value resides. In this case, it is the "part" table, and the "status_code" field.

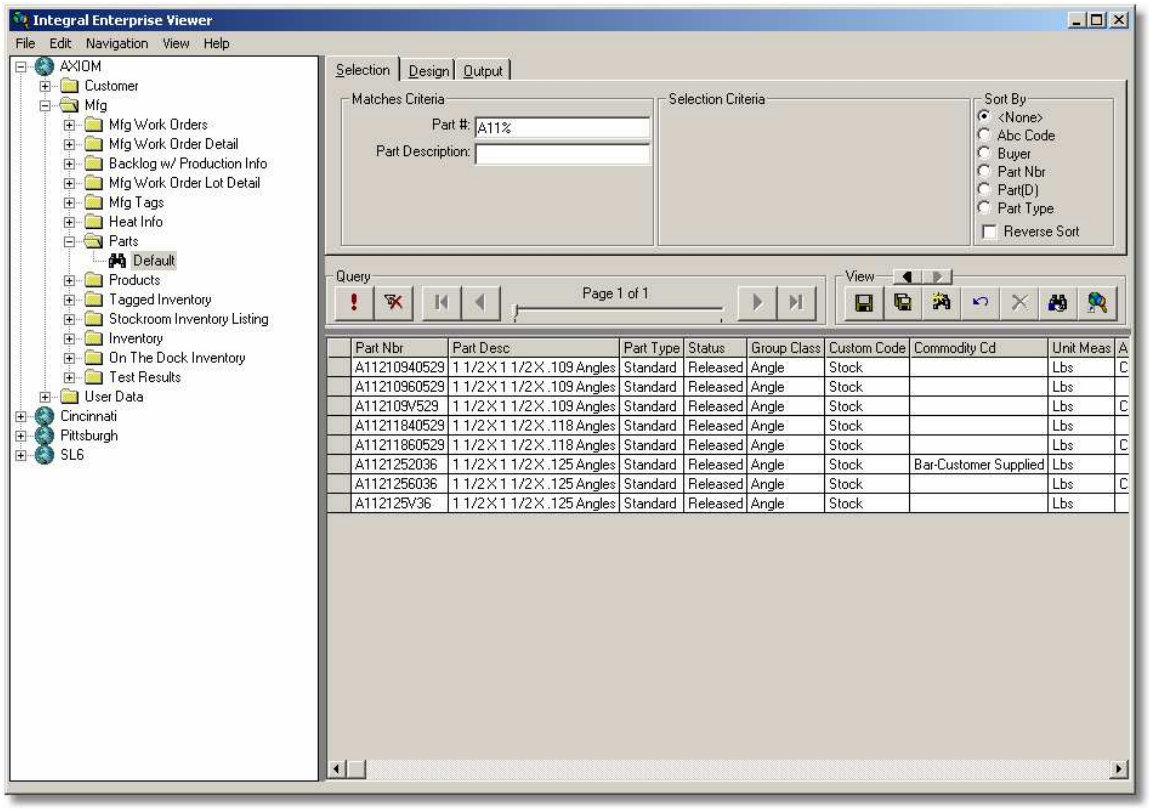
So in the IEV developer tool, I first delete the existing "status_code" field – it no longer needs to be referenced. Then, I add a new field simply called "Status". Again, this will be a SQL Scalar function with a return type of "chr":



And copy the function into the field (and edit the table and field name):



And as you can see, I can now use this inquiry in IEV:



And if I wished, I could use this new field in my Sort By and/or filters.