$DXM – DBX Dictionary Maintenance Utility

1.   Introduction

The information about the structure of a DBX Speedbase database is stored in a DBX data dictionary. Before you can actually create your DBX database, or compile a program using a DBX database, you must first generate a DBX dictionary.

DBX dictionaries are generated from DBX meta-dictionaries (a specialised Speedbase Database). The $DXM utility allows you to create and maintain DBX meta-dictionaries, to generate a DBX dictionary from them, and to produce copy libraries containing the information held in the generated dictionaries.

DBX dictionaries have the same naming conventions as their standard non-DBX variants, and can be distinguished from them by a file type of D2 instead of DD.

2.   Installation

$DXM is installed with the Speedbase Development Utilities.

3.   Database Restrictions

Standard Speedbase database limitations such as the number of record types that may be defined in a database and the number of Master/servant and GVA relationships that may be defined have been significantly extended for DBX databases.

The number of Tables (Record types) that can be defined for a given DBX database is 512.

The number of indexes held in a DBX database is unlimited. Each table's record may contain up to 64 indexes, and each index may be composed of up to 16 segments. The length of each index key cannot be more than 123 bytes.

Note: Pervasive SQL does not permit more than 119 index segments for any one Table. If you define a table that requires more than 119 segments to define its indexes, then the dictionary may only be processed using Microsoft SQL.

The number of Master records that may be defined for any given servant record for a DBX database is 16.

There is no restriction to the number of GVA fields that may be declared for a record. The size of each Data Record's Systems Area (which contains GVA values) is unrestricted. The number of GVF fields declarable for any record type has no specific limit.   A field maybe both a GVF and GVA.


Whenever the Speedbase Database Manager retrieves a record, it copies significant fields from the target record into a special area known as the KE (Key Extract area. This area is 4K in size. The maximum number of segments defined for each record type (i.e. field references required to define Index, Master access Key and GVF relations) is 512 bytes in size. Given that Pervasive SQL files are limited to 119 segments, this unlikely to present a practical limitation.

Note that all GVF fields, Index Segments and Master Access key fields **must** be declared within the first 127 field definitions of each record type.

A DBX database can hold up to 2G of records per table and therefore the DBX record number field is a 4 byte PIC 9(9) C integer.

4. Running $DXM

When $DXM is run an initial menu is displayed.
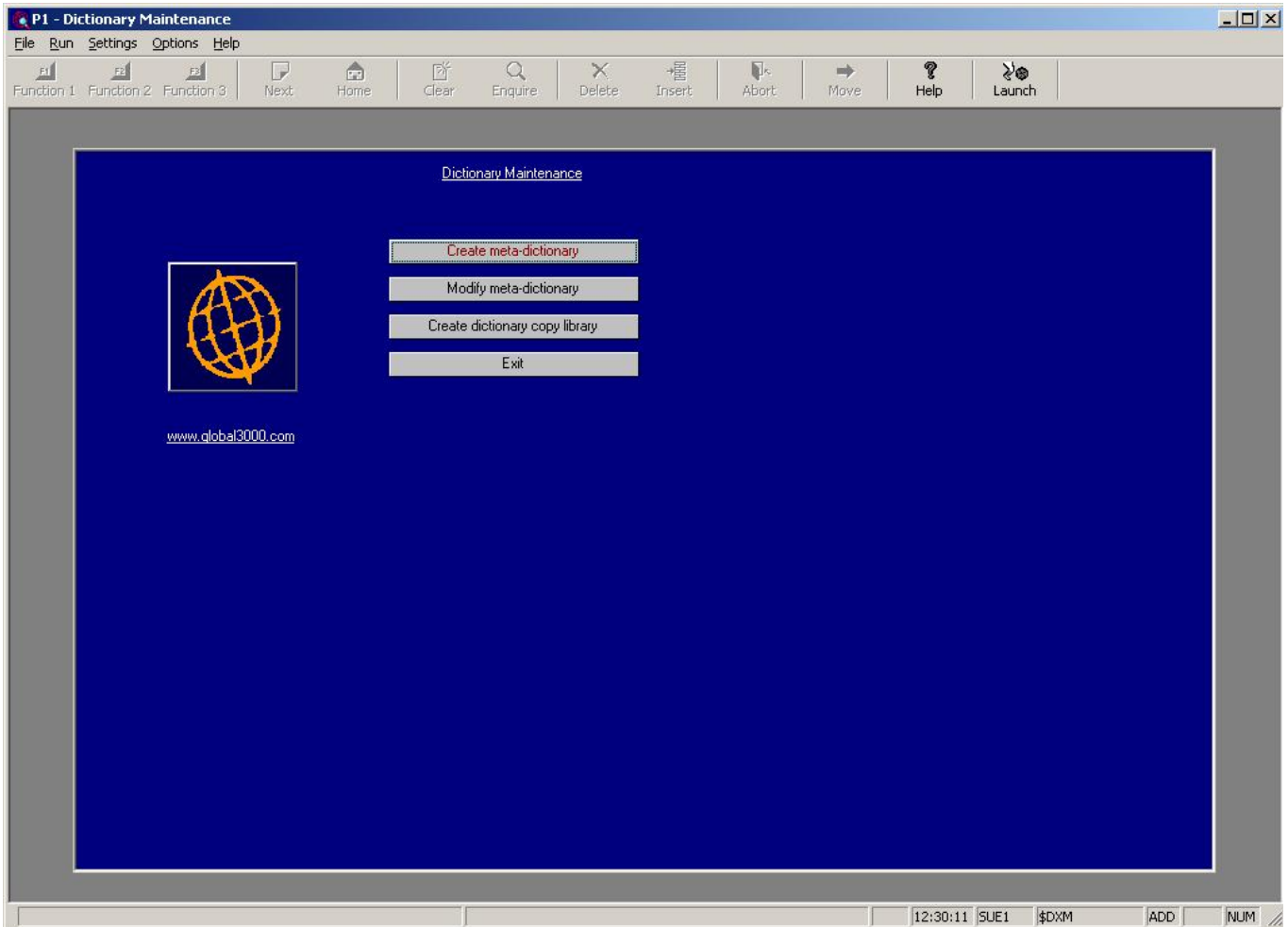
The 'Create Meta-dictionary' option allows you to create a new meta-dictionary, the 'Modify Dictionary' option to manipulate the dictionary tables, and the 'Create Copy Library' option to build a copy library from an existing dictionary.

5. Create Meta Dictionary

The 'Create Meta Dictionary' option from the main menu enables you to build an empty meta-dictionary database. You are first asked for the number of meta-dictionary elements you require:
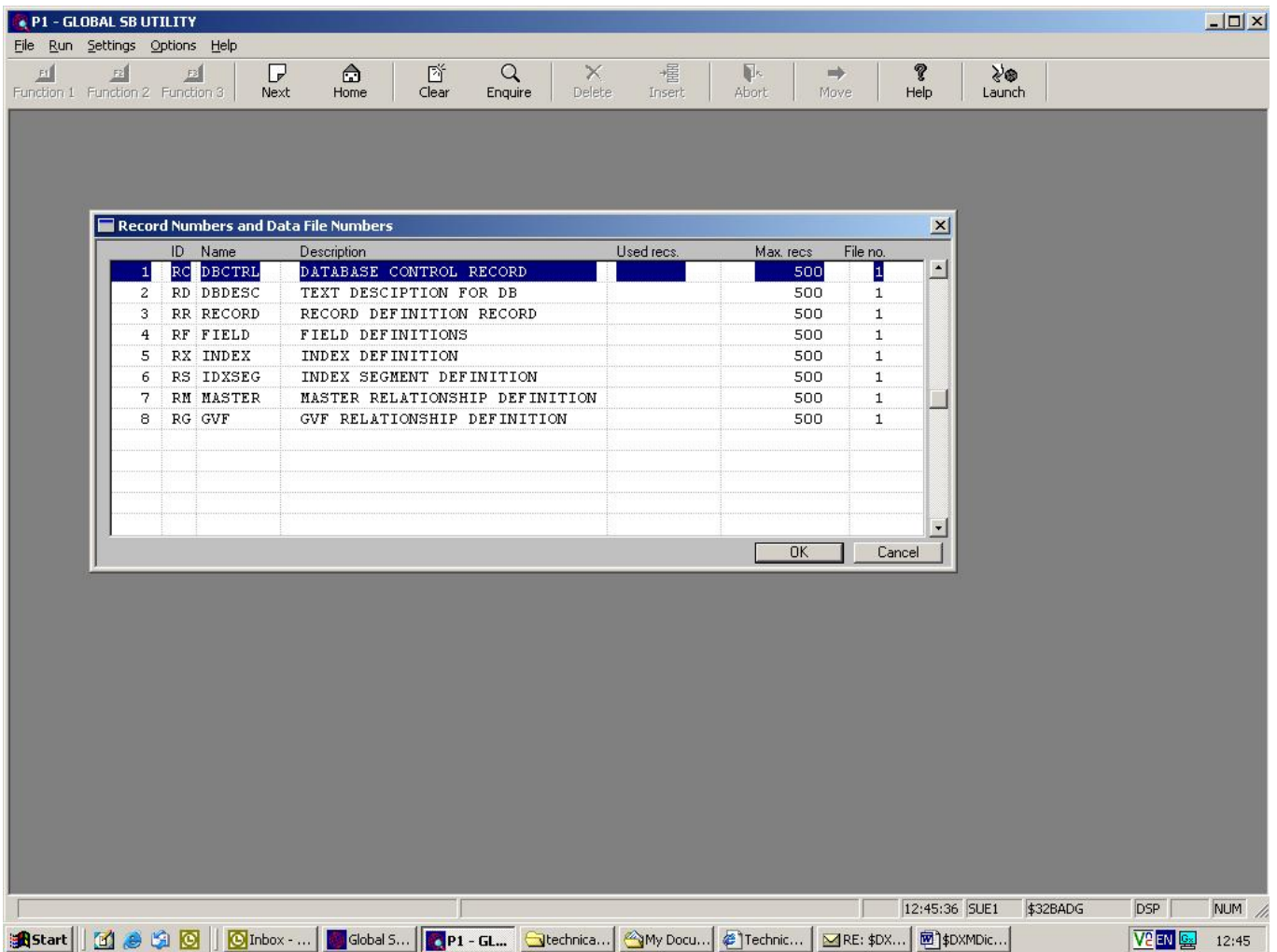
The meta-dictionary database records are as follows:

- Database control records. These contain general meta-dictionary control data and you should not need more than one of these.
- Text description records. A record is required for each line of text database description, record comments and other text description in the meta-dictionary
- Record definition records.   These contain general information about the record tables in the dictionary. A single record is needed for each table that you require.
- Field definition records. A record is required for each field defined in the dictionary to hold all the information on the field.
- Index definition records. These contain information about all the indexes in the database. A single record is required for each index defines.
- Index segment definition records. A record is required for each index segment in the database to hold segment details.
- Master servant relationship records. The master/servant relationship records are required to describe the relationship between a master record and its servants. You may require several records for each relationship.
- GVF Relationship definition records are used to define group value field and group value accumulator (GVA) relationships. A record will be required for each relationship.

It is preferable to be generous in the allocation of the maximum number of each type of record, since you would not wish to run out of space whilst amending the meta-dictionary. Once you have entered the maximum number of each type of record, you should press the 'OK' button and proceed to enter the name of the meta-dictionary you want to create, and the unit on which to create it. We suggest you adopt the convention that the meta-dictionary name is a lower case version of the final dictionary name.

6. Modify Meta-Dictionary

To modify an existing meta-dictionary you should select the 'Modify meta-dictionary' option from the main $DXM menu. A window asking for the name and unit of the meta-dictionary to be amended will appear and should be replied to. If the meta-dictionary is empty you will be establishing a new meta-dictionary otherwise you will be modifying an existing dictionary.

6.1 Establishing a New Meta-Dictionary

If the dictionary is empty and has not been amended before, $DXM will display a menu window as follows:
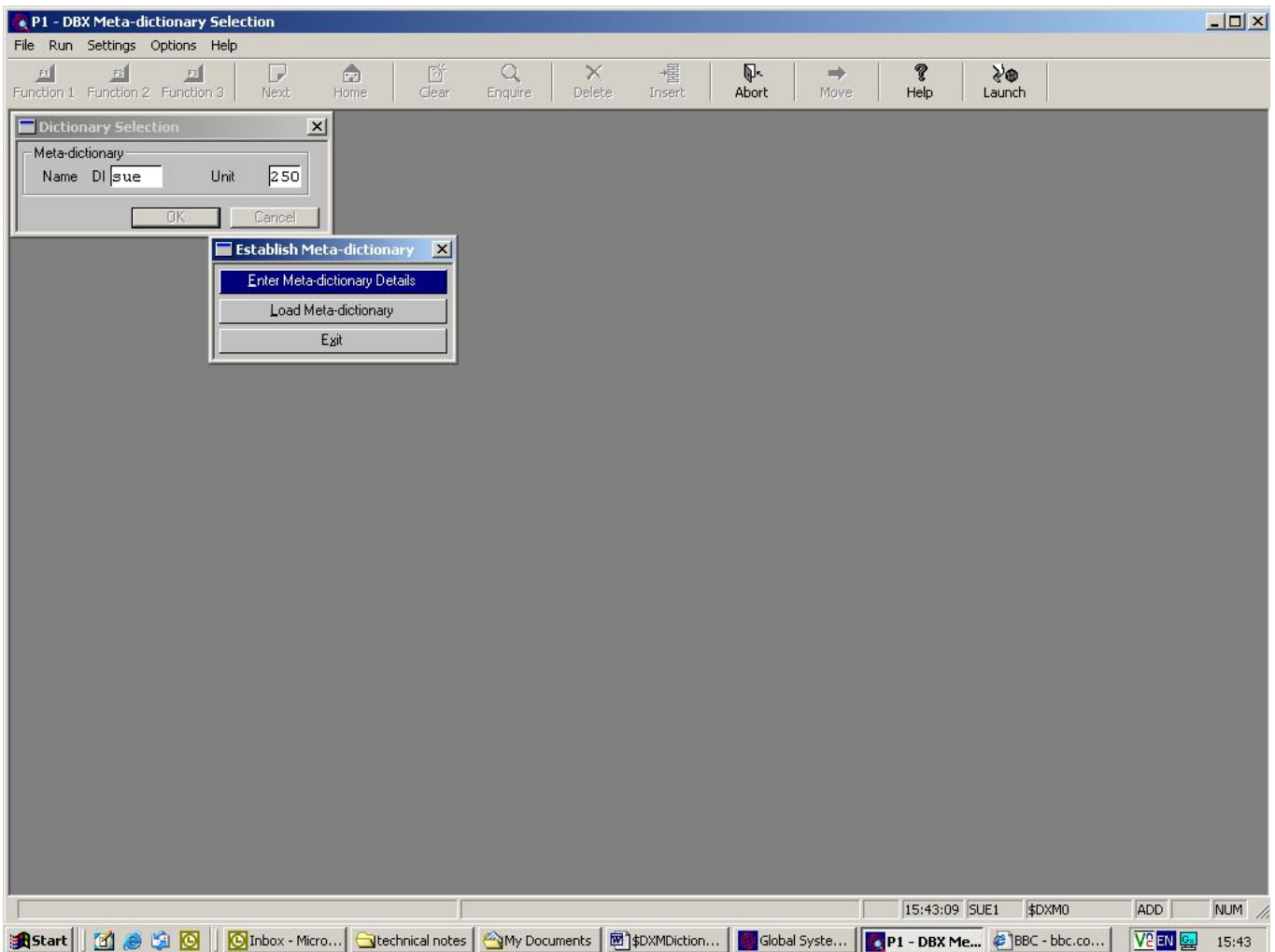
If you wish to simply create a new dictionary, without using another dictionary as a template, then you should select the 'Enter Meta-dictionary' option.
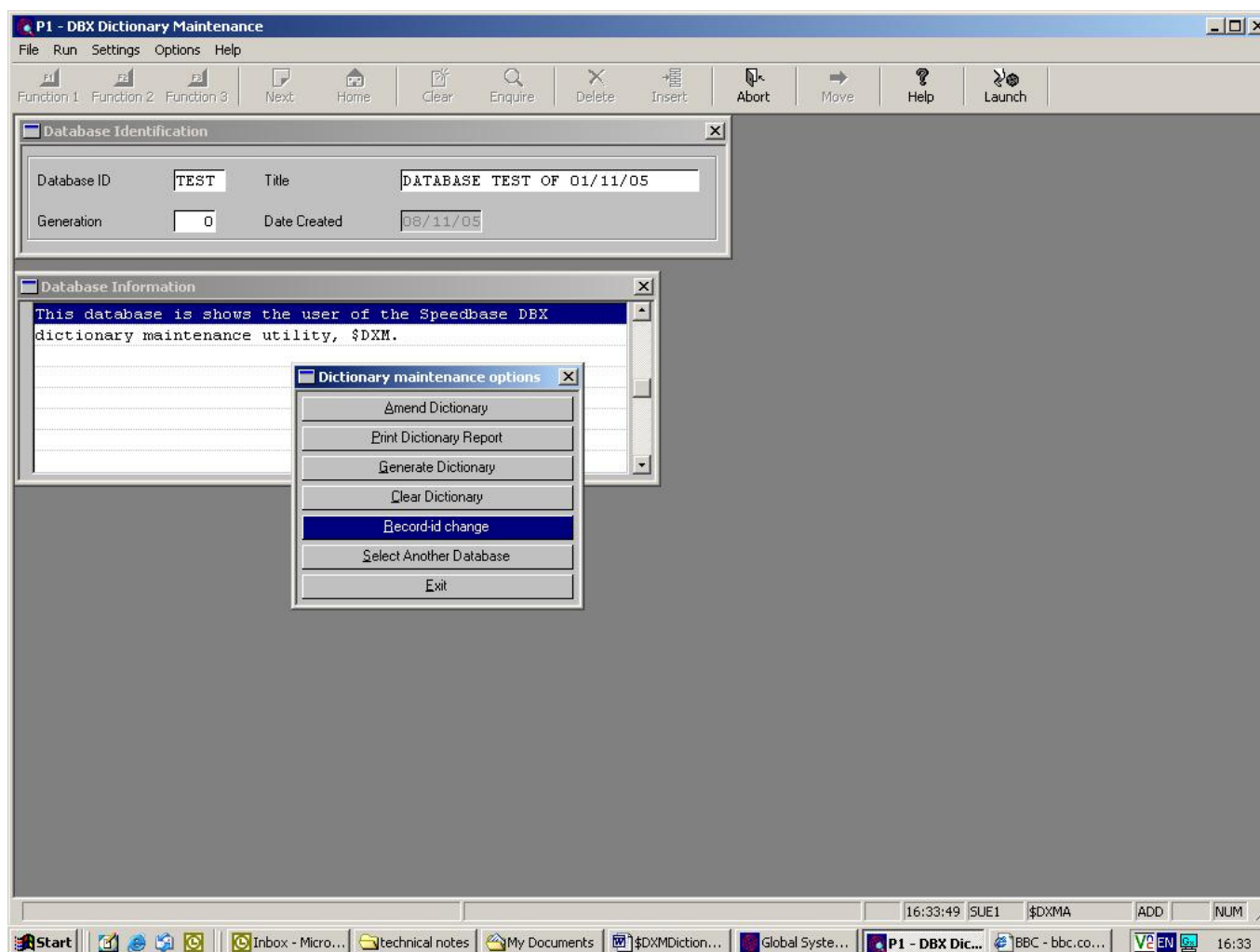
The 'Load Meta-dictionary' option allows you to load the tables of other dictionaries into the new meta-dictionary. You will be asked for the details of the dictionary you want to load. You may load a

a single DBX dictionary and any number of standard, non-DBX, dictionaries provided there are no record clashes and you do not exceed the meta-dictionary capacity that you specified when the meta-meta-dictionary was created. If you need to consolidate dictionaries that currently have clashing record-ids, you should first rename the offending records before attempting to merge the resulting dictionaries.

You will now automatically enter the 'Amend Dictionary' phase.

6.2 Amending an Existing Meta-Dictionary

If the meta-dictionary details have previously been entered the 'Dictionary Maintenance' options list will now be shown:

6.2.1 Amend Meta Dictionary

The 'Amend Dictionary' entry allows you to change the description and tables held in the meta-dictionary. The details in the 'Database Identification' and 'Database Information' windows (as shown above) can be amended.

The database-id identifies the database and is used in the ACCESS statement within the calling program. You would normally set this to be the same as the final dictionary name. The database-id id should not begin with "$$" as this is reserved for the specialised 'Memory Access Method' dictionaries. The database title and information are for descriptive purposes only. The generation number increases for every amendment of the dictionary and should not generally be modified. If,

however, you have previously made an error in the database definition and wish to back-track generation numbers, it is possible to do so by modifying the generation number. Great care should should be taken in doing this as two different versions of the same database with the same generation generation number could cause confusion. When changing generation numbers, it should be noted noted that the generation number you enter will increase by one when the dictionary generation option (described later) is selected.

Pressing the 'OK' button will now show a window listing the record types defined in the meta-dictionary:
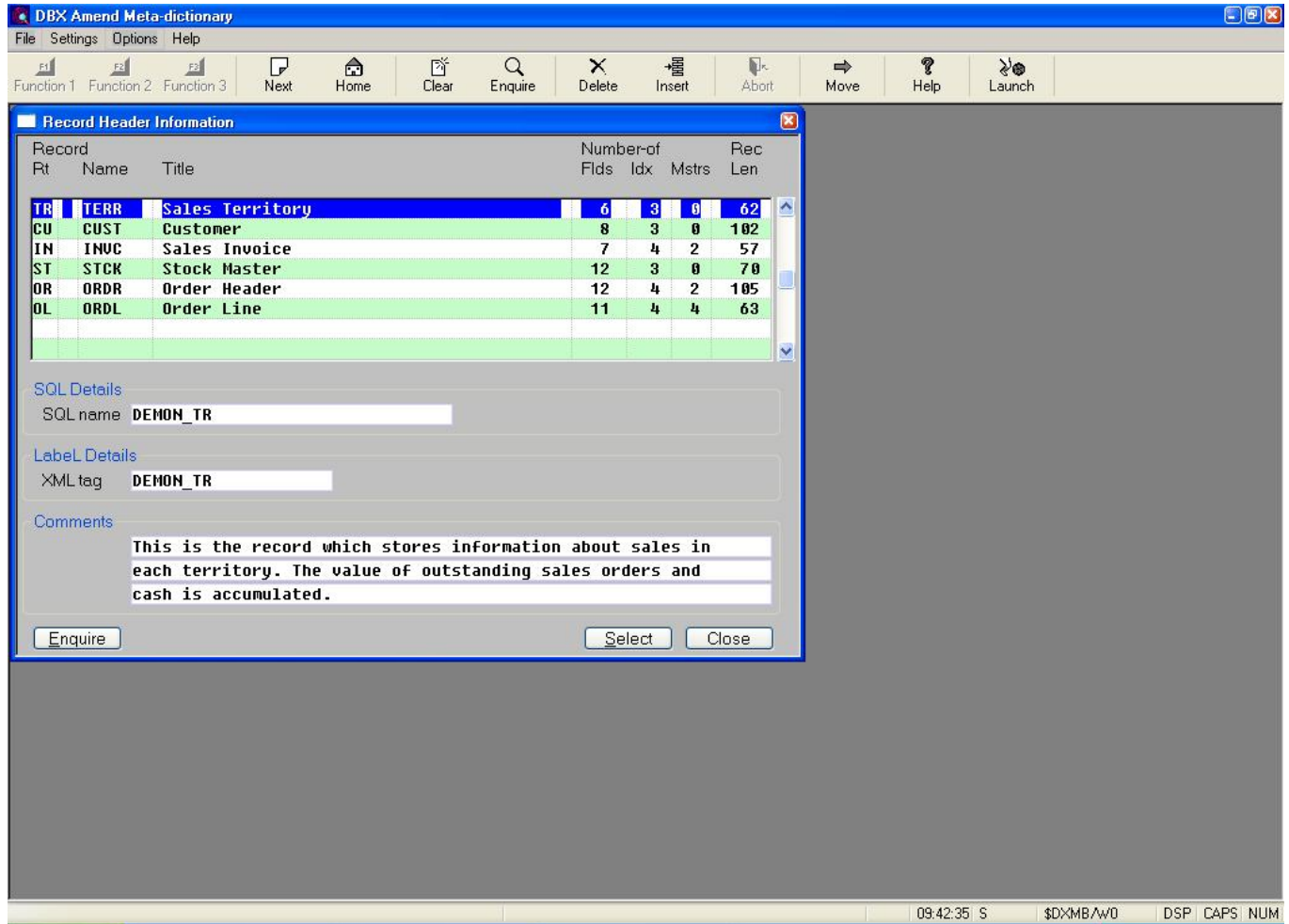
You may amend the record name, title, SQL name, XML tag, and the description of the records shown in the window. The SQL name is required and the first 14 characters must be unique within the database. The XML tag is currently for descriptive purposes but care should be taken when allocating the XML tags as they may be used in the future. $DXM supplies a default of the database name appended with the record name as for both the SQL name and XML tag. The SQL name is the table or field-name shown when SQL queries are later performed on the database. Note that the external names chosen must comply with the naming conventions required by the database engine you choose when you create a DBX database.

New record sets may be inserted by selecting the 'Insert' button on the tool bar or by adding records records to the end of the list. Record hierarchy is important, and servant record sets should not be

defined before master record sets. The 'Move' button is available to allow you to move a record's position.

The 'Enquire' button on the tool bar allows you to view the record sets in a different order.
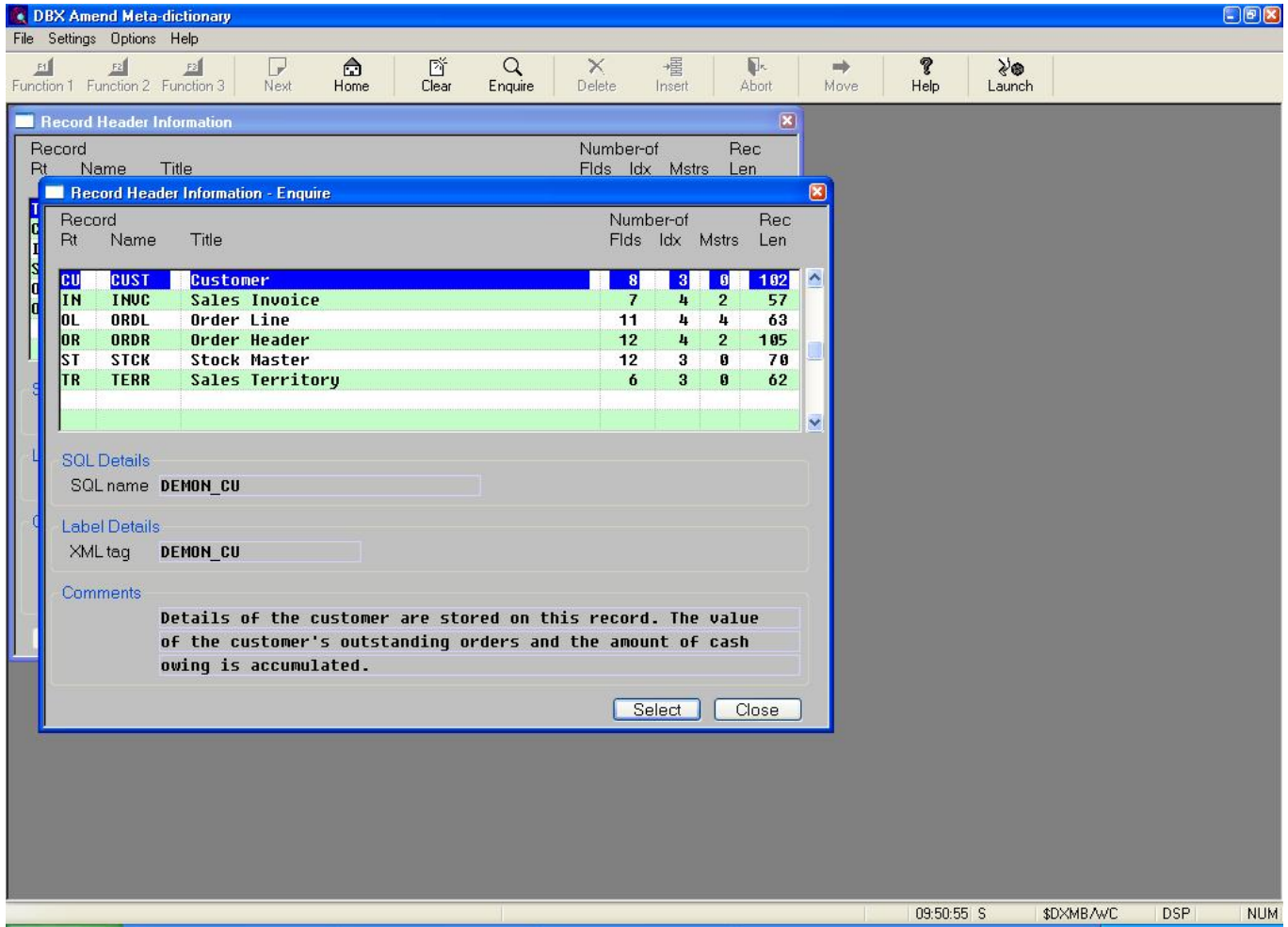

You may not amend the record details in this window but can only view the records.

To amend further details of a record set, you should select that record set by positioning on it using the cursor keys or mouse, and pressing the 'Select' button.

The next window shows which records act as masters to the current record. If you add a new master in the window and the appropriate master access key fields are not present in the current record, then when you select the 'Next' button to continue, you will get a message asking if you want the link fields created. If you reply positively, $DXM adds the appropriate fields to the current record. Otherwise you must delete the master relationship before continuing.
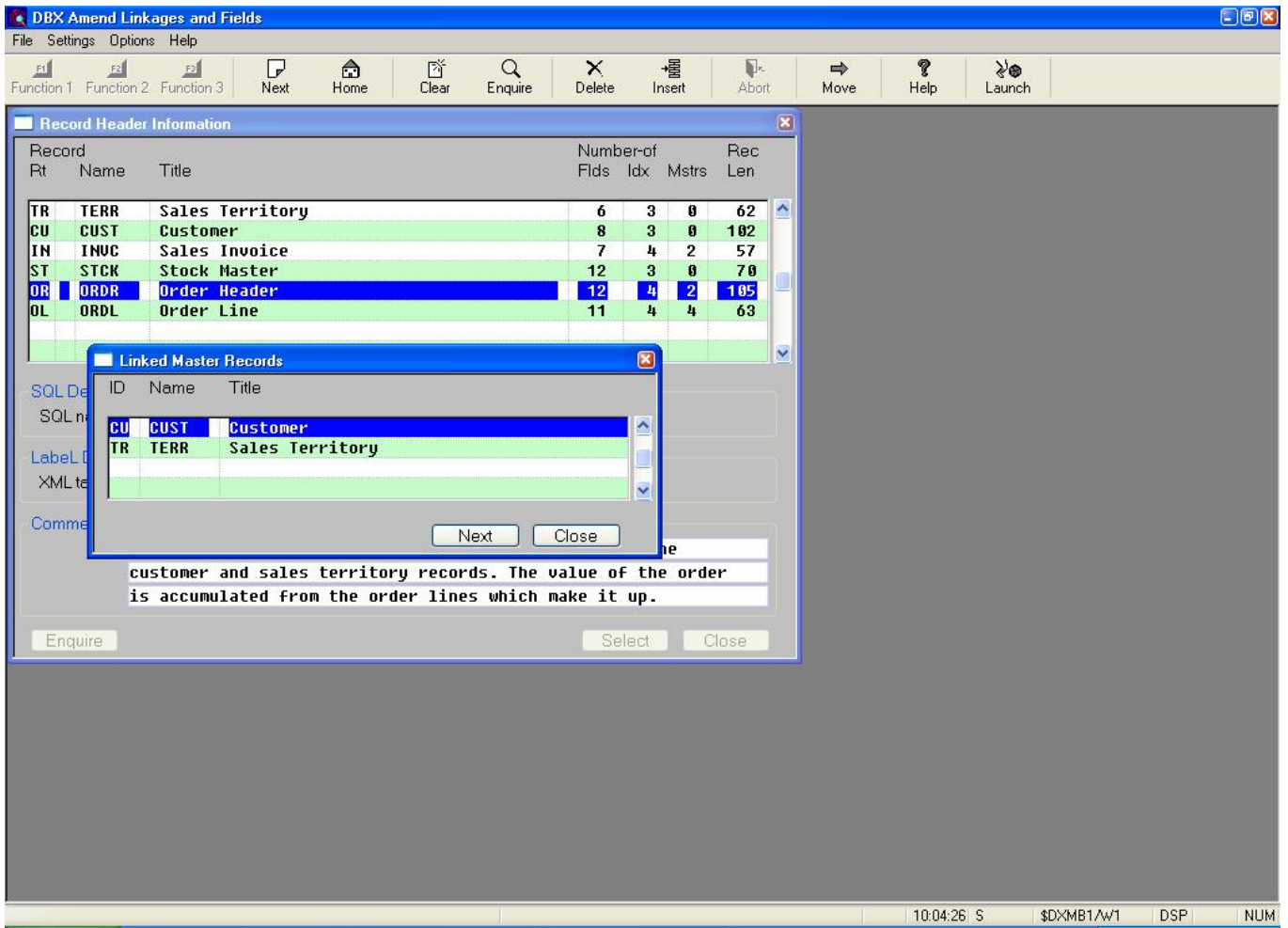
Selecting the 'Next' button from the 'Linked Master Record' window will show the field name table for the current record.
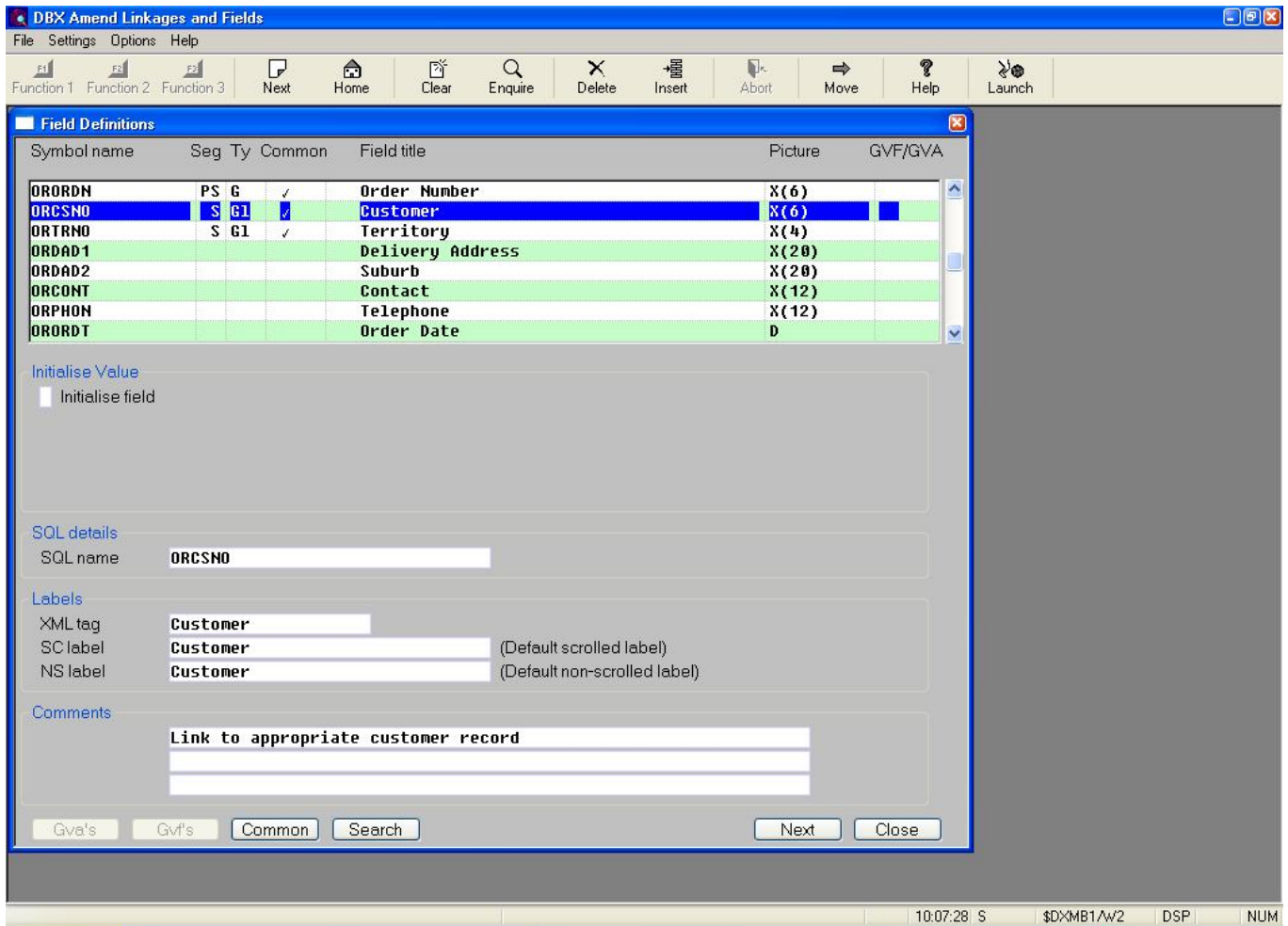
The fields that make up the primary index of the record are marked 'P' in the 'Segment' column. The segment column field may be amended to indicate if a segment is part of the primary index. To change the order of the segments in a primary index you should make use of the move function, changing the order of the fields themselves.

If the field is part of at least one secondary index it is marked 'S' in the 'Segment' column. Secondary indexes are defined in later windows. Fields often exist in more than one record in the master/servant hierarchy. These fields are known as global fields, and are marked with a G in the 'Type' column.

The parameters of a global field, e.g. its name and picture clause must be maintained by you across the master servant relationship. You can avoid having to do this by ticking the 'Common' for the field and all its related fields. If the 'Common' option is set all fields of the same name in the database which also has the common option ticked, will be modified automatically when an individual field common field is amended.

You can view the fields in a common group by pressing the 'Common' button.
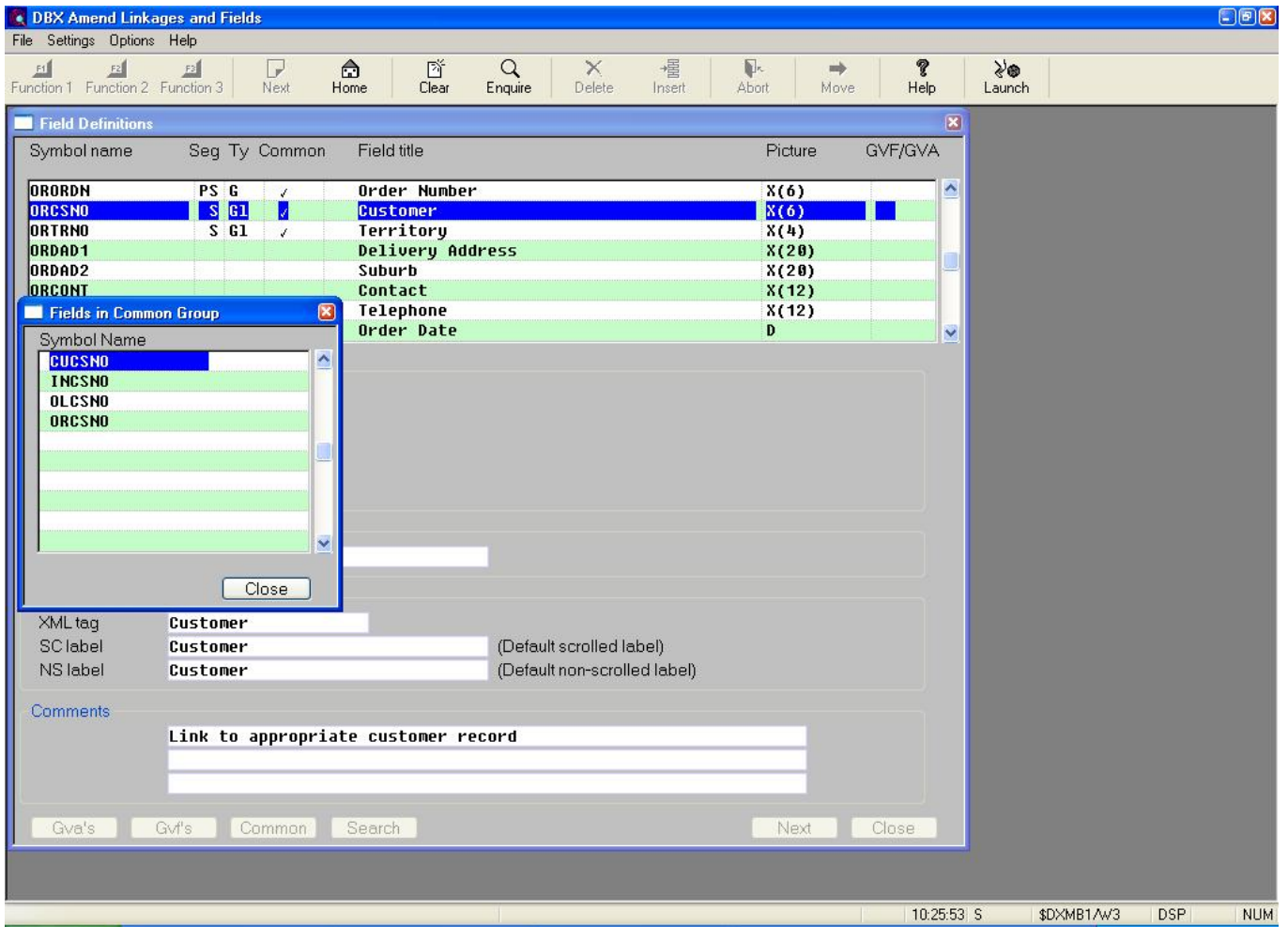
The 'Select' button shows all the fields in the database which have the same name.
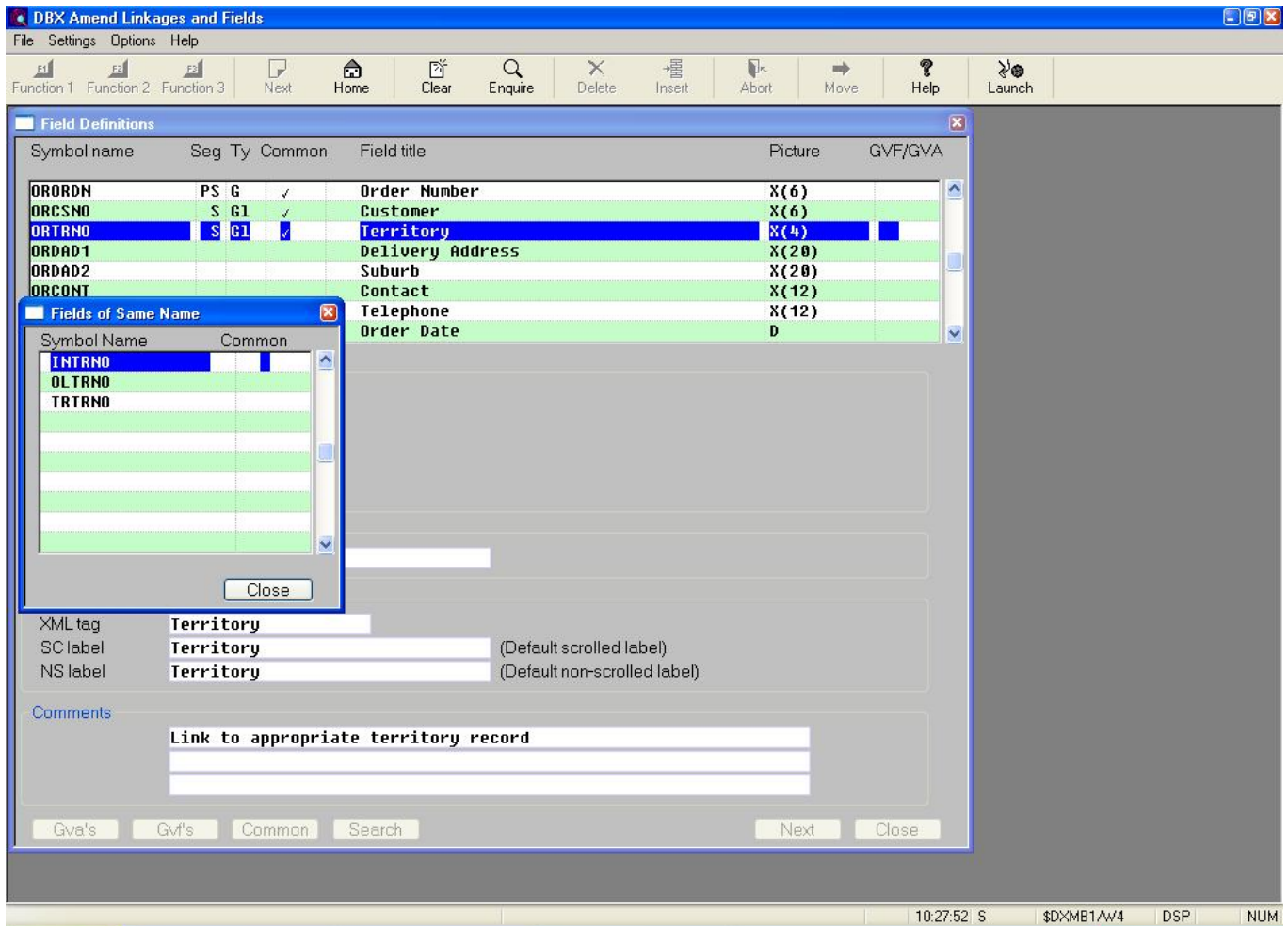
Ticking the 'Common' column allows you to include this field in the common group without having to go back and amend their individual record details.

Global fields provide the links between a record and its masters and comprise the master access key. If a field is part of any master access key it is marked as G1 in the 'Type' column. Other records, which are not part of the current master servant relationship, may contain fields of the same name as the fields in the current record. These fields, however, will not be considered as global fields.

Fields which are value accumulators (GVAs) are marked with an 'A' in the GVF/GVA column. If you position on a GVA field and select the 'GVAs' button, you will be shown the fields contributing to the GVA.
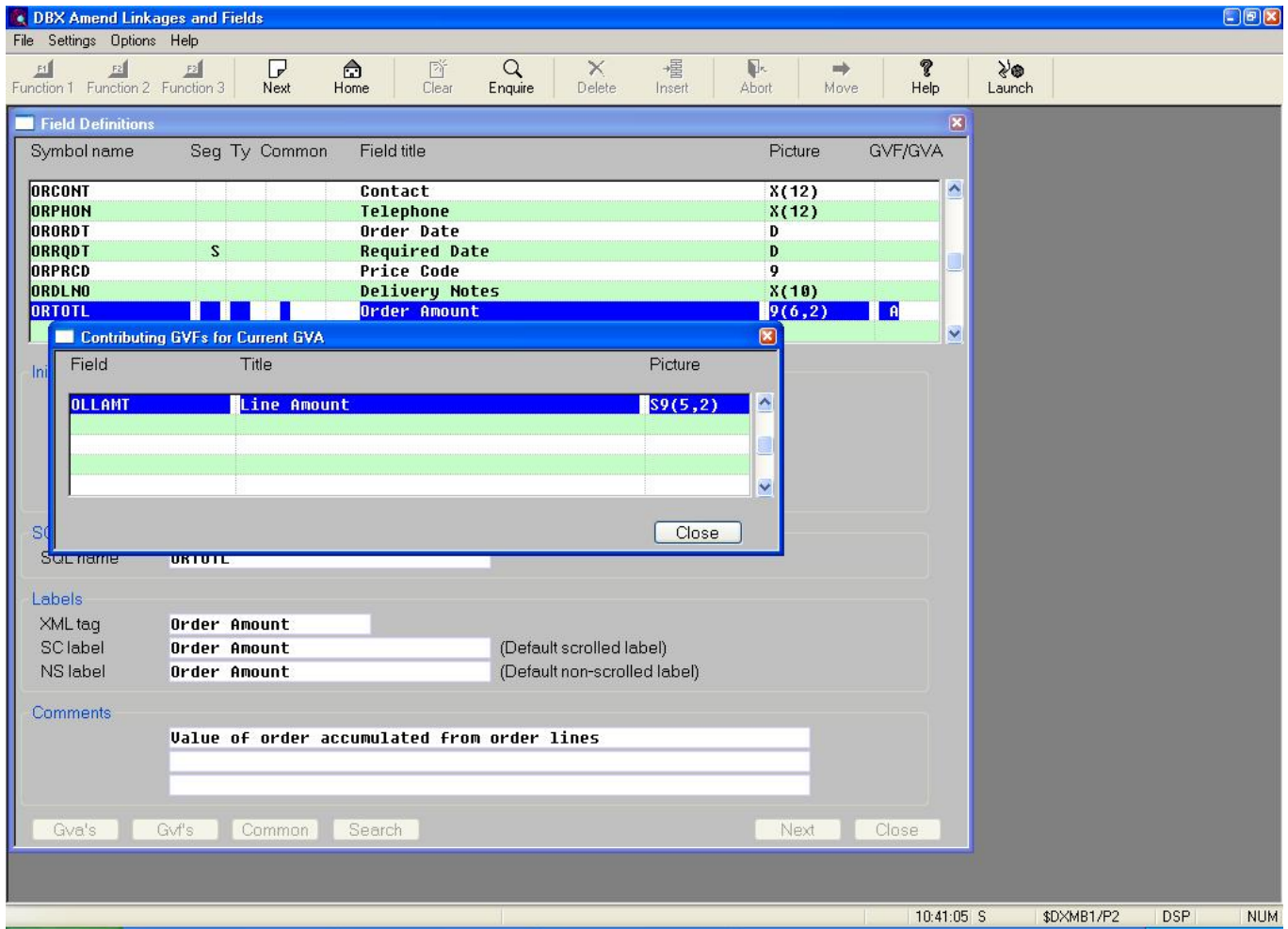
If a field contributes to a GVA field in another record 'F' will appear in the GVA/GVF column. If you position on the field record and select the 'GVFs' button, a table of the GVA fields that the GVF is added to is shown.
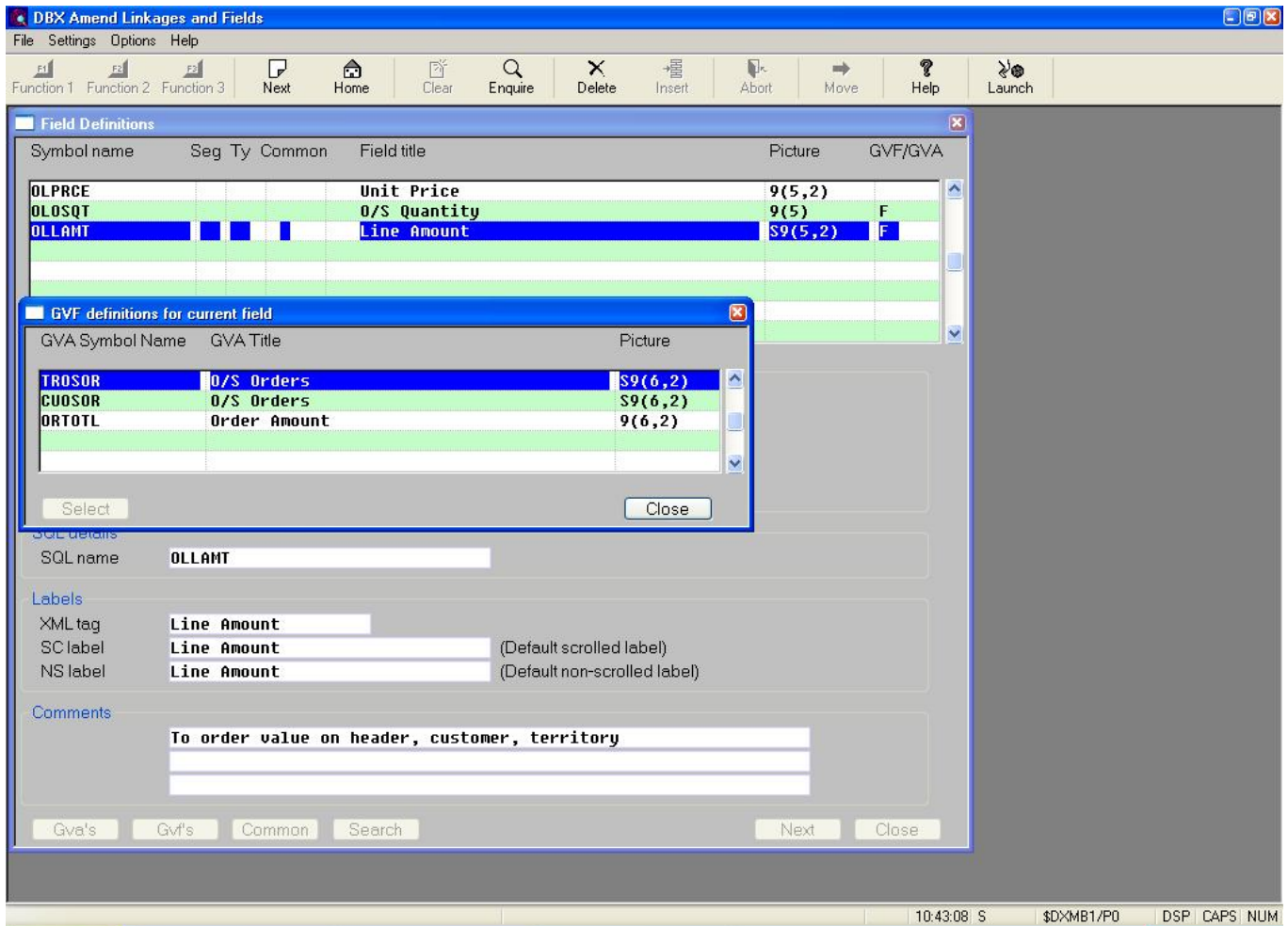
A field may be both a GVF and a GVA. For example item costs can be accumulated into sub-totals, which can then be accumulated into a grand-total. The sub-total field will be both a GVF and a GVA. Global master access fields (G1) cannot be GVAs.

Ticking the initialise box allows you to set a value which will be given to the field when the INITIALISE verb is used on the record set in a program. You may not initialise GVA fields.

For character fields the 'Byte value clear' option is available.
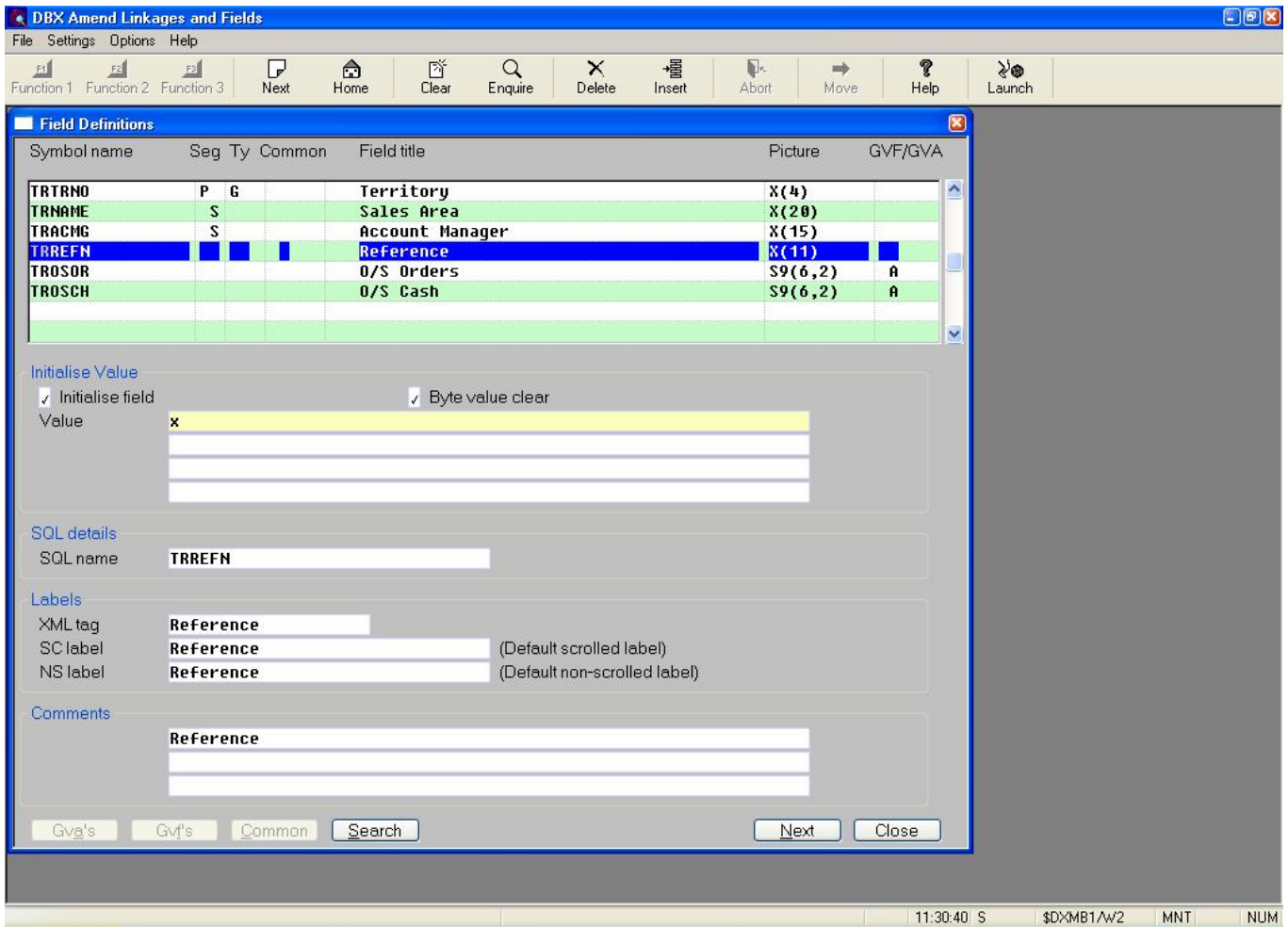
Ticking the "Byte value clear' option will clear the entire field to the single byte value specified when the INITIALISE verb is issued. In this example the field TRREFN would be initialised to "xxxxxxxxxxx". The byte value may either be expressed as a single character (in this case x) or as a single hexadecimal character beginning with #, for example #00 to initialise the field to LOW-VALUES. By default the INITIALISE verb will initialise all character fields to SPACES and numeric fields to 0.

The SQL field names in the field table window must be supplied and the first 14 characters of the SQL field names must be unique throughout the database. For a new field entry the SQL name is defaulted to the field title. This SQL name is the table or field-name shown when SQL queries are later performed on the database. Note that the external names chosen must comply with the naming conventions required by the database engine you choose when you create a DBX database.

The label fields with the exception of the XML tag are used in the window creation facility of the Speedbase editor to indicate the label that you would like to appear in the window for the selected field from the database. The XML tag is for future use but care should be taken in allocating this description.

Pressing the 'Next' button from the field list window causes the table of indexes for that record set to be shown.
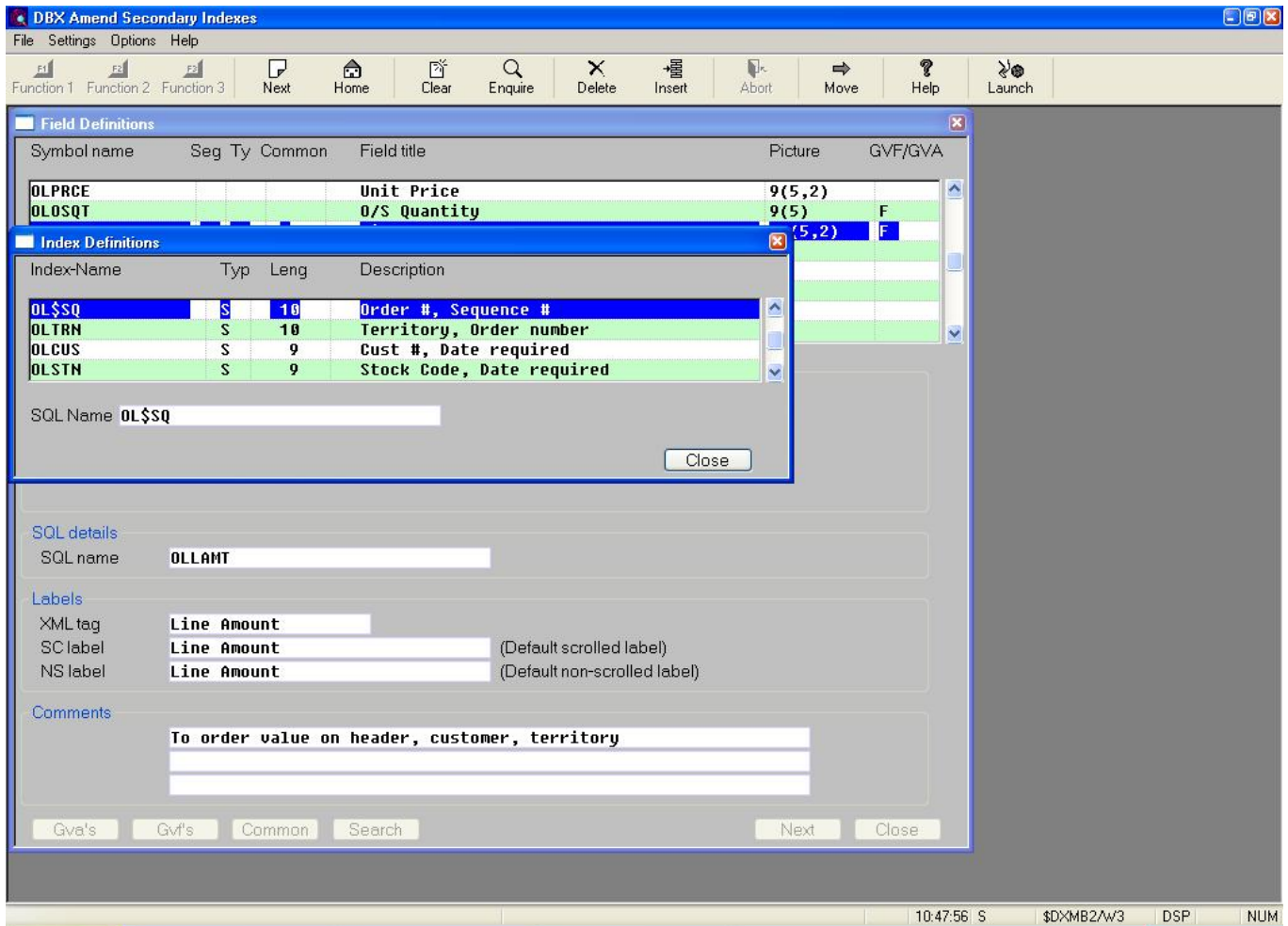
Primary indexes are indicated by a 'P' in the 'type column' and secondary indexes by an 'S'. To amend an index segment of a secondary index TAB through the name, type, and description fields. The index segment window will then be shown. Primary index segments are displayed and amended in the field window.
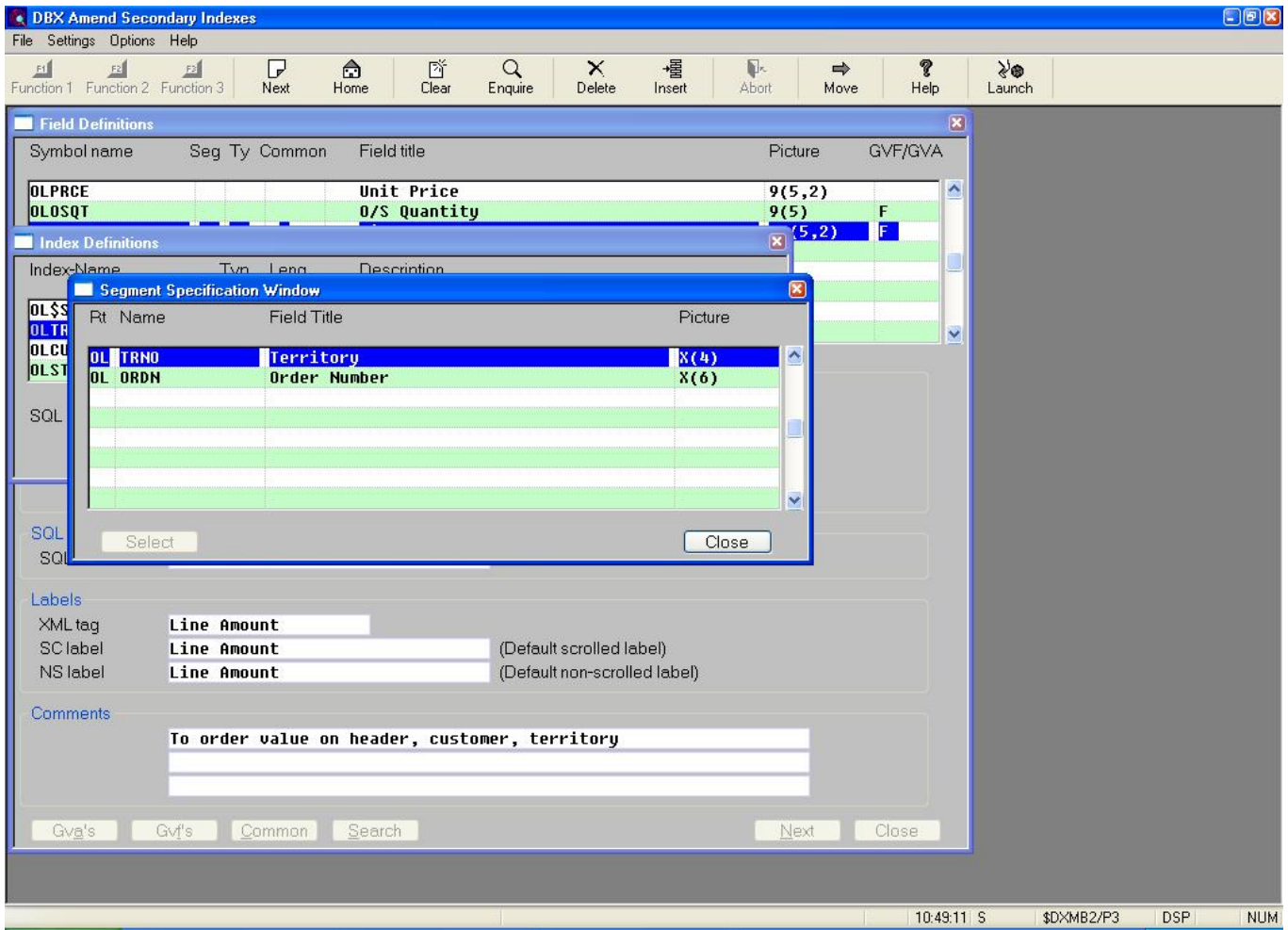
When adding indexes you may press the select button to display the current list of fields and to select one.
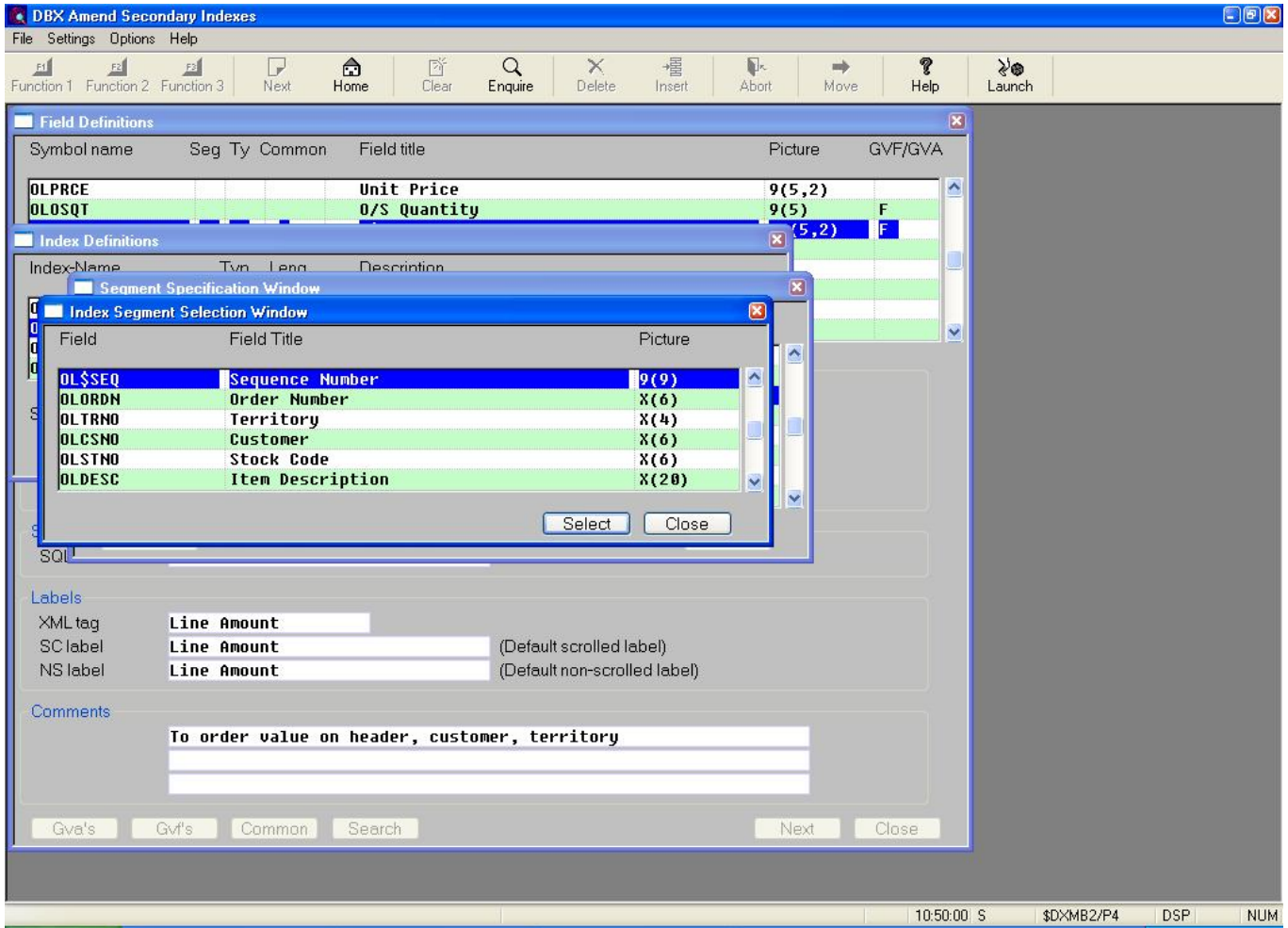
6.2.3 Print Meta Dictionary

To print the dictionary report, select the 'Print Dictionary Report' entry from the list of Dictionary Maintenance options. $DXM will ask if you require alphabetic ordering for the print report.

The report will be printed to unit $PR.

6.2.4 Generating the Dictionary

Once you have completed the amendments to the meta-dictionary you may generate the dictionary itself. To do this you must select the 'Generate Dictionary' entry from the list of 'Dictionary Maintenance' options. You will be asked for the name and unit of the output dictionary. After the dictionary has been generated you will be offered the option to print the dictionary.

6.2.5 Clear Dictionary
The clear dictionary option empties the meta-dictionary of its contents and should only be used if you are absolutely certain. If you select this option you will be asked to confirm that you want to clear the meta-dictionary.

6.2.6 Record-id Change
This option allows you to change the name of a record-id. You will be asked to enter the record-id of the existing record and the two characters of the new record-id.

6.2.7 Select another Database
Selecting this option will return you to the enter meta-dictionary window from whish you can update a further meta-dictionary.

7. Create Dictionary Copy Library
To create a dictionary copy library you must select this entry from the $DXM main menu. You will be asked for the dictionary name and unit together with the name and unit of the output copy library. The dictionary supplied must be a generated dictionary and not a meta-dictionary.
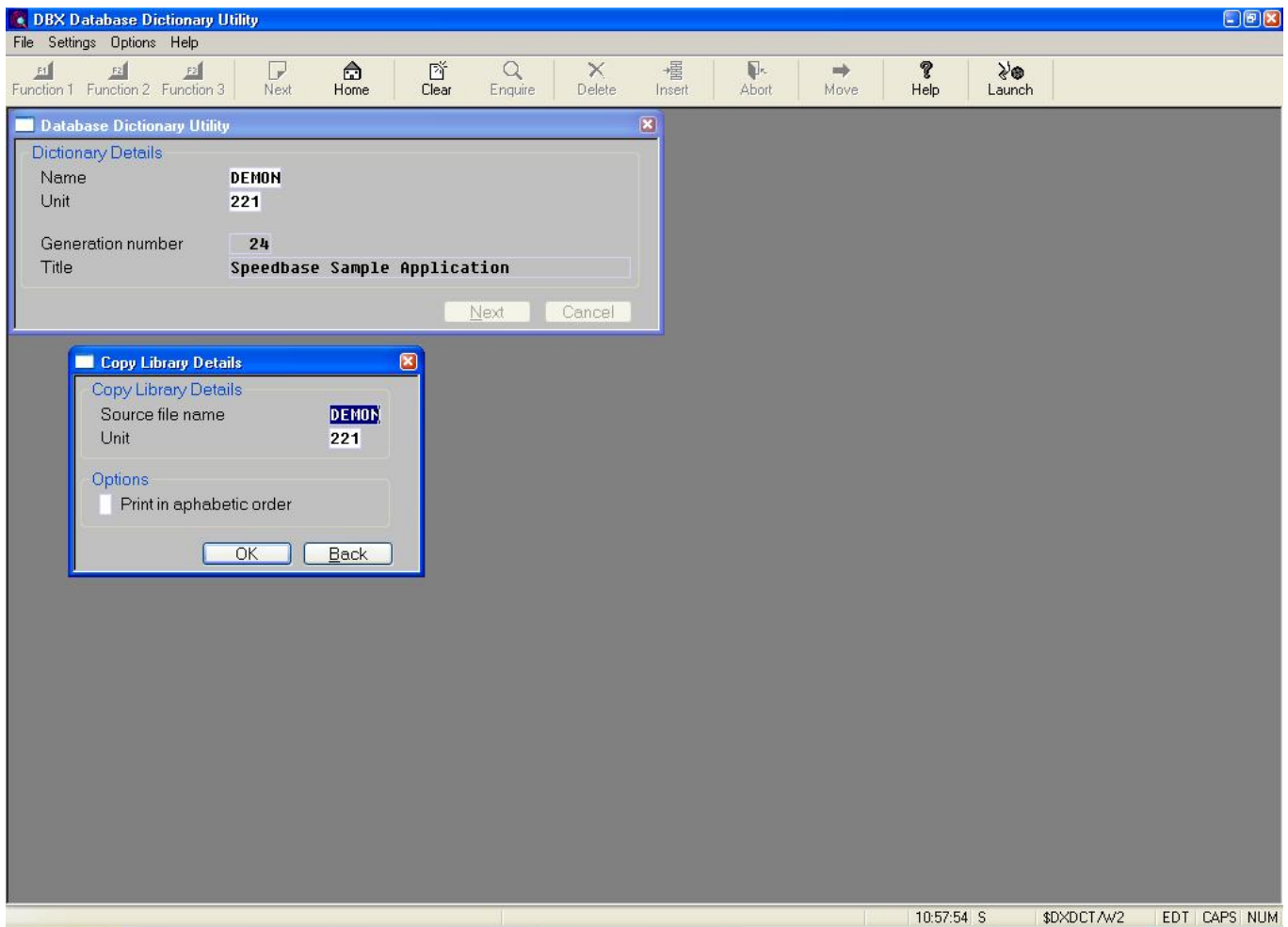
You may optionally choose to create the copy library with the records in alphabetic order.

The copy libraries can be used to compile with.

8.   See Also

$DXU   DBX Database Maintenance Utility
Speedbase Database Access Verbs
DBX INITIALISE verb