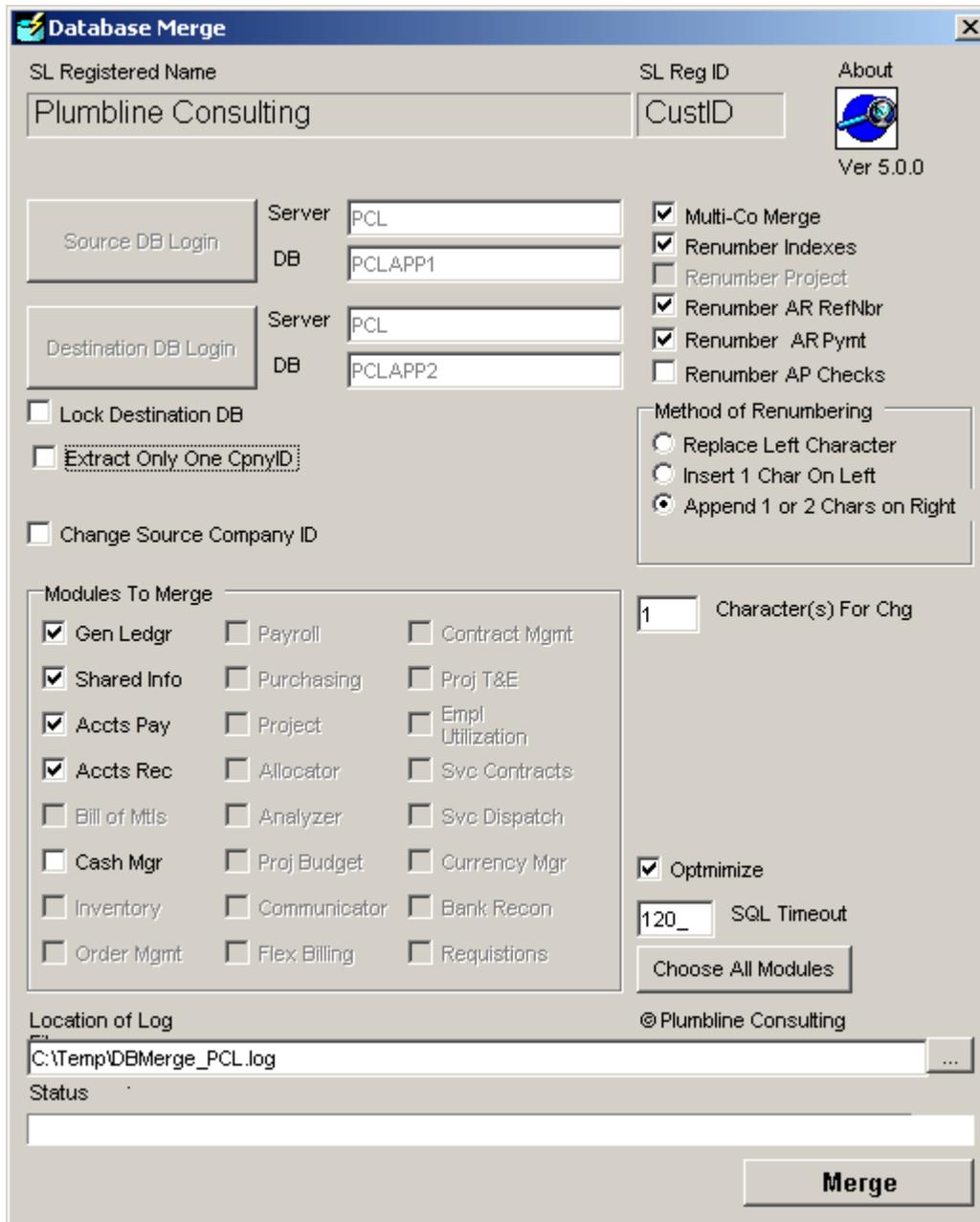


## DBMerge

This program merges one or more SIV 4.x or greater database(s) into another SIV 4.x or greater database to create a multi-company database. The source database and the destination database must be the same Dynamics SL version, have the same current period, and has the same base currency.



The screenshot shows the 'Database Merge' application window. It features a title bar with the application name and a close button. The main interface includes several sections:

- SL Registered Name:** A text field containing 'Plumblin Consulting'.
- SL Reg ID:** A text field containing 'CustID'.
- About:** A button with a globe icon and the text 'Ver 5.0.0'.
- Source DB Login:** A button with a text field for 'Server' (PCL) and 'DB' (PCLAPP1).
- Destination DB Login:** A button with a text field for 'Server' (PCL) and 'DB' (PCLAPP2).
- Options:** A series of checkboxes including 'Lock Destination DB', 'Extract Only One CpanyID', and 'Change Source Company ID'. A 'Method of Renumbering' section contains radio buttons for 'Replace Left Character', 'Insert 1 Char On Left', and 'Append 1 or 2 Chars on Right' (which is selected).
- Modules To Merge:** A grid of checkboxes for various modules such as 'Gen Ledgr', 'Shared Info', 'Accts Pay', 'Accts Rec', 'Bill of Mtis', 'Cash Mgr', 'Inventory', 'Order Mgmt', 'Payroll', 'Purchasing', 'Project', 'Allocator', 'Analyzer', 'Proj Budget', 'Communicator', 'Flex Billing', 'Contract Mgmt', 'Proj T&E', 'Empl Utilization', 'Svc Contracts', 'Svc Dispatch', 'Currency Mgr', 'Bank Recon', and 'Requisitions'.
- Character(s) For Chg:** A text field containing '1'.
- Optimize:** A checked checkbox.
- SQL Timeout:** A text field containing '120\_'.
- Choose All Modules:** A button.
- Location of Log:** A text field containing 'C:\Temp\DBMerge\_PCL.log'.
- Status:** A large empty text area.
- Merge:** A large button at the bottom right.

## DBMerge

### Installation

This program runs only on computers that Microsoft approves for running Dynamics SL IV/SL, which at the time of this writing is Windows XP, Windows Vista, Windows 7, Windows Server 2003, and Windows Server 2008. 32 bit or 64 bit will work. Install the DBMerge.exe program to any folder deemed appropriate – many people use \Program Files\DBMerge. If running on SL version 6.5 or earlier, the SIV client must be installed on the computer that will run DBMerge. Installing the SL 6.5 or earlier client will install the VB6 runtime components which are also used by the DBMerge program. If merging

version 7 or later DB's, the FRx client can be installed to load the needed OCx's and DLL's or we can provide a VB6 install utility that installs the VB6 runtime components.

If you get the following error message when running the DBMerge program, the VB6 components are not installed on the computer running the DBMerge.



Users may create a shortcut on the Windows menu or use the Windows/Run dialog or Windows Explorer to run the program.

### **Copyright**

The DBMerge program has been copyrighted since 2003 and is protected under copyright laws of the United States of America.

### **Running the program**

Note: please read the sections in the last page of the instructions on corrupt indexes or index errors. These problems are the most common tech support calls we get and most of the time we find that the person calling had not read that part of the instructions.

The program must be run for each source database that will be merged into the destination database. The program dialog screen is shown above:

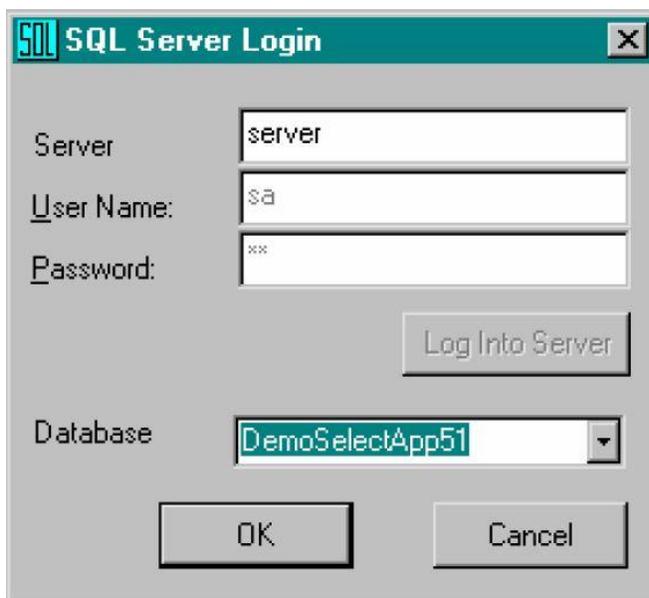
### *Renumber Indexes*

See Method of Renumbering below.

## DBMerge

### *Source DB Login*

Brings up the SQLServer dialog box to choose the source database, as shown below:

The image shows a screenshot of the 'SQL Server Login' dialog box. The title bar is green with the SQL logo and the text 'SQL Server Login'. The dialog has a grey background. It contains several input fields: 'Server' with the text 'server', 'User Name:' with 'sa', and 'Password:' with three asterisks. There is a 'Log Into Server' button to the right of the password field. Below these fields is a 'Database' dropdown menu showing 'DemoSelectApp51'. At the bottom are 'OK' and 'Cancel' buttons.

### *Destination DB Login*

Brings up the SQLServer dialog box to choose the destination database. NOTE: it is more expeditious to use the largest of all the databases to be merged as the destination DB.

### *Change Source Company ID*

This allows changing the company ID as it is merged. For example, change the source company ID from BeanoMeals to BM.

### *New Company ID*

The new company ID to be assigned to the source database being merged. In the above example the characters BM would be placed into this screen.

### *SQL Timeout*

## DBMerge

Set the number of seconds to wait for a response from SQLServer when the program makes a request. If a response is not received within the timeout period, the program is aborted. The default is 30 seconds.

### *Modules To Merge*

Choose which modules to merge.

### *Extract only one CpanyID*

This allows the DBMerge to select only one company from the source database. The company ID to be selected is set in the text box to the right of the check box.

### *Multi-co Merge*

Indicates to the program that a multi-company DB is being created or that data is added to an existing company.

### *Tables To Merge*

Choose which tables to merge.



### *Method of Renumbering*

To prevent duplications in the unique indexes, some fields used in indexes must be changed as they are copied to the destination. The Dynamics SL database schema does not always use Company ID as a column in a unique index but it does quite often use Batch number and reference number. Plus, there are some tables, such as APAdjust, that do not use Company ID but use RefNbrs to point to other data in the database. There is a high probability that in two databases that some of the fields used

## DBMerge

in a compound unique index will be the same. Therefore, this merge program gives the capability to change the value of these fields so as to maintain referential integrity in the data being transferred without violating the unique index. Fields changed, depending on the table, are: Alloc\_Batch, APRefNo, ARBatNbr, BatNbr; Batch\_ID, ChkNbr; ContractID, Detail\_Num, DocNbr, Draft\_Num, InvcNbr, OrdNbr, PONbr, Project, Project\_Billwith, RcptNbr; RefNbr; ReqNbr, ShipperID, and SOOrdNbr.

If there is absolute surety that there will be no duplication in any of these fields in the databases being merged, then leave unchecked the Renumber Indexes checkbox.

If the Renumber Indexes checkbox is checked, there are three ways to change the numbers:

- **Replace Left Character**  
The program will replace the leftmost character of BatNbr's, RefNbr's and so on with the character specified in the Character(s) For Change box. Note there are 11 boxes, one for each numeric digit with which a value could start and one for BatNbr/RefNbr's that start with an alphabetic character. Using the example in the screen above, refnbr's starting with 0 will have their leftmost digit replaced with an A. Refnbr's starting with 1 will have their leftmost digit replaced with a B, and so on. This method is the easiest to understand and implement and will work in about 50% of the circumstances.
- **Prefix 1 Character On Left**  
The program will insert the characters shown in the replacement characters to the left of BatNbr's, RefNbr's and so on. Note there are 11 boxes showing replacement characters, one for each numeric digit with which a source value could start and one for BatNbr/RefNbr's that start with an alphabetic character. Using the example in the screen above, refnbr's starting with 0 will have an A concatenated with the original number. Refnbr's starting with 1 will have a B concatenated with the original number, and so on. Note that if this method is used, some screens may need to be customized to expose all characters of the batch or reference number.
- **Append 1 or 2 Right Characters**  
The character(s) specified in the Character(s) For Change box will be appended to the right of all the characters in the existing number. If two appending characters are specified and the number of existing characters is greater than 8 then the replacement characters are appended to the leftmost 8 characters of the existing reference number. If one appending character is specified and the number of existing characters is greater than 9 then the replacement character is appended to the leftmost 9 characters. Suppose there were two batch numbers in the source database numbered 000100 and 100100. If the characters "AS" are

## DBMerge

entered into the Character(s) For Change box, then the result would be 000100AS and 100100AS. This works nearly every time but the disadvantage is that some of fields in each SIV screen will need to be customized to show all 10 characters of the new number.

### *Renumber Project*

Check this box if it is desired to renumber projects in the same manner as done in BatNbr, RefNbr, etc.

### *Renumber AR RefNbr*

Check this box if it is desired to renumber AR Reference Numbers, which are invoice numbers, payment check numbers, and so on. Some sites want to maintain the same numbers in the destination DB as were used in the source DB so they leave this box unchecked. This can be done as the RefNbr is not required to be renumbered in any of the AR tables except RefNbr. And the RefNbr table does not have to be merged in current versions of SL. However, not renumbering might be confusing if there will be several invoices for the same customer with the same invoice number.

### *Renumber AR Pymts*

Check this box if it is desired to renumber AR payments. This would allow the exact check numbers to show on statements as were submitted for payment.

### *Renumber AP Checks*

Check this box if it is desired to renumber AP checks. This would allow the exact check numbers to show on in the destination DB that were created in the source DB.

### *Lock Destination DB*

Check this box if it is desired to prevent a user from trying to log into the destination DB while the merge is taking place.

### *Location of Log File*

The program creates a log file and this is the location of that file.

### *Optimize Indexes*

If this is checked, clustering is not used when indexes are rebuilt on tables that have a primary index defined on more than one column. Tables that have a primary index based on one column, such as Vendor, are rebuilt with a clustered attribute regardless of whether *Optimize Indexes* is checked.

The standard Dynamics SL database has clustering on each table in the database. As documented in SQL Books Online, Microsoft does not recommend using a cluster

## DBMerge

attribute for a primary index that is based on more than one column. Removing the cluster attribute substantially increases performance for inserts and deletes. Dynamics SL does not require the cluster attribute, as per the following message from the Dynamics SL partner forum:

It is the opinion of Plumblin Consulting, LLC that removing the cluster attribute will nearly always improve SQLServer performance on tables that have primary indexes based on multiple columns or on tables that have frequent inserts or deletes. See SQL2000 Books Online and [http://msdn.microsoft.com/library/default.asp?url=/library/enus/createdb/cm\\_8\\_des\\_05\\_5h6b.asp](http://msdn.microsoft.com/library/default.asp?url=/library/enus/createdb/cm_8_des_05_5h6b.asp).

### **Demo Mode or Registered Mode**

When in demo mode, the program processes every other source record in most tables. That is, one row is written to the destination for every two read from the source. When in demo mode the program screen will say Demo on top of the screen.

If the program is in registered mode, the registration is imbedded into the exe and the name and SL registration of the company will show on the top of the program screen.

### **Merging Master Tables**

Master file tables, like Account, Vendor, and Customer, are not company specific. Rows with duplicate keys are not inserted into master tables of the destination DB. For example, if a record for VendID "AAA001" exists in the destination database Vendor table, then the vendor record for VendID "AAA001" from the source database would not be written to the destination. And in the same vein, if the vendor record does not exist in the destination, then it would be copied from the source and written to the destination. Balances for VendID "AAA001" would be added to the destination balances in the appropriate tables, such as AP\_Balances, if the company ID's are the same. If the company ID's are different in the source and destination then in tables like AP\_Balances there would be two records in the destination, one for each company ID. Company integrity is maintained. Duplicate checking is done using the primary key for the table.

However, If the duplicate key actually points to data that are different, i.e, "AAA001" is the vendor ID of AAA of Oklahoma in the source DB and "AAA001" is the vendor ID of AAA of Texas in the destination, then using a key change utility to change the vendor ID's in the source DB would be prudent. That will allow the data for both AAA of Oklahoma and AAA of Texas to reside in the destination DB as separate entities.

### **Indexes**

## DBMerge

Note in the discussion above, that the DBMerge program uses the primary key of the destination table when merging Master tables and other tables in which numbers from the source table are added to the corresponding destination tables. Therefore, the DBMerge program expects the tables to have the indexes that SL originally installed. DBMerge will not function correctly if the standard SL indexes do not exist when merging tables as discussed above. The program will issue an error message if the destination table for a merge operation has no indexes.

### **SNotes**

All note data from the SNotes table are transferred for the transaction tables that are merged if the SNote table is chosen for the module being merged. i.e. notes applicable to GLTran records are transferred.

For Master Table data, only the notes for the records that are merged from the source database are transferred; notes for duplicate data are not transferred. For example, if account 1234 does not exist in the destination database, Account table data and SNote table data for account 1234 is transferred from the source database to the destination database. If account 1234 exists in the destination database, then no data from the Account table or the SNote table are transferred from the source database.

### **Inventory Tables**

If the inventory module was merged, then upon conclusion of merging all the databases, the following IN Integrity Check options should be run: Rebuild Item Quantity and Costs, Rebuild OnOrder Quantities, and Rebuild Inventory Plan.

### **Use Copies of Source DB's**

Do not use the live source databases as DBMerge may make temporary changes to the source databases, as explained below.

### **Changes to Source Database**

In some tables, such as AcctHist and EarnDed, data is either inserted into the destination table or values from a given source row are added to a row in the destination that has the same key. In certain situations, the DBMerge program will insert a temporary column, named RowNumber, into the destination and sometimes the source tables. Temporary indexes are built on that column for the table(s) being merged. This allows the merge to be performed in a much faster manner than would be possible using the indexes that originally exist on the table. At the conclusion of the merge of the table, the temporary indexes are deleted and the temporary column is dropped from the source and destination tables. **However**, if some error occurs during the merge and it does not run to conclusion, those additional indexes and columns will not be removed. This is because the code that removes them is run upon successful merge of the data. If an error occurs, the code does not get a chance to be run. Therefore, both the source database and the destination database will need to be

## DBMerge

restored before continuing the merge. The probability of subsequent problems will be less if copies of the live source databases are used for the DBMerge.

### System DB Tables

The destination and source databases must be attached to a valid SL system database because some of the tables in the system database are read by the DBMerge program. However, only tables in the destination application DB are permanently changed. No system database tables are changed by DBMerge and the appropriate system DB tables, such as Company, Domain, and InterCompany, should be changed manually to reflect how the companies are merged into one database.

To get the AcctXRef changed in the system DB after a merge, a user should open the Account screen in SL, add some account, say 00000, save it, and then delete it. That will cause the system to repopulate the AcctXRef table. Do the same things with the SubAccount screen to repopulate the SubXRef table.

### Destination Database

An existing database or an empty database, created by Dynamics SL Database Maintenance, can be used for the destination database in the first merge. If the user chooses an existing database, then the best performance of this program will occur when the largest database is the destination and the smaller databases are the source. **Note** again that the destination database must be attached to a valid system database before the DBMerge is initiated.

If an empty database is used as the destination database, then the following SQL statements should be run on the destination DB before the first merge. This is necessary because Dynamics SL DB Maintenance, when it creates application databases, populates the FlexDef and SegDef tables. The DBMerge treats SegDef and FlexDef like all master tables; it will not move data from the source to the destination if records with the same key exist in the destination.

To prepare an empty DB, run the following in the SQL command processor:

```
Delete from segdef  
Delete from segdefheader  
Delete from flexdef
```

It is **important** to **make a SQLServer backup** of the destination database after every successful merge. This will save time if a subsequent merge fails due to index problems brought about by the method of renumbering, as shown in *Errors in changing indexes*.

### Move Individual Companies to another Database

DBMerge, originally designed to combine databases, can be used to create separate databases. The program is actually a vehicle that moves data from one DB to another.

## DBMerge

The program provides the ability to designate the Company ID for the data to be moved. Some tables have a company ID and others do not. For the tables that have a company ID, only data for the designated company ID will be moved. For tables that do not have a company ID but which can be linked to tables that do have a company ID, such as Customer, only the appropriate data for the table is moved. For tables that do not have a company ID and cannot be absolutely linked to tables that have a company ID, such as terms, all the data is moved. If the destination DB is empty, then the effect of running the DBMerge with a company ID designation is to create a new single-company database.

To accomplish this move process, choose a company ID when setting the parameters of the merge.

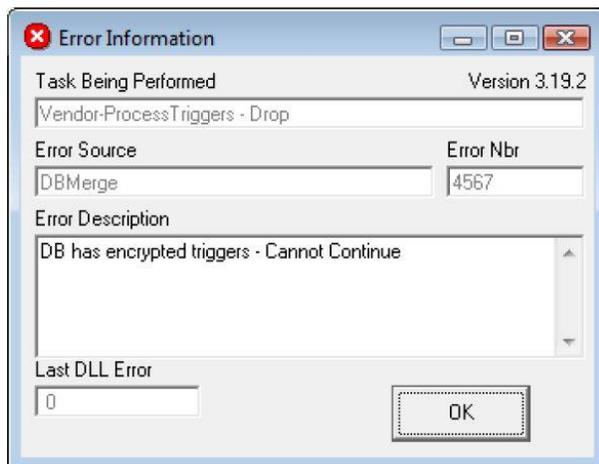
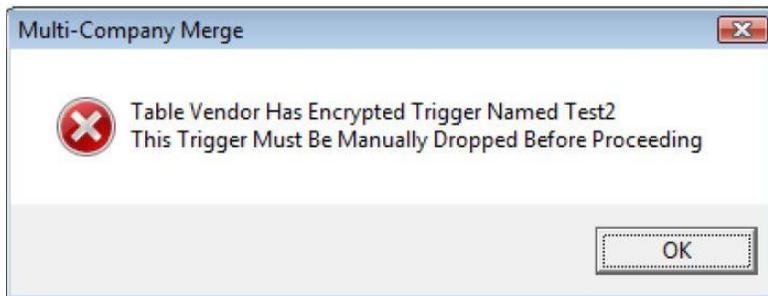
### Encrypted Triggers

There are no encrypted database triggers in the standard SL database. However, some third party ISV's and customizations add encrypted triggers to the database. The DBMerge drops all triggers in a destination database before merging in data from the source. Before dropping the triggers, it makes a copy in memory of the trigger SQL script and then uses that copy to rebuild the trigger on the merged table. However, DBMerge cannot read the SQL script of an encrypted trigger. Therefore it cannot drop the encrypted trigger and rebuild it. It is the responsibility of the DBMerge user to manually drop the encrypted trigger(s) and then rebuild them after the merge is finished. Should any encrypted triggers be found by DBMerge, the following message will appear upon logging into the destination database.



If the encrypted triggers are not deleted the system will throw the following message and error. If this happens, the destination database will have to be reloaded and the merge started again.

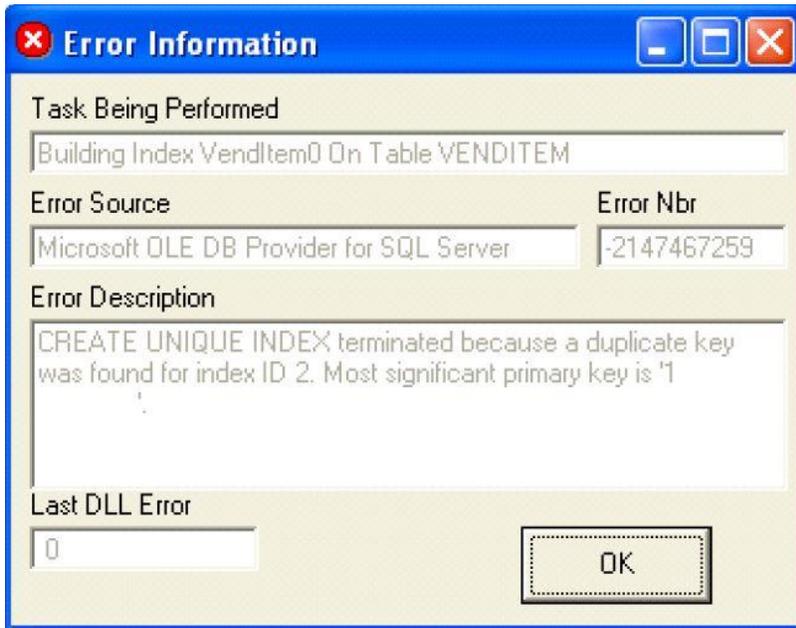
## DBMerge



### Errors in changing indexes

The screen below shows the error message that arises when there are duplicate values in the values used in a unique index. In almost all cases this error is caused by using a renumbering method that created a duplicate value. The user should restore a backup of the destination database made prior to attempting the merge that caused the error and do the merge again with a different method of renumbering.

## DBMerge



If the index error is on the RefNbr table when the AR module is being merged, the situation is usually when the Reference Numbers in AR are 9 or 10 digits long. Because the RefNbr field is limited to ten characters none of the renumbering schemes can create a unique new number in all circumstances. The RefNbr table is not used in current versions of SL as it was used in the past and if it were not merged it would have negligible effect on the final merged DB. Also we have found that in most circumstances, there are few duplicates in ten digit RefNbr's in the source and destination DB so the user can probably uncheck the *Renumber AR RefNbr checkbox* which will move the RefNbr data unchanged to the destination DB. Before doing so, the following SQL statements can be used to find any duplicates and delete them before doing the merge.

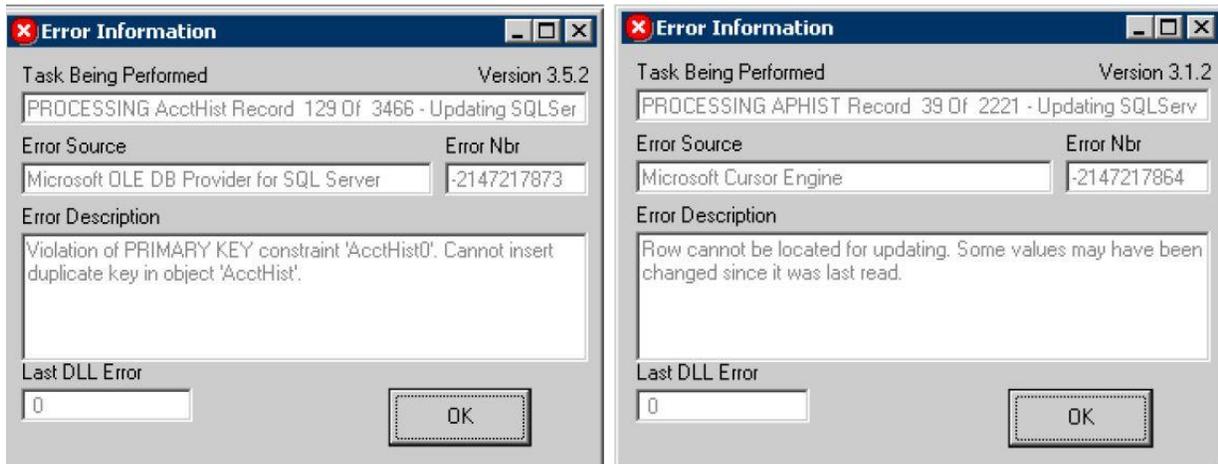
```
Select source.RefNbr From SALES1APP.dbo.RefNbr source
      Join USPROD.dbo.RefNbr dest on source.RefNbr = dest.RefNbr
```

```
Delete from USPROD.dbo.RefNbr Where RefNbr In (
Select source.RefNbr From SALES1APP.dbo.RefNbr source
      Join USPROD.dbo.RefNbr dest on source.RefNbr = dest.RefNbr)
```

### Data Corruption

If an error like one of the following is received while processing, indexes on the destination table are probably corrupt.

## DBMerge



There are a number of techniques which can be used to fix data corruption. One of the more common is setting the database to single user and running, for the table in question, the following statement:

```
DBCC CHECKTABLE ('APHIST', REPAIR_REBUILD)
```

Below is an old method (from the dBase days) that we find works 98% of the time when the repair methods in SQL and Management Studio do not fix the corruption. Although the following script was not created to fix a problem that DBMerge encountered, the same technique can be used to fix the tables that are causing DBMerge not to work correctly. As an aside, after running the following script, reports that ran in 7 minutes ran in 2 seconds.

```
ALTER TABLE [dbo].[StubDetail] Drop CONSTRAINT [StubDetail0]
--Drop all other indexes too
Drop Index StubDetail.StubDetail1 --If it exists

Select * Into StubDetailBkup From StubDetail

Delete from StubDetail --Do NOT drop table

Alter Table StubDetailBkup Drop Column tStamp
Alter Table StubDetail Drop Column tStamp

Insert [StubDetail] Select * From StubDetailBkup

Alter Table StubDetail Add tStamp TimeStamp

Create Index [StubDetail0] On [dbo].[StubDetail]
([Acct] ASC,[Sub] ASC,[ChkNbr] ASC,[DocType] ASC,
[StubType] ASC,[Typeld] ASC,[WrkLocld] ASC)
```

We know that VAR's and users are skeptical that their servers and/or databases have corruption. But it can exist. We have many circumstances in which VAR's will attempt to

## DBMerge

rebuild indexes and there will still be index errors as shown in the error messages above. We have had a policy for several years in that we will examine the tables with the problems if the data is zipped and sent to us. If we find the problem is due to incorrect code in our program then we will promptly fix the code at no additional charge. However, if we find the error is due to data or corrupt indexes, then we will charge a modest consulting fee to fix the data or indexes. We have done this several times and in every case but two, the problem was due to corrupt indexes or data which we were able to repair.

### **Bulk Copy Option (version 7 SQL) and Simple Recovery and Single-User**

If the destination database does not have the “**select into/bulkcopy**” option (SQL7) or Simple Recovery(SQL2000/2005/2008) turned on, this program will turn it on. The program will also change the recovery mode to Simple so as to not build a very large log file during the merge. After merging is complete, the recovery mode for the database will be set to Full. If the value of the option, *Lock Destination DB*, is checked by the user the destination DB will be marked as **Single-User** during processing and not Single-User upon conclusion.