THRU-PUT DATABASE UTILITY (DBUTIL)

USER GUIDE

Thru-Put 7.0

This document describes the various configuration options in DBUtil and how to set them.

Last Updated: Nov 25, 2025

Table of Contents

DBUtil Overview

DBUtil (Database Utility) is a tool you use to prepare the database for all modules in the Thru-Put product suite. DBUtil takes shopfloor and manufacturing data input from your ERP system and audits and processes to store in a local database system. You can provide input to DBUtil from set of text files or from another database. Before you start using DBUtil, you need to configure DBUtil for your environment. This document summarizes the steps you need to follow to configure and use various commands in DBUtil Program to Build database. For detailed understanding of the database table schema written and stored by DBUtil, see the Database Reference Manual..

Most of the configurable settings for DBUtil are modified in the Download Configuration dialog. Select File > Download Configuration. This chapter discusses this configuration in general terms.

- Download configuration
- Download files
- Order board

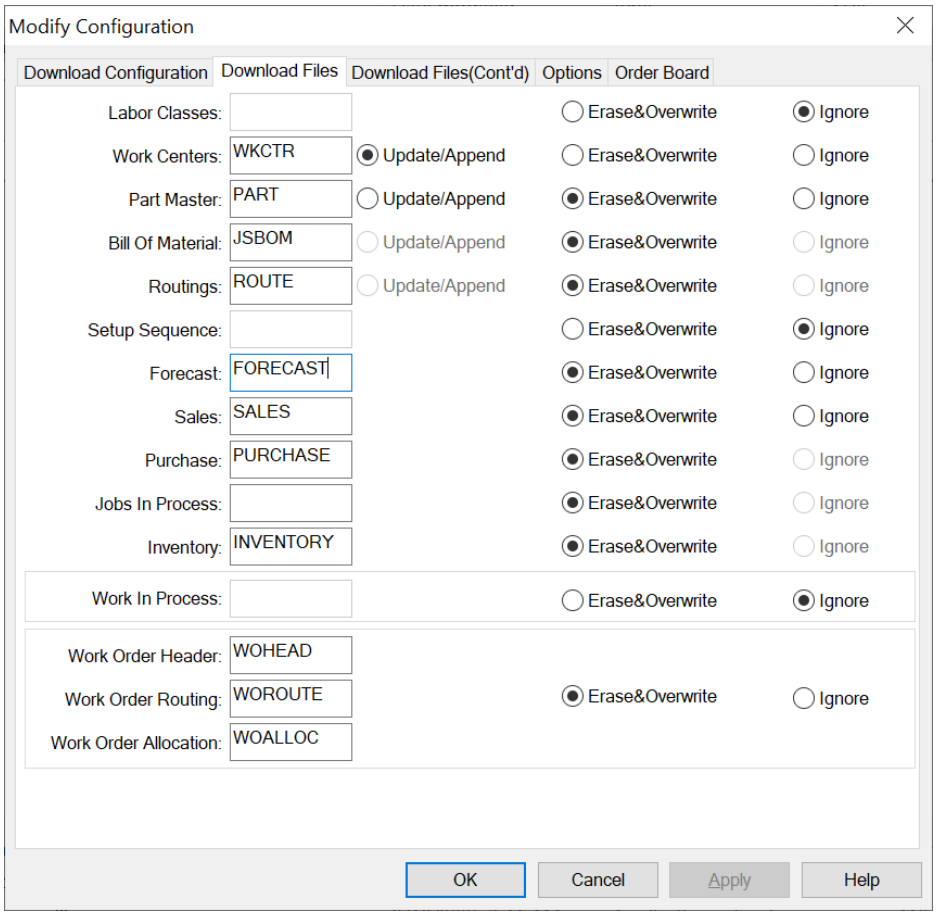
Download Configuration

If you use text files to import data into the ThruPut database, type in the name of the path of the directory in which the download data is stored into the Download Data Directory field. If you use a database and import the data directly into the ThruPut database, the Download Data Directory field contains the path in which the SQL files are located.

ThruPut can use an Access, SQL or Oracle database to store its data. If you choose to use Access, retain the database type and the ODBC setup. Select or configure the data source name (DSN) you want to use to build or update your database. If you choose to use Oracle or another database, configure the native Oracle connection, then change the database type and OBDC setup. See the Database Reference Manual for more information.

ThruPut records all errors and messages encountered during the DBUtil build into an error log file stored in the download data directory. You can specify how many sessions ThruPut should log files to archive in the Number of Log files to Archive field.

Download Files



The MRP download data files are the input to the DBUtil program. Download your MRP data files with a .DAT extension into the download directory. Type in the name of the data file you are downloading in the field next to the Thru-Put data file names. The following points should be kept in mind:

- The .DAT extension should not be entered in the file names specified.
- You can use the file names already present for the SAMPLEDB project as an example.
- The .DAT file does not have to exist at this point, but you will need it to create format files and build the database.

This table shows the various options available when you create a download file.

Option	**Description**
Update/Append	Select Update if a record in the data file exists in the database table (for example,

	the key fields match). The record is updated with the new data in the input data file. Select Append to insert a new record into the database table. When you select to update or append the table, Thru-Put reads and compares the input data to the existing table. Building the database using this option will take longer than using the Erase & Overwrite option.
Erase and Overwrite	Default. When you select this option, the existing table is eliminated and rebuilt.
Ignore	If this table exists in the database, it is not touched.

Since Update/Append option reads the existing table and compares with the input data it could take a longer time to complete the database build with this option than the erase and overwrite mode.

In order to use Thru-Put in a multiple-plant environment, you must configure two additional input files: the Part cross-references file, and the Transit Time file.

These inputs are used currently in interplant applications of the Customer Service Workbench. They are input to DBUtil through two data files called PART_XREF (for part cross-reference) and TRANSIT_TIME (for transit time between plants). In the Thru-Put database, these inputs are stored in two tables called SC_PART_XREF and SC_TRANSIT_TIME respectively. The Thru-Put database also maintains another table called SC_PLANTS, which contain the network and security information of the supply chain users. The Thru-Put Administrator (ResAdmin.exe) creates this table.

To maintain referential integrity in the database, DBUtil imposes restrictions on the setting of the above flags.

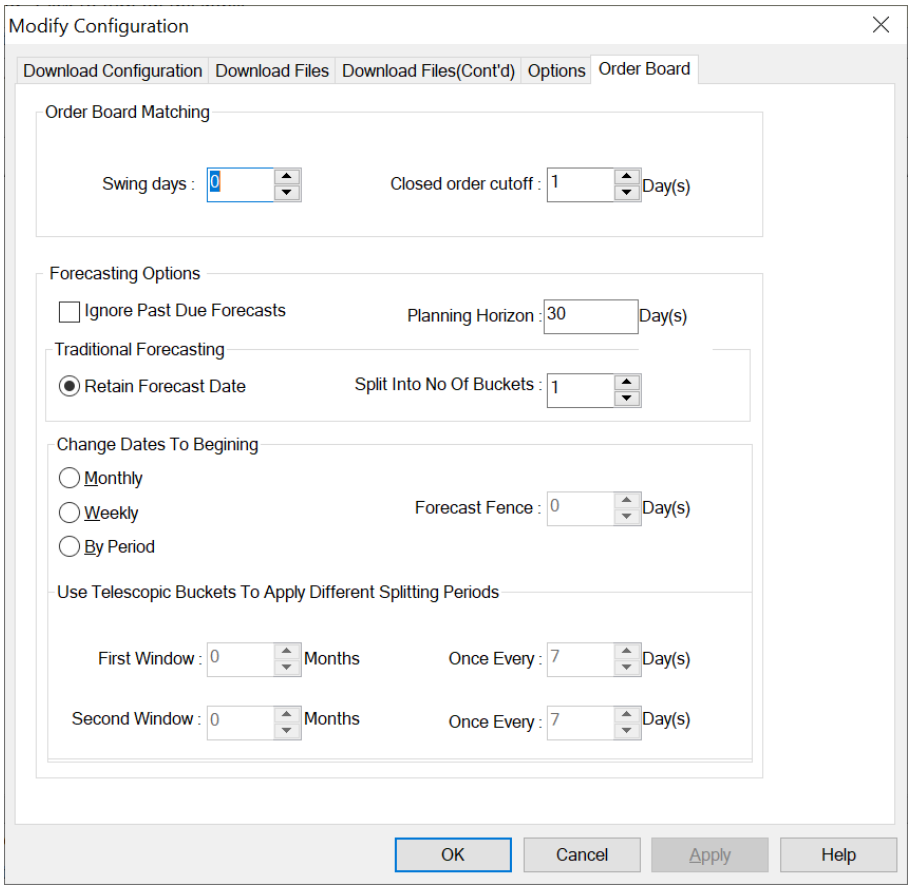
- If the Work Center table is set to Erase & Overwrite mode, then the Routing Table has to be set to the Erase & Overwrite mode, since it is possible that the Work Center information in the old routing records are inconsistent with the new Work Center data.
- If Part table is set to Erase & Overwrite mode, the BOM and Routing tables have to be refreshed.
- If either the Work Center table or the Part table is set to Erase & Overwrite mode, a filename must be specified for the Routing file for the above mentioned reasons.
- The WIP, Inventory, Supply, and Purchase are always set to Erase & Overwrite mode.

Options Tab

The following options are available in DBUtil:

Option	Category	Description
With jobsteps	BOM	Check this box to include the job step of the parent that consumes the child
Allow duplicates	BOM	In some cases, a parent part can have multiple instances of the same child at the same job step. This is common occurrence in semi conductor industries. Check this box to have DBUtil total all the children. If you do not check this box, duplicate children are rejected.
Required	Material Analysis	Check this box to make material analysis a required factor. Selecting this option has no impact on the output of the download.
Display errors	Loading Errors	An error message appears each time the process encounters a problem
Log errors	Loading Errors	Thru-Put writes the error messages to a log file. You should select this option unless you have a very small data set.
Buyer MRP Dock-to-stock	Default lead times	Default is 0. Specify global values for these parameters. If you do not map these fields to a part format file, Thru-Put uses the parameters you set here.
Forecast allow duplicates		Check this box to allow duplicate forecast records in the forecast table. Otherwise, duplicate forecast records are rejected.

Order Board



Option	Description
Swing days	Is specified in calendar days and it is a measure of buffer penetration that the plant is willing to withstand to maintain stable schedules.
Closed order cutoff	Specifies how many days before the download date the cutoff date occurs. All closed orders before the cutoff date will be ignored when Thru-Put is calculating the SHIPMENT table.

Refer to the order board matching chapter of the concepts section for more information on these parameters.

The options on the Order Board tab relate to forecast issues. Two methods are supported for forecast splitting and consumption.

- Traditional
- Periodic

See the Database Reference Manual section entitled “Order Board tab” for more information.

Traditional & Periodic Forecast

Traditional forecasts allow only monthly buckets for forecast consumption. Periodic forecasts do not import past-due buckets into the Thru-Put database.

Option	Type of forecast	Description
Split until next forecast	Traditional and periodic	Check this box to allow Thru-Put to spread the forecast evenly between the current and next forecast for the same part. Thru-Put determines frequency and interval by the number of buckets. If you do not check this flag, Thru-Put uses the periodic forecasting option.
No of Buckets	Traditional and periodic	Type in the number of buckets into which you want to split the forecast.
Forecasting Method: monthly or weekly	Periodic	forecast represents a monthly or weekly forecast. Determines the forecast splitting and consumption window.
Forecast fence	Periodic	The amount of time, in default calendar days. Use if you do not want to build any forecast orders during this period. If you select this option, Thru-Put moves all forecast orders to a date beyond this fence.
Splitting period	Periodic	Allows you to use the telescopic

		splitting function of DBUtil. You can split the forecast into finer buckets in the first period and more granular buckets in the second period. See the notes that follow.

Notes:

Forecast First Window: A value in months starting from the forecast fence ending date. Specifies the range of the first window when you choose to split a forecast. If this value is zero, no splitting occurs. If this value is between 1 and 100, all forecasts with a target date that value are split and distributed on the dates specified in the **Once every<days>** field in the same line. The exact number of buckets or split forecasts is based upon the number of working days in the default calendar in the monthly or weekly period.

Forecast Second Window: A value in months starting from the forecast fence ending date. Specifies the range of the second window when you choose to split a forecast. If this value is less than the value you specified in the first window, no second gradation exists. If this value is greater than the value you specified in the first window, all forecasts falling between the First Window+1 month to the Second Window are split. Target dates are distributed on dates specified in the **Once every<days>** field in the same line. The exact number of buckets or split forecasts is based upon the number of working days in the default calendar in the monthly or weekly period. The spread period in the second window must be greater than that specified in the first window.

For example, perform a monthly forecast. Specify:

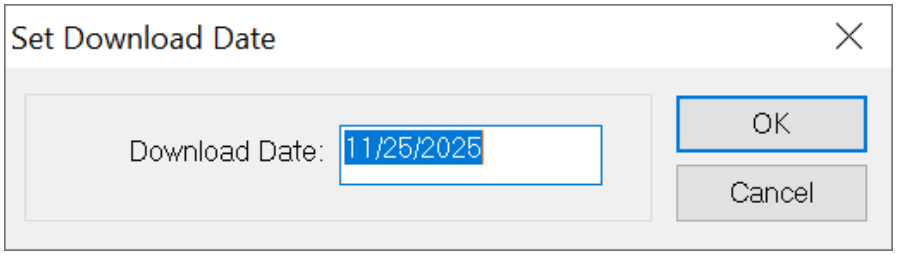
- 3 in the first window
- Once every <1> day
- 6 in the second window
- Split once every 7 days.

The forecast fence date falls in October. ThruPut sets forecast splitting as follows:

October, November, December	Split daily
January, February, March	Split every seven workdays
Remaining future months	Do not split

Set download date

Primarily used in testing and validation. Manipulating this date on a production database is not recommended. The download date defaults to the current date, but you can change this date via the File menu item “Set Download Date”. The download date is now displayed on the title bar, e.g. “Thru-Put DBUtility - (11/25/2025)”.



Building a database

Thru-Put requires a plant model to reference during the decision-making process. XREFPlant modeling is explained more thoroughly in “Implementing Thru-Put Manufacturing” section. It is important to properly build your database. Follow these steps to construct your database:

1. Configure DBUtil to build the Thru-Put database with your data.

2. Download your MRP or ERP data. Ensure that each file has a .DAT extension. XREF See “Input files” section for more information.

3. Map the input data fields to the Thru-Put schema using DbUtil > Prepare Format. Sample flat file input data records are provided. 4.Select File > Build Database. Upon successful completion of the build, Thru-Put renames the .DAT files to .OLD.

Working with sample data

You may want to see an example before attempting to map your own plant’s data. Thru-Put installs sample data files in the \SAMPLEDB subdirectory of your install folder. The download configuration, download file names, and download options in DBUtil are set for the SAMPLEDB during installation.

To build the sample database, start DbUtil and select File > Build Database.

You can create the Thru-Put database in any ODBC-compliant database such as desktop databases like Microsoft Access, or Microsoft SQL Server. Microsoft Access is the default database for Thru-Put. During installation, an ODBC Data Source is automatically established (as are ODBC Desktop Drivers, such as drivers for Microsoft Access).

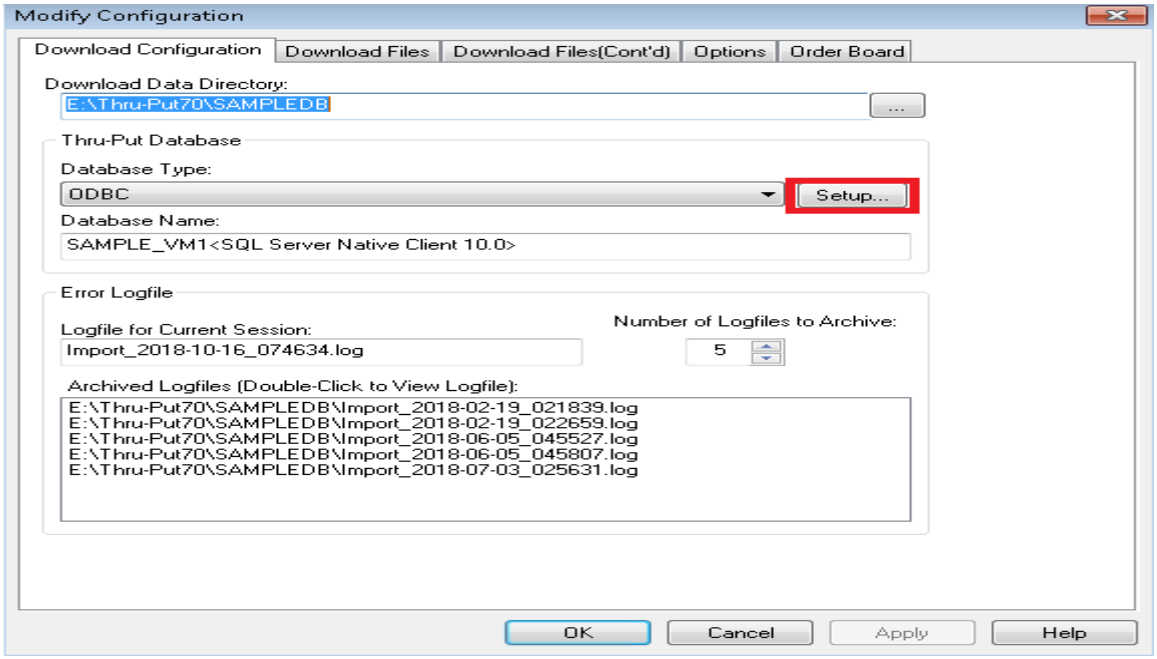
When you run DbUtil or Thru-Put for the first time, the data source is set to point to the \DB directory and DBUtil and Thru-Put are configured to use this data source. If you plan to use a database other than Access, you must have an ODBC driver for the database.

Adding or modifying a data source

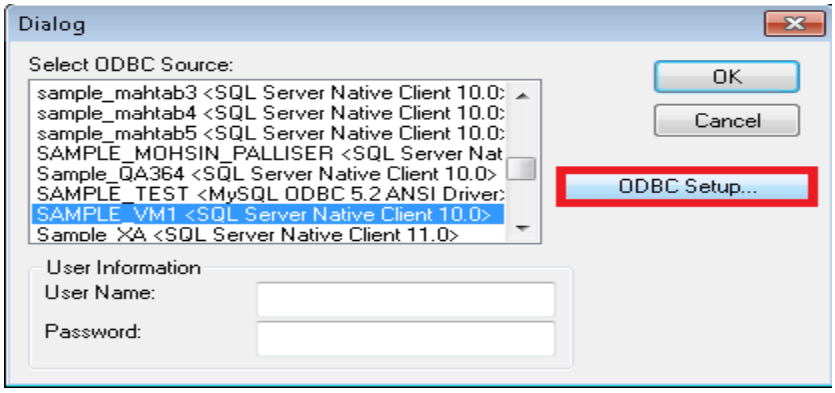
After you install Thru-Put, if you have no database in the install directory, you should modify the data source to point it to a different database. You can change the location of an existing database or you can create a new database. To build a Thru-Put database, create an ODBC data source for the database. To add or configure the data source, please follow below steps:

1. Start DbUtil (Please choose ‘Run as Admin’ Option). Select File > Download Configuration. Click on the **Download Configuration** tab if it is not already visible. A dialog box appears.

2. Select ODBC in the Database Type field and click **Setup**… as marked in the red.

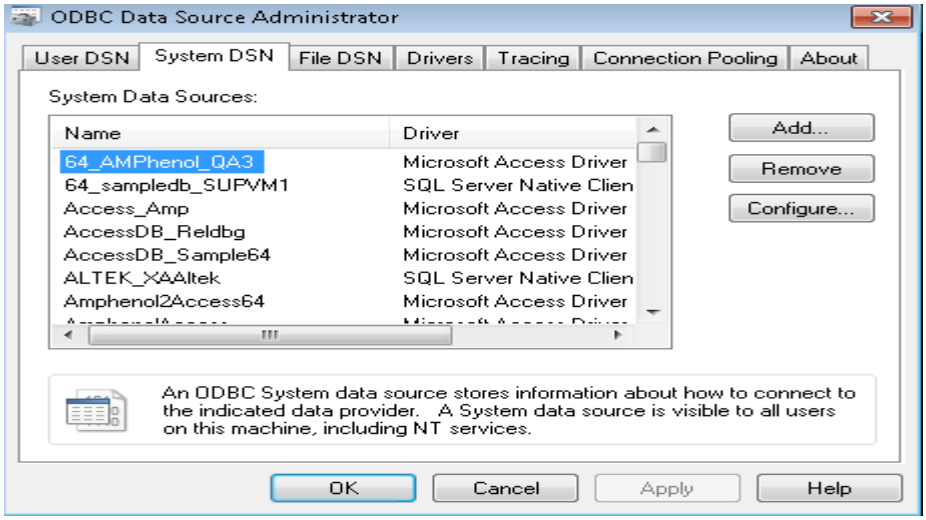


3. The following dialog box appears. Click ODBC Setup.. from this dialog.

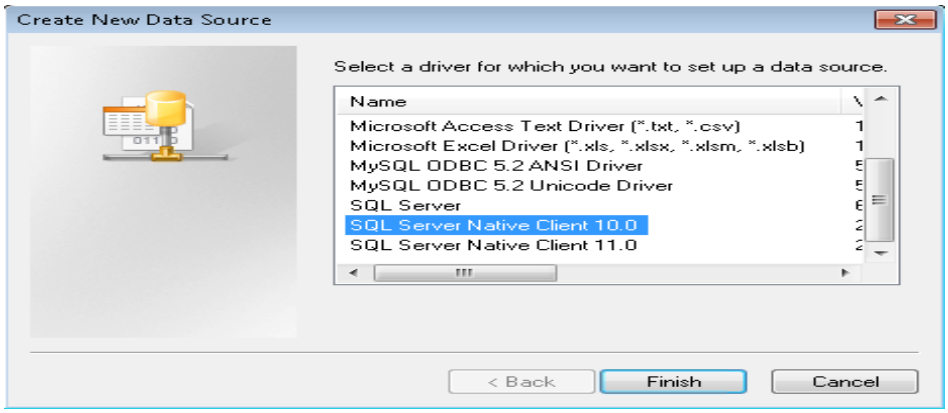


4. On clicking ODBC Setup button, a dialog box pop will be shown as seen in the below figure where you need to select **System DSN** tab and press the Add button to select a driver for which you want to set up a data source name (DSN).

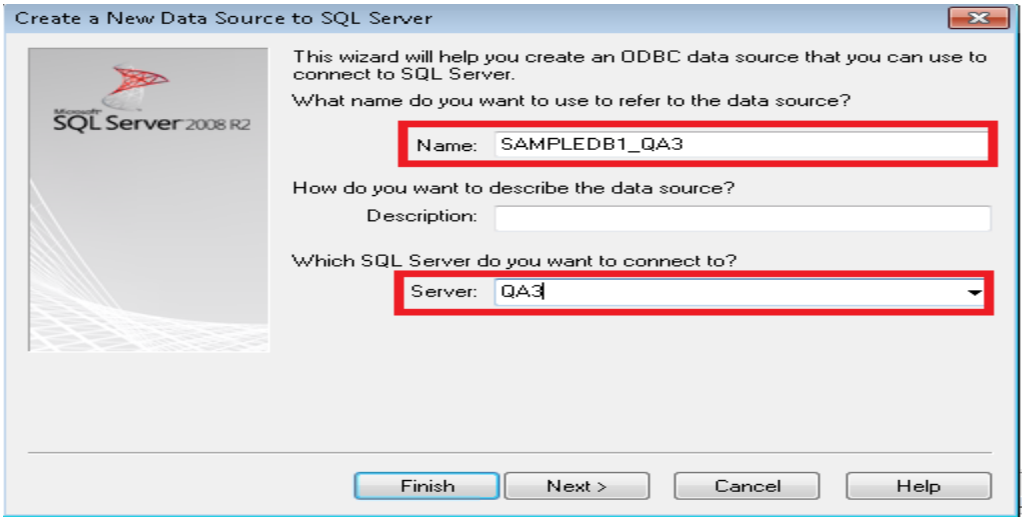
   Note: Make sure you have selected **System DSN** for the SQL Server ODBC driver with SQL native Client 10.0 or above.



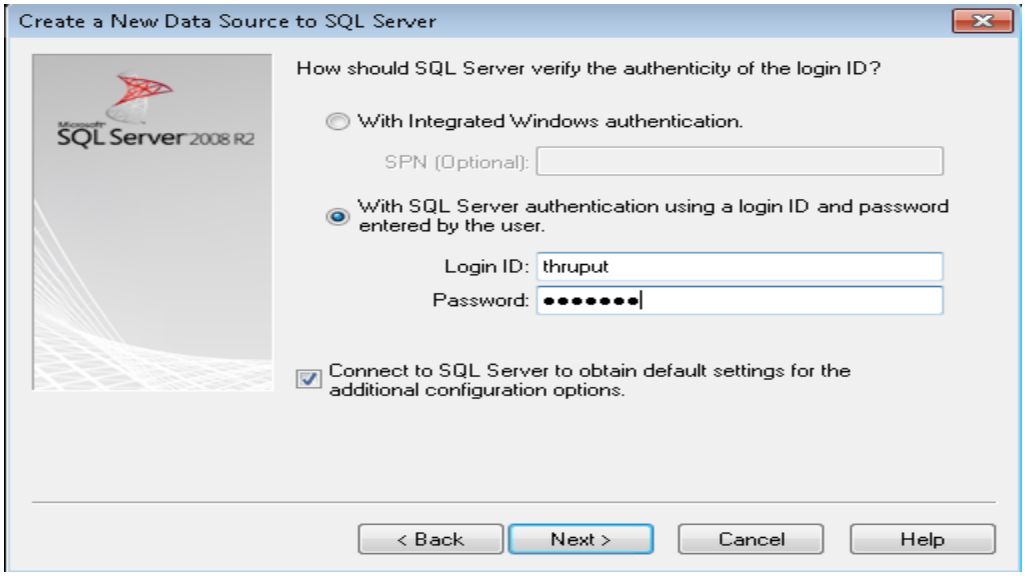
5. On Pressing the **Add** button, you will get the dialog as shown in the below figure where you need to select SQL Server Native Client 10.0 (or SQL Server Native Client 11.0) and then press **Finish** button.



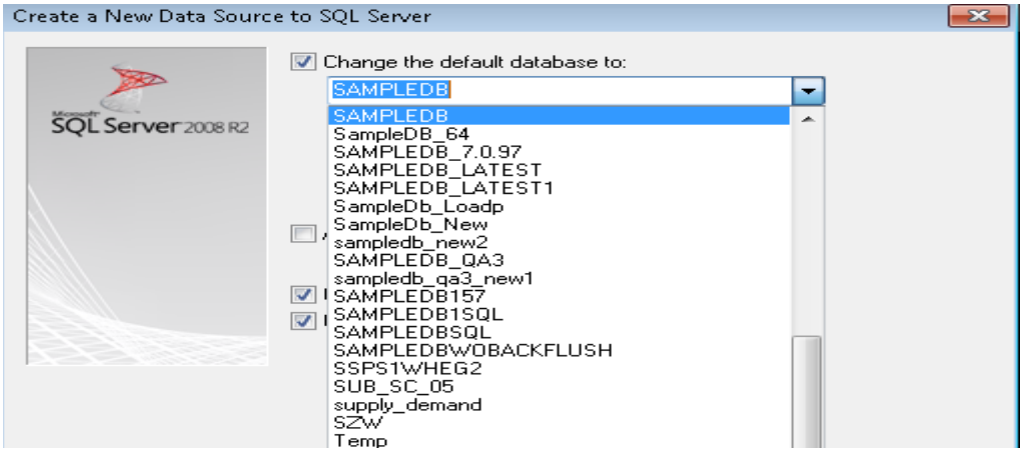
6. On clicking **Finish** button, you will get a dialog box where you need to give the name of the data source name (DSN) and to which server you want to connect to and then click **Next** button.



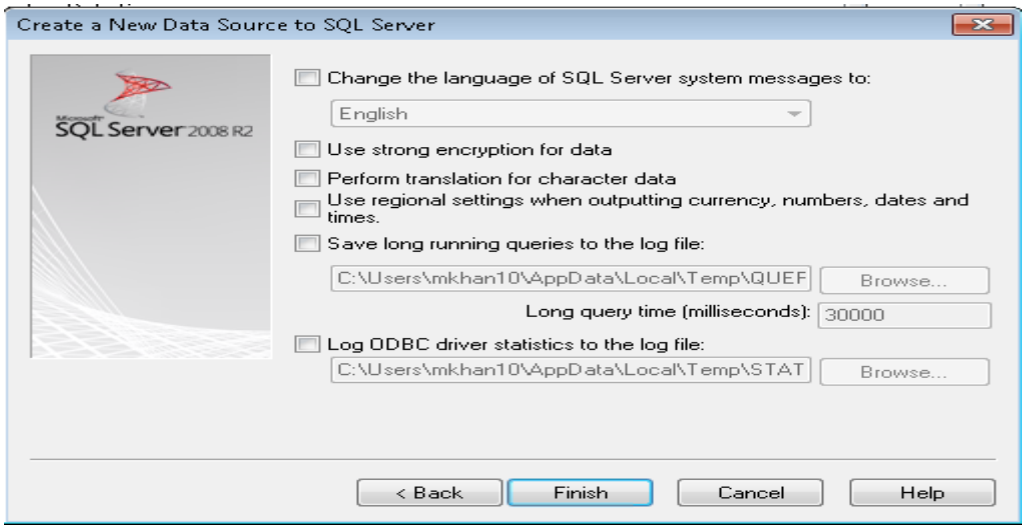
7. On clicking **Next** button, you will get a dialog box where you need to provide the *Login ID* and *Password* selected radio button option as shown in the figure given below and then press Next button.



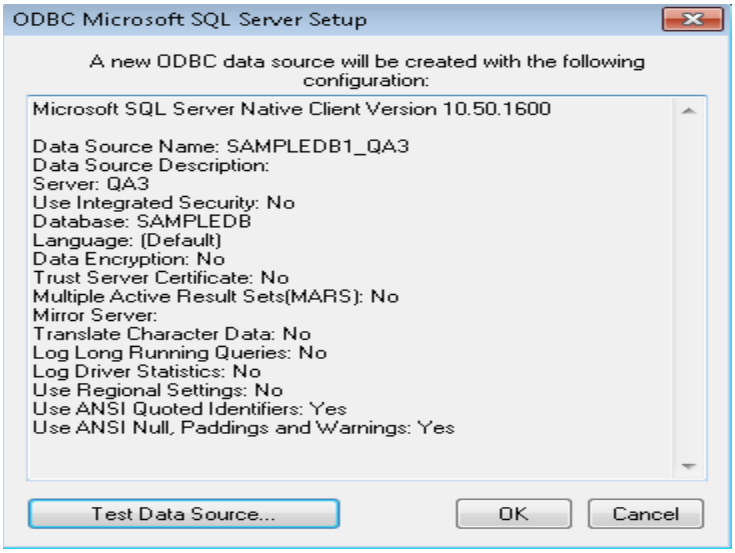
8. On clicking **Next** button, you will get a dialog box where you need to select the database which you want with the Data Source Name(DSN) shown in the figure given below.



9. Click **Next** button, next pop-up dialog will appear.



10. On clicking **Next** button, you will get a dialog as shown in the figure given below and then you can click on **Test Data Source** button and then hit the OK button to check the connectivity status. At last, click **Ok** to create the DSN.



11. Click OK. The OBDC Data Source Administrator display appears.

12. Click OK.

13. Select the entry you just created or modified, and click OK.

Tracking errors

Thru-Put logs errors to an error log file in the download data directory. Monitor the error log closely while you become acquainted with Thru-Put. It helps you evaluate the quality of your database builds. The software generates both warning messages and errors. The error messages detail the reasons for the error and the action the software took to overcome the error. Sometimes, these messages do not significantly affect your plant model. In that is the case, proceed to a scheduling session. If the messages can lead to incorrect schedules, correct the errors in your host system and download the data again. Repeat the process until you have a satisfactory data set.

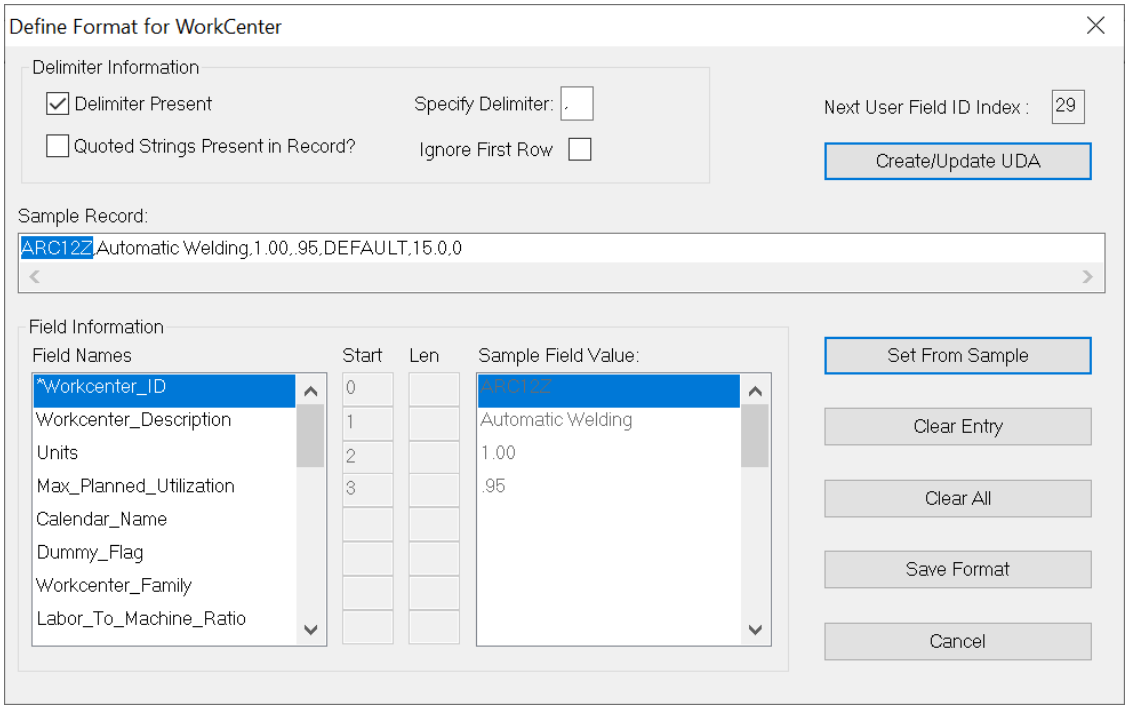
Preparing format files

In order to use your format files (.FMT), you must download your data files and specify the download file names in the download filer configuration window. Create format files for each of your data files to tell DBUtil their layout using one of the following methods:

- Automatic preparation
- Manual preparation

Automatic preparation

Select File > Prepare Format Files from DBUtil. A list of format file appears. Select the format file name from the list. The following display appears:



Select one of the following format files and click Save Format:

- WorkCenter
- Part
- Bill of Material
- Routings
- Work In Process
- Work Order Header
- Work Order Routing
- Work Order Allocation
- Inventory
- Jobs In Progress
- Purchase
- Forecast
- Sales

The Define Format display appears. Map the fields in your download data files to the input fields required by DBUtil. You can choose fixed-width formats or you can use delimiters to separate the field information.

Delimiter information	Select one of the following options. • Delimiter Present

<table>
<tr><td></td><td>• Specify Delimiter

• Quoted Strings Present in Record?

If the fields in your data file are delimited, select Delimiter Present and type in the delimiter in the Specify Delimiter field. Otherwise, SCM assumes your file has fixed-length fields. If the strings in your download are enclosed in quotes, for example, "PART-1", select Quoted Strings Present in Record?.</td></tr>
<tr><td>First User Field ID Index</td><td>The first field number available for you to download additional data to be used for custom reports.</td></tr>
<tr><td>Sample Record</td><td>A sample data record from your data file appears in the Sample Record field. To see a sample record:

• Ensure you have a real or sample data file in the format which you intend to download.

• Check that you specified the data file name in the Download Files page against the correct file name.

• Rename the *.OLD files to *.DAT if you have already successfully built the database</td></tr>
<tr><td>Field Information options</td><td>
<table>
<tr><td>Field Name</td><td>Description</td></tr>
<tr><td>Field names</td><td>Name for the field in this table in the SCM database</td></tr>
<tr><td>Start</td><td>input data string from which this field starts. Applies only if you use fixed width formats rather than delimited texts.</td></tr>
<tr><td>Len</td><td>Start length of the field. See the input file sections in the document to see how much length each field can be. Applies only if you use fixed-width formats rather than delimited texts</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td><table>
<tr><td>Sample field value</td><td>A sample field value extracted from the first line of the data file.</td></tr>
</table></td></tr>
<tr><td>Prepare format files dialog buttons</td><td><table>
<tr><td>Option</td><td>Description</td></tr>
<tr><td>Set from Sample</td><td>Select highlighted values from the sample record and map these to appropriate field names.</td></tr>
<tr><td>Clear entry</td><td>Remove the currently mapped column or field. Unmaps the current field and does not extract data for the column.</td></tr>
<tr><td>Clear All</td><td>Remove all mapped columns from the format file.</td></tr>
<tr><td>Save format</td><td>Create or update the Format file (.FMT)</td></tr>
<tr><td>Cancel</td><td>Cancel all changes.</td></tr>
</table></td></tr>
<tr><td>Mapping the fields</td><td>The field-mapping procedure differs, depending on whether or not the fields are delimited. If the fields are delimited:

1. Click on the field in the Field Names list that you want to map.

2. Click any part of the field in the Sample Record. You can scroll the Sample Record field if required.

3. Then click Set From Sample. The offset of the field (zero-based) appears in the Start box against the field name and the sample data appears in the Sample Field Values list.

If the fields are not delimited:

1. Click on the field in the Field Names list that you want to map.

2. Click the start of the field in the Sample Record by and drag to the end of the field. You can scroll the Sample Record field if required.

3. Then click Set From Sample. The offset of the field (zero- based)</td></tr>
</table>

	appears in the Start box against the field name. The length of the field appears in the Length box. The sample data appears in the Sample Field Values list. **Note**: All mandatory field names are marked with an asterisk. You must map fields
Saving the format	1. Click Save Format. A dialog box appears telling you that you may not have mapped several optional fields. 2. Click Yes to leave these fields unmapped.

Note: The format files have the same filename prefix as the corresponding data file and a .FMT extension. They appear in the Download directory.

Manual preparation

You can manually create format files as fixed-length field files. Each format file must have the same name as a corresponding data file, and an extension of .FMT. For example, the format file for PART.DAT must be named PART.FMT.

Store format and data files in the same directory. You should maintain a separate directory for input data and format files. You must use the following conventions when you manually create the format files.

Convention	**Used for**
[ ]	Optional information
{ }	Required information
“ “	First line
{Field name}{white space}{Field sequence in file structure}{white space{Number of characters in data file preceding this field}{white space}{Field length}	Second and all succeeding lines

For example, the format file for a part file containing two fields: Part_ID and Part_Type. Part_ID is 10 characters long and starts at column 1. Part_Type is 1 character long and starts at column 11. It should appear as follows:

```
""
Part_ID 1 0 10
Part_Type 2 10 1
```

Delimited fields format file

You can specify to delimit the input files by modifying the format file as follows for the first line:

"{delimiter}"[white space]{1, if text fields are enclosed in double quotes or 0, if text fields are not enclosed in double quotes}

Modify the second and succeeding lines (one line for every field in download file) as follows:

{Field name}{white space}{Field sequence}{white space}{Number of fields in data file preceding

For example, the format file for a part file containing tw fields: Part_ID and Part_Type. Part_ID is the second field and Part_Type is the first field. The fields are delimited by a comma, and are not enclosed in double quotes.

"," 0 Part_ID 1 1 Part_Type 3 0

Input Files

Data files contain information that must be imported into the SCM database. These files can have names up to eight characters long and the extension must be DAT. The file names are specified in DbUtil Configuration. See XREF (refer to the chapter on Configuring DBUtil). The data fields can be fixed length or delimited by any single character. Decide the approach you want to use and convey your information through the format files.

Each row in the data file must contain one record. Each record must contain one set of values for all the fields. The following areas document the required fields and their maximum lengths. Fields longer than the maximum length will be truncated. You can use a smaller length for any of the fields by specifying the length in the format files. The fields described in this section are intended for downloading from a host system or for creating input data for SCM. After scheduling, SCM creates additional fields and tables.

Once the data and format files are ready on your PC, you can run DbUtil to build the SCM database. The input (or format) files are important when you configure your MRP or ERP database to work with SCM.

The input files are:

- Bill of Material (BOM)
- Forecast

- Inventory
- Jobs in Progress (JIP)
- Labor Classes
- Part Master
- Purchase
- Routings
- Sales
- Sequence Dependent Setup
- Supply Chain Cross-Reference (SC_PART_XREF)
- Supply Chain Transit Time (SC_TRANSIT_TIME)
- Work in Process
- Work Centers
- Work Order Allocation
- Work Order Header
- Work Order Routing.

Some fields are marked with an asterisk (*) in the file summary tables. They are required by SCM in order to perform accurate processing. If you do not specify any of these fields, SCM generates an error message and further processing is terminated.

User-Defined Attributes (UDAs)

You can add and populate fields to input database tables in SCM; however, SCM will not use these fields. You can import data from these fields into the DBAs in DBUtil.

Use this feature to create custom reports or create links with other databases or applications.

Warning: Each time you change the .UDA file, you must re-create the database files using DBUtil in the Erase & Overwrite mode. Unexpected results may occur if the files are not re-created.

Manually Specifying user-defined fields for a table

Create a file with the name of the database table (not the import data file) and a .UDA extension in the database directory (for example, create WKCTR.UDA for the workcenter table).

Create the fields using the following template using a semi-colon (;) to separate each part of the description as indicated:

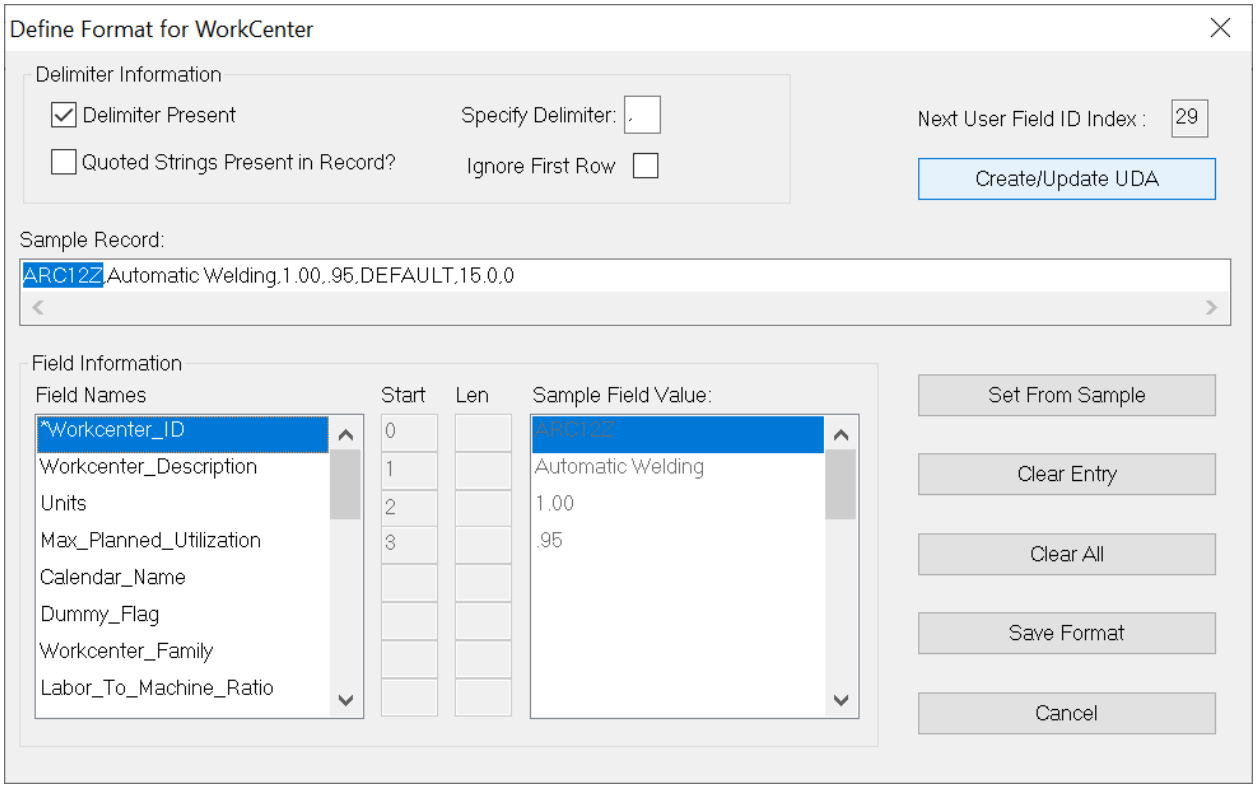
First line: Number of fields

Second line and onwards (one line for each field): Field name; field type; display properties; length;

For example, to create three fields named NEWFIELD1, NEWFIELD2, and NEWFIELD3, the first of alphanumeric type and length 13, the second of numeric type, and the third a date field, the .UDA file looks like this:

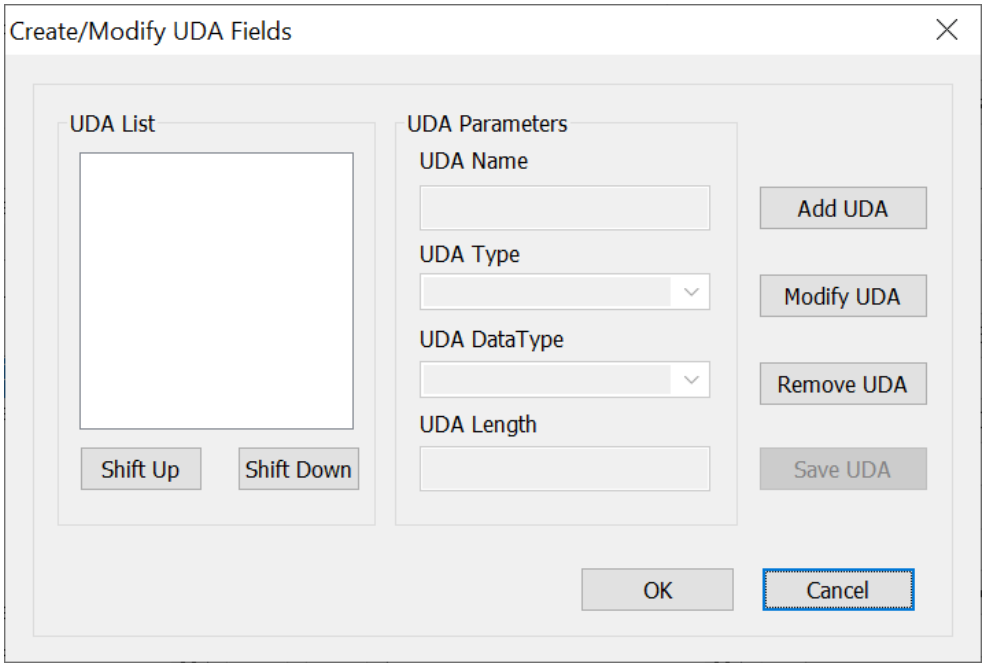
```
2

NEWFIELD1; alphanumeric; char;13;

NEWFIELD2; numeric; double;0;

NEWFIELD3; date; date; 0;
```

Automatically creating/updating user defined fields for a table



To automatically create/update UDA fields, user can click on “Create/Update UDA” button in above dialog.

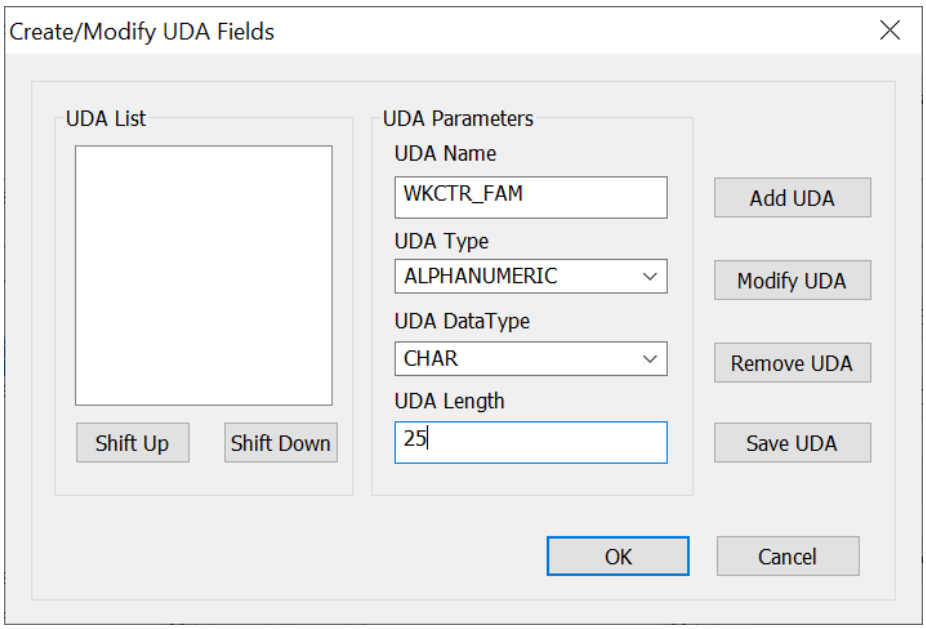
This will open below dialog:



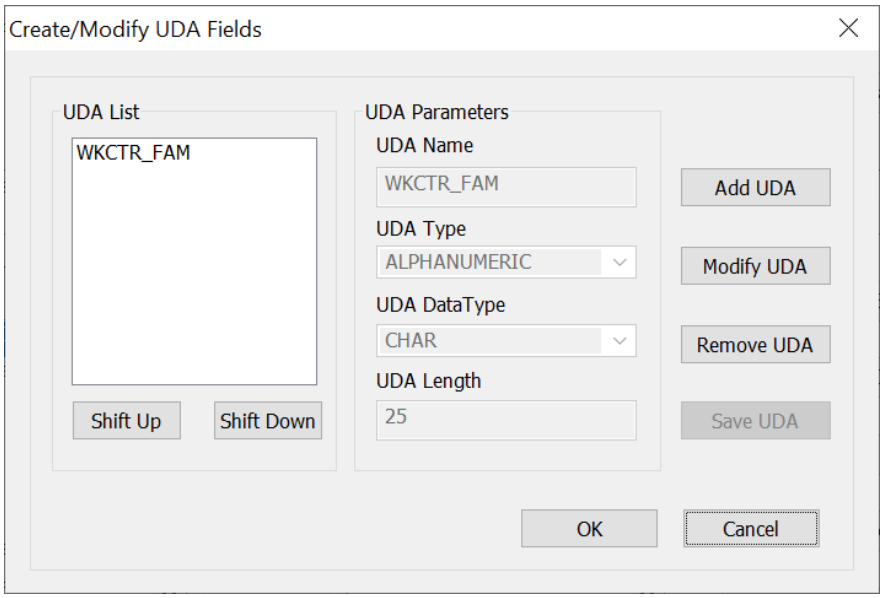
From this dialog, user can Add new UDA field, Modify any existing UDA field or Remove existing UDA field. Once changes are done, then using Save UDA, can save the changes to *.uda file.

Lets try adding a new UDA. For this, follow below steps:

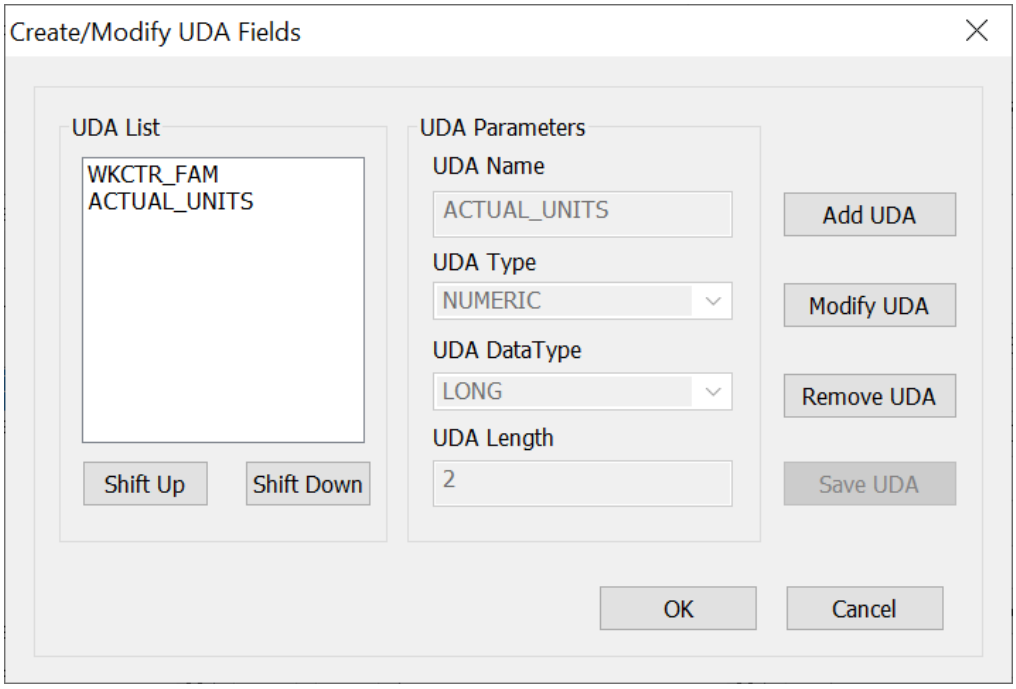
1. Click on Add UDA. This will enable all fields in UDA parameters section. Key-in values for UDA fields. An exmaple is shown below:



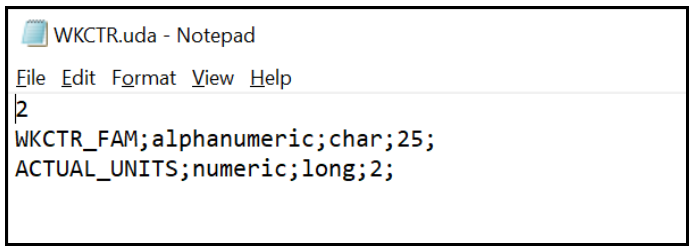
2. Click Save UDA. This will save this field in WKCTR.UDA file and will also list under UDA fields as shown below:



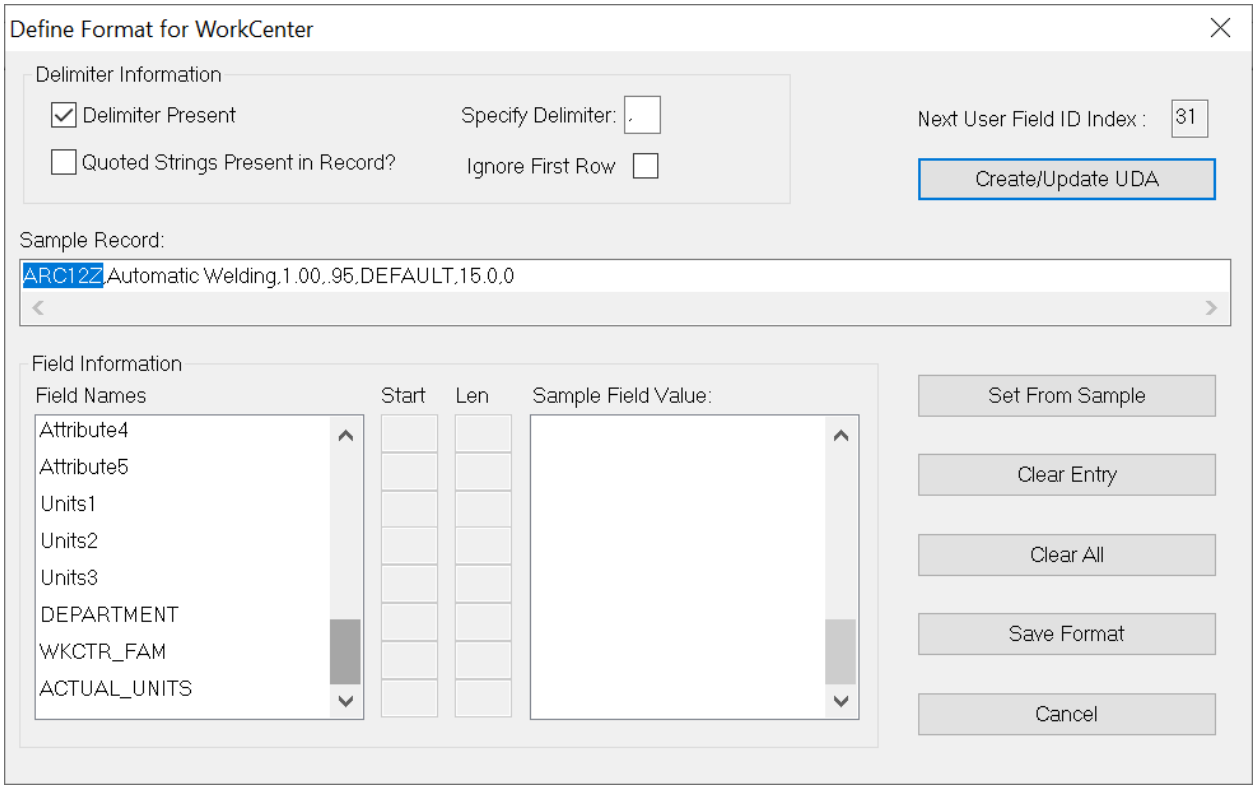
3. Similarly, we have added another integer type field for another UDA field named ACTUAL_UNITS as shown below:



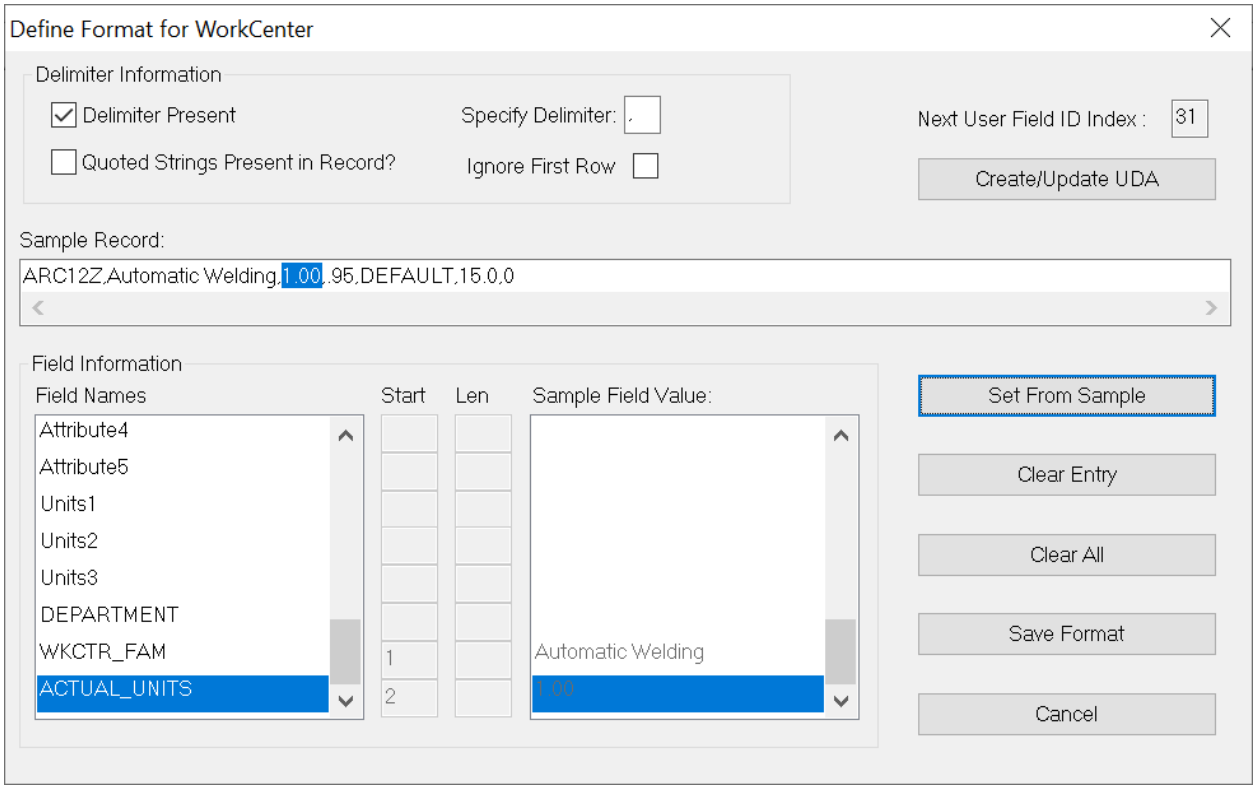
4. Once all UDA fields are added here, click OK button. You can verify WKCTR.UDA (in your Input Data directory folder) for these fields as shown below:



5. You will also see these UDA field in format file dialog at the end as shown below:



6. User can map these fields like other standard fields from above dialog as displayed below, Save Format and close this dialog:



7. Now building the database will correctly add these 2 new fields at the end in WKCTR table and data will also get populated from .dat files for these UDA fields.

Importing data into user-defined fields

Edit the format file for the database manually to specify the offsets for UDAs. Do not include a space between the semi-colons in the .UDA file. The format for user-defined fields starts after the regular fields. You can find the information from the prepare format screen for that table in DbUtil.

Specify the formats for all user fields imported from the data file in the ascending order of field IDs. Map the fields one-to-one with the fields defined in the .UDA file. The field ID for the first user field should be one more than the number of fixed fields in the table. In other words, if there are 13 standard fields in the Workcenter table, the new fields should have the IDs 14 and 15

PartImport From Multiple .Dat Files

Currently customers download data from ERP system in a single .dat file for each master table and then run DBUtil to import the same in corresponding master table in Thru-Put local database (MS Access or SQL Server). But quite often customer may be getting data downloaded from multiple sources for same master table. As an example, customer may be fetching BUYER related data (such as BUYER_ID, BUYER_LEAD_TIME) from one source for PART table; while the rest of the fields for PART table may be fetched from another source. In such a case, customer will be having 2 different .dat files for building PART table. But since DBUtil expects only one single .dat file for PART; so customer will need to manually copy-paste the data from both dat files into one single dat file and feed to DBUtil.

DBUtil will parse and import data from multiple dat files for a given master table. Currently, this feature is available for PART table and Forecast only;

There is an additional value introduced for existing config parameter **PartOverride** under [ERP Systems] as below:

PartOverride=2;<***Second dat file name without any extention***>

Points to note here are:

a) PartOverride value of 2 indicates that PART table data needs to be imported from multiple dat files. This is followed by a semi-colon (;) and then follows second dat file name which contains some of the columns data for PART table.

b) First dat file name is picked from standard **PartFile** config parameter under [**Download Files**] section as:

   [Download Files]

   PartFile=PART

c) The fmt file name for second dat file should be same as its dat file name except that it will have an extension of .FMT. E.g. if we have PartOverride=2;PART_NEW, then it indicates second dat file name is PART_NEW.dat; and similarly fmt file used for parsing this second dat file should be named as PART_NEW.fmt.

d) For regular data import from one dat file only, user can simply have PartOverRide=0 in config file.

e) The mapping mentioned in fmt files for both first and second dat file should be correct and in sync with data we have in .dat files.

f) All these .dat and .fmt files should reside in the directory mentioned in **InputFileDir** config parameter under [***Download Files***] section.

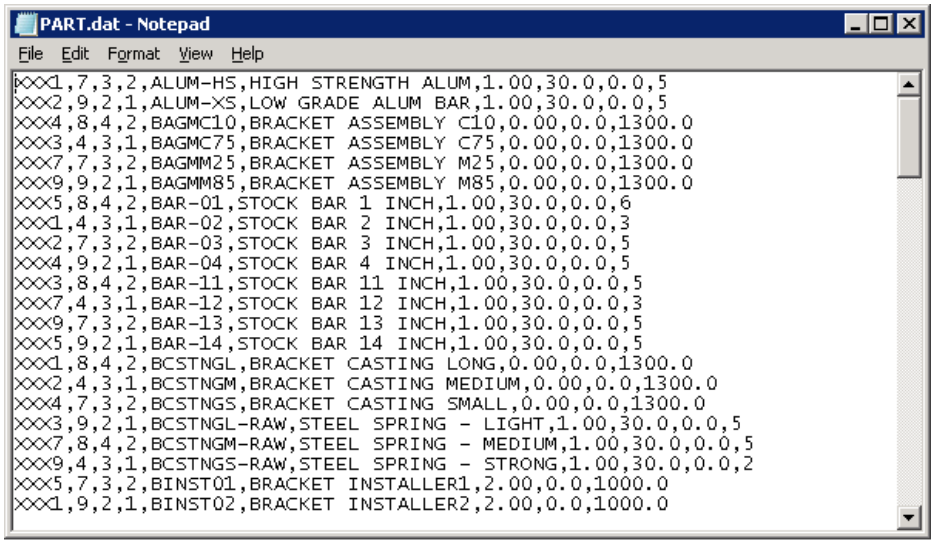
g) Another most important point is to make sure that data for primary fields is available and mapped in both set of .dat and .fmt files. E.g. For Part Table, we have Part_ID as Primary field, so both set of .dat files for PART should have data for PART_ID column so that DbUtil can correctly update the same for respective record in PART table.

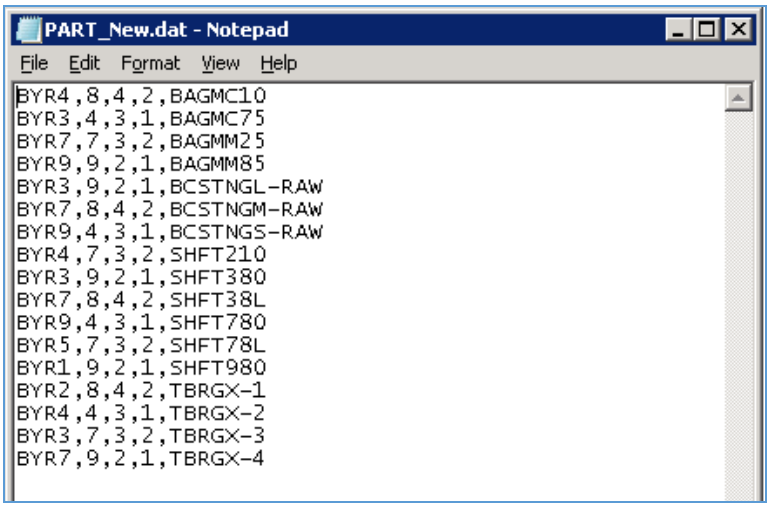
Once this input data has been correctly setup, user needs to click on Build Database option in DBUtil. It will first parse data from first set of .dat file for PART table (file name as mentioned in **PartFile** flag) and will hold it in memory. Then it will check PartOverride flag value; if its value has been mentioned as 2, then it will check file name for second .dat file for PART table. Then it will parse second dat file for PART table and update the corresponding PART records in memory for the columns read from second dat file. Once that is done, it will write the data in memory to PART table in Thru-Put local database.

Example

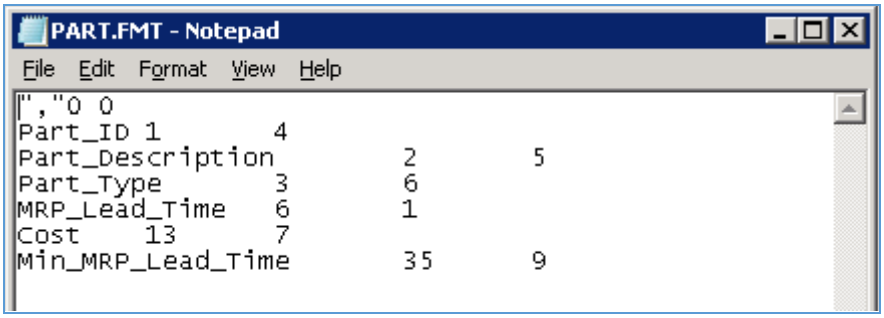
Input Data:

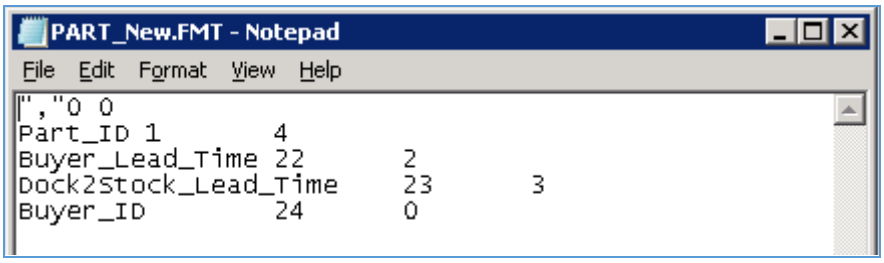
a) Let us assume we have below 2 set of .dat file for PART table. The files are named as PART.dat and PART_NEW.dat. Here PART.dat is primary file which contains data for all records. PART_NEW.dat file is secondary file which may or may not contain for all records. Here for this example, in PART_NEW.dat file, we have data for Buyer_ID, Buyer_Lead_Time and Dock2Stock_Lead_Time fields for some of the parts; along with PART_ID as that is primary key. Data for rest of the fields is in PART.dat file which contains all the records.



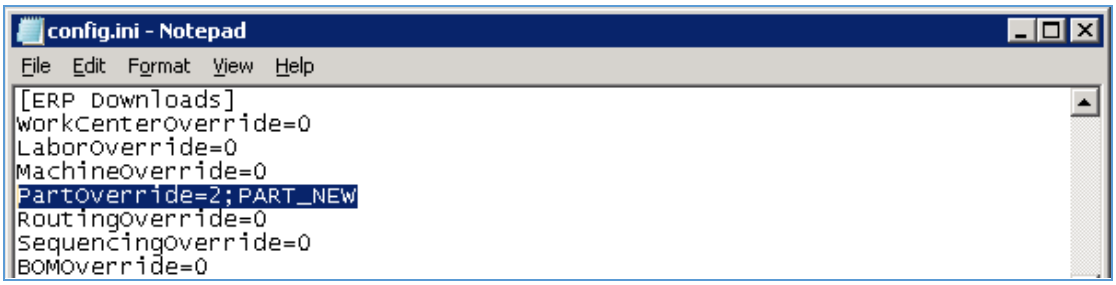


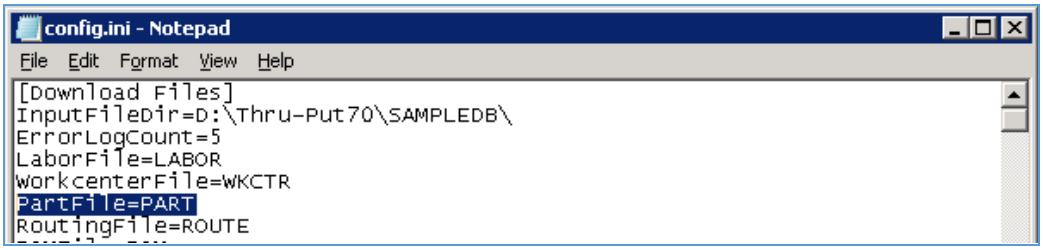
b) The corresponding fmt files for these 2 set of dat files are shown below and are named as PART.fmt and PART_NEW.fmt:



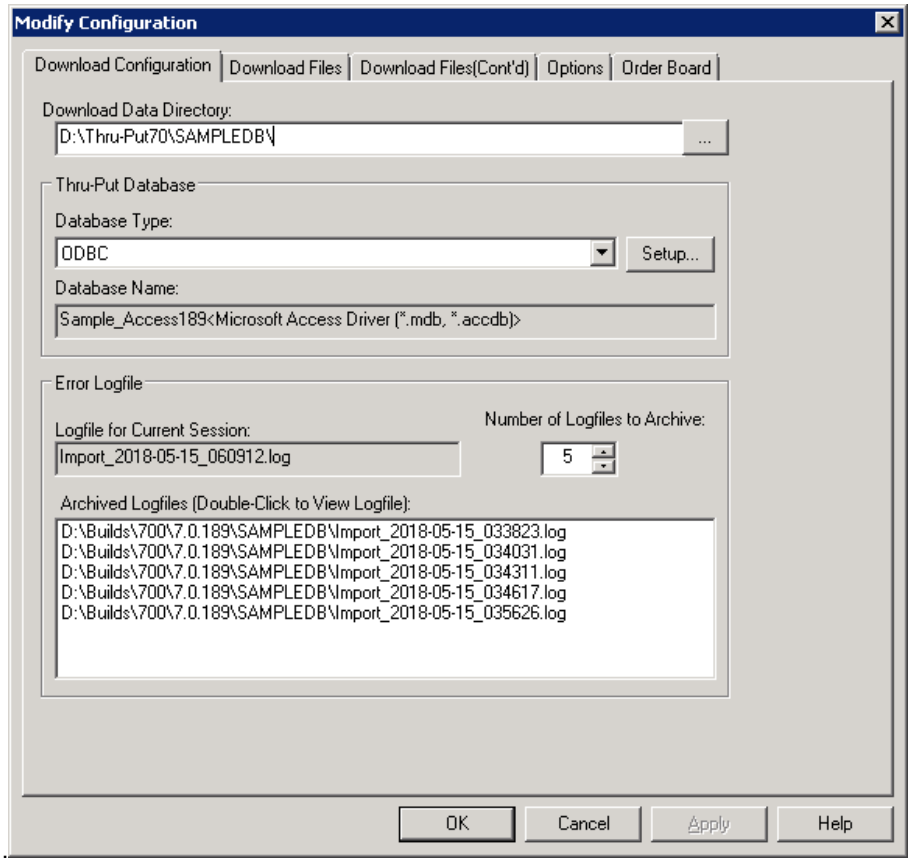


c) On config.ini file, we have below values for **PartOverRide** and **PartFile** flags:



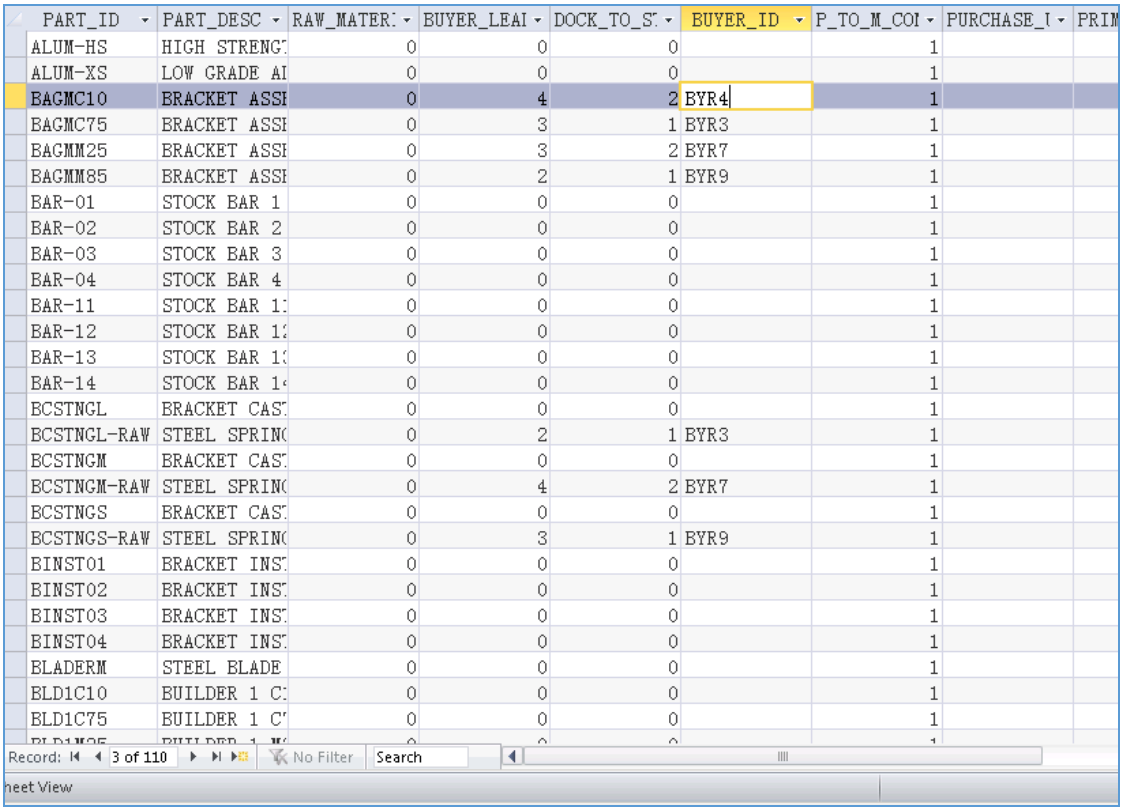


d) Once above data is set, open DBUtil and make sure **InputFileDir** points to the directory where all .dat and .fmt file resides. For our example, we have the .dat and .fmt file in D:\Thru-Put70\SampleDB folder; which we can correctly see under Download Data directory:



e) Click OK in above dialog. Now click on **Build DB** button from the top menu which should start building of database. Once that completes fine, exit out of DbUtil.

f) Now go to Thru-Put local database and open PART table. We can correctly see BUYER_ID, BUYER_LEAD_TIME and DOCK2STOCK_LEAD_TIME correctly updated for the parts in PART_NEW.dat file:



PART_ID	PART_DESC	RAW_MATER	BUYER_LEAI	DOCK_TO_S	BUYER_ID	P_TO_M_COI	PURCHASE_I	PRIM
ALUM-HS	HIGH STRENG	0	0	0		1		
ALUM-XS	LOW GRADE AI	0	0	0		1		
BAGMC10	BRACKET ASSI	0	4	2	BYR4	1		
BAGMC75	BRACKET ASSI	0	3	1	BYR3	1		
BAGMM25	BRACKET ASSI	0	3	2	BYR7	1		
BAGMM85	BRACKET ASSI	0	2	1	BYR9	1		
BAR-01	STOCK BAR 1	0	0	0		1		
BAR-02	STOCK BAR 2	0	0	0		1		
BAR-03	STOCK BAR 3	0	0	0		1		
BAR-04	STOCK BAR 4	0	0	0		1		
BAR-11	STOCK BAR 1	0	0	0		1		
BAR-12	STOCK BAR 1	0	0	0		1		
BAR-13	STOCK BAR 1	0	0	0		1		
BAR-14	STOCK BAR 1	0	0	0		1		
BCSTNGL	BRACKET CAS	0	0	0		1		
BCSTNGL-RAW	STEEL SPRIN	0	2	1	BYR3	1		
BCSTNGM	BRACKET CAS	0	0	0		1		
BCSTNGM-RAW	STEEL SPRIN	0	4	2	BYR7	1		
BCSTNGS	BRACKET CAS	0	0	0		1		
BCSTNGS-RAW	STEEL SPRIN	0	3	1	BYR9	1		
BINST01	BRACKET INS	0	0	0		1		
BINST02	BRACKET INS	0	0	0		1		
BINST03	BRACKET INS	0	0	0		1		
BINST04	BRACKET INS	0	0	0		1		
BLADERM	STEEL BLADE	0	0	0		1		
BLD1C10	BUILDER 1 C	0	0	0		1		
BLD1C75	BUILDER 1 C	0	0	0		1		

Record: 3 of 110 | No Filter | Search

heet View

Extracting data directly from ERP database

In the prior sections, it was assumed that text data was extracted first from ERP database and then input to Thru-Put DB Utility. That was useful when data is in multiple systems in multiple formats. However, if the all the ERP data that is required in Thru-Put for planning are stored in a single Relational databases such as IBM DB2, Oracle or Microsoft SQL Server, it might be more optimal to connect to the ERP Database using ODBC DSN connection and extract the required data from views or tables.

Following sections provide details on using ODBCERP Integration for extracting data from ERP system and building Thru-Put database.

ODBCERP Integration is used to extract data from any ERP system to Thru-Put database by making use of SQL queries to fetch the data from ERP system. Here we will focus on below steps which are needed as part of ODBCERP Integration:

- Configuring config.ini to use DBUtil for ODBCERP Integration
- Configuring DBUtil for the files which need to be downloaded.
- Writing data extraction SQL for each of the input file download we have selected in Dbutil.
- Creating format files
- Use DBUtil to build database

Pre-Requisites

This document assumes that you have an existing version of Thru-Put software is already installed on your computer.

Configuring config.ini to use DBUTIL for the ODBCERP interface

1. First open the config.ini for the warehouse you have set up. If you have not set up the plant/warehouse yet, then this file would be in the root directory of the Thru-Put installation folder, i.e., this same folder would also contain the files Resonance.exe and DBUTIL.exe

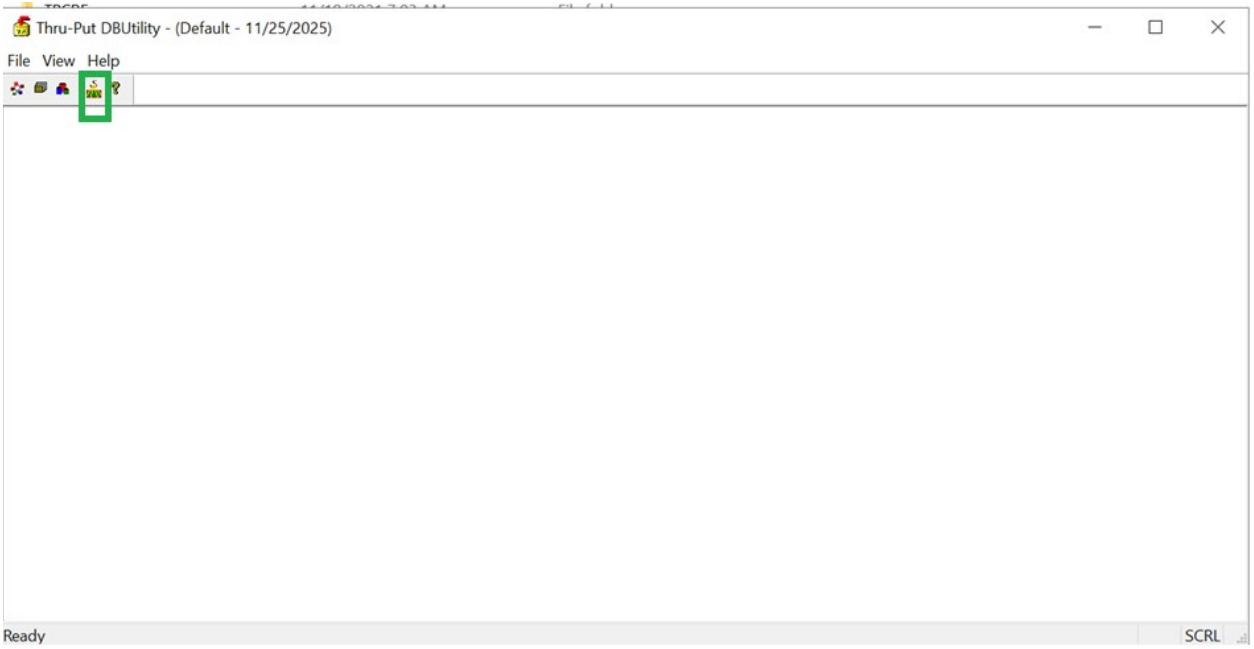
2. Set the following in the system section

[System]

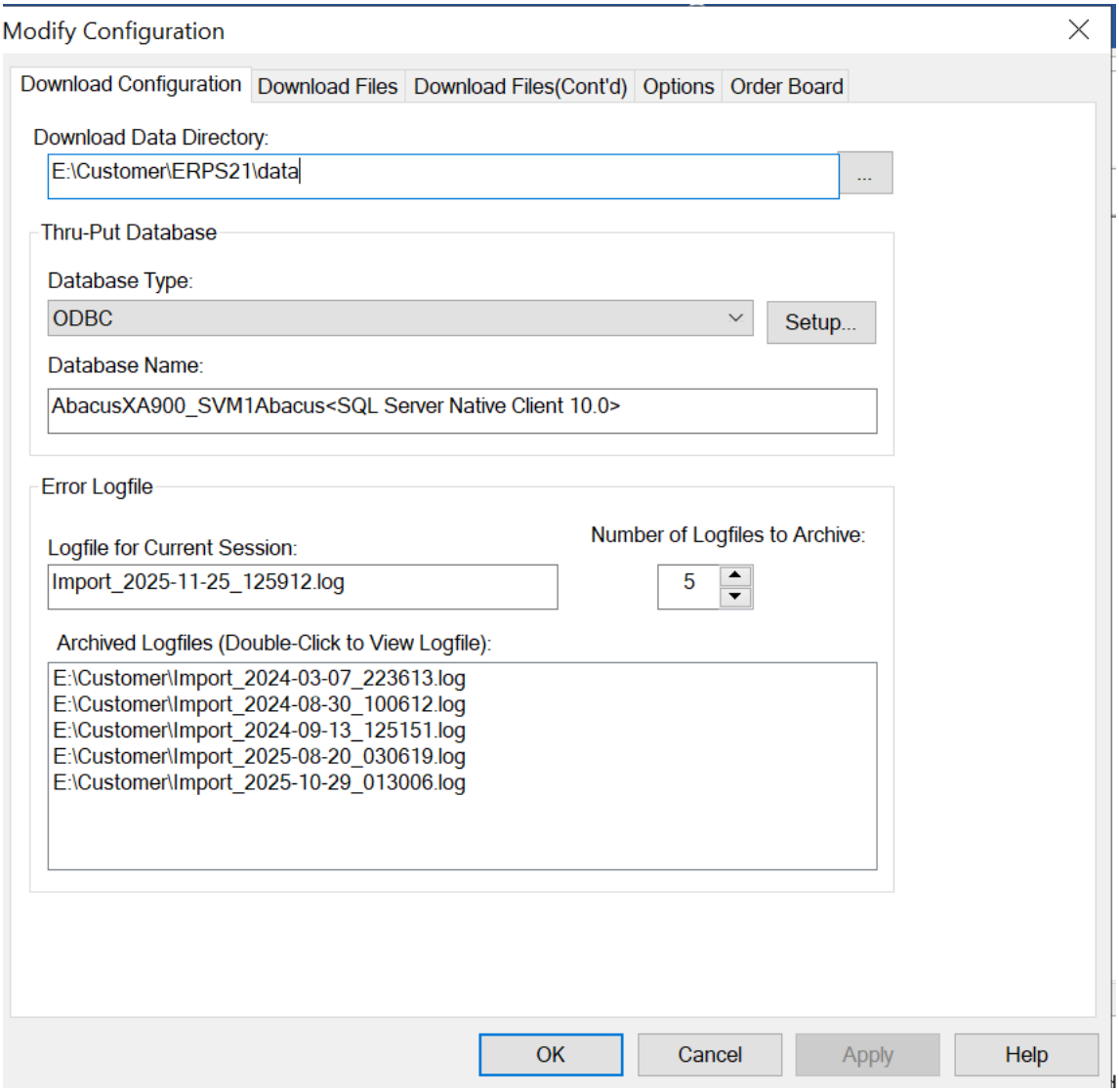
ERPSystem=ODBCERP

3. Save the changes and exit out of config.ini

Run DBUTIL by going through Start->Programs->Thru-Put->Database Utility. Now, you should see the third icon in the tool bar enabled (Box lifting icon) as shown below:



Go to File->Download Configuration. You will see following dialog:



Change the download directory to point to ERP sub folder as shown above. Specify target data source location where Thru-Put table schema to be stored. Hit OK button to save the changes. Exit out of DBUTIL. This step ensures that settings related to the ERP interface get written to the config.ini.

1. Again open up config.ini again.

2. Set the following in the OA_ERP settings

[OA_ERP]

UserName= [ERP Database Server user name]

Password=[Password to log onto the ERP server]

ConnectString= [DSN Name which has been setup for ERP server]

SQLFile= [The query file containing the download queries for fetching data from ERP server]

ERPSystem=ODBCERP

There will other parameters under the OA_ERP section, do not worry about them for now. An example of above settings is shown below:

3. Save the changes and exit out of config.ini

Please see below example showing OA_ERP section values:

[OA_ERP]

UserName=R23AVPUSR

Password=R23AVPUSR

ConnectString=S21_IntegrationSQL

SQLFile=D:\Thru-Put70\ERPS21\S21ERPQuery.sql

ConsumeForecastInResonance=1

Preparing DB-Mapping Query file (download SQL file)

This file consists of set of queries that map the ERP database schema with Thru-Put database schema. There is one query per data input file of Thru-Put such as PART, ROUTE, SALES etc. Each of these queries has a standard start signature and end signature. For example, part query is enclosed between PART_QUERY_START and PART_QUERY_END.

Here is the list of query templates for different input files:

Plan Control file (PPIPLN) used for creating RESPARA table:

PLANNING_PARA_QUERY_START

<Please provide select query here>

PLANNING_PARA_QUERY_END

Workcenter Table:

WKCTR_QUERY_START

<Please provide select query here>

WKCTR_QUERY_END

Part Table:

PART_QUERY_START

<Please provide select query here>

PART_QUERY_END

Route Table:

ROUTE_QUERY_START

<Please provide select query here>

ROUTE_QUERY_END

JSBOM Table:

BOM_QUERY_START

<Please provide select query here>

BOM_QUERY_END

SALES Table:

SALES_QUERY_START

<Please provide select query here>

SALES_QUERY_END

FORECAST Table:

FORECAST_QUERY_START

<Please provide select query here>

FORECAST_QUERY_END

PURCHASE Table:

PURCHASE_QUERY_START

<Please provide select query here>

PURCHASE_QUERY_END

WOHEAD Table:

WOHEAD_QUERY_START

<Please provide select query here>

WOHEAD_QUERY_END

WOALLOC Table:

WOALLOC_QUERY_START

<Please provide select query here>

WOALLOC_QUERY_END

WOROUTE Table:

WOROUTE_QUERY_START

<Please provide select query here>

WOROUTE_QUERY_END

Used for updating QOH in PART Table:

FGI_QUERY_START

<Please provide select query here>

FGI_QUERY_END

SC_CUSTOMER_MASTER Table:

CUSTOMER_MASTER_QRY_START

<Please provide select query here>

CUSTOMER_MASTER_QRY_END

SC_CUSTOMER_DETAILS Table:

CUSTOMER_DETAIL_QRY_START

<Please provide select query here>

CUSTOMER_DETAIL_QRY_END

SC_VENDOR_MASTER Table:

VENDOR_MASTER_QRY_START

<Please provide select query here>

VENDOR_MASTER_QRY_END

SC_VENDOR_DETAILS Table:

VENDOR_DETAIL_QRY_START

<Please provide select query here>

VENDOR_DETAIL_QRY_END

SOURCING_RULES TABLE:

SOURCING_RULE_QRY_START

<Please provide select query here>

SOURCING_RULE_QRY_END

WKCTR_MAINT Table:

WKCTR_MAINT_QRY_START

<Please provide select query here>

WKCTR_MAINT_QRY_END

SC_PART_XREF Table:

SC_PART_XREF_QUERY_START

<Please provide select query here>

SC_PART_XREF_QUERY_END

SC_PLANTS Table:

SC_PLANTS_QUERY_START

<Please provide select query here>

SC_PLANTS_QUERY_END

SC_PART_CUSTOMER Table:

SC_PART_CUSTOMER_QUERY_START

<Please provide select query here>

SC_PART_CUSTOMER_QUERY_END

SC_PART_VENDOR Table:

SC_PART_VENDOR_QUERY_START

<Please provide select query here>

SC_PART_VENDOR_QUERY_END

SC_TRANSIT_TIME Table:

SC_TRANSIT_TIME_QUERY_START

<Please provide select query here>

SC_TRANSIT_TIME_QUERY_END

DEMAND_QUERY_START

<Please provide select query here>

DEMAND_QUERY_END

LABOR_QUERY_START

<Please provide select query here>

LABOR_QUERY_END

PART_PROCESS Table:

PART_PROCESS_QUERY_START

<Please provide select query here>

PART_PROCESS_QUERY_END

WIP Table:

WIP_QUERY_START

<Please provide select query here>

WIP_QUERY_END

JIP Table:

QIP_QUERY_START

<Please provide select query here>

QIP_QUERY_END

SETUP_SEQUENCE Table:

SETUP_SEQUENCE_QUERY_START

<Please provide select query here>

SETUP_SEQUENCE_QUERY_END

SC_PLAN_PROFILE Table:

PLAN_PROFILE_QRY_START

<Please provide select query here>

PLAN_PROFILE_QRY_END

Preparing DB-Mapping Query for upload (Upload file)

Any update query which needs to be executed on the ERP server should be entered between below standard start and end signature tags. ***If there are multiple UPDATE queries; then those should be separated by semi-colon (;)****. Single UPDATE query need not be delimited by semi-colon; but it will be a good practice to end every query in this UPDATE section with a semi-colon for uniformity.*

TRACK_UPDATE_QUERY_START

<UPDATE Query 1>;

<UPDATE Query 2>;

TRACK_UPDATE_QUERY_END

Please note, semi-colon should NOT be used in any SELECT query used for any input file download.

Examples of SQL File Content:

Please refer to few examples of SELECT and UPDATE queries used in one sample SQL file:

WKCTR_QUERY_START

```
SELECT
mach12                                                  AS WORKCENTER_ID,
concat(concat(mach12,' '),mdes12)                       AS WORKCENTER_DESC,
case when dsl112 <> 0 then
dsc112 / dsl112
else 1 end                                              AS UNITS,
case when stef12 > 100 or stef12 = 0 then
1.0
else stef12/100 end                                     AS UTILIZATION,
''                                                      AS CALENDAR_ID,
'0'                                                     AS DUMMY_FLAG,
ifnull((select min(work16) from R4ST3F2.msp16 where cono16=a.cono12 and mach16=a.mach12),' ')
                                                        AS WORKCENTER_FAMILY,
1                                                       AS LABOR_MC_RATIO,
macq12 * 3600                                           AS FORCED_QUEUE_TIME,
0                                                       AS SETUP_TIME,
```

```
0                                                   AS RUN_TIME_PER_BATCH,
0                                                   AS IDLING_TIME_PER_BATCH,
0                                                   AS DRUM_NUMBER,
0                                                   AS OVERTIME_RATE,
0                                                   AS MIN_BUFFER_TIME,
0                                                   AS TYPICAL_BUFFER_TIME,
0                                                   AS SETUP_PERCENTAGE,
cono12                                              AS LOCATION_ID,
'0'                                                 AS RESOURCE_TYPE,
''                                                  AS ATTRIBUTE1,
''                                                  AS ATTRIBUTE2,
''                                                  AS ATTRIBUTE3,
''                                                  AS ATTRIBUTE4,
''                                                  AS ATTRIBUTE5
from R4ST3F2.msp12 a
where cono12 = 'Z1' and RSTS12 <> 'D'
```

WKCTR_QUERY_END

PLANNING_PARA_QUERY_START

```
select
        cast(cono02 as char(3)),
        cast(modn00 as char(3)),
        cast(VARCHAR_FORMAT(timestamp_format(char(psed02+19000000),
'YYYYMMDD'),'DDMMYYYY') as char(8)),
```

```
        cast(VARCHAR_FORMAT(timestamp_format(char(pscd02+19000000),
'YYYYMMDD'),'DDMMYYYY') as char(8)),

        cast(VARCHAR_FORMAT(timestamp_format(char(pstf02+19000000),
'YYYYMMDD'),'DDMMYYYY') as char(8)),

        cast(VARCHAR_FORMAT(timestamp_format(char(pssd02+19000000),
'YYYYMMDD'),'DDMMYYYY') as char(8)),

        '0',

        cast(VARCHAR_FORMAT(timestamp_format(char(pscd02+19000000),
'YYYYMMDD'),'DDMMYYYY') as char(8)),

        '1',

        cast(rsfz02 as char(5)),

        '000',

        '1',

        '00000'

        from

        R4SMVF2.mvp02, R4ST3F2.msp00

        where cono02 = 'Z1' and cono02=cono00
```

PLANNING_PARA_QUERY_END

TRACK_UPDATE_QUERY_START

```
UPDATE R4SMVF2.mvp02

    SET XTDT02 = CAST(CONCAT(1, SUBSTR(REPLACE(cast(CURRENT DATE as char(10)), '-', ''), 3)) as
char(7)),

       XTTM02 = CAST(REPLACE(cast(CURRENT TIME as char(10)), ':', '') as char(6))

    WHERE CONO02='Z1';
```

TRACK_UPDATE_QUERY_END

IMPORTANT POINTS:

1. *All update queries should be inside **TRACK_UPDATE_QUERY_START** and **TRACK_UPDATE_QUERY_END** block in the SQL file.*
2. ***If there are multiple UPDATE queries; then those should be separated by semi-colon**. Single UPDATE query need not be delimited by semi-colon; but it will be a good practice to end every query in this UPDATE section with a semi-colon for uniformity.*
3. *SELECT queries should not end with semi-colon.*
4. *For PPIPLN.dat file, the date format used in the query should match with **AsciiDateFormat** used in config file; else data in RESPARA table won't be created correctly.*

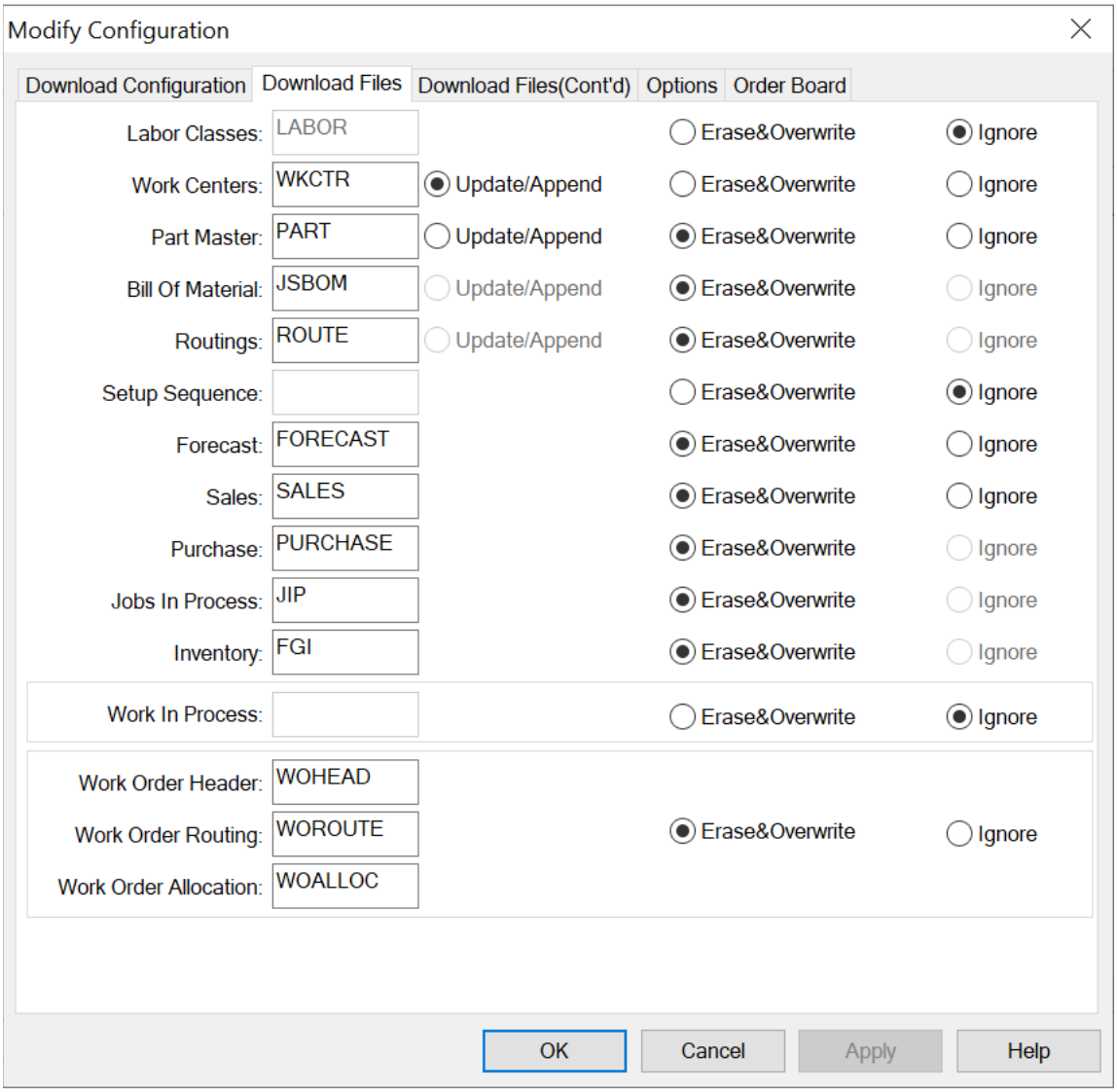
So far, we have:

- Have setup Source and target Data-Source;
- Configured DBUtil to use ERPSystem as ODBCERP;
- Have written data extraction SQL for each of the download we have in Dbutil.

Next step is to create format files that will map fields in ERP system with fields in Dbutil.

Configuring DBUtil for SQL download

Goto File Menu, Download Configuration and click on Download Files tab. Using this tab, one can decide which files to download. Make sure all the input files names have the same syntax as shown below. Make the button for DEMAND as ignore (This option for download is present for backward compatibility reasons and should not be used for the download of SALES and FORECAST). This will highlight the button for SALES and FORECAST. If you choose not to download FORECAST, you can set it to IGNORE.



The following points should be kept in mind:

Table 1: Download Files tab options

Option	Description
Update/Append	Update: If a record in the data file exists in the database table (i.e., the key fields match), then the record in the table is updated with the new data in the input data file. Append: A new record in the data file is inserted into the table.

Erase & Overwrite	When this option is selected, the existing table is dropped and rebuilt, i.e. all correct data records are inserted into the table. This is the default option.
Ignore	This table will not be built as long as this option is set.

To maintain referential integrity in the database, restrictions are imposed by DBUtil on the setting of the above flags.

If the Work Center table is set to Erase & Overwrite mode, then the Routing Table has to be set to the Erase & Overwrite mode, since it is possible that the Work Center information in the old routing records are inconsistent with the new Work Center data.

If Part table is set to Erase & Overwrite mode, the BOM and Routing tables have to be refreshed.

If either the Work Center table or the Part table is set to Erase & Overwrite mode, a filename must be specified for the Routing file for the above mentioned reasons.

The WIP, Inventory, Supply, and Purchase are always set to Erase & Overwrite mode.

Once we have identified for which tables we need to download the data; we will need to write our SQL file (mentioned against **SQLFile** config parameter under **OA_ERP** section). This file will contain the queries for extracting data from ERP database. Please refer to below section for the same.

Creating Format Files for SQL download

In order to use your format files (.FMT), you must download your data files and specify

the download file names in the download file configuration window. Select File >

Prepare Format Files. A list of format files appears. Select from the following:

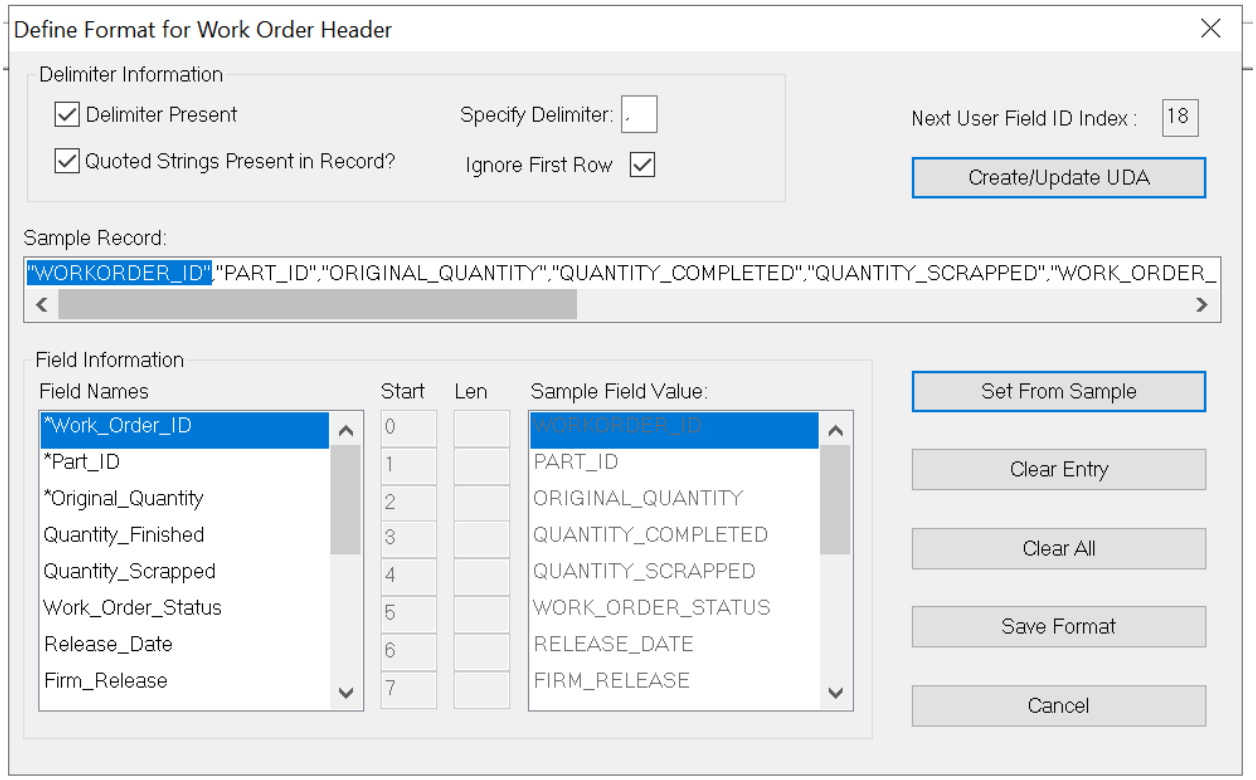
- Workcenter
- Part
- Bill of Material
- Routings
- Work in Process
- Workorder Header

- Workorder Routing
- Workorder Allocation
- Inventory
- Jobs in Progress
- Sequence Setup
- Purchase
- Labor Classes
- Forecast
- Sales
- Plant Transit Time
- Part-Plant Xref
- Plants
- Customer Master
- Customer Details
- Vendor Master
- Vendor Details
- Sourcing Rules
- Workcenter Maint

The Define Format display appears. Map the fields in your download data files to the input fields required by DBUtil. You can choose fixed-width formats or you can use delimiters to separate the field information.

For example, select the format file **Workorder Header** from the list. The following display

appears:



Create format files for each of your data files to tell DBUtil their layout by clicking on each filed and selecting the data from the sample. This method is same as the preparing format file for text input data. You can also add user defined attributes (UDA) just like the way you did text input extraction. Include those UDA fields to SQL file that goes with the input data. For example. For routing download you will have ROUTE.UDA that contains the additional fields you want to extract from your ERP database and at the end of Routing extraction SQL statement, you will have the ERP field name where you will get these data from.

Once format files are in place, then use Dbutil to extract data from a ODBC data source.

Building Thru-Put database/ Running DBUtil

Running DBUtil daily is same whether you use text file as input or SQL query for data extract.

Go to File->Build complete database. DBUtil will directly connect to ERP data-source mentioned under OA_ERP section; will fire SELECT queries provided in SQL File (file name mentioned as **SQLFile** config parameter in OA_ERP section) to extract data for the input files configured in DBUtil; will generate dat files corresponding to each input file and will load and build the corresponding table in Thru-Put database.

Checking errors during download

All integration related errors will be written to a text file called ERPIntegration.log in ERP sub folder (folder selected in the Download File configuration dialog on page 11). If there were any data related issues or download validation information will be written to import<datatimestamp>.log file in the same folder.

Batch Mode

Batch mode is common for both types of inputs i.e. text data and SQL query input. You can use batch mode commands as a part of a windows batch routine to run every night to capture a current snapshot of data from your ERP system to the Thru-Put database. The following commands are available to let you run DBUTIL in the batch mode.

Command	Description
DBUTIL –b	Build Complete Database Example: DBUTIL –b –n700_Sample where 700_Sample is plant name.
DBUTI -b -s	Building complete Database but skipping Order Board Matching Process. Example: DBUTIL –b –S –n700_Sample where 700_Sample is plant name.
DBUTI -bi	Updating The Database from ION Example: DBUTIL –bi –n700_Sample where 700_Sample is plant name.
DBUTIL –u –g	Upload Schedules to ERP and set Group ID needed for connecting to ERP Database while uploading ERP schedule Example: DBUTIL –u –g1 –n700_Sample where 1 after –g represents Group ID and 700_Sample is plant name.
DBUTIL –d	Set Download Date when starting DBUTIL. Example1: DBUTIL -dtoday+<No Of Days> (For Setting the Download Date Dynamically.) Example2: DBUTIL -d06/18/2014 (For Setting the Download Date statically.)
DBUTIL -i or -iwsp or -iwps	For updating INVENTORY

or -ipsw or -ipsw or -iswp or -ispw	Example: DBUTIL –i –n700_Sample where 700_Sample is plant name.
DBUTIL –iw	For Updating Table WOHEAD,WOALLOC,WOROUTE,JIP and FGI only. Example: DBUTIL –iw –n700_Sample where 700_Sample is plant name
DBUTIL -ip	For Updating Table PURCHASE and FGI only. Example: DBUTIL –ip –n700_Sample where 700_Sample is plant name.
DBUTIL –is	For Updating Table SALES and FGI only. Example: DBUTIL –is –n700_Sample where 700_Sample is plant name.
DBUTIL –iwp or –ipw	For Updating Table WOHEAD,WOALLOC,WOROUTE,JIP ,FGI and PURCHASE. Example: DBUTIL –iwp –n700_Sample where 700_Sample is plant name
DBUTIL –iws or –isw	For Updating Table WOHEAD,WOALLOC,WOROUTE,JIP ,FGI and SALES. Example: DBUTIL –iws –n700_Sample where 700_Sample is plant name
DBUTIL –ips or -isp	For Updating Table PURCHASE,SALES and FGI only Example: DBUTIL –ips –n700_Sample where 700_Sample is plant name.

NOTE: 1. You can use either '/' or '-' as Prefix before the arguments.

2.Command line argument is case-insensitive.

In batch mode, the configuration is based on the last settings saved in the CONFIG.INI file. All status and error messages are automatically directed to the log file.