**BellHawk®**

**Real-Time Work-in-Process Data Collection & Materials Tracking Software**

**www.BellHawk.com**

# User Manual for the DEX2 Interface to BellHawk

## Table of Contents

## 1. Introduction

The BellHawk DEX (Data Exchange) interface is now the standard interface to the BellHawk Work-in-Process Data Collection and Materials Tracking software. It replaces all prior interfaces, which directly communicated with the BellHawk database. These are now reserved for use of the BellHawk software development team.

The DEX interface comes in two primary versions:

- DEX2 - which is intended for installation on Windows Workstation based PCs (currently Windows 10 Pro or IIOT operating system) and is intended for use as a regular Windows program under user control for exchanging data between BellHawk running at a remote data center and a SQL Server Express database installed on the PC.
- MDEX - which is based on the MilramX automated data exchange platform and is intended for installation on a Windows Server computer. It is intended for unattended 24x7 operation under remote control through a web-browser interface.

The intention is that users can test out their interfaces or custom report generation using a Windows Workstation based PC and then transition to the use of MDEX for regular operational use, if needed. This is made possible by the use of identical DEX databases and DTOs (Data Transfer Objects) in both versions.

This User Manual is for the PC version of DEX. For details about how to install DEX2 on a PC, please see the DEX2 Installation Guide. Please also see other manuals for documentation of how to install and use MDEX

DEX only requires a standard external outbound Internet connection to communicate with the BellHawk website, such as you would use with any web-browser. It does not require any special "holes" in the organization's Internet firewall for inbound communications, as all inbound data is fetched from BellHawk in response to requests from DEX.

There can be multiple DEX and MDEX interfaces communicating with a single copy of BellHawk at the same time. This enables custom interfaces and custom reporting to be implemented in multiple manufacturing plants and warehouses in different geographic locations. Each copy of DEX requires its own BellHawk Device Access License (DAL) login.

These remote DEX and MDEX data exchanges can present a significant processing load to the BellHawk database. As such, mechanisms are in place to throttle the speed of data exchanges, to ensure that small amounts of data can be exchanged quickly but larger amounts of data are exchanged slowly but reliably.

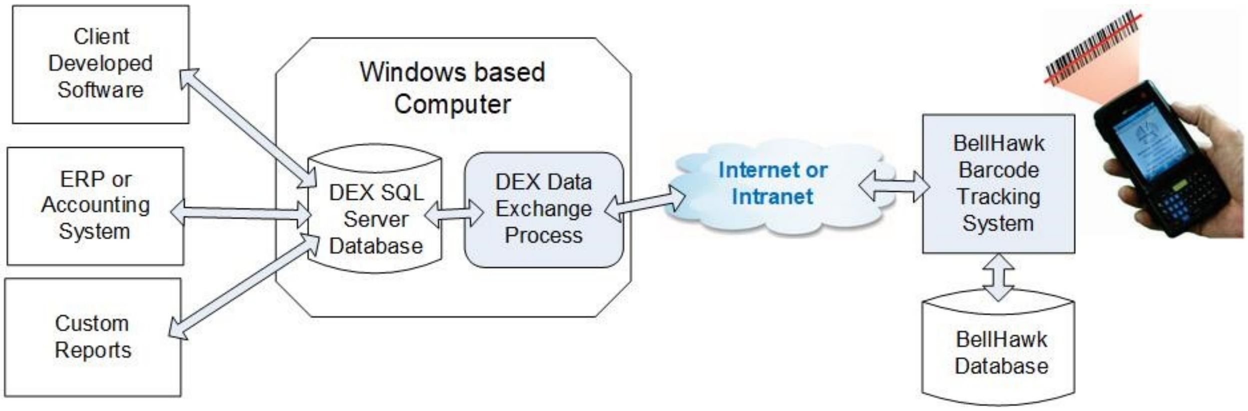
When BellHawk is being used as part of the BellHawk Online Service these throttling mechanisms are used to ensure that the use of DEX by any one user does not "hog" the processor to the detriment of other clients using a shared BellHawk Online server. As a result, users should not expect rapid transfers of large amounts of data, such as inventory snapshots, and should plan to do these overnight.

For IT departments installing BellHawk on their own servers, these throttling mechanisms can be used to tune the system so that web-service transfers do not overwhelm the use of the system for barcode data collection.

The source code of the DTOs is available, as part of a DEX developer kit, for use by .Net programmers who wish to modify the DTOs. This includes modifying the DTOs so they directly exchange data, using ODBC adaptors, with the databases of their ERP or accounting systems rather then indirectly through the DEX store and forward interface.

These customized DTOs can then be integrated into a custom MilramX interface between BellHawk and the client's ERP or accounting system.

## 2. DEX2 Overview



The DEX2 interface provides a simple to use interface for exchanging data with BellHawk. DEX consists of a SQL Server database and a DEX data exchange process.

Data written into tables in the DEX database are automatically transferred to the corresponding tables in the BellHawk database by the DEX data exchange process. Similarly, data entered into BellHawk is automatically transferred to corresponding tables in the DEX database.

The tables in the DEX database are structured in a well-documented tabular format (think Excel spreadsheet), with a set of self-contained records and no indirect references. This makes it easy for users to develop their own custom reports, using the contents of the DEX database. It also makes the development of interfaces to exchange data with ERP, accounting, CAD, and other systems very straightforward.

This is in contrast to the BellHawk database itself, which is designed for rapid transactional processing of barcode scanning data from a large number of mobile computers. This requires a complex database organization with many indirect references, which makes using the BellHawk database itself for reporting or data exchange interfaces much more complex than using the DEX interface.

DEX is designed to exchange data periodically (every 5 minutes or longer) with BellHawk, in the background, without interfering with the rapid response needed for efficient barcode scanning and operational user interaction.

Data transfers are done one record at a time, with error detection and correction. These record transfers are throttled to one every few seconds, so as not to interfere with operational use of BellHawk. As a result, large volumes of data can take a significant time to transfer.

The benefit of this throttling is that users of the DEX interface do not have to worry about interfering with their barcode data collection and operational use of BellHawk when transferring large volumes of data.

Because DEX transfers one data record at a time, this solves the problem of Internet connections timing out when attempting to transfer large volumes of data at the same time to or from BellHawk. Also, because of built-in record and character level error detection and correction, many of the problems of transferring data over inherently unreliable Internet connections are solved.

Users can start out using DEX2 to develop custom reports and/or interfaces and then transition to using MDEX for operational use. This is especially valuable when implementing shared reports using software such as SSRS (SQL Server Reporting Service) which require the DEX database to be continuously updated without human intervention. It is also essential when implementing automated data exchange interfaces with ERP, accounting, CAD and other systems.

MDEX can also be run on a Windows Server in the same data center as BellHawk along with the ERP or other system with which BellHawk will automatically exchange data. In this case MDEX will exchange data directly with the BellHawk database, which is more efficient that communication over the Internet.

## 3. Benefits of DEX

Running software such as BellHawk "in the Cloud" has many advantages, whether it is being run on computers at a secure data center managed by a third party, or is being run at a client's data center, or on a private Cloud based server.

The disadvantage is that this makes the BellHawk database inaccessible for implementing automated data exchange interfaces with systems such as ERP and accounting systems or for clients creating custom reports.

The DEX interface automatically exchanges data between the BellHawk database and the local DEX SQL Server database, which is formatted such as to be user friendly for interface and custom report creators.

This approach also overcomes a major disadvantage of directly using the BellHawk database, which is structured for rapid response to many users doing barcode scanning at the same time. This structure, while good for rapid data capture, is not very user friendly for interface or report generation. The DEX approach also avoids the possibility of users inadvertently damaging the BellHawk database and ending up with a non-working BellHawk system, as a result.

Also, the DEX interface is well documented whereas the structure of the BellHawk database is only documented in a manner usable by knowledgeable software developers.

Some uses of the DEX interface include:

1. Automatically exchanging data with an ERP or accounting system.

2. Generating custom reports using software such as Access, Excel, Crystal Reports, or SSRS based on data from the BellHawk database.
3. Exchanging data with EDI and shipping systems.
4. Transferring data from CAD or other engineering design systems into BellHawk.
5. Interfacing BellHawk with process control equipment and machines.
6. Generating large screen shop-floor displays showing performance dashboards.

There can be multiple copies of the DEX interface communicating with BellHawk from different plants, warehouses, and data centers. This enables BellHawk to be interfaced to a variety of systems and used by people doing custom reporting from multiple geographically separated locations.

One major advantage of the DEX interface is that it isolates and protects the BellHawk database from possible damage from reporting and data exchange software which would be possible if they directly interacted with the BellHawk database. This also helps ensure compliance with requirements such as CFR 21 Part 11, which require that users not be able to modify data once it is captured without an audit trail being present.

## 4. Data Exchange Overview

The data that can be exchanged consists of the following groups:

- Current status data. This includes the status of work orders and the contents of the containers table. These DEX tables are updated as changes occur in the BellHawk database.

- Data for transfer to ERP and accounting systems. These include aggregated shipments and receipts, as well as nightly inventory snapshots. These are periodically transferred from BellHawk at intervals set by the DEX user.

- Transaction history data. This DEX data includes containers of material received and picked, material moved, material consumed and produced on work orders, containers of material shipped, labor and machine time consumed against work orders, and the change in quality control status of containers of material. New records are added whenever a transaction is recorded in BellHawk.

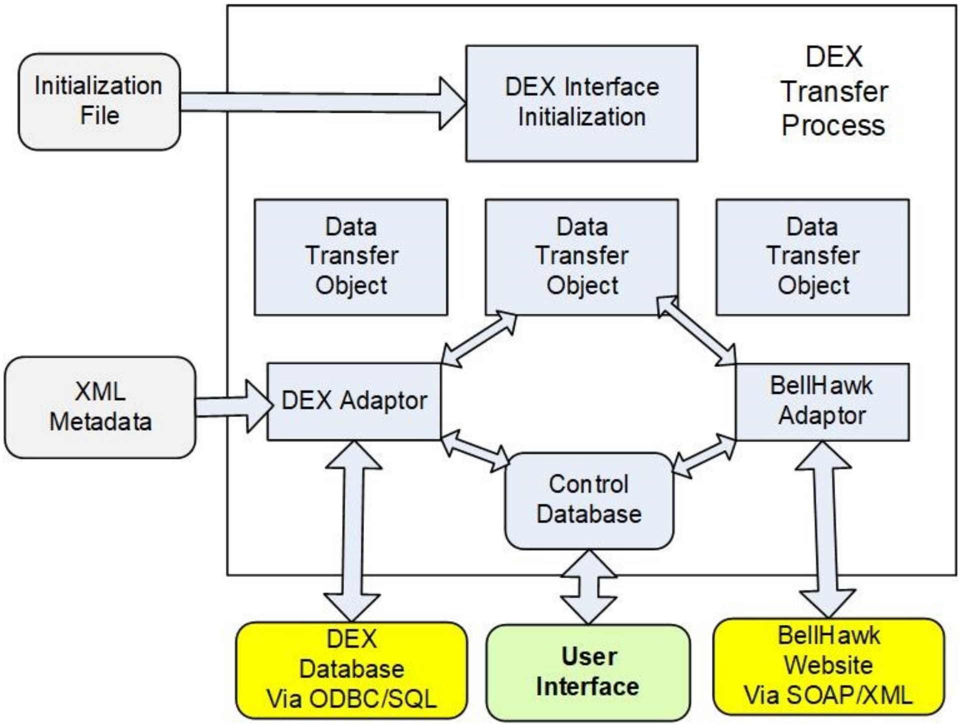
- Setup data for BellHawk - this is the same data as that which can be imported in HLDO (High Level Data Object) format through the Excel Setup data interface into BellHawk. The primary function of these transfers is to enable setup data to be transferred from ERP and accounting systems into BellHawk but this can also be a convenient way of manually transferring setup data into BellHawk.

- Setup Data from BellHawk. This includes the same set of data objects as can be transferred to BellHawk but they are transferred from BellHawk into a separate set of DEX tables, which are updated automatically as changes are made in BellHawk. These are useful for including in reports, especially when the setup data is imported directly into BellHawk or is automatically updated in BellHawk from another system.

- Orders for BellHawk. These are the same Purchase Orders, Pick Orders, Work Orders, and Ship Orders, which can be directly entered into BellHawk. They enable orders to be automatically sent to BellHawk directly from ERP and other systems.
- Orders from BellHawk. These are copies of the Purchase Orders, Pick Orders, Work Orders, and Ship Orders directly entered into BellHawk or imported into BellHawk from another system. These are written into a separate set of tables from the DEX tables used for sending orders to BellHawk and are automatically updated whenever changes are made in the BellHawk database. These are intended for reporting but are also useful for transferring order data to other systems, such as operational parameters for process control systems.

Please note that the data which can be exchanged depends on the BellHawk options in use.

## 5. DEX Internal Architecture



The DEX transfer process, which is embedded in the DEX2 program and is used by MDEX, contains a set of Data Transfer Objects (DTOs) which are essentially subroutines that exchange specific data objects, such as Work Orders, between tables in the DEX database and BellHawk (through its remote Web-Services interface). These DTOs are scheduled according to data setup in the Control Database through the User Interface.

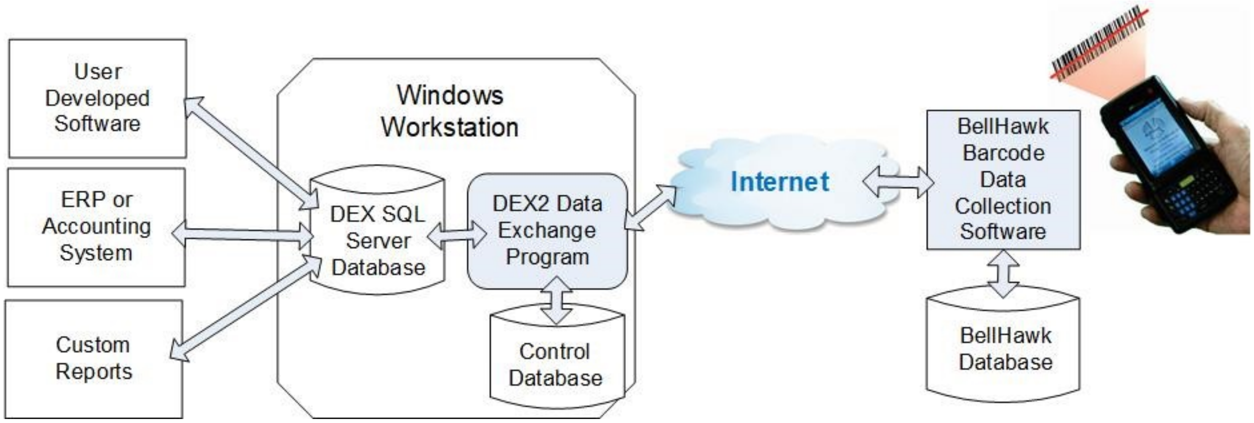
The User Interface for the DEX2 program shows the status of each transfer and any errors that may have occurred. This same data can be viewed using the MDEX web-browser user interface.

Connection information with BellHawk (through its URL) and login information for the DEX and Control SQL Server databases are setup in an initialization file, which is an editable text file, which is read by the transfer process when it is started running. The translation between the DEX database tables and HLDOs (High-Level-Data-Objects) used by BellHawk are read from XML metadata files, which are supplied with the DEX software.

In DEX2, the DTO scheduler is integrated within the DEX2.exe program. In MDEX, it is integrated within the Launcher, and the Transfer function only executes one DTO or DTO sequence. In either case, the scheduler runs DTOs or DTO sequences based on when they are scheduled to run. If there is more than one DTO which can be run then it chooses the most important DTO to run. If there are multiple DTOs with the same importance then it gives precedence to the DTO which has been waiting longest to run.

Parameters such as when or how often to run a DTO or sequence of DTOs, and their importance are set by users through the DEX2 user interface or the MDEX web-browser interface. They can also be set by external computer programs with direct access to the control database.

## 6. The DEX2 Program



DEX2 is a computer program which automates the exchange of data between the BellHawk database and the DEX2 database. It is designed so that this computer program, along with the DEX database, can be installed by users on their Windows PCs without needing assistance from their organization's IT staff.

DEX2 uses the free Microsoft SQL Server Express database server for its databases and only needs the normal outbound external Internet access granted to a PC. This is because all data is either pushed to BellHawk or pulled from BellHawk, thus avoiding the need for creating "holes" in the Internet firewall, which could be a security risk.

As described in a subsequent section of this handbook, users can use the user interface of DEX2 to control what data is exchanged with BellHawk and how often these transfers occur. This control information is stored in a Control Database. It is also where communication errors and other monitoring data are logged.

The DEX2.exe data exchange program is intended to be run as a regular computer program, under the control of a user, on a Windows PC or Workstation. This makes this version of the DEX interface easy to install and run for users with limited IT skills. But, while useful for exchanging data with BellHawk and extracting data for reporting, under the control of a user, this version is not intended for running unattended for long periods of time.

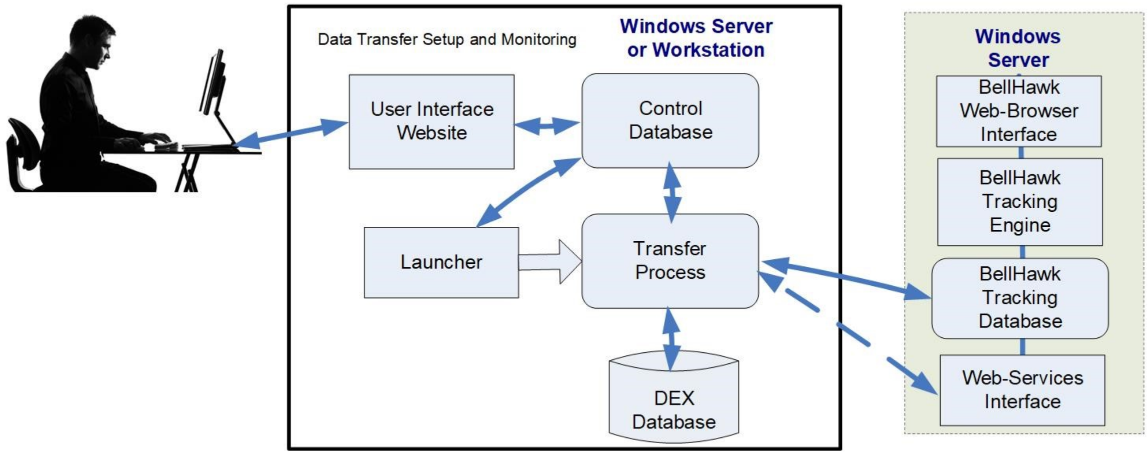
The reasons for this are:

1. All user interaction with respect to communications errors is handled through the DEX2 program user interface. DEX2 assumes that, if a communications error with either BellHawk or the DEX database occurs, or some other data transfer error occurs, then the user will fix the problem and restart the DEX2 program.
2. DEX2, like any Windows user program, will stop working when the PC on which its is running is rebooted, which can take place due to frequent automated Microsoft updates to the Windows operating system.
3. DEX2 will stop working or throw errors, if left running for multiple days, as it will run out of memory resources, just like any other Windows User Program. This is due to the design of the Windows Operating System which limits its ability to recover and reissue memory resources without restarting each executable program. This may require the user to reboot the PC on which DEX is left running, if the PC has run out of resources.
4. Communications over the Internet are inherently unreliable. A message sent over the Internet can be dropped by any one of the many routers in the path between the PC running DEX and the BellHawk server and back-again. If transmission fails, then DEX2 simply puts an entry in its log file and assumes that the DTO will be run again at a later time.

These issues preclude the use of the DEX2.exe interface program being used for long-running data exchange interfaces, such as automatically exchanging data with ERP or accounting systems which need to run unattended 24x7. They also preclude the use of DEX2.exe to automatically collect data in the DEX interface 24x7, which will then be used for report viewing by multiple users based on the use of SSRS (SQL Server Reporting Service) or some other reporting software.

## 7. Running DEX as a Service

If the DEX interface is needed to run unattended 24x7 for long periods of time then the DEX interface should be run as part of the MilramX software platform, as shown below. In this MDEX configuration, the DEX database and MilramX are installed on a Windows Server or Windows Workstation computer.



Instead of the DTOs (Data Transfer Objects), which perform the data transfer, being embedded in the DEX2.exe user program they are now embedded in a DEX Transfer Process. This runs as a Windows Service under the control of the MilramX Launcher program, which also runs as a Windows Service. As the Launcher is run as a Windows Service, it automatically restarts whenever the server is rebooted, unlike the DEX2 program which has to be manually restarted.

The Launcher runs the Transfer process to run individual DTOs and DTO sequences based on the contents of the Control database. This is the same Control Database which is used by the DEX2.exe program to control when, and in what order, it runs the DTOs.

The difference is that, instead of DEX2 running continuously, the transfer function only runs for the execution of a single DTO or DTO sequence, when the transfer process terminates itself, thus returning all its resources to the operating system.

This enables the DEX transfer function to run 24x7 without ever running out of memory resources or bringing the computer to its knees by hogging all the memory resources. Also, if there is an error in communications, which can cause both DEX2 and the DEX transfer function to hang, waiting on some communications resource to become available, the MilramX Launcher will kill the hung process, after a specified period of time, and then retry the data exchange later.

Because it is intended to run unattended, the MilramX version of DEX uses an IIS based web-server interface. This enables users, through the web-browser interface, to set how often each DTO is to be run and with what importance/priority. This interface also enables these users to monitor the transfers and to see if there have been any data transfer errors.

IT staff, who maintain the MilramX server can also be alerted by Email when communication problems occur, and can come in remotely over the Internet or their organizations private Intranet, examine the causes of the errors, and take appropriate corrective actions, without needing to physically come to the computer.

These features all enable the DEX interface to be run 24x7 for long periods of time, without human intervention. The MilramX software platform, however, does need an IT person, with significant knowledge and experience in Windows Server, to install this software. It is not something that can be done by a typical end user, whereas a DEX2 installation can be done themselves by a user who is reasonably "computer savvy".

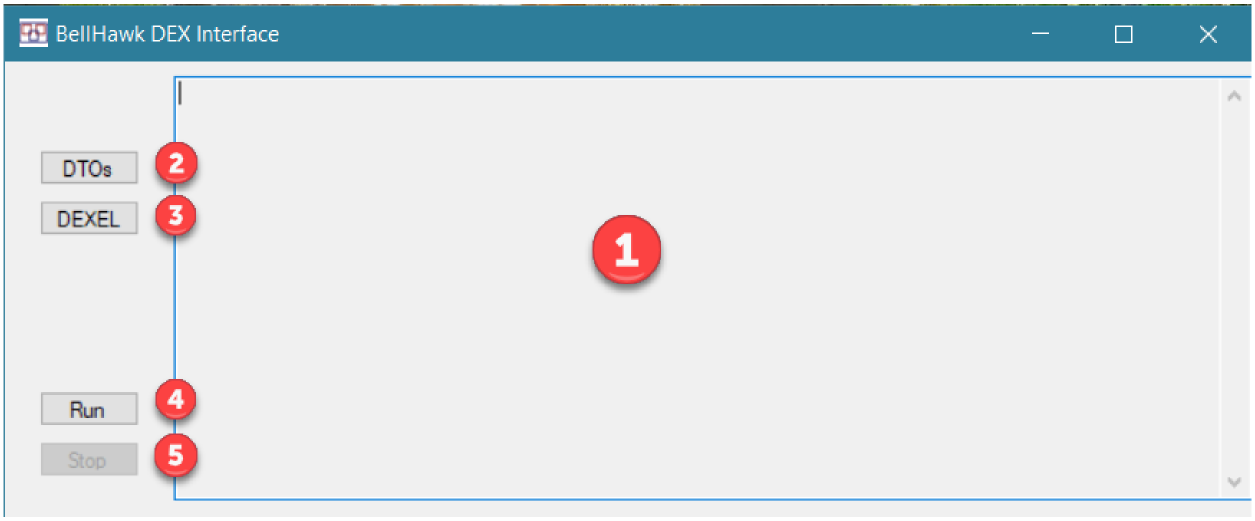
Because both the DEX2.exe computer program and the MDEX, MilramX based version of the DEX interface use an identical DEX database and Control database, clients can start out by using DEX2 to develop their own interfaces and reports. They can then transition to using the MilramX version of DEX for operational use without needing to change their reporting mechanism or interfaces.

# 8 Using the DEX2 User Interface

## *8.1 Running the DEX2 Program*

After DEX2 and the DEX and DEXControl SQL Server databases are installed and setup, as per the DEX2 Installation Guide, an icon will appear on the user's PC desktop, as shown here.

Selecting this icon brings up the DEX2 user interface main screen.



This screen has a scrolling text area (1) where DEX2 shows which DTOs are running and reports any warning messages or errors.

The [DTOs] button (2) leads to a screen which enables users to set up and schedule DTO transfers.

The [DEXEL] button (3) is used to setup DTO sequences, that is a set of DTOs which are running in a specific order to ensure that prerequisite data is transferred first.

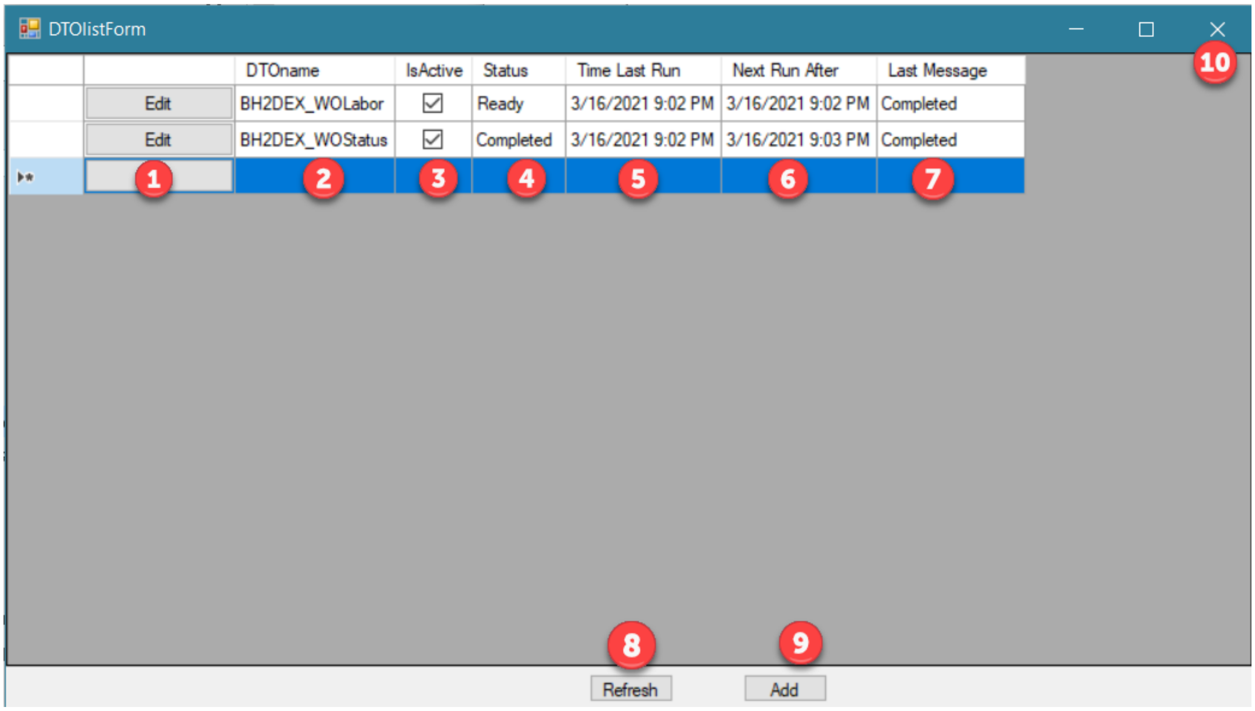
The [Run] button (4) is used to start the scheduling and execution of enabled and active DTOs and the [Stop] button (5) is used to stop their execution.

Note that the Stop button will not interrupt the execution of a DTO but will wait until the DTO has completed running before ceasing to schedule any more DTOs. The same applies to closing the DEX2 program using the [x] symbol at upper right of the screen, which will allow the running DTO to complete execution before exiting. Because of this both actions can take some time to complete (please be patient).

Note that this only applies to individual DTOs and DTOs within a DTO sequence and not to the whole sequence itself.

### *8.2 Selecting DTOs to Run*

When the [DTOs] button is selected, the following screen appears (in addition to the main DEX2 screen) which lists all the scheduled DTOs.



| | DTOname | IsActive | Status | Time Last Run | Next Run After | Last Message |
|---|---|---|---|---|---|---|
| Edit | BH2DEX_WOLabor | ☑ | Ready | 3/16/2021 9:02 PM | 3/16/2021 9:02 PM | Completed |
| Edit | BH2DEX_WOStatus | ☑ | Completed | 3/16/2021 9:02 PM | 3/16/2021 9:03 PM | Completed |

For each DTO in the schedule, which can include DTO sequences, this screen lists:

1) An Edit button, to set the scheduling parameters for the DTO

2) The DTO name. Transfers to BellHawk begin with DEX2BH and transfers from BellHawk begin with BH2DEX.

3) A flag showing whether the DTO is active, which means that it can be run by the scheduler. If not set to active this DTO will not be run. This can be turned on and off by clicking on the IsActive flag or on the Edit screen.

4) The Status shows as Ready to be run (waiting in the DEX DTO scheduler queue) , Running, or Completed. The Completed status changes to Ready when the DTO reaches its next scheduled time to be run.

5) The Date and Time the DTO was last run.

6) The Date and Time the DTO is scheduled to run next.

7) Any informational, warning, or error message output from the DTO last time it was run. In the case of an error or warning message, more details can be obtained from the daily log file generated by DEX for this purpose.

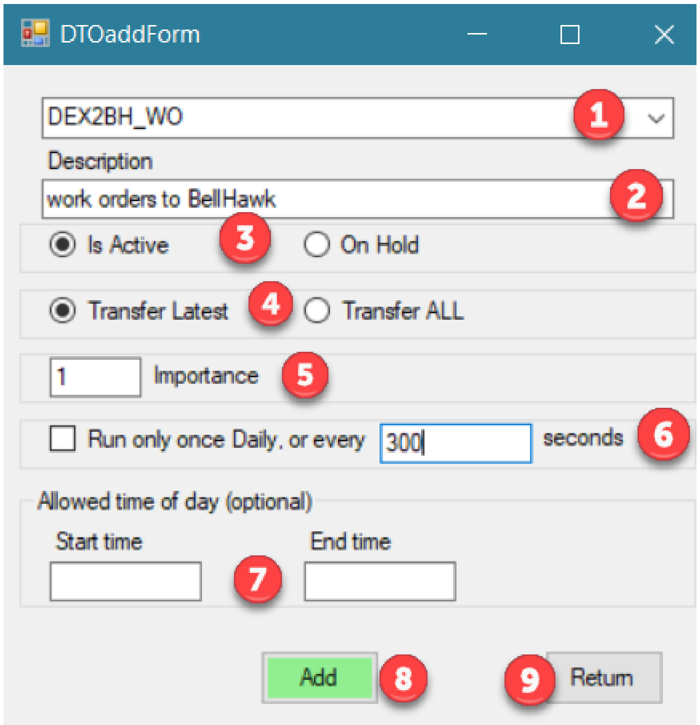
In addition, there is a [Refresh] button (8) to refresh the screen, which is useful to see latest times and messages, if the screen has been open for a while.

There is also an [Add] button (9) to add DTOs to the schedule. DEX is distributed with some 60 different DTOs not all of which will be required by any one user. When initially distributed there are no DTOs or DTO sequences in the schedule. Users can add the DTOs they need using this Add button.

This screen will remain open until it is closed using the [x] icon (10) at top right or the main DEX2 screen is closed.

### 8.3 Add Screen for DTOs

When the [Add] button is selected the screen shown at right appears. On this screen:



DTOaddForm

DEX2BH_WO

Description: work orders to BellHawk

● Is Active ○ On Hold

● Transfer Latest ○ Transfer ALL

1 Importance

□ Run only once Daily, or every 300 seconds

Allowed time of day (optional)

Start time

End time

1) There is a drop-down list of available DTOs and DTO sequences to add to the schedule from which a DTO can be selected. Note that only needed DTOs should be added, as each added DTO places an additional processing load on DEX and BellHawk.

2) Once selected the description of the selected DTO appears here.

3) The DTO can be marked as Active (ready to run) or On Hold, in which case it will not be scheduled to run.

4) Whether the DTO should transfer just the latest entries in the source database table (for operational use) or All the entries (useful for testing).

5) Sets the scheduling importance (1 to 100) of the DTO or DTO sequence A . This determines which DTO or DTO sequence to run when there is more than one DTO in the scheduling queue is available to be run at the same time. In this case, the DTO or DTO sequence with the highest importance value is given priority.

6) Whether to run the DTO just once a day or to run the DTO on a periodic basis. Note that the period has a minimum value of 5 minutes.

7) Sets the times of day within which the DTO can be run. This can be useful to limit large transfers, such as inventory snapshots from BellHawk to evening hours, when neither BellHawk nor the user's PC are in use for other purposes.

8) [Add] button used to add the selected DTO to the list of scheduled DTOs. This button returns back to the same screen, after adding the selected DTO to enable adding more DTOs to the schedule.

9) A [Return] button to close this screen and return to the last of DTOs in the schedule.

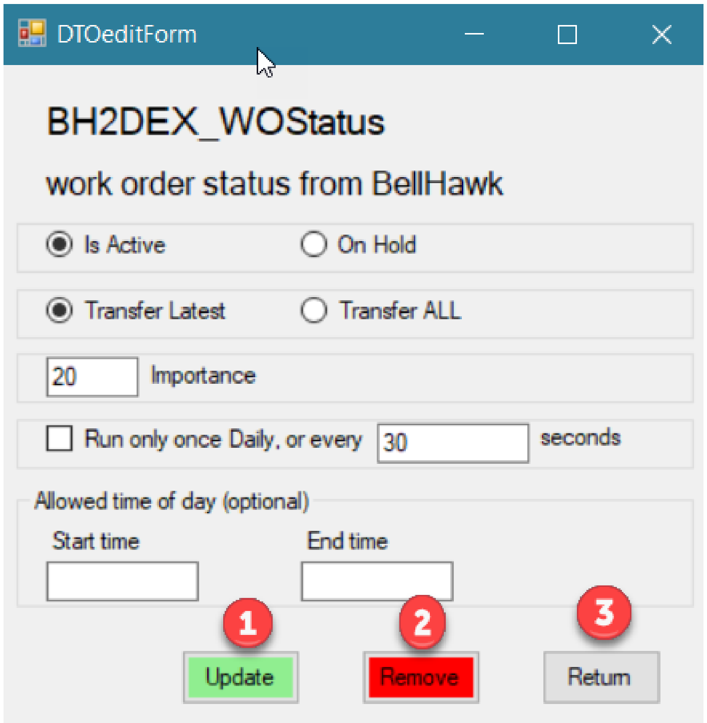
### 8.4 Edit Screen for DTOs

When the [Edit] button is selected for a DTO in the schedule, a similar screen to that used to Add DTOs to the schedule appears.

This enables the scheduling parameters for the DTO to be changed and then the scheduling parameters updated by selecting the [Update] button (1).

If a DTO is to be removed from the transfer scheduling queue then the DTO's [Edit] button is selected followed by the [Remove] button (2) which then returns to the list of DTOs in the schedule, with the designated DTO removed.

The [Return] button (3) simply closes this screen and returns to the list of DTOs in the schedule.



DTOeditForm

BH2DEX_WOStatus

work order status from BellHawk

● Is Active ○ On Hold

● Transfer Latest ○ Transfer ALL

20 Importance

□ Run only once Daily, or every 30 seconds

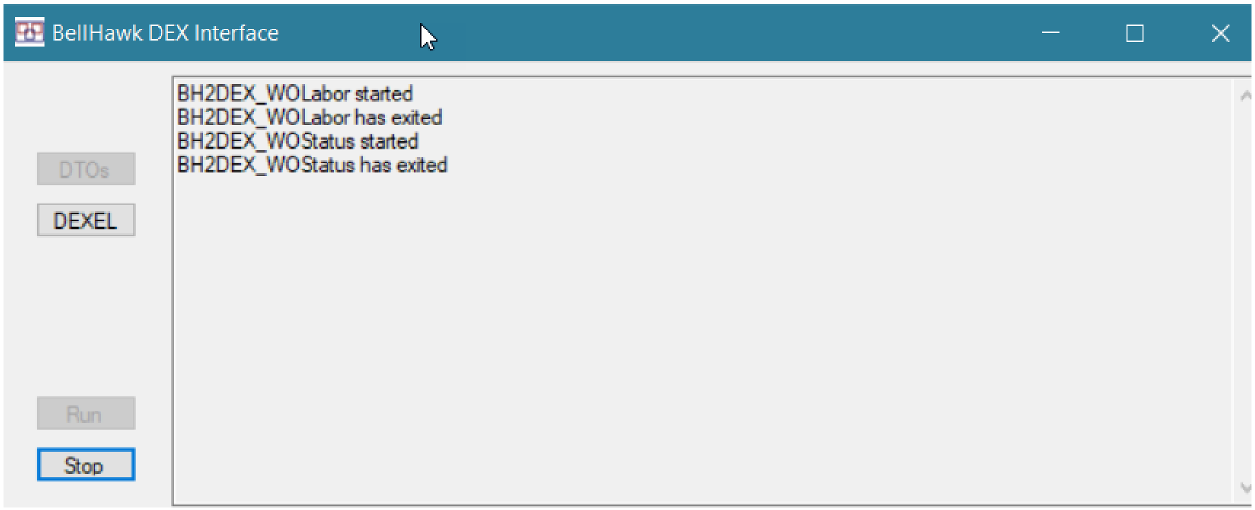
Allowed time of day (optional)

Start time

End time

### 8.5 Running the DTOs

When the [Run] button on the main DEX2 screen is selected the main panel shows which DTO is running along with any warning or error messages produced by the DTO.



BellHawk DEX Interface

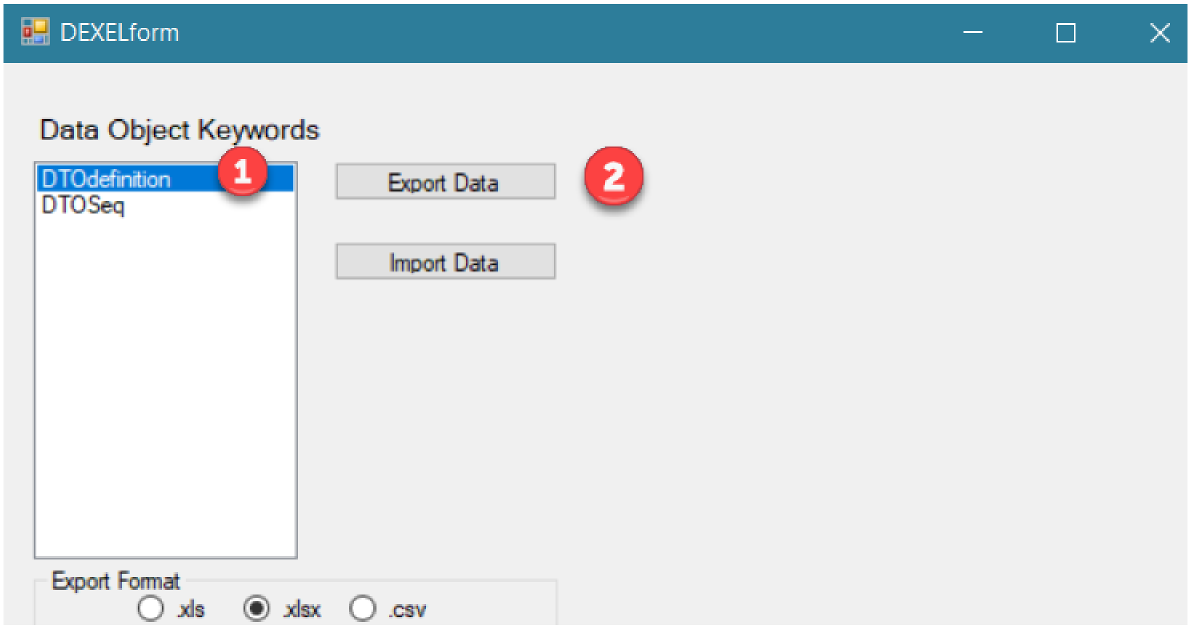
```
BH2DEX_WOLabor started
BH2DEX_WOLabor has exited
BH2DEX_WOStatus started
BH2DEX_WOStatus has exited
```

Execution can then be stopped using the [Stop] button.

### *8.6 Setting up DTO Sequences using DEXEL*

DTO sequences enable the user to set up sequences of DTO executions, so as to ensure that the BellHawk and DEX databases are updated in the correct sequence. This for example can be used to ensure that an Item definition is transferred before the Ship Order and Line which references the Item.

When the [DEXEL] button is selected on the main DEX2 screen, the following screen appears.



DEXELform

Data Object Keywords

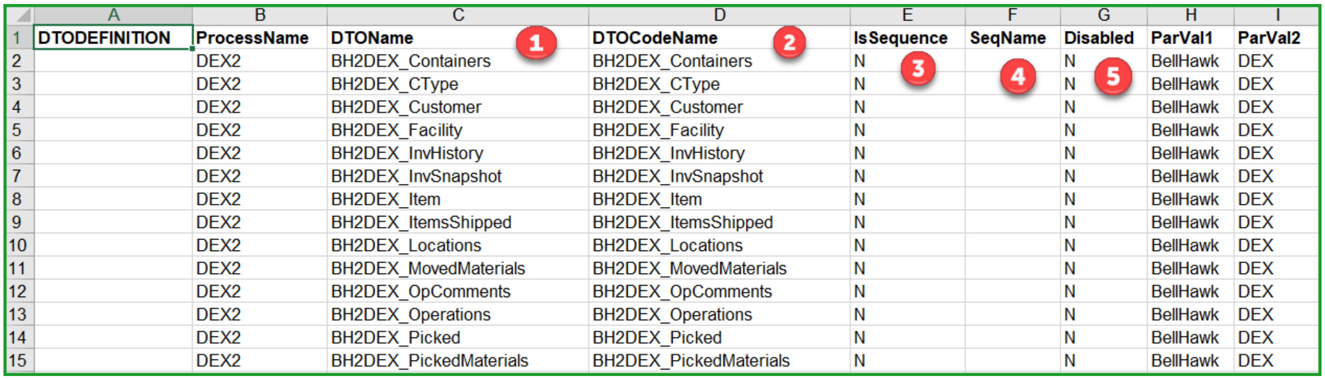
- DTOdefinition
- DTOSeq

Export Data

Import Data

Export Format ○ .xls ● .xlsx ○ .csv

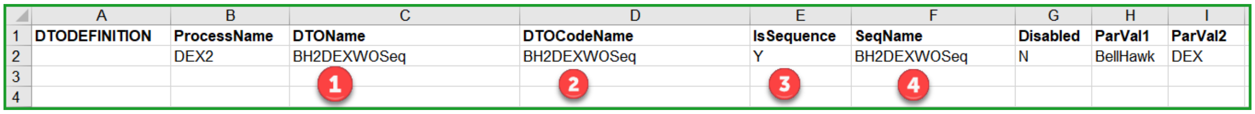
The first step to using sequences to select the DTOdefinition button (1) and selecting the Export Data button. The user will then be requested to select a directory and file name to receive the export, which will look like:



| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | DTODEFINITION | ProcessName | DTOName | DTOCodeName | IsSequence | SeqName | Disabled | ParVal1 | ParVal2 |
| 2 | | DEX2 | BH2DEX_Containers | BH2DEX_Containers | N | | N | BellHawk | DEX |
| 3 | | DEX2 | BH2DEX_CType | BH2DEX_CType | N | | N | BellHawk | DEX |
| 4 | | DEX2 | BH2DEX_Customer | BH2DEX_Customer | N | | N | BellHawk | DEX |
| 5 | | DEX2 | BH2DEX_Facility | BH2DEX_Facility | N | | N | BellHawk | DEX |
| 6 | | DEX2 | BH2DEX_InvHistory | BH2DEX_InvHistory | N | | N | BellHawk | DEX |
| 7 | | DEX2 | BH2DEX_InvSnapshot | BH2DEX_InvSnapshot | N | | N | BellHawk | DEX |
| 8 | | DEX2 | BH2DEX_Item | BH2DEX_Item | N | | N | BellHawk | DEX |
| 9 | | DEX2 | BH2DEX_ItemsShipped | BH2DEX_ItemsShipped | N | | N | BellHawk | DEX |
| 10 | | DEX2 | BH2DEX_Locations | BH2DEX_Locations | N | | N | BellHawk | DEX |
| 11 | | DEX2 | BH2DEX_MovedMaterials | BH2DEX_MovedMaterials | N | | N | BellHawk | DEX |
| 12 | | DEX2 | BH2DEX_OpComments | BH2DEX_OpComments | N | | N | BellHawk | DEX |
| 13 | | DEX2 | BH2DEX_Operations | BH2DEX_Operations | N | | N | BellHawk | DEX |
| 14 | | DEX2 | BH2DEX_Picked | BH2DEX_Picked | N | | N | BellHawk | DEX |
| 15 | | DEX2 | BH2DEX_PickedMaterials | BH2DEX_PickedMaterials | N | | N | BellHawk | DEX |

This file can be edited to add Sequence DTOs and reimported using the [Import Data] button.

To do this, delete all the rows except the header row and the first DTO record and then edit this record, as follows:



| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | DTODEFINITION | ProcessName | DTOName | DTOCodeName | IsSequence | SeqName | Disabled | ParVal1 | ParVal2 |
| 2 | | DEX2 | BH2DEXWOSeq | BH2DEXWOSeq | Y | BH2DEXWOSeq | N | BellHawk | DEX |
| 3 | | | | | | | | | |
| 4 | | | | | | | | | |

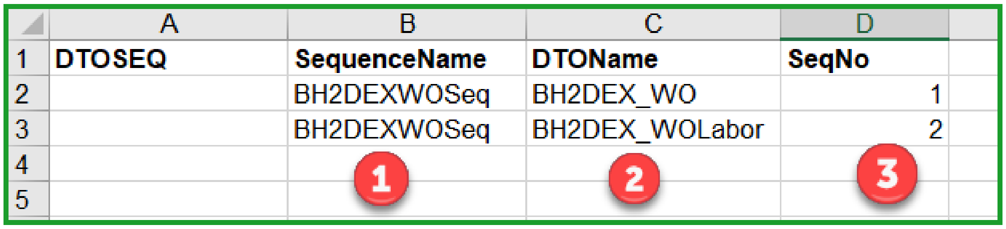
- Change the DTO name in column C (1) to a new DTOName
- Change the DTO code name in column (2) to be the same (it is ignored in the case of a sequence) - but must be present.
- Then set IsSequence (3) to Y
- Add the Sequence Name (4) in Column F.

The edited file can then be reimported by selecting the [Import Data] button on the DEXEL screen. Note that existing entries, identified by the Process Name and DTO Name will be overwritten and any new entries added.

Also, any DTOs that you want to be disabled, because the data is predefined, even if they appear in a DTO Sequence, can be reimported with Disabled (5) set to Y.

Any existing entries not in the import file will not be modified. To delete any unwanted existing entry put a D in column A for that record and reimport

Once the Sequence Name for the DTO Sequence has been setup, then the DTOs in the sequence can be defined by selecting the DTOsequence entry (1) on the DEXEL screen and selecting [Export Data] to export this setup file, which appears below, after editing:

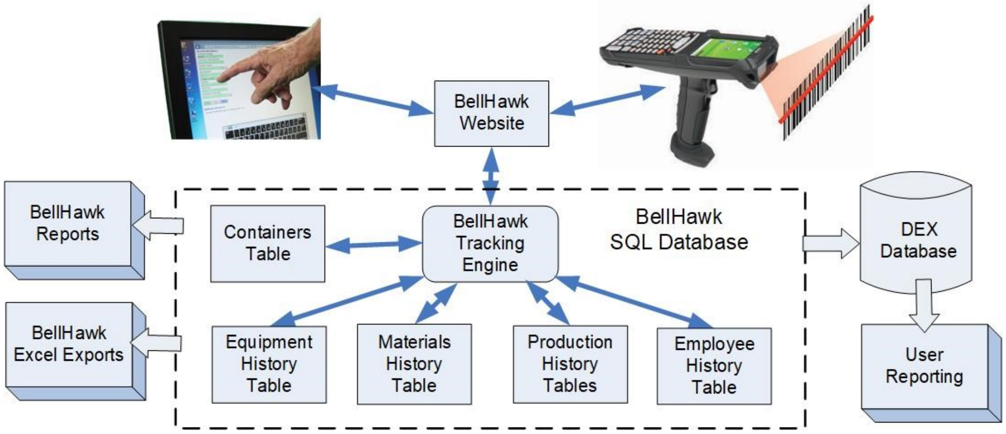


| | A | B | C | D |
|---|---|---|---|---|
| 1 | DTOSEQ | SequenceName | DTOName | SeqNo |
| 2 | | BH2DEXWOSeq | BH2DEX_WO | 1 |
| 3 | | BH2DEXWOSeq | BH2DEX_WOLabor | 2 |
| 4 | | | | |
| 5 | | | | |

In this file, the SequenceName (1) must be the same as the SeqName in the DTO Definition file. This is followed by the name of the DTO (2) to be run and the sequence in which the DTOs are to be run.

After this file is edited and reimported using the [Import Data] button on the DEXEL screen, then the sequence can be executed using the DTO Name given to the sequence, just like any other DTO.

## 9. Generating Custom Reports using DEX



BellHawk maintains the status of all active materials in its containers table. It also captures the transaction history in a set of tables relating to materials receipt, movement, usage, production and shipment as well as the equipment and labor times used in production operations.

From these tables BellHawk produces a set of standard reports that cover most standard operations and materials tracking requirements. Also "reports" such as barcoded receiving and picking sheets, as well as purchase, work, and ship orders can be customized in a limited way, such as by using the client's logo instead of the BellHawk logo, and by adding fields to the header and lines on these "reports".

These standard reports, along with related Excel exports, usually meet the requirements of most organizations for reporting real-time operational status. Users can then create their own reports using report generation software such as Access, Crystal Reports and Excel using the contents of the DEX database, which contains copies of the data in all of these BellHawk tables.

The PC based version of DEX makes it easy for users to run DEX2 to transfer the data they need from BellHawk into the DEX database. They can then run the reports they need by linking software such as Access, Excel and Crystal reports to the appropriate tables in the DEX database.

BellHawk has the capability to integrate external web-based reporting into its standard reports. To do this, users will create need to create their custom reports using SSRS (SQL Server Reporting Services), or equivalent, and install these under IIS on a Windows Server so that they are accessible through a remote URL.

These custom reports can then be integrated into BellHawk through setup data imported into BellHawk, using Excel files, by the BellHawk Systems Administrator. This same feature can be used to replace existing BellHawk reports but not Excel exports.

For details about integrating custom reports into BellHawk, please see the BellHawk Reports User Manual, which is available from www.BellHawk.com.

## 10. DEX Technology Notes

Users can link report generation programs, such as Crystal reports, to the DEX database in order to generate custom reports. More sophisticated users, can integrate this database into a reporting scheme using SSRS or a similar reporting software platform to generate web-accessible reports, which can then be integrated into the list of standard BellHawk reports.

Users of Excel can link Excel to the DEX tables and use these to generate Excel based reports and graphs, as well as import and export comma delimited files for manual data exchange with other systems.

Users of Microsoft Access can link Access tables to the DEX tables and be able to develop reporting and analysis programs in VBA as if DEX were a local Access database.

ERP and accounting systems which use SQL databases, can use stored procedures within the ERP system to write data to the DEX database and to read data from this database, whenever data is to be exchanged.

DEX comes with a full set of DTOs ready to use. Normally the setup and control of the DTO scheduling is done through the DEX2 program or the MDEX web interface. For programmers, such as those who wish to develop special code in .Net, the Control database is fully  accessible and can be used to schedule the transfers under control of their own program.

There is no need to setup any special "holes" in the network firewall of the plant or warehouse in which DEX is installed, as DEX uses a standard "outbound" Internet connection, just like any other PC on the network. This avoids a major security risk and a need to involve IT in the installation of DEX2.

There can be multiple copies of DEX, running at different geographic locations, interacting with BellHawk at the same time. This enables different departments, in different locations, to easily develop their own custom reports and Excel exports. It also enables integration with multiple external systems at different locations.

The overall processing load on the BellHawk database and server may, however, require a server with a dedicated processor for running BellHawk and its database, if the processing load becomes too high or the transfers are too slow.

For those situations where DEX is required to run reliably 24x7 then it is recommended that the Server version of DEX be used once all the transfers have been tested. The Server version consists of the same set of DTOs but implemented within the MilramX framework Here the same set of DTOs are used but they are integrated into a transfer function, which is run automatically under the control of the MilramX launcher.

Control of the server version of DEX is performed through a web-browser, instead of a desktop user interface This allows remote control and intervention by IT staff, should problems arise Also the server version can send alert Emails or Text Messages to IT staff when problems with data transfers occur.

## 11. Installing DEX

Installation and setup of the PC version of DEX is described in the DEX2 Installation Guide which is available on www.BellHawk.com.

Installation and setup of the MDEX Server version of DEX is described in the MilramX Installation Guide, which is available from www.BellHawk.com.

## 12. Field Data Types

Every field in each table has a specified field data type. These field types are used to check that a data value being set for a parameter is valid; for example, that a field type of DATE does indeed contain a string value that can be converted to a date without error. They are also used to check that the value of a parameter being retrieved from the database is valid and does not contain invalid characters for the data type.

Note that BellHawk only accepts extended ASCII characters (for reasons of real-time performance) and not Unicode or UTF8 characters. For user convenience DEX does contain a Character Translation table (tblChars) which can be used to translate from Unicode characters into a sequence of extended ASCII characters.

The valid field types are:

TEXTID – These are used to specify parameters that are lookup parameters and also text foreign key parameters whose value that must match key values in other tables. These can contain any ASCII character except non-printing (control) characters and the percent (%), comma (,) and double quote (“) characters. Unlike TEXT fields they cannot contain an empty string.

TEXT – Contains any ASCII characters except for non-printing (control) characters. An empty string "" is also a valid TEXT string.

INTEGER – Contains numeric digits. May be prefixed by + or – sign.

FLOAT – Can be prefixed with an optional + or – sign followed by one or more numeric digits. These may be followed by a decimal point and one or more numeric digits. This may be followed by an “e” or “E” for an exponent, followed by an optional + or – sign, followed by one or more numeric digits. An example is 34.79e-12

DECIMAL – May be prefixed by an optional + or – sign, followed by one or more numeric digits, followed optionally by a decimal point and one or more numeric digits. An example is +34.87.

DATE – must be in any valid date format for the system being used. Examples are 10/31/2009 and 2009-10-31.

DATETIME – must be in any valid time data format for the .Net system being used. An example is 10/29/2009 3:35PM.

MLTEXT – Multi-Line Text. Same rules as for text except these can contain embedded carriage return or new line characters.

YNBOOL – Must be one of the characters “Y” or “N”, and used to specify the yes/no binary data types found throughout BellHawk.

JSON - a set of parameters in JSON format - such as{"Color":"Blue","Size":"Large"}

Please note that the maximum length for each of these fields is specified in the HLDO definition for each parameter for the object in question, which can be exported in Excel format using the DEX web browser interface DEXEL capability.

## 13. Handshaking for Record Transfer

**Records sent to BellHawk**

Each table is for sending a specific type of data, such as Item or PO Header. When a new record is written or an existing record updated then the DateLastMod field is updated by a trigger on the database. This field is used by DEX to send just the latest updates to the table to BellHawk.

To aid in correctly ordering transfers, there is a TransferStatus on records to be written. This is set to "W" when the record is ready to be transferred (typically when the record is written into the DEX table) and is set by DEX to "T" if the transfer is completed successfully or "E" if not.

It is important to update previous records, rather than creating new records for the same data, when updating BellHawk. This is done by updating the existing record and setting the TransferStatus to "W" again, when DEX will attempt to transfer the record again.

Please see the separate section on Error handling for more details on this topic.

DEX tracks the last time it transferred data from each table in the tblParameterStore table in the Control database and compares this with the table

Setting the IsDeleted flag to 1 will cause the corresponding record in BellHawk to be marked as deleted. Please note that records are never deleted from the BellHawk database but are simply marked as deleted (and therefore invisible) for referential integrity.

Also note that changes to these records is a "one-way-street" and that changes to the corresponding data made directly in BellHawk does not update these records in the DEX database.

**Records Sent from BellHawk**

When BellHawk runs it collects historical transaction data in a set of history tables. DEX long-polls the BellHawk web-services interface looking for changes to any of these history tables. When an update is found, the history data is used to update the related tables in the DEX database.

When a new record is written into the DEX database by DEX, the DateLastMod field is set by a trigger on the database table to the time that the record was updated, so that a user's program can pick-up changes and transfer these to another program, such as an accounting or ERP system, or use the latest updates for reporting purposes.

Also, DEX sets the IsTransferred field on each record to "N". Optionally this can be updated by a user's program to "Y" when transferring the data to another application.

Inventory snapshots are the exception to the rule that transfers from BellHawk take place whenever there is an update to the corresponding table in BellHawk. These transfers occur whenever the Inventory Snapshot DTO is run (typically once a day at night) and overwrites the previous (night's) inventory snapshot data in the DEX database.

This inventory snapshot can be used to update the inventory in an ERP or accounting system, which will then keep a historical record of the inventory at the close of business on each day.

## 14. Tables for Transfer of Setup Data to BellHawk

### *Locations*

Table/DTO DEX2BH-Locations

1. LocationCode (TEXTID) - required  the BellHawk location code which is either the location barcode or, in the case of a Generic location, the lookup name.
2. LocationDescription (TEXT) – required name of location – appears on reports and some screens.
3. FacilityCode (TEXTID) - optional -must match a Facility Code (such as for Production, Warehouse, etc.) already setup in BellHawk, otherwise transfer will be ignored.
4. WorkCenterCode (TEXTID) - optional work center corresponding to location.
5. ERPCode (TEXTID) - optional - set to location code in ERP system that corresponds to the BellHawk Location Code. Inventory at location is reported back to ERP system in terms of ERPCode. Allows for translation from ERP location code to barcode on shelf or rack or generic name of BellHawk location. There can be multiple BellHawk locations for a single ERP code, if needed.
6. IsGeneric (YNBOOL) - required - if Y then will be displayed in the location selector list - note that there should only be a few locations that are Generic (such as Receiving Dock, Shipping Dock, etc.) otherwise location lookups will become very slow in BellHawk.
7. IsReceiving (YNBOOL) - required - if this is a Generic location, list this in the list of generic locations for receiving. Used to further restrict lookup list when receiving. Material handlers can still scan any location barcode to record put-away location.
8. IsWithdrawalLocation (YNBOOL) - if Y, any materials and non-reusable containers moved to this location are removed from inventory. Optional - defaults to N.
9. IsPicking (YNBOOL) - optional - if Y then materials can be picked from here, otherwise they will be excluded from locations considered as available for picking materials by Pick Orders.
10. IsMRB (YNBOOL) - Is this a material review board location?
11. IsQC (YNBOOL) - Is this a QC Quarantine location?
12. IsShipping (YNBOOL) Is this a shipping location?

13. IsSubcontractor (YNBOOL) Is this a subcontractor location?
14. SortCode (TEXTID) - optional - this is a way of grouping a set of locations into Zones, with the same sort code, and then organizing zones, in alphabetically ascending order, into a sequence so that material handlers can pick efficiently. Defaults to LocationCode.
15. ProjectCode (TEXTID) - optional - common stock materials moved to this location are designated for a specific project/job. Also materials purchased or made for a specific project/job cannot be moved to a location designated for a different project.
16. IsDeleted (YNBOOL) - to set IsDeleted flag on record in BellHawk
17. DateLastMod - (DATETIME) - set by DEX DTO whenever record is updated
18. TransferStatus (TEXTID) This is set to "W" when the record is ready to be transferred (typically when the record is written into the DEX table) and is set by DEX to "T" if the transfer is completed successfully or "E" if not.

### *Customer*

Table/DTO name DEX2BH-Customer

Fields:

1. CustomerNumber (TEXTID)– required unique alphanumeric identifier for the customer – used in drop-down lists
2. CustomerName (TEXT) – name of customer – used on reports and some screens
3. SeqNo (INTEGER) - sets order in drop-down lists
4. IsFacility (YNBOOL) set to Y if "customer" is really another company plant
5. IsCreditHold (YNBOOL) - is customer on credit hold - cannot ship orders
6. CreditLimit (FLOAT) - for warnings when creating new ship orders
7. BalanceDue (FLOAT) - for warnings when creating new ship orders
8. Notes (TEXT)
9. UDP (JSON)
10. DateLastMod (DATETIME)
11. TransferStatus (TEXTID)
12. IsDeleted (YNBOOL)
13. Note that CustomerNumber MYCOMPANY is reserved for the host company plant. This is so that products can be made for MYCOMPANY and inventory can belong to MYCOMPANY, as opposed to some other customer. An appropriate customer name can then be associated with MYCOMPANY.

### *Supplier*

Table/DTO Name: DEX2BH-Supplier

BellHawk prerequisites MTS or RTOPS + PO or WMS

Fields:

1. SupplierNumber (TEXTID) – required unique alphanumeric identifier for the supplier – unique identifier.
2. SupplierName (TEXT) – name of supplier – used on reports and some screens
3. SeqNo (INTEGER) - sets order in drop-down lists
4. IsFacility (YNBOOL) set to Y if "supplier" is really another company plant
5. DateLastMod (DATETIME)
6. TransferStatus (TEXTID)
7. IsDeleted (YNBOOL)

### *Project*

Specifies a Project/Job Code for composite reporting of all materials, POs, work orders, pick orders, and ship orders for a specific customer job.

Table/DTO Name: DEX2BH-Project

BellHawk Prerequisites (PROJECT)

Fields:

1. ProjectCode - (TEXTID) - required - typically used for overall project/job number
2. ProjectName - (TEXT) - project/job short description
3. ProjectDescription - (MLTEXT)
4. CustomerNumber (TEXTID) - required customer number field
5. DateWanted - (DATETIME) - optional
6. IsOpen - (YNBOOL) required - set to Y if open or N if the project is to be closed
7. IsReleased - (YNBOOL) required - set to Y if the project is released and N if to be transferred in unreleased status.
8. DateReleased - (DATETIME) - optional - when released if already released
9. DateLastMod (DATETIME)
10. TransferStatus (TEXTID)
11. IsDeleted (YNBOOL)

### *Item*

Table/DTO Name: DEX2BH-Item

BellHawk pre-requisites MTS, RTOPS

Fields:

1. ItemNumber (TEXTID) – required - the unique internal item number for the part or the route to make one or more parts. May also refer to a non-inventoried item such as waste or shipping charges. Item numbers may have an “>” symbol appended, followed by the revision number. This is the unique identifier for the item table.
2. ItemDescription (TEXT) – part description to appear on reports, barcode labels, and screens - not required but should be included.
3. UPC – UPC/GTIN number (TEXTID) – this optional barcode enables the UPC or GTIN (GS1 Global Trade Identification Number) barcode on an item to be scanned to identify the item when doing transactional data entry.

4. Category – item Category Code field (TEXTID) – such as RAW, INTERMEDIATE, FINISHED – defaults to DEFAULT to enable items with no specified category in an external system to be imported. Must match Category Code previously setup in BellHawk.

5. MaterialType – (TEXTID) optional field used as an additional sort field for reporting and item selection – examples include steel, acid, and packaging. Must match previously setup Material Code in BellHawk

6. UnitCost – (FLOAT) optional standard cost for item in primary UOM – used as default on PO line data entry. Also used as default value for activity-based costing calculations.

7. UnitPrice – (FLOAT) optional price per unit in primary UOM - used as default on ship order lines.

8. UOM – (TEXTID) required primary unit of measure nomenclature (such as ea, ins, gals) in which the quantity of this item in a container is tracked. Must match an item nomenclature previously setup in BellHawk

9. UOM2 (TEXTID) – optional secondary unit of measure nomenclature (such as lbs) in which the quantity of this item in a container is tracked.

10. UOMConversionFactor (FLOAT) – the floating point number you multiply quantities in the primary unit of measure by to get quantities in the secondary unit of measure. This is required if UOM2 is specified.

11. IsInventoried (YNBOOL) – defaults to Y for being an inventoried item. Set to N for items that are not tracked in inventory, such as shipping charges.

12. IsIndividual (YNBOOL) – this is set to Y for items that have their own “license-plate” tracking barcodes as opposed to having a quantity of materials in (or on) a container with a tracking barcode.

13. IsSerialized (YNBOOL) – this is set to Y to tell BellHawk to collect a serial number in addition to the tracking barcode on an individually barcoded item or container.

14. DefaultLocation (TEXTID) - this optional field is the default put-away location for an item. It is automatically filled in on a Move transaction but can be over-ridden.

15. LotControl (INTEGER) – required - every batch of materials entered into BellHawk through a transaction is given an internal lot number, which may span multiple containers of material. If LotControl = 0 then external lot numbers are not requested or displayed for these items. If LotControl = 1 then a lot number is automatically generated based on a prefix and a sequence number setup on a BellHawk System Administrators screen. If LotControl = 2 then the lot number is requested manually in all transactions. This setting is also used when the lot number is extracted along with the item number by scanning a composite GS1 standard barcode.

16. QCEvery (INTEGER) – required - used with the BellHawk Quality Control option. If QCEvery =0 then, when entering materials into inventory, mark them as automatically having passed inspection and do not check their QC status. This may be used for packing

materials and the like that are not subject to formal QC inspection. If QCEvery =1 then set every part with this item number as needing inspection before usage.

17. EstProdLife – (INTEGER) - required - estimated product life. If EstProdLife=0 do not request expiration date on entering materials into inventory. If > 0 then request expiration date when entering a new container of material into BellHawk.
18. MinInventoryQty. (FLOAT)This is the level used for inventory replenishment warnings on screens, reports and for generating alerts - optional- defaults to 0.
19. Comments – (TEXT) this is for notes about the item – which appear on the Item Master setup screen in BellHawk - optional.
20. ToBeChecked (YNBOOL) – required - this sets whether the import needs to be checked (and possibly modified) after import into BellHawk (y) or whether the record is ready for use (N).
21. DateLastMod (DATETIME)
22. TransferStatus (TEXTID)
23. IsDeleted (YNBOOL)

### *Work Centers*

Table/DTO Name: DEX2BH-WorkCenters

BellHawk prerequisites SPTS or RTOPS

Fields:

1. WcCode - required - Work Center Code (TEXTID) must be unique amongst all work centers
2. WorkCenterName - Work Center Name (TEXT) - optional
3. SeqNo - Order in Drop-Down List (INTEGER) - optional
4. DateLastMod (DATETIME)
5. TransferStatus (TEXTID)
6. IsDeleted (YNBOOL)

### *Operation*

Table/DTO Name: DEX2BH-Operation

BellHawk prerequisites SPTS or RTOPS + QC, if QC options are used

Fields:

1. OperationCode - required - Operation Code (TEXTID) - must be unique amongst all operations.

2. Description - Operation Description (TEXT) - optional
3. WcCode - Work Center Code (TEXTID) - Required - Work Center in which Operation Takes Place - must correspond to a WCCode for aa Work Center in BellHawk
4. IsRework - Optional Rework Operation (YNBOOL) - allows in parts that have failed QC
5. IsQC - optional - if QC Operation - allows in parts awaiting inspection (YNBOOL)
6. IsTAR - Is Test, Assembly or Repair Operation - required- (YNBOOL). If IsTAR = N then the primary materials recorded into the operation are consumed, that is, withdrawn from inventory. Also materials recorded out from the operation are entered as new materials into inventory. If IsTAR= Y then the materials with the primary item number being made on the work order are not consumed upon being recorded into the operation but are simply moved to the location where the operation takes place. Similarly on recording materials out from the operation the materials are simply moved from the operation location to the destination specified in the material-out transaction. IsTAR = Y generally applies to operations where non-destructive testing or repair takes place. It also applies to assemblies that move through a sequence of operations where parts are steadily added to the assembly at each operation.
7. LocationCode - Location code for Operation (TEXID) - required if IsTAR=Y.
8. DateLastMod (DATETIME)
9. TransferStatus (TEXTID)
10. IsDeleted (YNBOOL)

### *Container Type*

Table/DTO DEX2BH-CType

1. ContainerType (TEXTID) Unique code identifying Type of Container
2. ContainerDescription (TEXT) for use in reports
3. IsMultiUse (YNBOOL) - can it contain different materials
4. IsReusable (YNBOOL) - retain container after it becomes empty
5. IsSingleCustomer (YNBOOL) can only hold materials for a single customer
6. TareWeight (FLOAT) - average weight of container itself
7. TareUnitNotation (TEXTID) - UOM Notation for container weight - must be already defined in BellHawk.
8. SeqNo (INTEGER)- controls order of listing in drop-down list
9. IsDeleted (YNBOOL) to set IsDeleted flag on record in BellHawk database.
10. DateLastMod - (DATETIME) - set by DEX DTO whenever record is updated

11. TransferStatus (TEXTID) This is set to "W" when the record is ready to be transferred (typically when the record is written into the DEX table) and is set by DEX to "T" if the transfer is completed successfully or "E" if not

### *Shipper*

Used to send list of possible shippers, such as UPS or FedEx.

DTO/Table Name: DEX2BH-Shipper

Fields:

1. ShipperCode (TEXTID) - Unique Code Identifying Shipper
2. ShipperDescription (TEXT) - Longer Form for Reports
3. IsDeleted (YNBOOL) - to set IsDeleted flag on record in BellHawk
4. DateLastMod - (DATETIME) - set by DEX DTO whenever record is updated
5. TransferStatus (TEXTID) This is set to "W" when the record is ready to be transferred (typically when the record is written into the DEX table) and is set by DEX to "T" if the transfer is completed successfully or "E" if not.

## 15. Tables for Transfer of Orders to BellHawk

### *Purchase Order Header*

Table/DTO Name: DEX2BH-PO

BellHawk prerequisites MTS or RTOPS + PO or WMS

Fields:

1. PONumber (TEXTID) Number. Required.– Purchase Order Number - unique identifier
2. SupplierNumber (TEXTID) – supplier/vendor number for this PO. Required.
3. ProjectCode (TEXTID) - optional field for job/project code.
4. DateWanted – (DATE) Date wanted – Optional.
5. IsCustomerOwned - (YNBOOL) - Y= Customer Owned; N= Owned by MYCOMPANY required.
6. OwnedBy (TEXTID) – code of customer owning material - required - set to MYCOMPANY if not customer owned.
7. IsReleased - (YNBOOL) – Y=Yes PO is released for receiving materials; N= No – do not receive materials against this PO until it is released. Defaults to N.
8. IsClosed (YNBOOL) - used to set PO as closed if IsClosed =Y and open if IsClosed = N. Receipts cannot be recorded in BellHawk against closed POs, neither can changes to PO lines.

9. IsCancelled (YNBOOL) used to set PO as cancelled if IsCancelled = Y and open if IsCancelled = N. Receipts cannot be recorded in BellHawk against cancelled POs, neither can changes to PO lines.
10. DateLastMod (DATETIME)
11. TransferStatus (TEXTID)

### *Purchase Order Line*

Table DTO Name: BH2DEX-POL

BellHawk prerequisites MTS or RTOPS + PO or WMS

Fields:

1. PONumber - (TEXTID) Required – Matches Purchase Order Header Number.
2. LineNumber - (TEXTID) Required. PO Line Number such as 1,2, 3a, 3b, etc.
3. ItemNumber – (TEXTID) BellHawk Item Master Part Number. Required
4. ItemDescription - (TEXT) - Item Description from PO – Description of Item ordered.
5. Quantity (FLOAT)– Quantity ordered of BellHawk part number - required
6. UOM – (TEXTID) Units of measure notation (such as ea, lbs, ft etc.) for quantity of parts ordered. Required. Must match previously defined UOM nomenclature in BellHawk - required.
7. UnitCost – (FLOAT) Unit cost in specified UOM for PO Line.
8. DateWanted – (DATETIME) Date PO Line Item wanted. Optional.
9. IsReleased - (YNBOOL) – Y=Yes PO is released for receiving materials; N= No – do not receive materials against this PO until it is released in BellHawk. Defaults to N.
10. IsClosed (YNBOOL) - used to set PO line as closed if IsClosed =Y and open if IsClosed = N. Receipts cannot be recorded in BellHawk against closed PO lines, neither can changes to the PO line.
11. IsCancelled (YNBOOL) used to set PO line as cancelled if IsCancelled = Y and open if IsCancelled = N. Receipts cannot be recorded in BellHawk against cancelled PO lines, neither can changes to the PO line.
12. UDP (JSON)
13. DateLastMod (DATETIME)
14. TransferStatus (TEXTID)

Note that the PO Line entries are uniquely identified by the PONumber and LineNumber combination.

### Pick Order Header

DEX Pick Orders are used to pass pick orders from an ERP system to BellHawk. They consist of a Pick Order Header record and one or more Pick Order Line records. Please note that these are intended for picking "kits" of materials for production and not for picking sales orders.

Table/DTO Name: DEX2BH-PickOrderHeader

BellHawk prerequisites MTS or RTOPS + PICK or WMS

Fields:

1. PickOrderNumber (TEXTID) Required
2. PickOrderType (TEXTID) Required - "MT" = Move Ticket, "WO" = Pick for Work Order, "SO" = Pick for Ship Order
3. Order (TEXTID) Order Number - Optional - Work Order or Move Ticket Number - defaults to move ticket.
4. Description (TEXT) - Optional - Can be used for Work Order Step Code
5. ProjectCode (TEXTID) Project Code - Optional -but must correspond to a Project Code in BellHawk if specified.
6. FromFacility (TEXTID) - Optional - Must Correspond to a Facility in BellHawk if specified
7. ToFacility (TEXTID) - Optional - Must Correspond to a Facility in BellHawk if specified
8. ToLocation (TEXTID) - Optional - Destination Location Code for Pick Order
9. IsComplete (YNBOOL) - Required used to set Pick Order as complete in BellHawk if IsComplete = Y and open if IsComplete = N. Picks cannot be recorded in BellHawk against completed Pick Orders, neither can changes to Pick Order lines.
10. IsCancelled (YNBOOL) Required used to set Pick Order as cancelled in BellHawk if IsCancelled = Y and open if IsCancelled = N. Picks cannot be recorded in BellHawk against cancelled Pick Orders, neither can changes to Pick Order lines.
11. DateTimeWanted (DATETIME) - Optional  Date Time Wanted
12. DateLastMod (DATETIME)
13. TransferStatus (TEXTID)

### Pick Order Lines

Table/DTO Name: DEX2BH-PickOrderLines

BellHawk prerequisites MTS or RTOPS + PICK or WMS

Fields:

1. PickOrderNumber (TEXTID) required - belongs to this header record.
2. LineNumber (TEXTID) required uniquely identifies this specific record within Pick Order
3. ItemNumber (TEXTID) required -must match an Item Number in BellHawk
4. Quantity (FLOAT) - required - quantity to pick
5. UOM - required -must match an available UOM nomenclature for Item Master part.
6. IsComplete (YNBOOL) -required - used to set Pick Order line as complete in BellHawk if IsComplete = Y and open if IsComplete = N. Picks cannot be recorded in BellHawk against completed Pick Orders, neither can changes to Pick Order lines.
7. IsCancelled (YNBOOL) required - used to set Pick Order as cancelled in BellHawk if IsCancelled = Y and open if IsCancelled = N. Picks cannot be recorded in BellHawk against cancelled Pick Orders, neither can changes to Pick Order lines.
8. UDP (JSON)
9. DateLastMod (DATETIME)
10. TransferStatus (TEXTID)

Note that the Pick Order Line entries are uniquely identified by the PickOrderNumber and LineNumber combination.

### *Work Orders*

Table/DTO Name: DEX2BH-WO

BellHawk prerequisites SPTS or RTOPS

Fields:

1. WONumber - Work Order Number - (TEXTID) - required must be unique
2. TravelerBarcode - Appears as WO Barcode on Traveler (TEXTID) - optional - set to WONumber by default - if provided, must be unique amongst all traveler barcodes
3. WOType - Work Order Type W = Batch/Service; M=Processing (TEXTID) required
4. ItemNumber - Item Part Number to be made (TEXTID) - not required for Batch/Service work orders; required for Processing work orders.
5. Instructions - Work Order Instructions (MLTEXT) - optional
6. QtyToBeMade - Quantity to be made (FLOAT) in primary UOM for Item - not required for Batch/Service work orders; required for Processing work orders.
7. LotNumber - Lot Number for Parts to be Made( TEXTID) - optional - required if Item is lot controlled and lot number is manually specified. Not required for batch/service work orders.

8. DateWanted - Date Wanted (DATE) -required
9. Importance - WO Importance (TEXT) Standard, Rush etc. - optional - will be set to Standard importance by default - if supplied this must match a standard importance value previously setup in BellHawk
10. StartDate - Planned Start Date (DATE) - optional
11. IsReleased - Is WO Released (YNBOOL) set to N, if to be released through BellHawk, set to Y if already released.
12. IsCustomerOwned - Are Items made Customer Owned (YNBOOL) -Y/N -- not required for Batch/Service work orders; required for Processing work orders.
13. CustomerNumber - For Customer Number (TEXTID) - required - must match customer code in BellHawk
14. ProjectCode - Project Code (TEXTID) -optional - but, if provided, must match an open project in BellHawk.
15. IsCancelled (YNBOOL) - used to flag BellHawk that the Work Order is cancelled.
16. UDP (JSON)
17. DateLastMod (DATETIME)
18. TransferStatus (TEXTID) W/T/E

### *Work Order Route Step*

Table/DTO Name: DEX2BH-WOStep

BellHawk prerequisites SPTS or RTOPS

Fields:

1. TravelerBarcode - required - WO Traveler Barcode (TEXTID)
2. StepCode - required - Step Code (TEXTID) - must be unique within WO - used for step barcode on traveler.
3. StepNumber - required - Step Number (NUMID) - sets sequence of operation steps within WO
4. OperationCode - required - Operation Code (TEXTID)
5. Instructions - optional - Operation Instructions (MLTEXT)
6. UDP (JSON)
7. DateLastMod (DATETIME)
8. TransferStatus (TEXTID)

9. IsDeleted (YNBOOL)

Note that the Work Order Route Step entries are uniquely identified by the TravelerBarcode and StepCode combination.

### *Work Order Step Part In*

Table/DTO Name: DEX2BH-WOStepPartIn

BellHawk prerequisites RTOPS

Fields:

1. TravelerBarcode - required WO Traveler Barcode (TEXTID)
2. StepCode - Step Code (TEXTID) - required -must be unique within WO
3. ResCode - Resource Code (TEXTID) -required - must be unique within WO Step
4. SeqNo - required - Sequence number in which parts appear (INTEGER)
5. ItemNumber - required Item Number for Part In (TEXTID)
6. Quantity - required Part-In Quantity (FLOAT) in primary UOM for Part-In
7. IsWIP - Is Part-In WIP materials? (YNBOOL) required
8. DateLastMod (DATETIME)
9. TransferStatus (TEXTID)
10. IsDeleted (YNBOOL)

Note that the Work Order Step Part In entries are uniquely identified by the TravelerBarcode, StepCode, ResCode combination.

### *Work Order Step Labor*

Optional - Used to report on estimated versus actual labor needed for WO.

BellHawk prerequisites SPTS or RTOPS

Table/DTO Name: DEX2BH-WOStepLabor

Fields:

1. TravelerBarcode - WO Traveler Barcode (TEXTID) -required
2. StepCode - Step Code (TEXTID) - must be unique within WO - required
3. ResCode - Resource Code for each Labor Item (TEXTID) must be unique within WO Step - required
4. SeqNo - Sequence number in which labor resources appear (INTEGER) required

5. LaborClassCode - Labor Class Code (TEXTID) must match setup in BellHawk - required
6. Hours - Estimated time to perform operation (FLOAT) required
7. DateLastMod (DATETIME)
8. TransferStatus (TEXTID)
11. IsDeleted (YNBOOL)

Note that the Work Order Step Labor entries are uniquely identified by the TravelerBarcode, StepCode, ResCode combination.

### *Work Order Step Part Out*

Table/DTO Name: DEX2BH-WOStepPartOut

BellHawk prerequisites RTOPS

Fields:

1. TravelerBarcode - WO Traveler Barcode (TEXTID) required
2. StepCode - Step Code (TEXTID) - must be unique within WO - required
3. ResCode - Resource Code (TEXTID) to identify Part Out - must be unique within WO Step - required
4. SeqNo - Sequence number in which parts out appear (INTEGER) - required
5. ItemNumber - Item Number for Part in (TEXTID) - required
6. Quantity - Expected Part Out Quantity (FLOAT)- required
7. UOM - Default Unit of Measure nomenclature for Part Out (TEXTID) -required
8. IsWIP - Is Part Out WIP materials? (YNBOOL) - required
9. DateLastMod (DATETIME)
10. TransferStatus (TEXTID)
12. IsDeleted (YNBOOL)

Note that the Work Order Step Part Out entries are uniquely identified by the TravelerBarcode, StepCode, ResCode combination.

### *Ship Orders Headers*

DEX Ship Orders (SOs) are used to pass ship orders from the ERP or sales order system to BellHawk. They consist of a Ship Order Header record and one or more Ship Order Line records. Please note that while ship orders may be the same as sales orders, in an ERP, sales order, or accounting system, they may also correspond to one shipment or release under a "blanket" purchase order or contract.

Table Name: DEX2BH-SO

BellHawk prerequisites MTS or RTOPS + SO

Fields:

1. ShipOrderNumber - Ship Order Number (TEXTID) - Required - must be unique
2. CustomerNumber - Customer Number (TEXTID) - Required
3. DateWanted - Date Wanted (DATE) Optional
4. ShipDate - Date to be Shipped (DATE) Optional
5. SODescription - optional Ship Order Notes (MLTEXT) 100 characters max
6. BillToStreet1 Bill To Street1(TEXT) - Note that the BillTo information is optional
7. BillToStreet2 Bill To Street2 (TEXT)
8. BillToCity - Bill To City (TEXT)
9. BillToState - Bill To State (TEXT)
10. BillToZip - Bill To Postal Code (TEXT)
11. BillToCountry- Bill To Country (TEXT)
12. ShipToStreet1 - Ship To Street1 (TEXT) Note that the Ship To information is Optional
13. ShipToStreet2 - Ship To Street2 (TEXT)
14. ShipToCity - Ship To City (TEXT)
15. ShipToState - Ship To State (TEXT)
16. ShipToZip - Ship To Postal Code (TEXT)
17. ShipToCountry - Ship To Country (TEXT)
18. ImportanceCode - Priority Codename - (TEXTID) Optional default to Standard
19. IsReleased - Set to Y for immediate release - (YNBool) Optional - defaults to N
20. Facility - Facility to Ship From (TEXTID) - optional
21. ContractNumber - Contract Number (TEXTID) - optional
22. ReleaseNumber - Release Number Under Contract - (TEXTID) - optional
23. ShipperCode - such as UPS (TEXTID) - optional - must match Shipper Code in BellHawk
24. ShipperDescription - Shipper Description (TEXT) optional - 50 characters max
25. FOB - Free on Board Statement - (TEXT) optional - 50 characters max

26. Terms - Terms Statement - (TEXT) optional - 50 characters max
27. SalesPerson - Sales Person Name- (TEXT) optional - 50 characters max
28. IsComplete - Set to Y to Close - (YNBOOL) defaults to N
29. IsCancelled - Set to Y to Cancel - (YNBOOL) defaults to N
30. dtCompleted - Date Completed (DATE) required if IsComplete = Y
31. dtCancelled - Date Cancelled (DATE) required if IsCancelled = Y
32. ShipFromStreet1 - ShipFromStreet1 (TEXT) - Note Ship From fields are optional
33. ShipFromStreet2 - ShipFromStreet2 (TEXT)
34. ShipFromCity - ShipFromCity (TEXT)
35. ShipFromState - ShipFromState (TEXT)
36. ShipFromZip - ShipFromZip (TEXT)
37. ShipFromCountry- ShipFromCountry (TEXT)
38. CustOrderNumber - optional Customer Order Number (TEXT)
39. CustOrderDate - optional Customer Order Date (DATE)
40. ProjectCode - optional Project Code for Order (TEXTID)
41. UDP (JSON)
42. DateLastMod (DATETIME)
43. TransferStatus (TEXTID)

Note that while many fields are marked as optional they may be required for other purposes, such as generating Packing Lists or Bills of Lading or for generating Barcode Labels.

### *Ship Order Lines*

Table Name: DEX2BH-SOL

BellHawk prerequisites MTS or RTOPS + SO

Field Names:

1. ShipOrderNumber - Ship Order Number (TEXTID) Required - must match header record
2. ShipLineNumber - Line Number (TEXTID) Required - must be unique within Ship Order
3. SeqNo - Sequence Number (Integer) - required - to appear within Ship Order
4. ImportanceCode - Priority (TEXTID) - optional defaults to Standard
5. ItemNumber - Item Number (TEXTID) - required

6. ItemDescription - Item Description (MLTEXT) - optional max 255 characters
7. UOM - Order Line UOM - (TEXTID) - required - must match an available UOM for Item Nomenclature in BellHawk
8. Quantity - Order Quantity (FLOAT) - required
9. UnitPrice - Order Unit Price (DECIMAL) - required
10. IsComplete - Set to Y to Close - (YNBOOL) defaults to N
11. IsCancelled - Set to Y to Cancel - (YNBOOL) defaults to N
12. dtCompleted - Date Completed (DATE) required if IsComplete = Y
13. dtCancelled - Date Cancelled (DATE) required if IsCancelled = Y
14. IsReleased - Is Released (YNBOOL) defaults to N
15. IsMakeToOrder - to be made to order (YNBOOL) defaults to N
16. DateWanted- Date Wanted (DATE) - defaults to overall SO Date Wanted
17. ProjectCode - optional Project Code (TEXTID)
18. UDP(JSON)
19. DateLastMod (DATETIME)
20. TransferStatus (TEXTID)

## 16. Tables for Transfer of Setup Data from BellHawk

Note that each of these has an ID and an IsDeleted field, which indicates whether the data object instances are active or not.

If a setup object instance is marked as deleted in BellHawk then it cannot be reactivated but a new record may be created with the same Work Center Code, Location Code, Item Number etc., lookup code.

This may result in multiple entries with the same lookup code in the BellHawk database tables, but at most, only one will have IsDeleted = N and the rest will have IsDeleted= N. In such a case then all the entries will be transferred to the corresponding DEX table but only one will have IsDeleted = N.

Active instances of setup objects in BellHawk may be edited, which may result in the instances in DEX also being updated. To facilitate only updating the latest instance of an object, with IsDeleted=N, as opposed to other instances with the same lookup code but with IsDeleted=Y, the ID number of each object instance in its respective table is also exported to DEX.

### *Work Centers*

BellHawk Prerequisite SPTS or RTOPS

Table/DTO BH2DEX-WorkCenters

Fields

1. ID (INTEGER) - Unique identifier
2. WcCode (TEXTID)
3. WorkCenterName (TEXT)
4. IsDeleted (YNBOOL)
5. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
6. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Operations*

BellHawk Prerequisite SPTS or RTOPS

Table/DTO BH2DEX-Operations

Fields

1. ID (INTEGER) Unique identifier
2. OperationCode (TEXTID)
3. WcCode (TEXTID)
4. Description (TEXT)
5. IsTAR (YNBOOL) - is this a test, assembly, or repair operation
6. IsRework (YNBOOL)
7. IsQC (YNBOOL)
8. LocationCode (TextID)
9. UDP (JSON)
10. IsDeleted (YNBOOL)
11. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
12. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Facility*

Table/DTO Name DEX2BH-Facility

1. ID (INTEGER) - Unique identifier

2. FacilityCode (TEXTID)
3. FacilityDescription (TEXT)
4. SeqNo (INTEGER) - sets order in drop-down lists
5. IsDeleted (YNBOOL) -sets record in BellHawk as deleted
6. DateLastMod (DATETIME)
7. TransferStatus (TEXTID)

### *Locations*

Table/DTO BH2DEX-Locations

1. ID (INTEGER) - Unique identifier
2. LocationCode (TEXTID) the BellHawk location code which is either the location barcode or, in the case of a Generic location, the lookup name.
3. LocationDescription (TEXT) –name of location – appears on reports and some screens.
4. FacilityCode (TEXTID) - Facility Code
5. WcCode (TEXTID) - optional work center corresponding to location.
6. ERPCode (TEXTID) - set to location code in ERP system that corresponds to the BellHawk Location Code. Inventory at location is reported back to ERP system in terms of ERPCode. Allows for translation from ERP location code to barcode on shelf or rack or generic name of BellHawk location. There can be multiple BellHawk locations for a single ERP code, if needed.
7. IsGeneric (YNBOOL) - required - if Y then will be displayed in the location selector list - note that there should only be a few locations that are Generic (such as Receiving Dock, Shipping Dock, etc.) otherwise location lookups will become very slow in BellHawk.
8. IsReceiving (YNBOOL) - required - if this is a Generic location, list this in the list of generic locations for receiving. Used to further restrict lookup list when receiving. Material handlers can still scan any location barcode to record put-away location.
9. ERPCode (TEXTID) - optional - set to location code in ERP system that corresponds to the BellHawk Location Code. Inventory at location is reported back to ERP system in terms of ERPCode. Allows for translation from ERP location code to barcode on shelf or rack or generic name of BellHawk location. There can be multiple BellHawk locations for a single ERP code, if needed.
10. IsWithdrawalLocation (YNBOOL) - if Y, any materials and non-reusable containers moved to this location are removed from inventory. Optional - defaults to N.
11. IsPicking (YNBOOL) - optional - if Y then materials can be picked from here, otherwise they will be excluded from locations considered as available for picking materials by Pick Orders.

12. IsMRB (YNBOOL) - Is this a material review board location?
13. IsQC (YNBOOL) - Is this a QC Quarantine location?
14. IsShipping (YNBOOL) Is this a shipping location?
15. IsSubcontractor (YNBOOL) Is this a subcontractor location?
16. SortCode (TEXTID) - optional - this is a way of grouping a set of locations into Zones, with the same sort code, and then organizing zones, in alphabetically ascending order, into a sequence so that material handlers can pick efficiently. Defaults to LocationCode.
17. ProjectCode (TEXTID) - optional - common stock materials moved to this location are designated for a specific project/job. Also materials purchased or made for a specific project/job cannot be moved to a location designated for a different project.
18. UDP (JSON)
19. IsDeleted (YNBOOL) - from BellHawk database
20. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
21. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### Container Type

Table/DTO BH2DEX-CType

1. ID (INTEGER) - Unique identifier
2. ContainerType (TEXTID) Unique code identifying Type of Container
3. ContainerDescription (TEXT) for use in reports
4. IsMultiUse (YNBOOL) - can it contain different materials
5. IsReusable (YNBOOL) - retain container after it becomes empty
6. IsSingleCustomer (YNBOOL) can only hold materials for a single customer
7. TareWeight (FLOAT) - average weight of container itself
8. TareUnitNotation (TEXTID) - UOM Notation for container weight - must be already defined in BellHawk.
9. UDP (JSON);
10. SeqNo (INTEGER)- controls order of listing in drop-down list
11. DateLastMod - (DATETIME) - set by DEX DTO whenever record is updated
12. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### Customer

Table/DTO Name: BH2DEX-Customer

Fields:

1. ID (INTEGER) - Unique identifier
2. CustomerNumber (TEXTID)– unique alphanumeric identifier for the customer – used in drop-down lists
3. CustomerName (TEXT) – name of customer – used on reports and some screens
4. IsPlant (YNBOOL) set to Y if "customer" is really another company plant
5. IsCreditHold (YNBOOL) - is customer on credit hold - cannot ship orders
6. CreditLimit (FLOAT)  - for warnings when creating new ship orders
7. BalanceDue (FLOAT) - for warnings when creating new ship orders
8. Notes (TEXT)
9. UDP (JSON)
10. IsDeleted (YNBOOL)

11. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
12. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Supplier*

Table/Supplier Name: BH2DEX-Supplier

Fields:

1. ID (INTEGER) - Unique identifier
2. SupplierNumber (TEXTID) – required unique alphanumeric identifier for the supplier – unique identifier.
3. SupplierName (TEXT) – name of supplier – used on reports and some screens
4. IsPlant (YNBOOL) set to Y if "supplier" is really another company plant
5. IsDeleted (YNBOOL)
6. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
7. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Project*

Table/DTO Name: BH2DEX-Project

Fields:

12. ProjectCode - (TEXTID) - required - typically used for overall project/job number
13. ProjectName - (TEXT) - project/job short description
14. ProjectDescription - (MLTEXT)
15. CustomerNumber (TEXTID) - required customer number field
16. DateWanted - (DATETIME) - optional
17. IsOpen - (YNBOOL) required - set to Y if open or N if the project is to be closed
18. IsReleased - (YNBOOL) required - set to Y if the project is released and N if to be transferred in unreleased status.
19. DateReleased - (DATETIME) - optional - when released if already released
20. IsDeleted (YNBOOL)
21. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
22. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Item*

Table/DTO Name: BH2DEX-Item

Fields:

1. ID (INTEGER) - Unique identifier
2. ItemNumber (TEXTID) – required - the unique internal item number for the part or the route to make one or more parts. May also refer to a non-inventoried item such as waste or shipping charges. Item numbers may have an ">" symbol appended, followed by the revision number. This is the unique identifier for the item table.
3. ItemDescription (TEXT) – part description to appear on reports, barcode labels, and screens - not required but should be included.
4. UPC – UPC/GTIN number (TEXTID) – this optional barcode enables the UPC or GTIN (GS1 Global Trade Identification Number) barcode on an item to be scanned to identify the item when doing transactional data entry.
5. Category – item Category Code field (TEXTID) – such as RAW, INTERMEDIATE, FINISHED – defaults to DEFAULT to enable items with no specified category in an external system to be imported. Must match Category Code previously setup in BellHawk.
6. MaterialType – (TEXTID) optional field used as an additional sort field for reporting and item selection – examples include steel, acid, and packaging. Must match previously setup Material Code in BellHawk
7. UnitCost – (FLOAT) optional standard cost for item in primary UOM – used as default on PO line data entry. Also used as default value for activity-based costing calculations.
8. UnitPrice – (FLOAT) optional price per unit in primary UOM - used as default on ship order lines.
9. UOM – (TEXTID) required primary unit of measure nomenclature (such as ea, ins, gals) in which the quantity of this item in a container is tracked. Must match an item nomenclature previously setup in BellHawk
10. UOM2 (TEXTID) – optional secondary unit of measure nomenclature (such as lbs) in which the quantity of this item in a container is tracked.
11. UOMConversionFactor (FLOAT) – the floating point number you multiply quantities in the primary unit of measure by to get quantities in the secondary unit of measure. This is required if UOM2 is specified.
12. IsInventoried (YNBOOL) – defaults to Y for being an inventoried item. Set to N for items that are not tracked in inventory, such as shipping charges.
13. IsIndividual (YNBOOL) – this is set to Y for items that have their own "license-plate" tracking barcodes as opposed to having a quantity of materials in (or on) a container with a tracking barcode.

14. IsSerialized (YNBOOL) – this is set to Y to tell BellHawk to collect a serial number in addition to the tracking barcode on an individually barcoded item or container.
15. DefaultLocation (TEXTID) - this optional field is the default put-away location for an item. It is automatically filled in on a Move transaction but can be over-ridden.
16. LotControl (INTEGER) – required - every batch of materials entered into BellHawk through a transaction is given an internal lot number, which may span multiple containers of material. If LotControl = 0 then external lot numbers are not requested or displayed for these items. If LotControl = 1 then a lot number is automatically generated based on a prefix and a sequence number setup on a BellHawk System Administrators screen. If LotControl = 2 then the lot number is requested manually in all transactions. This setting is also used when the lot number is extracted along with the item number by scanning a composite GS1 standard barcode.
17. QCEvery (INTEGER) – required - used with the BellHawk Quality Control option. If QCEvery =0 then, when entering materials into inventory, mark them as automatically having passed inspection and do not check their QC status. This may be used for packing materials and the like that are not subject to formal QC inspection. If QCEvery =1 then set every part with this item number as needing inspection before usage.
18. EstProdLife – (INTEGER) - required - estimated product life. If EstProdLife=0 do not request expiration date on entering materials into inventory. If > 0 then request expiration date when entering a new container of material into BellHawk.
19. MinInventoryQty. (FLOAT)This is the level used for inventory replenishment warnings on screens, reports and for generating alerts - optional- defaults to 0.
20. Comments – (TEXT) this is for notes about the item – which appear on the Item Master setup screen in BellHawk - optional.
21. UDP (JSON)
22. IsDeleted (YNBOOL)
23. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
24. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Shipper*

DTO/Table Name: BH2DEX-Shipper

Fields:

1. ID (INTEGER) - Unique identifier
2. ShipperCode (TEXTID) - Unique Code Identifying Shipper
3. ShipperDescription (TEXT) - Longer Form for Reports
4. IsDeleted (YNBOOL) - IsDeleted flag from BellHawk

5. DateLastMod - (DATETIME) - set by DEX DTO whenever record is updated
6. IsTransferred (TEXTID This is set to "W" when the record is ready to be transferred (typically when the record is written into the DEX table) and is set by DEX to "T" if the transfer is completed successfully or "E" if not.

## 17 Tables for the Transfer of Order Data from BellHawk

### *Purchase Order Header*

Table/DTO Name: BH2DEX-PO

Fields:

1. PONumber (TEXTID) Number. Required.– Purchase Order Number - unique identifier
2. SupplierNumber (TEXTID) – supplier/vendor number for this PO. Required.
3. ProjectCode (TEXTID) - optional field for job/project code.
4. DateWanted – (DATE) Date wanted – Optional.
5. IsCustomerOwned - (YNBOOL) - Y= Customer Owned; N= Owned by MYCOMPANY required.
6. OwnedBy (TEXTID) – code of customer owning material - required - set to MYCOMPANY if not customer owned.
7. IsReleased - (YNBOOL) – Y=Yes PO is released for receiving materials; N= No – do not receive materials against this PO until it is released. Defaults to N.
8. IsClosed (YNBOOL) - used to set PO as closed if IsClosed =Y and open if IsClosed = N. Receipts cannot be recorded in BellHawk against closed POs, neither can changes to PO lines.
9. IsCancelled (YNBOOL) used to set PO as cancelled if IsCancelled = Y and open if IsCancelled = N. Receipts cannot be recorded in BellHawk against cancelled POs, neither can changes to PO lines.
10. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
11. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Purchase Order Line*

Table DTO Name: DEX2BH-POL

Fields:

1. PONumber - (TEXTID) Required – Matches Purchase Order Header Number.

2. LineNumber - (TEXTID) Required. PO Line Number such as 1,2, 3a, 3b, etc.
3. ItemNumber – (TEXTID) BellHawk Item Master Part Number. Required
4. ItemDescription - (TEXT) - Item Description from PO – Description of Item ordered.
5. Quantity (FLOAT)– Quantity ordered of BellHawk part number - required
6. UOM – (TEXTID) Units of measure notation (such as ea, lbs, ft etc.) for quantity of parts ordered. Required. Must match previously defined UOM nomenclature in BellHawk - required.
7. UnitCost – (FLOAT) Unit cost in specified UOM for PO Line.
8. DateWanted – (DATETIME) Date PO Line Item wanted. Optional.
9. IsReleased - (YNBOOL) – Y=Yes PO is released for receiving materials; N= No – do not receive materials against this PO until it is released in BellHawk. Defaults to N.
10. IsClosed (YNBOOL) - used to set PO line as closed if IsClosed =Y and open if IsClosed = N. Receipts cannot be recorded in BellHawk against closed PO lines, neither can changes to the PO line.
11. IsCancelled (YNBOOL) used to set PO line as cancelled if IsCancelled = Y and open if IsCancelled = N. Receipts cannot be recorded in BellHawk against cancelled PO lines, neither can changes to the PO line.
12. UDP (JSON)
13. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
14. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

Note that the PO Line entries are uniquely identified by the PONumber and LineNumber combination.

### *Work Order*

Table/DTO Name: BH2DEX-WO

BellHawk prerequisites SPTS or RTOPS

Fields:

1. WONumber - Work Order Number - (TEXTID)
2. TravelerBarcode - Appears as WO Barcode on Traveler (TEXTID)
3. WOType - Work Order Type W = Batch/Service; M=Processing (TEXTID)
4. ItemNumber - Item Part Number to be made (TEXTID) - only for M work order types
5. Instructions - Work Order Instructions (MLTEXT)

6. QtyToBeMade - Quantity to be made (FLOAT) in primary UOM for Item - only for M work order types
7. QtyMade - Quantity already made (FLOAT) in primary UOM for Item - only for M work order types
8. LotNumber -  Lot Number for Parts to be Made( TEXTID) - only for M work order types
9. DateWanted - (DATE)
10. Importance - WO Importance (TEXT) Standard, Rush etc.
11. IsReleased - Is WO Released (YNBOOL)
12. DateReleased (TIMEDATE)
13. PlannedStartDate - (DATE)
14. IsStarted (YNBOOL)
15. DateStarted (TIMEDATE)
16. IsCustomerOwned - Are Items made Customer Owned (YNBOOL) only for M work order types
17. CustomerNumber - For Customer Number (TEXTID)
18. ProjectCode - Project Code (TEXTID) -optional - but, if provided, will match an open project in BellHawk.
19. IsCancelled (YNBOOL) - used to flag BellHawk that the Work Order is cancelled.
20. DateCancelled (TimeDate)
21. IsClosed (YNBOOL)
22. DateClosed (YNBOOL)
23. UDP - (JSON)
24. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
25. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

Note that Work Orders cannot be deleted; they can only be marked as Cancelled or Closed.

### *Work Order Route Step*

Table/DTO Name: BH2DEX-WOStep

BellHawk prerequisites SPTS or RTOPS

Fields:

13. TravelerBarcode - required - WO Traveler Barcode (TEXTID)
14. StepCode - Step Code (TEXTID) - unique within WO - used for step barcode on traveler.
15. WONumber - Work Order Number - (TEXTID)
16. StepNumber - Step Number (NUMID) - sets sequence of operation steps within WO
17. OperationCode - Operation Code (TEXTID)
18. Instructions - Operation Instructions (MLTEXT)
19. UDP - (JSON)
20. IsDeleted (YNBOOL) - Record Marked as IsDeleted in BellHawk
21. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
22. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

Note that:

- The Work Order Route Step entries are uniquely identified by the TravelerBarcode and StepCode combination.
- Deleted Work Order Steps need to be transferred from BellHawk to DEX.

### *Work Order Step Part In*

Table/DTO Name: BH2DEX-WOStepPartIn

BellHawk prerequisites RTOPS

Fields:

10. TravelerBarcode - required WO Traveler Barcode (TEXTID)
11. StepCode - Step Code (TEXTID) - required -must be unique within WO
12. ResCode - Resource Code (TEXTID) -required - must be unique within WO Step
13. WONumber - Work Order Number - (TEXTID)
14. SeqNo - required - Sequence number in which parts appear in BellHawk (INTEGER)
15. ItemNumber - Item Number for Part In (TEXTID)
16. Quantity - Part-In Quantity (FLOAT) in primary UOM for Part-In
17. IsWIP - Is Part-In WIP materials? (YNBOOL)
18. UDP- (JSON)
19. IsDeleted (YNBOOL) - Record Marked as IsDeleted in BellHawk

20. DateLastMod (DATETIME)
21. TransferStatus (TEXTID)

Note that:

- The Work Order Route Step resource entries are uniquely identified by the TravelerBarcode,  StepCode, and ResCode combination.
- Deleted Work Order Step resources need to be transferred from BellHawk to DEX.

### *Work Order Step Labor*

Optional - Used to report on estimated versus actual labor needed for WO.

BellHawk prerequisites SPTS or RTOPS

Table/DTO Name: BH2DEX-WOStepLabor

Fields:

9. TravelerBarcode - WO Traveler Barcode (TEXTID) -required
10. StepCode - Step Code (TEXTID) - must be unique within WO - required
11. ResCode - Resource Code for each Labor Item  (TEXTID) must be unique within WO Step - required
12. WONumber - Work Order Number - (TEXTID)
13. SeqNo - Sequence number in which labor resources appear (INTEGER) required
14. LaborClassCode - Labor Class Code (TEXTID)
15. Hours - Estimated time to perform operation (FLOAT) required
16. IsDeleted (YNBOOL) - Record Marked as IsDeleted in BellHawk
17. DateLastMod (DATETIME)
18. TransferStatus (TEXTID)

Note that:

- The Work Order Route Step resource entries are uniquely identified by the TravelerBarcode,  StepCode, and ResCode combination.
- Deleted Work Order Step resources need to be transferred from BellHawk to DEX.

### *Work Order Step Part Out*

Table/DTO Name: BH2DEX-WOStepPartOut

BellHawk prerequisites RTOPS

Fields:

11. TravelerBarcode - WO Traveler Barcode (TEXTID)
12. StepCode - Step Code (TEXTID)
13. ResCode - Resource Code (TEXTID) to identify Part Out
14. WONumber - Work Order Number - (TEXTID)
15. SeqNo - Sequence number in which parts out appear (INTEGER) - required
16. ItemNumber - Item Number for Part in (TEXTID) - required
17. Quantity - Expected Part Out Quantity (FLOAT)- required
18. UOM - Default Unit of Measure nomenclature for Part Out (TEXTID) -required
19. IsWIP - Is Part Out WIP materials? (YNBOOL)
20. UDP - (JSON)
21. IsDeleted (YNBOOL) - Record Marked as IsDeleted in BellHawk
22. DateLastMod (DATETIME)
23. TransferStatus (TEXTID)

Note that:

- The Work Order Route Step resource entries are uniquely identified by the TravelerBarcode,  StepCode, and ResCode combination.
- Deleted Work Order Step resources need to be transferred from BellHawk to DEX.

### *Ship Orders Headers*

Ship Orders consist of a Ship Order Header record and one or more Ship Order Line records. Please note that while ship orders may be the same as sales orders, in an ERP, sales order, or accounting system, they may also correspond to one shipment or release under a "blanket" purchase order or contract.

Table Name: BH2DEX-SO

Fields:

1. ShipOrderNumber - Ship Order Number (TEXTID) - Required - must be unique
2. CustomerNumber - Customer Number (TEXTID) - Required
3. DateWanted - Date Wanted (DATE) Optional
4. ShipDate - Date to be Shipped (DATE) Optional
5. SODescription - optional Ship Order Notes (MLTEXT) 100 characters max

6. BillToStreet1 Bill To Street1(TEXT) - Note that the BillTo information is optional
7. BillToStreet2 Bill To Street2 (TEXT)
8. BillToCity - Bill To City (TEXT)
9. BillToState - Bill To State (TEXT)
10. BillToZip - Bill To Postal Code (TEXT)
11. BillToCountry- Bill To Country (TEXT)
12. ShipToStreet1 - Ship To Street1 (TEXT) Note that the Ship To information is Optional
13. ShipToStreet2 - Ship To Street2 (TEXT)
14. ShipToCity - Ship To City (TEXT)
15. ShipToState - Ship To State (TEXT)
16. ShipToZip - Ship To Postal Code (TEXT)
17. ShipToCountry - Ship To Country (TEXT)
18. ImportanceCode - Priority Codename - (TEXTID) Optional default to Standard
19. IsReleased - Set to Y for immediate release - (YNBool) Optional - defaults to N
20. Facility - Facility to Ship From (TEXTID) - optional
21. ContractNumber - Contract Number (TEXTID) - optional
22. ReleaseNumber - Release Number Under Contract - (TEXTID) - optional
23. ShipperCode - (TEXTID) - optional - such as UPS - must match Shipper Code in BellHawk
24. ShipperDescription - Shipper Description (TEXT) optional - 50 characters max
25. FOB - Free on Board Statement - (TEXT) optional - 50 characters max
26. Terms - Terms Statement - (TEXT) optional - 50 characters max
27. SalesPerson - Sales Person Name- (TEXT) optional - 50 characters max
28. IsComplete - Set to Y to Close - (YNBOOL) defaults to N
29. IsCancelled - Set to Y to Cancel - (YNBOOL) defaults to N
30. dtCompleted - Date Completed (DATE) required if IsComplete = Y
31. dtCancelled - Date Cancelled (DATE) required if IsCancelled = Y
32. ShipFromStreet1 - ShipFromStreet1 (TEXT) - Note Ship From fields are optional
33. ShipFromStreet2 - ShipFromStreet2 (TEXT)

34. ShipFromCity - ShipFromCity (TEXT)
35. ShipFromState - ShipFromState (TEXT)
36. ShipFromZip - ShipFromZip (TEXT)
37. ShipFromCountry- ShipFromCountry (TEXT)
38. CustOrderNumber - optional Customer Order Number (TEXT)
39. CustOrderDate - optional Customer Order Date (DATE)
40. ProjectCode - optional Project Code for Order (TEXTID)
41. UDP (JSON)
42. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
43. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Ship Order Lines*

DTO/Table Name: BH2DEX-SOL

BellHawk prerequisites MTS or RTOPS + SO

Field Names:

1. ShipOrderNumber - Ship Order Number (TEXTID) Required - must match header record
2. ShipLineNumber - Line Number (TEXTID) Required - must be unique within Ship Order
3. SeqNo - Sequence Number (Integer) - required - to appear within Ship Order
4. ImportanceCode - Priority (TEXTID) - optional defaults to Standard
5. ItemNumber - Item Number (TEXTID) - required
6. ItemDescription - Item Description (MLTEXT) - optional max 255 characters
7. UOM - Order Line UOM - (TEXTID) - required - must match an available UOM for Item Nomenclature in BellHawk
8. Quantity - Order Quantity (FLOAT) - required
9. UnitPrice - Order Unit Price (DECIMAL) - required
10. IsComplete - Set to Y to Close - (YNBOOL) defaults to N
11. IsCancelled - Set to Y to Cancel - (YNBOOL) defaults to N
12. dtCompleted - Date Completed (DATE) required if IsComplete = Y
13. dtCancelled - Date Cancelled (DATE) required if IsCancelled = Y

14. IsReleased - Is Released (YNBOOL) defaults to N
15. IsMakeToOrder - to be made to order (YNBOOL) defaults to N
16. DateWanted- Date Wanted (DATE) - defaults to overall SO Date Wanted
17. ProjectCode - optional Project Code (TEXTID)
18. UDP(JSON)
19. DateLastMod - DATETIME - set by DEX DTO whenever record is updated
20. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

## 18. Tables for the Transfer of Status Data from BellHawk

### *Inventory History Snapshot*

This DTO generates a new snapshot of the inventory in the BellHawk database every time it is run.

Table Name: BH2DEX-InvHistory

BellHawk Prerequisites MTS or RTOPS

Fields:

1. ItemNumber - TEXTID BellHawk Item Number
2. ItemCategory - TEXTID for sub-selecting data exports
3. IsCustomerOwned (YNBOOL)
4. CustomerNumber (TEXTID) - MYCOMPANY or customer number, if customer owned.
5. FacilityCode (TEXTID) the BellHawk facility in which the inventory is located.
6. PrimaryQty -(FLOAT) Total quantity of Item at the facility where the Item is located in primary UOM for Item
7. PrimaryUnitCost - (FLOAT) Average unit cost of all materials of type Item in the Facility in primary UOM for Item
8. PrimaryUOM - (TEXTID) primary UOM nomenclature for Item
9. SecondaryQty - FLOAT Total quantity of Item at the facility where the Item is located in secondary UOM for Item
10. SecondaryUnitCost - (FLOAT) Average unit cost of all materials of type Item in the Facility in secondary UOM for Item
11. SecondaryUOM - (TEXTID) primary UOM nomenclature for Item

12. Snapshot (INTEGER) - unique snapshot number - YYDDDNN - where DDD is day of year and NN starts at 00 and is incremented for each snapshot on a specific day
13. DateLastMod - DATETIME - set by DEX DTO
14. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

A new snapshot, with a unique snapshot number, will be created each time the Inventory History DTO is run.

### *Containers*

Table/DTO BH2DEX-Containers

This will be updated whenever there is a change in the BellHawk Containers Table.

1. ContainerCode (TEXTID) Container LPN barcode or internal code assigned by BellHawk to virtual container within a parent container. May change over time, as container is relabeled
2. ContainerUOI (TEXTID) unique GUID code assigned to container - remains invariant over time.
3. ContainerType -(TEXTID) Container Type code
4. ParentContainer (TEXTID) Parent Container Barcode (if container is inside a parent container)
5. Location (TEXTID) Location Code - Location Code or User Badge Barcode
6. LocationType (INTEGER) LocationType 0=Location 1=Employee
7. ItemNumber (TEXTID) Item Master Part Number
8. UID (TEXTID) such as UPC or GTIN code, used for product identification
9. PrimaryQty (FLOAT) Quantity in Primary UOM
10. Primary UOM (TEXTID) - Nomenclature (such as EA, LBS, or FT)
11. SecondaryQty (FLOAT) Quantity in Secondary UOM
12. Secondary UOM (TEXTID) - Nomenclature (such as EA, LBS, or FT)
13. LotNumber (TEXTID) Lot Number
14. SerialNumber (TEXTID) Serial Number
15. ExpirationDate (DATE) Expiration Date
16. UnitCost Unit Cost
17. TareWeight Tare Weight
18. GrossWeight Gross Weight
19. IsEmpty (YNBOOL) Set to Y when Container is Empty
20. IsVirtual (YNBOOL) Set to Y when this is a Virtual Container
21. IsIndividual (YNBOOL) Set to Y for Individually Barcoded Item with primary quantity=1
22. IsWIP (YNBOOL) Set to Y for contains work-in-process materials

23. IsPending (YNBOOL) Set for Pre-Printed Label w/material not yet entered into inventory and for ASN material that is yet to be received.
24. InTransit (YNBOOL) Set to Y to indicate container has been shipped but is still being tracked by BellHawk.
25. ProjectCode (TEXTID) Was purchased or made for this project code.
26. OwnedByCustomer (TEXTID)Customer Number if Customer Owned
27. ForCustomer (TEXTID) Material Designated for this Customer Number
28. QCStatus (TEXTID) A=Auto Approved; P=Passed Inspection; W=Waiting Inspection; F=Failed Inspection
29. QCDate (DATETIME) Date and Time QC Status was last changed.
30. PickOrder (TEXTID) Pick Order Number - was picked on this order
31. PickOrderLine (TEXTID) Pick Order Line Number - was picked against this order line
32. ShipOrder (TEXTID) Was shipped against this Ship Order Number
33. ShipOrderLine (TEXTID) Was shipped against this Ship Order Line Number
34. PONumber (TEXTID) Purchase Order Number against which this container was received.
35. POLine (TEXTID)Barcode of PO Line this container was received against
36. SupplierNumber (TEXTID) Supplier Number
37. Supplier Lot (TEXTID) Supplier Lot Number
38. WorkOrder (TEXTID) Work Order Number that produced container
39. WorkOrderOperation (TEXTID) Work Order Operation that produced container
40. ContainerSeqNo (INTEGER) Container sequence number
41. ContainerSeqTotal (INTEGER) Expected Total Number of Containers in Sequence
42. Description (TEXT) Container description/notes
43. MaterialCost (FLOAT) - total cost of materials in container
44. LaborCost (FLOAT) accumulated labor cost for making materials in container
45. Machine Cost (FLOAT) accumulated machine time cost for making materials in container
46. UDP (JSON) User Defined Parameters
47. DateTimeCreated (DATETIME) Date Time container Received or Produced

48. DateTimeLastChanged (DATETIME) - Date that container record data was last changed.
49. IsDeleted (YNBOOL) - set to Y to indicate that container is not available for transactional processing.
50. DateTimeLastCycleCount (DATETIME) when last cycle count was performed.
51. DateTimeLastAudit (DATETIME) when container was last audited.
52. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
53. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

Please note that not all fields will be filled out with data in each record as many depend on the BellHawk options licensed and the transactions performed on the container.

### *Work Order Status*

A record will be written in the DEX table when the WO traveler header record is created in BellHawk. This record will then be updated whenever the status of the work order changes.

Table/DTO Name: BH2DEX-WOStatus

BellHawk Prerequisites SPTS or RTOPS

Field Names:

1. WONumber - Work Order Number - (TEXTID)
2. TimeCreated - (DATETIME) Date and Time record created
3. IsReleased - Y/N (YNBOOL)
4. TimeReleased - (DATETIME) Date and Time Released, or Null if not
5. IsStarted - Y/N (YNBOOL) - has work started on this WO (Labor, Materials, or Machine time recorded in for WO)
6. TimeStarted - (DATETIME) Date and Time Released, or Null if not
7. IsCompleted - Y/N (YNBOOL) - has manager marked WO as Complete
8. TimeCompleted - (DATETIME) Date and Time completed, or Null if not
9. IsCancelled - Y/N (YNBOOL) - has manager marked WO as Cancelled
10. TimeCancelled - (DATETIME) Date and Time cancelled, or Null if not
11. UDP (JSON) - from the traveler header - contains UDP data for the work order itself
12. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
13. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is complete

### *Work Order Progress*

Reported every time an operator records a Stop Work or End Work with the Completed flag checked.

Table/DTO Name: BH2DEX-WOProgress

BellHawk Prerequisites SPTS or RTOPS

This is generated by the WOStatus DTO which also creates entries in the WOStatus table in DEX - there is not a separate WOProgress DTO

Field Names:

1. ID (INTEGER) - uniquely identifies this record in the table so that it can be updated when the corresponding data in BellHawk is changed.
2. WONumber - Work Order Number - (TEXTID)
3. StepCode - Step Code (TEXTID) - Step Barcode from Traveler or NULL if Operation Code Scanned
4. Operation - Operation Code (TEXTID)
5. IsStarted - Y/N (YNBOOL)
6. TimeStarted - (DATETIME) Date and Time Started, if started, or Null if not
7. IsOperatorComplete - Y/N (YNBOOL) - has operator checked completed flag?
8. TimeOpComplete - (DATETIME) Date and Time operator marked operation as complete, if operator complete, or Null if not.
9. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
10. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed

### *Shipments*

These records track the status of Shipments by Shipment Number from the time that they are created to the time that the dock door is closed and the shipment is completed. The shipment numbers are also referenced in Items Shipped and Shipped Containers

Table Name: BH2DEX-Shipments

BellHawk Prerequisites MTS or RTOPS + SO and SDO or WMS

Fields:

1. ID (INTEGER) For Updating Shipment Record
2. ShipmentNumber (TEXTID) - Shipment Number

3. Description (TEXT)
4. Status (TEXTID)
5. DateTimeOpen (DATETIME) Date Time when Dock Opened
6. DateTimeShipped - (DATETIME) Date Time Shipped
7. DockLocation - (TEXTID) Dock Door Location Code - optional
8. ShipperCode - (TEXTID) Shipper Code such as UPS - optional
9. TruckNumber (TEXTID) Truck Number - optional
10. Driver (TEXT) Driver Name - optional
11. IsDeleted (YNBOOL) If record is deleted
12. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
13. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

## 19. Tables for the Transfer of ERP Data from BellHawk

### *Receipts*

The purpose of receipt records is to update the ERP system after materials are received either from a simple Receive transaction or a Receipt against PO transaction.

These are exported from BellHawk after approval from the receiving supervisor.

Table Name: BH2DEX-Receipts

BellHawk Prerequisites MTS or RTOPS + PO

Field Names:

1. SupplierNumber (TEXTID) - Supplier Number
2. PO (TEXTID) - PO Number
3. LineNumber (TEXTID) - PO Line Number - if received against a PO
4. ItemNu mber (TEXTID) - Item Number
5. UOM (TEXTID) - Unit of Measure - for PO line item
6. Quantity (FLOAT) - Quantity in Unit of Measure
7. UnitCost (FLOAT) - Unit Cost (Possibly adjusted in BellHawk from Unit Cost on PO line)
8. ExportDate (DATE) - date approved for export from BellHawk by Receiving Supervisor

9. SeqNumber (NUMID) - This is incremented every time the Receiving Supervisor approves a batch of receipts to be exported to the ERP system.
10. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
11. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

Note that:

1. The receipt records are for aggregated quantities of each item, which may have been received in multiple containers in BellHawk. This is intended to provide data about receipts to ERP and Accounting systems.
2. DEX will transfer a group of receipts once they are approved in BellHawk. These will all have the same RefNumber.

### *Picked Materials Summary*

The purpose for this is to relay picked materials back to The ERP system from BellHawk so that raw materials inventory in The ERP system can be relieved as materials are picked.

Please note that these picked records will be generated when the pick order is marked as completed in BellHawk, even if not all items on the Pick order have been picked.

Table/DTO Name: BH2DEX-Picked

BellHawk Prerequisites MTS or RTOPS + PICK or WMS

Fields:

1. PickOrderNumber (TEXTID) belongs to this header record.
2. LineNumber (TEXTID) uniquely identifies this specific record within Pick Order
3. ItemNumber (TEXTID) Item Number
4. Quantity (FLOAT) - total quantity picked.
5. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
6. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Items Shipped*

The purpose of items shipped records is to update the ERP system after materials are shipped either from a simple Ship transaction or a Ship Picked transaction or using the Shipping Dock Option.

These are exported from BellHawk after approval from the receiving supervisor.

Table Name: BH2DEX-ItemsShipped

BellHawk Prerequisites MTS or RTOPS + SO

Field Names:

1. ShipmentNumber (TEXTID) - Shipment Number
2. CustomerNumber (TEXTID) - Customer Number
3. ShipOrderNumber (TEXTID) - SO Number
4. ShipLineNumber (TEXTID) - SO Line Number - if Shipped against a SO
5. ItemNumber (TEXTID) - Item Number
6. UOM (TEXTID) - Unit of Measure - for SO line item
7. Quantity (FLOAT) - Quantity in Unit of Measure
8. UnitCost (FLOAT) - Unit Price
9. ShipDate (DATE) When Shipped
10. WhenExported (DATETIME) - date approved for export from BellHawk by Shipping Supervisor
11. SeqNumber (NUMID) - This is incremented every time the Shipping Supervisor approves a batch of Shipments to be exported to the ERP system.
12. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
13. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

Note that:

1. These records are intended for updating an ERP or accounting system about items shipped against SO and SO lines imported into BellHawk.
2. The Shipped Items records are for aggregated quantities of each item, which may have been shipped in multiple containers in BellHawk.
3. DEX will transfer a Shipment once it is approved in BellHawk. All related entries will all have the same Shipment SeqNumber.

### *Inventory Snapshot*

These are intended for export to an ERP/accounting system and only include materials that are not customer owned. Quantities for each item are grouped by ERP location to support ERP systems that separately track materials by their location codes.

Table/DTO Name: BH2DEX-InvSnapshot

Fields:

1. ItemNumber - TEXTID BellHawk Item Number
2. ItemCategory - TEXTID for sub-selecting data exports
3. ERPCode - TEXTID from location table
4. Quantity - FLOAT Total quantity of Item at ERP location in primary UOM for Item
5. UnitCost - FLOAT Average unit cost of all materials of type Item in ERP location in primary UOM for Item
6. UOM - TEXTID primary UOM nomenclature for Item
7. DateLastMod - DATETIME - set by DEX DTO
8. IsDeleted (YNBOOL)
9. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

If there is no entry in a new snapshot corresponding to an existing entry, in the prior snapshot, then that entry is marked as IsDeleted = Y so these can be ignored in updating an ERP system.

## 20. Tables for the Transfer of Transaction History Data from BellHawk

These records are added incrementally as they are transacted in BellHawk. They are not overwritten with the exception of the Work Order Labor records, which are initially written when the operator starts work and updated when the operator ends work.

### *Received Materials*

Table Name: BH2DEX-ReceivedMaterials

BellHawk Prerequisites MTS or RTOPS

Fields:

1. ContainerUOI - Container Universal Identifier - such as "A68B06F8-B3D1-495C-BB2A-1B910BD6B6F1" (TEXTID) - a unique identifier for each container. This will be for the innermost real or virtual container holding the materials received.
2. ContainerCode - Container Barcode, if the container holding the received materials is a Type 1, single item use, container or this record is for an individually barcoded item. This field may also contain the tracking code assigned to virtual containers. This is so that this data can be correlated with the contents of the containers table in BellHawk.
3. Project (TEXTID) - project code, if any associated with received containers
4. PO  - Purchase Order Number against which this container was received (TEXTID)
5. POLine - Purchase Order Line Number against which this container was received (TEXTID)
6. Location - Location Code - (TEXTID) - may be "Receiving" or some other location

7. ItemNumber- Item Master Part Number (TEXTID) in container
8. LotNumber - optional - Lot Number (TEXTID) if applicable
9. SerialNumber Serial Number (TEXTID) for individually barcoded item
10. Quantity - Quantity of items in container in Primary UOM for Item (FLOAT)
11. ExpirationDate - Expiration Date (DATE)
12. UnitCost - Unit Cost from PO Line (DOLLAR)
13. WhenReceived - Time-Date Received (DATETIME)
14. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
15. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Picked Materials*

The purpose for this is to relay picked materials back to the DEX database from BellHawk so this can be used to immediately update an ERP system or used for reporting.

Please note that these picked records will be generated when the pick order is marked as completed in BellHawk, even if not all items on the Pick order have been picked.

Table/DTO Name: BH2DEX-PickedMaterials

BellHawk Prerequisites MTS or RTOPS + PICK

Fields:

7. PickOrderNumber (TEXTID) belongs to this header record.
8. LineNumber (TEXTID) uniquely identifies this specific record within Pick Order
9. Project (TEXTID) - project code, if any associated with picked items
10. SourceLoc - Source Location Code (TEXTID)
11. SourceUOI - Source Container UOI for innermost container holding material, real or virtual, before pick.
12. SourceContainerCode - Container Barcode, if the container holding the materials before the move is a Type 1, single item use, container or this record is for an individually barcoded item. This field may also contain the tracking code assigned to virtual containers. This is so that this data can be correlated with the contents of the containers table in BellHawk.
13. IsWholeContainer - Y/N (YNBOOL) Y=>Whole container picked
14. Item Number (TEXTID) Item Number

15. Quantity (FLOAT) - total quantity picked.
16. WhenPicked - Time-Date Picked (DATETIME)
17. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
18. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Moved Materials*

Table Name: MovedMaterials

BellHawk Prerequisites MTS

Fields:

1. SourceLoc - Source Location Code (TEXTID)
2. SourceUOI - Source Container UOI for innermost container holding material, real or virtual, before move.
3. SourceContainerCode - Container Barcode, if the container holding the materials before the move is a Type 1, single item use, container or this record is for an individually barcoded item. This field may also contain the tracking code assigned to virtual containers. This is so that this data can be correlated with the contents of the containers table in BellHawk.
4. ItemNumber- Item Master Part Number (TEXTID) in container
5. ItemCategory – (TEXTID) for sub-selecting data exports
6. LotNumber - optional - Lot Number (TEXTID) if applicable
7. Project (TEXTID) - project code, if any associated with moved items
8. SerialNumber Serial Number (TEXTID) for individually barcoded item
9. Quantity - Quantity of items in container in Primary UOM for Item (FLOAT)
10. DestLoc - Destination Location Code (TEXTID)
11. DestUOI - Destination Container UOI for innermost container holding materials after move. May be same as SourceUOI for a whole container move.
12. DestContainerCode - Container Barcode, if the container holding the materials after the move is a Type 1, single item use, container or this record is for an individually barcoded item. This field may also contain the tracking code assigned to virtual containers. This is so that this data can be correlated with the contents of the containers table in BellHawk. May be the same as the SourceContainerCode if a whole container moved.
13. WhenMoved - Time-Date Moved (DATETIME)
14. DateLastMod (DATETIME) - updated by DEX mechanism when record is written

15. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Work Order Labor*

Reports labor expended on Work Order operations - reported on each Stop or End Work transaction. These Work Order Labor records are initially written when the operator starts work and updated when the operator ends work.

Table/DTO Name: BH2DEX-WOLabor

Field Names:

1. ID (INTEGER) - uniquely identifies this record so that it can be updated when the corresponding data in BellHawk is changed.
2. WONumber - Work Order Number - (TEXTID)
3. StepCode - Step Code (TEXTID) - Step Barcode from Traveler or NULL if Operation Code Scanned
4. Operation - Operation Code (TEXTID)
5. EmpNum - Employee Number (TEXTID)
6. First - First Name (TEXT)
7. Last - Last Name (TEXT)
8. LaborRate - (DECIMAL)
9. EquipCode - (TEXTID) - if employee scanned in to equipment
10. EquipPhase - (TEXTID) "S" = Setup, "R' = Run, "D" = Down, "C"= cleanup
11. TimeStarted - (DATETIME) when started work
12. TimeEnded- (DATETIME) when ended work
13. IsComplete - Y/N (YNBOOL)
14. Qty - (Float) - Piecework Quantity or 0 if not entered
15. AllocatedHours (DECIMAL) - allocated hours to task
16. UDP (JSON) - UDP data captured on Stop Work and End Work transactions.
17. IsDeleted (YNBOOL) Previously entered record (such as on Start Work)
has been deleted
18. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
19. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Material into Work Order Operation*

Reports materials recorded into Work Order operation. There is one entry for each time a material is recorded in. This may be updated if edited by the production manager in BellHawk.

Table/DTO Name: BH2DEX-WOMaterialIn

BellHawk Prerequisites RTOPS

Field Names:

1. ID (INTEGER) - index from materials history table in BellHawk
2. WONumber - Work Order Number - (TEXTID)
3. StepCode - Step Code (TEXTID) -  Step Code for Traveler or NULL if Operation Code Scanned.
4. OperationCode - Operation Code (TEXTID)
5. EmpNum - Employee Number (TEXTID) - employee recording material in
6. ItemNumber (TEXTID) - Item Number
7. IsWIP (YNBOOL) - Is this a WIP Item
8. IsCustomerOwned (YNBOOL) Does this material belong to a customer
9. PrimaryUOM (TEXTID) - Primary Unit of Measure
10. PrimaryQty (FLOAT) - Quantity in Primary Unit of Measure
11. SecondaryUOM (TEXTID) - Secondary Unit of Measure
12. SecondaryQty (FLOAT) - Quantity in Secondary Unit of Measure
13. LotNumber (TEXTID) - Lot Number
14. SerialNumber(TEXTID) -SerialNumber
15. ExpirationDate (DATE) - ExpirationDate
16. ContainerUOI (TEXTID) - container from which material was taken
17. ContainerCode (TEXTID) - innermost barcode holding material
18. UnitCost (FLOAT) - Unit Cost of input material in Primary UOM
19. UnitMaterialCost (FLOAT) - Unit Material Cost of input material in Primary UOM
20. UnitLaborCost (FLOAT) - Unit Labor Cost of input material in Primary UOM
21. UnitMachineCost (FLOAT) - Unit Machine Cost of input material in Primary UOM
22. ProjectCode (TEXTID) - Project Code

23. dtStamp (DATETIME) date-time when transaction occurred
24. UDP (JSON) - UDP Data Captured in Material-In transaction
25. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
26. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Material out from Work Order Operation*

Reports materials recorded out from Work Order operation. There is one entry for each time a material is recorded out. This may be updated if edited by the production manager in BellHawk.

Table/DTO Name: DEX2BH-WOMaterialOut

BellHawk Prerequisites RTOPS

Field Names:

1. ID (INTEGER) - index from materials history table in BellHawk
2. WONumber - Work Order Number - (TEXTID)
3. StepCode - Step Code (TEXTID) -  Step Code for Traveler or NULL if Operation Code Scanned.
4. OperationCode - Operation Code (TEXTID)
5. EmpNum - Employee Number (TEXTID) - employee recording material in
6. ItemNumber (TEXTID) - Item Number
7. IsWIP (YNBOOL) - Is this a WIP Item
8. IsCustomerOwned (YNBOOL) Does this material belong to a customer
9. IsScrap(YNBOOL) - Is this scrap material
10. ScrapReason (TEXTID) - Reason code for scrap
11. PrimaryUOM (TEXTID) - Primary Unit of Measure
12. PrimaryQty (FLOAT) - Quantity in Primary Unit of Measure
13. SecondaryUOM (TEXTID) - Secondary Unit of Measure
14. SecondaryQty (FLOAT) - Quantity in Secondary Unit of Measure
15. LotNumber (TEXTID) - Lot Number
16. SerialNumber(TEXTID) -SerialNumber
17. ExpirationDate (DATE) - ExpirationDate

18. QCStatus (TEXTID) - of material output
19. QCReasonCode (TEXTID) - for output material
20. ContainerUOI (TEXTID) - container into which material placed
21. ContainerCode (TEXTID) - innermost barcode for container into which material is placed.
22. LocationCode (TEXTID) - location where material is placed.
23. UnitCost (FLOAT) - Unit Cost of output material in Primary UOM
24. UnitMaterialCost (FLOAT) - Unit Material Cost of output material in Primary UOM
25. UnitLaborCost (FLOAT) - Unit Labor Cost of output material in Primary UOM
26. UnitMachineCost (FLOAT) - Unit Machine Cost of output material in Primary UOM
27. ProjectCode (TEXTID) - Project Code
28. UDP (JSON) - UDP Data Captured in Material-Out transaction
29. dtStamp (DATETIME) date-time when transaction occurred
30. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
31. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Return Material from Work Order Operation*

Reports materials returned from Work Order operation. There is one entry for each time a material is returned. This may be updated if edited by the production manager in BellHawk.

Table/DTO Name: DEX2BH-WOMaterialReturn

BellHawk Prerequisites RTOPS

Field Names:

1. ID (INTEGER) - index from materials history table in BellHawk
2. WONumber - Work Order Number - (TEXTID)
3. StepCode - Step Code (TEXTID) -  Step Code for Traveler or NULL if Operation Code Scanned.
4. OperationCode - Operation Code (TEXTID)
5. EmpNum - Employee Number (TEXTID) - employee recording material in
6. ItemNumber (TEXTID) - Item Number
7. IsWIP (YNBOOL) - Is this a WIP Item

8. IsCustomerOwned (YNBOOL) Does this material belong to a customer
9. PrimaryUOM (TEXTID) - Primary Unit of Measure
10. PrimaryQty (FLOAT) - Quantity in Primary Unit of Measure
11. SecondaryUOM (TEXTID) - Secondary Unit of Measure
12. SecondaryQty (FLOAT) - Quantity in Secondary Unit of Measure
13. LotNumber (TEXTID) - Lot Number
14. SerialNumber(TEXTID) -SerialNumber
15. ExpirationDate (DATE) - ExpirationDate
16. ContainerUOI (TEXTID) - container into which material placed
17. ContainerCode (TEXTID) - innermost barcode for container into which material is placed.
18. LocationCode (TEXTID) - location where material is placed.
19. UnitCost (FLOAT) - Unit Cost of returned material in Primary UOM
20. UnitMaterialCost (FLOAT) - Unit Material Cost of returned material in Primary UOM
21. UnitLaborCost (FLOAT) - Unit Labor Cost of returned material in Primary UOM
22. UnitMachineCost (FLOAT) - Unit Machine Cost of returned material in Primary UOM
23. ProjectCode (TEXTID) - Project Code
24. UDP (JSON) - UDP Data Captured in Material-In transaction
25. dtStamp (DATETIME) date-time when transaction occurred
26. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
27. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Operator Comments*

These are the "post-it notes" comments made by operators when running a work order.

DTO/Table Name BH2DEX-OpComments

Fields:

1. ID - (INTEGER) index of comment
2. EmpNum - Employee Number making Comment
3. WONumber - (TEXTID) Work Order Number

4. dtStamp (DATETIME) date-time when transaction occurred
5. OpComment (MLTEXT)
6. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
7. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

### *Shipped Containers*

This is a list of the containers shipped using a simple ship, ship picked, or the shipping dock option. These records are intended for ASN support for EDI as well as performing detailed reporting about shipments.

Table Name: BH2DEX-ShippedContainers

BellHawk Prerequisites MTS or RTOPS

Fields:

1. ShipmentNumber (TEXTID)
2. ShipOrderNumber - Ship Order Number (TEXTID)
3. ShipLineNumber - Ship Order Line (TEXTID)
4. IsVirtual - (YNBOOL) Y = Has Tracking Barcode; N = Does not have Tracking Barcode.
5. IsParent - (YNBOOL) Y = Parent Container;  N= Not Parent Container
6. IsIndividual - (YNBOOL) Y= Individually barcoded N=Container with Quantity
7. TrackingBarcode - License-Plate Tracking Barcode (TEXTID) - if real, individual, or parent container.
8. ItemNumber -  Item Number (TEXTID)
9. Quantity - in container (FLOAT)
10. Lot - (TEXTID) Lot Number
11. SerialNumber - (TEXTID) Serial Number
12. ExpirationDate - (DATE) Expiration Date
13. UOM - UOM Nomenclature (TEXTID)
14. ParentBarcode - Parent Barcode (TEXTID) if any - typically SSCC barcode
15. ContainerType - Container Type Code (TEXTID)
16. GrossWeight - (FLOAT) Optional Gross Weight of Container
17. TareWeight - (FLOAT) Optional Tare Weight for Container

18. TareWeightUOM (TEXTID) UOM Nomenclature for container weights
19. DateLastMod (DATETIME) - updated by DEX mechanism when record is written
20. IsTransferred (YNBOOL) is set to N by DEX when writing the record. Can be updated to Y by ERP system to record transfer is completed.

These records are transferred as soon as the shipment is approved by the shipping supervisor.

## 21. Notes on Transfers to BellHawk

### *21.1 Preconditions*

It is expected that the following will be setup manually in BellHawk prior to the transfers:

- Units of Measure and Measure Types. All UOMs specified in the tables must have a matching UOM Nomenclature entry in BellHawk otherwise the transfer will be rejected as being in error.
- Item Categories
- Material Types
- Facilities

Item Transfers will be ignored, without error, if the Item Category does not correspond to an entry in BellHawk. Location transfers will be ignored if they do not correspond to a Facility in BellHawk.

When transferring Items, PO lines, Pick Order lines, Ship Order Lines, the DTOs will check that the UOMs correspond to a valid UOM nomenclature in BellHawk and will reject the transfer as an error if there is not a valid match. The same applies to Material Type Codes for Items.

### *21.2 Sequencing*

Sequencing: The data will be transferred from the tables to BellHawk in the following order:

1. Suppliers
2. Customers
3. Projects
4. Locations
5. Items
6. POs
7. PO Lines
8. Pick Order Headers
9. Pick Order Lines

10. Work Order Headers
11. Work Order Operations
12. Work Order

It is the responsibility of the ERP system export mechanism to make sure that entries are in place before the entries that reference them are placed in other DEX tables.

## 22. The Control Database

The Control database provides data used by the DEX program and its DTOs, including scheduling when DTOs run, logging errors, and performing a variety of ancillary functions. This data in this database can be setup by linking the tables with Microsoft Access or by an external program.

The Control database (named DEXControl) contains a number of tables that it is useful to know about:

- BC_CTL – this contains one entry per DTO with data about the name of the executable process and the DTO and parameters such as the Adaptors it uses. It is through this table each DTOs can be set into active state and the times when the DTOs are run are specified.
- tblTransferDef – This table contains the setup/control data for each DTO. It is provided pre-loaded for all the DTOs available with the supplied DEX interface.
- tblErrorLog – in addition to displaying errors and warnings on the DEX screen and writing errors and warnings in the daily log file, the DEX error handling routines write the errors into the Error Log table.
- tblParameterStore – this is where DEX stores the last access times for fetching data for each DTO through each adaptor. It is used to enable DTOs to get the latest updates to database tables without writing the code to find which entries have changed.

There are other tables within the Control database which are not currently used in the standard DEX interface. These tables are, however, used by MilramX and are used when DEX is run as system's process as part of MDEX.

## 23. Commentary

We are in the process of adding additional data objects which can be exchanged between DEX and BellHawk. Our aim is to make all the necessary data objects within BellHawk accessible to users of the BellHawk DEX interface.