Getting data in with the Data Transfer Utility

The **Data Transfer Utility** (DTU) provides a simplified path for getting data from a local computer to the JMS external queue by acting as a *local* directory crawler transport to ingest files from your desktop or from a mapped network drive (any file that is accessible via Windows Explorer). Rather than waiting for a scheduled job or a transport to process a web-based data stream, you control when to run the DTU on a *local* data subset. It is highly effective for getting sample data sets into DigitalEdge test and prototype systems. The Data Transfer Utility uses SSL to transport data securely into the JMS external node and uses available memory to ingest data.

For large production systems, you should specify a complete processing pipeline with **System Builder** rather than using the DTU as a desktop transport mechanism. Consult the DigitalEdge *Configuration Guide* for detailed instructions on configuring transports/data sinks/parsers/data models.

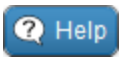
Before using the DTU, you must create an input data model either by doing a full configuration with the Data Model Editor, or by using the Data Model Creation Wizard to construct a simple data model from a flat file. You will need a data model, data sources, and data field mappings so the DTU can transport your data to a DigitalEdge processing pipeline. See the DigitalEdge *Configuration Guide* for details.

Your DigitalEdge system must be **Start**ed and running in the **OK** state before using the Data Transfer Utility.

The DTU can be used as a GUI application or through its command line interface.

Downloading the Data Transfer Utility

The DTU application is an executable JAR file, downloaded from the DigitalEdge **Management Console.**

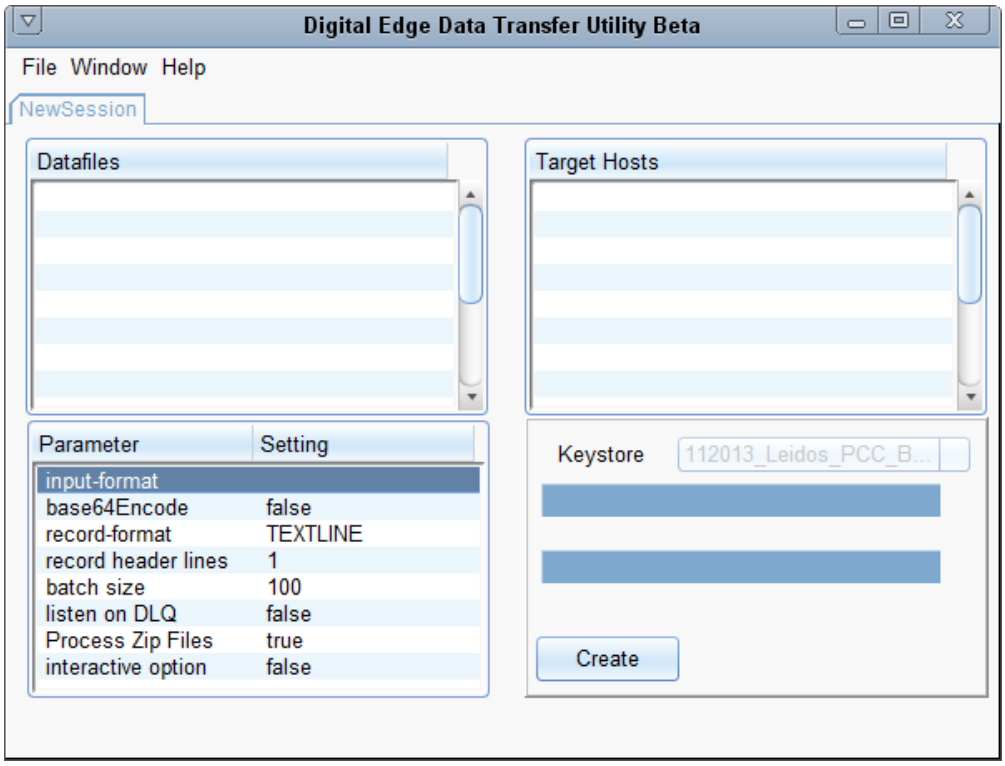
1. From the **Management Console**, click the **Help** icon 

to access the **Product Documentation** page.
2. On the **Data Transfer Utility** line, click Download  .
3. Click **Save File** to copy the DTU file to a directory with read/write access.
4. Click on the executable JAR file to run the DTU.

Using the Data Transfer Utility GUI

Use the GUI to define a data transfer job by specifying input files, destination hosts, keystores, record formats, and other options. You can create and save a transfer job as a "session" in the DTU to run immediately or repeatedly over time.

1. Open the Data Transfer Utility:



2. To create a new data transfer job, simply enter information into the **NewSession** tab.
3. In the left **Datafiles** panel, right-click anywhere to access the **Add/Delete** dialog box. Click **Add Row** to access the **Open** dialog box. Browse to select an input source data file or a directory. To use multiple input files in one data transfer session, repeat this process to add files or directories, one at a time. Or enter the data directly in the **Datafiles** panel without using the **Open** dialog box.

   If you specify a directory as the input, DTU will recurse into all subfolders.

   You can use wildcards in filename specifications (? = one character; * = multiple characters).

   To edit a file name, double-click on its row.

   To delete a file, right-click on a row to access the **Add/Delete** dialog box. Click **Delete Row**.

4. Specify the data model parameters for this job:
   - **input-format**: Specifies the data source that DTU uses to pull data off the input queue; enter as the Data Source name from the Data Model Editor. This parameter also identifies the data model and field mappings that are associated with this data source and which DigitalEdge will use to process your data once it is tranported into DigitalEdge by the DTU (DTU does not do data modeling or field mapping).

- **base64Encode**: Click to select true or false; indicates if the resulting records should be encoded in Base64; use true only with the Binary Parser. Default = false
- **record-format**: Specifies a data record type that helps to determine record boundaries in the source data. Click to select a type of record splitter to use: PCAP, JSON, TEXTLINE, NULL, or TEXTLINEWITHQUOTES. Default = TEXTLINE
- **record header lines**: The number of lines of header information in the source file(s); enter 0 if no header is present. Default = 1
- **batch size**: Defines the size of the destination JMS message batches. Default = 100 messages
- **listen on DLQ**: Click to select true or false; indicates if DTU should listen for messages on the Dead Letter Queue (DLQ). Messages/records go to the DLQ when they cannot be parsed. If this parameter value is true, the DTU will not self-terminate when the transfer session is completed, but will listen on the DLQ and output messages until you kill the DTU. Default = false
- **Process Zip Files**: Unzips and processes zipped files: ZIP, TAR, TAR.GZ, TGZ, TAR.BZ2, TBZ2, GZ, 7Z, RAR, BZ2. Default = true
- **interactive option**: Click to select true or false; Indicates if the DTU should pause between the processing of each JMS message. Default = false

5. In the right **Target Hosts** panel, right-click anywhere to access the **JMS Queue Selection** dialog box. Enter an IP address or DNS name for the target JMS external node that will receive the input data. You can find the IP address at: **Management Console** > **Systems** > **Process Groups** > **jms.external** > double-click for the IP addresses (use the **Copy** function to copy an IP address that you want to paste here). To use multiple queues in one data transfer session, repeat this process to add IP addresses or DNS names, one at a time. (Or enter the data directly in the **Target Hosts** panel without using the **JMS Queue Selection** dialog box.)
6. Select a keystore to encrypt the connection from your PC to the JMS node. Recommendation: Use the default keystore as both a keystore and truststore.

If you do not have a keystore, you can generate and download a keystore from the Management Console (**Management Console** > **Security** > **Download** > **Keystore**). Save the file in a keystore subdirectory under the DTU installation directory. The file must end with a .ks extension to be recognized by the utility.

7. Save your data transfer specifications in a "session" by clicking the **Create** button. DTU saves your session and names it sequentially (Session_0, Session_1, Session_2, etc.).
8. Once you have all options specified, you should test the connection before running the data transfer job.
   a. Click **Test** to verify the connection, SSL, and the existence of the specified files.
   b. Any warnings or errors will be reported above the **Test** button and saved in the **Logs** file.
   c. Click the **Logs** tab (or select **Window** > **Logs**) to review the text results.
   d. Correct any problems and retest the connection.

9. To run the session, fist be sure that your DigitalEdge system is started and running in the Management Console. Click the **Session_n** tab here, then click **Upload**. When the session runs, the input files are uploaded to the JMS queue and processed.
   a. Status is indicated by the two progress bars above the **Test** button.
   b. The first progress bar indicates overall progress.
   c. The second progress bar tracks the transmittal of individual files.
   d. Processing, warnings, and errors are captured in the **Logs** file. Click the **Logs** tab (or select **Window** > **Logs**) to review the session's history.
   e. If a record fails to be processed, it is sent to the Dead Letter Queue (DLQ) and recorded in the **Logs** file. Click the **DLQ** tab (or select **Window** > **DLQ**) to view records that ended up in the DLQ. You can also copy records from the DLQ to another file, edit them, and resubmit them to DigitalEdge.

Managing DTU sessions

Here is a quick reference list for using DTU functionality.

Data Transfer Utility: GUI Functionality	
Task	**GUI Function**
Create a new session (data transfer job)	**NewSession** tab
Run a session (push data files to the JMS queue)	**Upload**
Stop a running session	Close a **Session** tab, or **Cancel** a session, or **File** > **Exit**
Restore a previous session	**File** > **Open**
Create a new session from a similar session	1. **File** > **Open**. Select a previously saved session. 2. Edit the previous session (with new parameters or different data files) 3. The session is auto-saved with the session number and a timestamp as a filename.
View DTU processing logs	**Logs** tab, or **Window** > **Logs**
View the Dead Letter Queue (records that were not parsed)	**DLQ** tab, or **Window** > **DLQ**
Read DTU help instructions	**Help** > **Manual (PDF)**

Data Transfer Utility: GUI Functionality	
Task	**GUI Function**
Save a session	Click **Create** for DTU to number your session (Session_1, Session_2, etc.)

Using the Data Transfer Utility command line interface

The Data Transfer Utility (DTU) also provides a command line interface which can be called from any location. Use the following syntax:

```
java –jar DataTransferUtility.jar <arguments>
```

If no parameters are passed in, the DTU will run in GUI mode. Use the -?, -a, and -m parameters for information only; all other parameters will run the DTU. For example:

```
java –jar DataTransferUtility.jar -d 192.160.50.100
-k C:\Users\Smith\Downloads\ssl-keystore.ks --trustStore
C:\Users\Smith\Downloads\ssl-truststore
-f C:\Users\Smith\Documents\export.json
-s JSON -i mydata -l -z 100 -h 0
```

Here are the parameters available to the DTU, along with descriptions and arguments.

Data Transfer Utility: Command Line Interface		
Parameter	**Shortcut**	**Explanation**
--help	-?	Displays a list of DTU parameters and explanations
--availableRecordFormat	-a	Displays a list of the available record format types (for input with the --recordFormat parameter); when you use this parameter, DTU ignores all other parameters
--encode	-b	Use Base64 to encode the resulting records; used only with the Binary Parser
--testConnection	-c	Tests the connections and files but does not send any JMS messages yet
--destination <arg>	-d	List of comma-separated public IP addresses or DNS names for the target JMS external nodes
--files <arg>	-f	Path to the input source file or files (directory) that will be sent to JMS.
--recordHeaderLines <arg>	-h	Number of lines of header information in the source file(s); specify 0 if no header is present
--inputFormat <arg>	-i	Identifies the source and parser format that DTU

Data Transfer Utility: Command Line Interface		
Parameter	**Shortcut**	**Explanation**
		uses to pull data off the input queue; specified as the Data Source name from the Data Model Editor
--keyStore <arg>	-k	Path to the keystore that encrypts the connection from your PC to the JMS node; you can obtain a keystore in the Management Console (**Management Console** > **Security** > **Download** > **Keystore**)
--listen	-l	Listens for messages on the Dead Letter Queue (DLQ). Messages go to the DLQ when they cannot be parsed. If this flag is specified, the DTU will not self-terminate when completed, but will listen on the DLQ and output messages until you kill the DTU.
--md5	-m	Calculates the MD5 hash of the DTU file to verify that the file has not been tampered with; when you use this parameter, DTU ignores all other parameters
--zip	-p	Unzips and processes zipped files: ZIP, TAR, TAR.GZ, TGZ, TAR.BZ2, TBZ2, GZ, 7Z, RAR, BZ2
--recursive	-r	Recurses into subfolders if the input is a directory (specified in the --files parameter)
--recordFormat <arg>	-s	Specifies a record format type that determines record boundaries in the source data; used to split the data stream into logical records; to see the complete list of available record types, use --a
--trustStore <arg>	-t	Path to the truststore that authenticates the connection between your PC and the JMS node; you can obtain a truststore in the Management Console (**Management Console** > **Security** > **Download** > **Truststore**)
--interactive	-w	Pauses between JMS message batches (see --z) and files to review processing for each message; waits for input before proceeding
--mapInputModels	-x	A CSV string that specifies input model mapping parameters, including a Data Source name, the record format for splitting records, and the number of header lines. When you use this parameter, DTU ignores the individual parameters: --recordFormat

Data Transfer Utility: Command Line Interface		
Parameter	**Shortcut**	**Explanation**
		--inputFormat --recordHeaderLines Specify file mappings in the following format: fileName1:recordFormat1:recordHeaderLines1:inputFormat1,fileName2:recordFormat1:recordHeaderLines2:inputFormat2,fileName3:recordFormat3:recordHeaderLines3:inputFormat1 where: • fileName can include wildcards: ◦ ? represents a single character ◦ * represents multiple characters • recordFormat is the data record type to determine how to split multiple records on the appropriate boundaries • recordHeaderLines is the number of header lines that can be stripped out of the records • inputFormat must match a Data Source name as defined in the Data Model Editor White spaces are ignored.
--batchSize <arg>	-z	Sets the size of the JMS message batches; default = 100 messages


You can also embed DTU commands and parameters in a script to run from your local machine.