



Shipping Fulfillment

v 1.1

Contents

Overview of Order Fulfillment	2
Importing New Order Records with Sync	2
Requesting and Updating Orders via API	3

Overview of Order Fulfillment

As a business, it is imperative to your success to get orders to your warehouses or 3PL for fulfillment. Having a smooth order fulfillment process in place will help your business continue to draw customers and satisfy them with your products.

In this document we will discuss the different methods of providing order data to your Third-Party Logistics (3PL) for fulfillment.

Methods of obtaining order fulfillment data:

- Directly from the Sync Database
- API Call

Importing New Order Records with Sync

The Exigo Database Sync is constantly pulling new and updated customer and order records. There is often the need to move these changes into another system. An example may be an accounting package, or shipping fulfillment etc.

The sync has a simple way to snapshot change sets. This allows you freeze a set of records in a queue schema. These tables hold only the records which are new or have changed since the last time you checked. The tables included in this snapshot are Customers, Orders, OrderDetails, Payments, CustomerAccount, and CustomerSite.

This process allows a simple way to constantly pull in new data into another system or process.

QUEUE PROCESS FLOW

The sync database has a schema called 'queue'. To get started execute the following command:

```
EXEC queue.Refresh
```

This command will create the following tables and pull in the last 24 hours of changes:

- queue.Customers
- queue.Orders
- queue.OrderDetails
- queue.Payments
- queue.CustomerAccounts
- queue.CustomerSites

To get the next batch of changes, simply execute the same command again:

```
EXEC queue.Refresh
```

Every time you execute `queue.Refresh` it will replace the above tables with only the records which are new or have changed since the last time you ran.

If you need to go back to a specific point in time you can override the start date by supplying it as a parameter.

Example: `EXEC queue.Refresh '1/1/2017'`

This starts the process with all changes since Jan 1 2017. The next time you run `queue.Refresh` it will resume normal behavior and populate with only changed records since the last command run.

Freezing these records in the queue schema gives you the time to process these as a batch of changes. You could then schedule this whole process to run at an interval which makes sense to your routine.

Please submit a support ticket for any additional questions or training requests.

Requesting and Updating Orders via API

By utilizing the Exigo API infrastructure, you and your 3PL are able to pull all orders that need to be fulfilled directly into the necessary system.

To pull orders into the necessary system via API you or your 3PL will need to have an Exigo Username with API access. API access may be granted directly in Exigo Admin on the username in question.

General Order Lifecycle:

1. A customer places an order.
2. Payment is confirmed received and the Order is created in an Accepted status in Exigo.
3. The Order is then pulled into the 3PL system for fulfillment where a packing slip or invoice is printed.
4. The Order is then updated to a status of Printed by said 3PL.
5. The Order is shipped by the 3PL.
6. The Order is then updated to a status of Shipped and any applicable tracking numbers are added.
7. Order fulfillment is now complete.

Requesting Orders:

- API Call to use: `GetOrders`
- API Documentation: <http://api.exigo.com/3.0/ExigoApi.aspx?op=GetOrders>

As shown in the Exigo API Library Documentation mentioned above, you can request Orders based on a number of different qualifiers.

Examples of ways to identify which orders you want to pull in:

- CustomerID
- Specific list of OrderID's (*Comma delimited list*)
- OrderDate range
- OrderTypes
- OrderStatus
- WarehouseID
- Etc.

Updating Orders:

- API Call to use: [UpdateOrder](#)
- API Documentation: <http://api.exigo.com/3.0/ExigoApi.aspx?op=GetOrders>

**** If you only need to update the status of an Order at any given time you may do so using the [ChangeOrderStatus](#) API Call.*

As shown in the Exigo API Library Documentation mentioned above, you can use this API call to update Orders as needed as they move through the Order lifecycle.

This is generally the call used once a fulfillment slip has been printed to tell Exigo to update the status of a specific order to Printed.

Once said order has been shipped, it is then used again to update the status of an order to Shipped. At this point any applicable tracking numbers may be added to an order. If you have special business rules in place where a 3PL may be required to alter the Ship Method for an order, they may do so with this API call as well.