

# SMARTETAILING

## POS Sync API

Retailers can achieve substantial operational efficiencies when data and processes inherent in their POS system and website are integrated. This document contains the technical information needed to integrate a POS system with SmartEtailing's website management application.

### POS Sync and POS Specials

In the industries SmartEtailing serves, we provide retailers with a professionally prepared catalog library containing thousands of their industries' most popular products. Retailers simply need to review the lists of products and select those they wish to display in their website. Yet, as easy as this is, time needs to be invested by the retailer to continually update these selections and adjust prices as their inventory status changes.

POS Sync is designed to automate a substantial portion of this process by automatically updating the retailers website based on POS inventory status. Items from the SmartEtailing catalog library are automatically displayed and removed from the retailer's website based on current stock status and business rules they define. Product prices are also adjusted to match POS inventory.

To provide additional value from the POS inventory data, SmartEtailing also offers retailers a feature called POS Specials which produces a list of items on a Specials Page. The Specials page is independent from the product catalog and allows retailers to display every POS item that's currently in stock and on sale, grouped by store location (if more than one).

The POS data needed for these features, and the format/method in which it should be provided, is as follows:

#### Data Format

Data should be formatted into two tab-delimited text files (without column headers). The two files should be compressed into a single zip file named "catsync.zip". The file names and fields in the order they are required is as follows (Note: Field Name and Data Type is listed for informational purposes only):

**File Name:** inv.txt

| <b>Field Name</b> | <b>Description</b>  | <b>Data Type</b> |
|-------------------|---|------------------|
| vchsku            | SKU from POS  | varchar (150)    |
| vchmpn            | Manufacturer Part Number (MPN)                                | varchar (150)    |
| vchbrand          | Brand of Item   | varchar (150)    |
| txdescription     | Description of Item*  | varchar (500)    |
| dtsdate           | Date Item was Last Modified                                   | datetime         |
| flprice           | Regular Price   | money            |
| flsaleprice       | Sale Price  | money            |
| dtsalestart       | Sale Start Date   | datetime         |
| dtsaleend         | Sale End Date   | datetime         |
| iminreorder       | Minimum Re-Order Number                                       | int              |
| imaxreorder       | Maximum Re-Order Number                                       | int              |
| iboh              | Balance on Hand   | int              |
| ibohloc           | Balance on Hand per Location**                                | varchar (1000)   |
| iboo              | Balance on Order  | int              |
| dtlastsold        | Date Item was Last Sold                                       | datetime         |
| dtcreated         | Date Item was Created in POS                                  | datetime         |
| iaactive          | Flag if Item is Active (1) or Inactive (0)                    | int              |
| idispweb          | Flag to Display (1) or Not Display (0) in Website             | int              |
| vchgtn1           | GTIN: Global Trade Item Number (ie UPC, EAN) 8/12/13/14 digit | char (14)        |
| vchgtn2           | GTIN: Global Trade Item Number (ie UPC, EAN) 8/12/13/14 digit | char (14)        |

Any field in which a value is not available, leave null, followed by a tab delimiter (unless it's the last field of the table, in which case a carriage return follows instead of a tab).

All date fields should be formatted using "yyyy-mm-dd".

\* Description of Item: If the POS database contains separate sub-description fields (ie. size or color) that are important for the consumer to know, append this information to the description (txdescription) in an easy-to-read format such as separating each sub-description with a comma (ie. "Product Description, Color, Size").

\*\* Balance on Hand per Location: The ibohloc field should contain balance on hand information broken down by store location. Each store location with a balance on hand **greater than zero** for the given sku should be included in this field by entering the LocName (see loc.txt file spec), followed by two bars (||), followed by the balance on hand at that location. Separate multiple locations by three bars (|||). In the following example,

Culver City||8|||Ventura||3|||Boulder||12

there are three locations that have a balance on hand greater than zero (Culver City has 8, Ventura has 3, and Boulder has 12). Note: The iboh field for this sku should contain a value of 23 (the aggregated balance on hand of all locations).

Other than ibohloc, all fields in the inv.txt file are aggregated values from all store locations.

In order to minimize file size and reduce data transport bandwidth, it is preferred that SKUs only be added to the inv.txt file if a GTIN has been entered -- OR -- Manufacturer Part Number (MPN) has been entered --OR-- the item is currently on sale (a sale price other than zero is entered and today's date is equal to or between the sale start and sale end dates). Note: SmartEtailng will be mapping POS items with website catalog items based on a common GTIN or MPN. Please have a discussion with SmartEtailng about the definition of these numbers to make sure we both have the same understanding.

**File Name:** loc.txt

| <b>Field Name</b> | <b>Description</b>            | <b>Data Type</b> |
|-------------------|-------------------------------|------------------|
| vchLocName        | Name of Location <sup>+</sup> | varchar 150      |
| vchAddress        | Street Address                | varchar 150      |
| vchCity           | City                          | varchar 150      |
| vchState          | State                         | varchar 50       |
| vchZip            | Zip Code                      | varchar 50       |
| vchCountry        | Country                       | varchar 50       |
| vchPhone          | Telephone                     | varchar 50       |

<sup>+</sup> The POS admin screen where the retailer configures their SmartEtailng settings should provide a way for the retailer to enter descriptive text for each store location. This is the location name that needs to be entered in the ibohloc field defined in the inv.txt file spec.

### **Data Transport**

The zipped file catsync.zip, containing the two tab-delimited files inv.txt and loc.txt, will be sent to an FTP site hosted by SmartEtailng. Thus, the POS admin screen also needs to include a place for the retailer to enter the following parameters, which will be provided to them by SmartEtailng:

FTP Host Address  
FTP site User Name  
FTP site Password

A scheduling interface is also required on the POS admin screen. This interface should allow the retailer to schedule the frequency of data transmission. The frequency options should at a minimum include Daily (at a user specified time) and a way to invoke the process immediately (ie. Send Now).