Data Feed Source File Specification

Updated 6/23/2020

This document describes the baseline source file format and field requirements for most advertising and selling channels. We pick up your tab- or comma-delimited product source file (.txt or .csv) via FTP or HTTP from your servers or ours. We also import XML files generated according the [MyDataFeed's XML schema](#).

Using a Current Shopping Feed as Your Source File

If you already have an existing feed file, such as Google Shopping product feed, then you can most likely use it as your source file. You will simply need to host the file at an HTTP link or upload it to FTP.

NOTE: Once you import your source file, DO NOT change the file name and format. You may add new fields/columns to your source file (up to 219 fields/columns in total), but you should not rename or remove existing fields unless they are not being used in any of your feeds. Renaming or removing an existing field that is being used in your feeds can cause listings to be removed from your advertising/selling channels.

Sample source files:

- [Products.csv](#)
- [Products.xml](#)
- [Products.txt](#)

Comma/Tab delimited file format

- Source files must be provided in a UTF-8 encoded format
- The first row must contain the field/column names separated by tabs or commas
- Each product/SKU must be on a separate row/line within the file
- You must make sure that no line-breaks are contained within the field content, only between products/SKUs. Line-breaks within product content may cause the product line/row to be parsed incorrectly and fail to import.

Required/Suggested Fields

Below are the required and suggested fields/columns for creating your source file. These fields represent the most common product fields found across the most popular shopping sites. You may add more fields to your source file as needed, however our system can only import 219 fields. Please note, our sample files contain more example fields than are listed below.

NOTE: Field names do not have to be named the same as listed below.

Field Name	Description	Value
Unique ID	The value that uniquely identifies the product in the merchant's system, such as product SKUs **No duplicates**	**Required**
Name	The product name **Between 15 – 150 total characters, usually Brand Adjective Adjective Noun, use appropriate keywords**	**Required**
Description	The product long description **Up to 5,000 total characters, include relevant keywords**	**Required**
Price	The product price **Two decimal point format without $ sign**	**Required**
Merchant Category	The category the product belongs to in the merchant's internal system **Do not name it non-category names like: On Sale, Misc., Accessories, New Arrivals - be specific such as: Women's Shoes > Sandals or Ergonomic Office Chairs**	**Required**
URL	The URL to the product details page on your site **Must begin with http:// or https:// and contain same domain name**	**Required**
Image URL	The URL to a photo of the product **Must begin with http:// or https:// and end with .jpg, .tif, .png or .gif**	**Required**
Manufacturer	The manufacturer name of the product	**Required**
Manufacturer Part Number	Product part number from the Manufacturer	**Required**
Brand	The brand name of the product	**Required**
Keywords	Brief, relevant keywords or search terms for the product separated by commas **Up to 10 keywords**	Suggested
Shipping Price	The shipping price of the product **Generally shown is lowest price for ground shipping**	Suggested
Quantity	The quantity of products in stock	Suggested
Weight	The shipping weight of the product	Suggested
Condition	The condition of the product **New, Used, Refurbished, etc.**	Suggested
UPC	The UPC of the product **Must be 12-14 digits**	Suggested
Sale Price	The marked-down sale price of the product	Suggested

MyDataFeeds XML file format

The MyDataFeeds XML file format is ideal for platform integrations, although it can be used for a single store import. It is also the preferred format for large product set imports for stores with tens of thousands of SKUs, since it has a built-in paging mechanism to make the import process more efficient.

There are three XML node sections to this file's schema, **Fields**, **Products** and **Paging**; each is discussed below.

Fields node

The **Fields** node section lets our system know what product attributes to expect for your import. This node contains multiple child **Field** nodes (up to 219), each representing a product attribute to be imported from your system. Each Field node must contain a **name** attribute that contains the field name as a value.

For example:

```
<Fields>
    <Field name="UniqueID" />
    <Field name="Name" />
    <Field name="Description" />
    <Field name="Price" />
    More Field nodes...
</Fields>
```

Products node

The **Products** node section lets our system know what product attributes to expect for your import. This node contains multiple child **Product** nodes, each representing a product in your system. Each **Product** node must contain child nodes named according to the **Fields** node section, all fields from the **Fields** node section must be present in each **Product** node and the content within each product field node must be contained in <![CDATA[ ]]> to avoid parsing errors caused by HTML tags within the content.

For example:

```
<Products>
    <Product>
        <UniqueID><![CDATA[6816916]]></UniqueID>
        <Name><![CDATA[Apple iPod Video 30GB White 5.5 GEN]]></Name>
        <Description><![CDATA[With 5.5 generation iPods all your music at
        your fingertips you may never want to stop listening...]]></Description>
        <Price><![CDATA[237.99]]></Price>
        More fields...
    </Product>
    More Product nodes...
</Products>
```

Paging node (optional section)
The **Paging** node section allows our system to relay paging calls to your system, providing an efficient mechanism for large product set imports and eliminating server timeouts on large XML files.

IMPORTANT NOTE:

The **Paging** node contains three child nodes:
- **Start** – This is the starting index of the first product to be returned
- **Count** - The number of products to return
- **Total** – The total products to be returned for this import

You must populate each of these nodes as calls are made to your system.

For example:

```
<Paging>
      <Start>1</Start>
      <Count>100</Count>
      <Total>1000</Total>
</Paging>
```

Our system will make calls to your system via HTTP and will pass query string parameters that indicate which product page to return.

The HTTP call will be similar to the following:
http://www.MyStore.com/MyWebPage?start=1&count=100

To retrieve the second page of a 1000 products file the call will be similar to the following:
http://www.MyStore.com/MyWebPage?start=101&count=100

Suggestions
If you have over 25,000 items, the ideal way to import is by FTP, keeping in mind that:
- CSV (comma delimited) or TXT (tab delimited) is better over XML
- Select our FTP location for improved connectivity
- ZIP the Source File for a faster upload time