How To Get Started with MWS
IMPORTANT
Before starting any development activity, make sure that you have gone through the Registration Process Guide. The processes explained in the guide will help you with all pre-requisites before getting started with MWS.

Acronym:
Amazon MWS: Amazon Marketplace Web Service

Contents
Introduction..................................................................................................................................................................................................................................................................................................................3
1. Using Client Libraries..................................................................................................................................................................................................................................................4
2. Checking correctness of Feed Submission..................................................................................................................................................................................................................................................5
3. Creating Listings on Amazon..................................................................................................................................................................................................................................................................................................................................................................................7
3.1 How to match products with already existing catalogue? ..................................................................................................................................................................................................................................................................................................7
3.2 How to create new listings? ..................................................................................................................................................................................................................................................................................................................................................................10
3.2.1 Required XML feeds (Add a new product) ..................................................................................................................................................................................................................................................................................................................................................................10
3.2.2 Optional XML feed..................................................................................................................................................................................................................................................................................................................................................................10
3.3 How to submit partial updates to your listings? ..........................................................................................................................................................................................................................................................................................................................................................11
3.4 How to update price and inventory? ..........................................................................................................................................................................................................................................................................................................................................................11
4. Surface Errors..................................................................................................................................................................................................................................................................................................................................................................................12
5. Manage barcodes exception ..........................................................................................................................................................................................................................................................................................................................................................14
6. Check image quality..................................................................................................................................................................................................................................................................................................................................................................................18
7. Manage Variations..................................................................................................................................................................................................................................................................................................................................................................................19
8. Drive data quality..................................................................................................................................................................................................................................................................................................................................................................................23
9. Reports Generation..................................................................................................................................................................................................................................................................................................................................................................................27
10. How to test submission of queries on MWS? ..........................................................................................................................................................................................................................................................................................................................................................27
11. Throttling..................................................................................................................................................................................................................................................................................................................................................................................27
11.1 Recommended Synchronization Timing..........................................................................................................................................................................................................................................................................................................................................................27
Introduction

To start with the automation process with Amazon, below are the set of operations that we recommend any Developer to look at:

- Upload product listings
- Inventory and Price updates
- Reports Management

The MWS APIs have all the recommended functions to allow a Developer to automate all of the above operations. This guide has the necessary steps on how to get started with MWS integration along with the best practices a Developer should adopt when using the Amazon MWS APIs. This guide also expects the Developer to have basic knowledge of MWS. More information on MWS can be found here.

To begin with MWS, the pre-requisites for a Developer is to have a seller account and to use the same to register for a developer account. This whole process is explained in the MWS Registration Guide.

This guide assumes a basic knowledge of MWS as well as the main differences between the two file types (Amazon Feeds) used to push data to Amazon systems:

- **Flat Files**: A tab delimited text file which is both category and marketplace specific
  Documentation here
- **XML files**: One product feed for all the categories for all the Marketplaces. Five XML feeds (Product, Inventory, Price, Stock, Relationship) to create a Product
  Documentation here

In this document, unless explicitly stated, we refer to XML technology.
1. Using Client Libraries

A Developer can either use Amazon’s own client library or create his own client library as well for use with Amazon MWS. The Amazon client library contains codes for doing many common tasks when working with MWS. This allows the Developer to save time as well as ensure that all the requests and responses are well formatted. If the Developer decides to create his own client library, he has to make sure that the code is in the format expected by Amazon. To have more information on Amazon’s client libraries as well as how to create your own library, please follow this [link](#). Below is a list of some MWS APIs and links to their respective client libraries:

- Feeds API
- Reports API
- Products API
- Fulfillment Inventory API
2. Checking correctness of Feed Submission

As a Developer, you will be using the MWS Feeds API to submit the Sellers’ data to Amazon. To submit any feed to this API, you can use either XML or Flat Files. You can have more information on the MWS Feeds API here.

XMLs must be validated against the XSDs.

You should always validate your XMLs against Amazon XSDs before submitting them to Amazon. You can use a public parser (e.g. DOM, SAX) to verify that your XMLs match against the schemas.

Keep also in mind that this validation is a necessary but not sufficient condition to guarantee the correctness of your XMLs. Different normative across countries could lead to differences in listing characteristics in terms of mandatory fields and values admitted within the same field.

You should double check:

• if a field is mandatory in the sheet Data Definition of the category specific flat file for that marketplace.
• the valid values for a field in the sheet Valid Values of the Flat File of the category specific FF for that marketplace.
Example

You need to list a Product in the “Sports” category on the UK Marketplace using XML.

By looking at the **Sports XSD** you can see the valid values for the tag `ProductType`

```xml
<xsd:element name="ProductType" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="SportingGoods"/>
      <xsd:enumeration value="GolfClubHybrid"/>
      <xsd:enumeration value="GolfClubIron"/>
      <xsd:enumeration value="GolfClubPutter"/>
      <xsd:enumeration value="GolfClubWedge"/>
      <xsd:enumeration value="GolfClubWood"/>
      <xsd:enumeration value="GolfClubs"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

This list contains all possible values for the `<ProductType>` tag on the “Sports” category worldwide. It is possible that some of these values are not available on certain marketplaces.

To confirm what ProductType values you can use in the UK Sports category, you need to look up the ProductType column in the sheet Valid Values of the UK **Sports Category File**.

In this column you will read:

```
<table>
<thead>
<tr>
<th>Product Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SportingGoods</td>
</tr>
</tbody>
</table>
```

There is only one `<ProductType>` value admitted for the Sports category in the UK Marketplace.
That would mean that if you try create a product via XML in the Sports Category of the UK Marketplace by using a value not present in this list 

\[ \text{<ProductType>GolfClubs<ProductType>} \]

You will get an error even if this value is present in the XSD.

GolfClubs is a valid value for the <ProductType> tag globally, but not in UK.

You should map in your database all the Valid Values present in the XSDs, but then surface to your application interface only the valid ones according to the specific Marketplace and Category where your Seller wants to create their listing.

3. Creating Listings on Amazon

Creation of listings on Amazon can be done using either Flat Files or XML feeds. When a seller starts with the listings process, he can either match the product against an existing product in the Amazon catalogue or create new listings if the product is not already present.

Developing high quality Listings Creation functionality requires knowledge of a number of key aspects, both from the business and the technical side of Amazon.

Amazon does not allow duplicate products on the catalogue.

Before starting with the listing process, it should be noted that Amazon does not allow the creation of duplicate products that already exists in its catalogue. Therefore, once a product is created for the first time on the Amazon catalogue, all other sellers who would like to create listings to sell the same product should engage the existing product. Otherwise, their load as a new product goes waste. In some rare cases, it may happen that some product attributes that exist on the Amazon catalogue may have some errors. If the seller is sure of the correctness of his information about the product and sees a contradiction in the Amazon catalogue, the case can be reported to seller support who can drive a change to the database.

3.1 How to match products with already existing catalogue?

As a best practice, it is always recommended that before proceeding to list on Amazon, please check if the product is already present in the catalogue. This will prevent duplication and avoid wasting time in the
process of loading new product attributes. The MWS Products API offers all the required operations for a Seller to check if a product already exists in the Amazon catalogue. Please read more on the MWS Products API [here](#).

Your application should implement a logic to manage smoothly product mismatching edge cases.

When a Seller creates a new listing for a product already present on Amazon, but provides a key attribute of that product different than the one already on the Amazon catalogue, you will get an error code 8541 or 8542 in the processing report. These errors are also known as matching errors.

Your application should be able to parse the processing report, detecting matching errors and propose an interface to the Seller, highlighting these conflicts of information. You can read more on how to surface processing errors to your sellers in Section 4 of this document.

You can also consider implementing an additional functionality to highlight in advance to the Seller any potential matching conflict by retrieving all the information of an existing product through the Products API.

We do not recommend you to force the matching by retrieving the ASIN code from the Product API and creating that ASIN as `<StandardProductID>`. By doing this, it could ultimately create in the Seller's catalogue a different product than the expected one (e.g. when the Seller provides a valid EAN code, but for a different product)

In this case we suggest that you surface the product you are about to match and let your user confirm that it is exactly the product for which they want to create a listing. If the Products API returns more than one ASIN for a single barcode, we suggest you to surface the item with the lowest `<rank>` value.

Once a Seller has validated that the ASIN is a good match, in order to create a listing for the Seller based on this ASIN, populate the SKU and ASIN in the XML product feed and upload same using the Feeds API. Below is an example of how to populate the product feed to create a listing based on a match.
By using the ASIN instead of EAN, the listing will be created out of a product that already exists. All the other attributes that you send will be contributions to what already exists in Amazon catalogue.

**Seller Experience:**

The Seller submits a new listing. If the products matches to an existing product in the catalogue but one or more of the attribute values does not match to those present in the catalogue, we return errors. Your application can surface these errors and show the value present at Amazon.

The application should give the Seller the possibility to fix the issues and re-submit the product.

Should the Seller be sure of the correctness of their data, conflicting with the data on Amazon, your application should suggest that the Seller escalate the case to Seller support.
3.2 How to create new listings?

Once you are sure that the product the Seller wants to list does not exist in the Amazon catalogue, you can let the seller create a new product on Amazon. For this, you will need to submit the below feeds through the Feeds API.

3.2.1 Required XML feeds (Add a new product)

i. Products Feed – To create or update the product catalogue (101)
ii. Images Feed – To create or update the images (104)
iii. Stock Feed – To update the inventory (106)
iv. Price Feed – To update the prices (105)

3.2.2 Optional XML feed

i. Relationship Feed - To define the Parent/Child relationships in the case of products with variations (size, color, etc.) (102)

(You can find a video tutorial on how to create variations [here](#) and you can find additional information on this [here](#). You can have more information on Variation creation in section 7 of this document.

Important Notes:

a. You should only proceed to upload the other feeds after successful upload of the Products Feed.

b. Please follow the Amazon style guides to make sure that your solution is designed in a way to upload compliant data that would lead to successfully created listings. The style guides for different product categories can be downloaded from [here](#).

c. Please note that Amazon uses Browse Nodes to categorize products that makes it easier for customers to find listings on the marketplace. Browse Node is a mandatory parameter required in creating new products. You can use the Browse Tree Guide to see how to assign browse nodes. For more information on browse nodes, please follow this [link](#). You can also identify browse nodes using [this](#) Amazon tool.

d. After a successful upload, please advise the seller to check the ‘Inventory > Manage Inventory’ tab on Seller Central to see how the data uploaded is reflected on Amazon.
3.3 How to submit partial updates to your listings?

Once your product has been successfully uploaded on Amazon, you can add or partially edit certain attributes such as item weight, description, etc. This can be done by submitting a Product Feed using the ‘SubmitFeed’ operation from the Feeds API. The mandatory elements to submit for a partial update are OperationType as ‘PartialUpdate’, ‘SKU’ and ‘Title’. You can keep adding any other elements as per XSDs based on the values you want in your product. Below is an example of a PartialUpdate operation.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<AmazonEnvelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="amzn-envelope.xsd">
  <Header>
    <DocumentVersion>1.01</DocumentVersion>
  </Header>
  <Message>
    <MessageID>1</MessageID>
    <OperationType>PartialUpdate</OperationType>
    <Product>
      <SKU>SKU_value</SKU>
      <Title>Title of the product</Title>
      <ItemDimensions>
        <Length unitOfMeasure="IN">25.0</Length>
        <Width unitOfMeasure="IN">12.0</Width>
        <Height unitOfMeasure="IN">27.0</Height>
      </ItemDimensions>
      <ItemWeight unitOfMeasure="KG">8.00</ItemWeight>
    </ProductData>
    <ClassificationData>
      <HeelHeight unitOfMeasure="IN">1</HeelHeight>
    </ClassificationData>
  </Product>
</Message>
</AmazonEnvelope>
```

3.4 How to update price and inventory?

After the creation of the listing logic to allow Sellers to upload their products on Amazon, the next step is to add an offer to the listing. The Inventory and Price feeds can be used respectively to update the stocks and the prices of products on Amazon. You can have access to these feeds by following the link [here](#).

It is highly recommended that when you are developing the solution to allow the Sellers to update their product prices on Amazon that you recommend the Sellers to always set a Minimum and a Maximum Price for their product. These numbers set by Sellers should be reasonable. This is a very important step that ensures that the price range for a Seller’s product is always acceptable and this reduces risks that can happen due to unintentional wrong pricing updates that may lead to bad customer experience and also serious losses for Sellers. As a best practice, you should have in your software a way to prompt the
Seller to input these Minimum and Maximum prices if they are empty and submit them to Amazon through the Pricing Feed. From the Pricing Feed XSD, as seen below, the Minimum Price should be submitted using the ‘MinimumSellerAllowedPrice’ element and Maximum Price using the ‘MaximumSellerAllowedPrice’ element. Even though these elements are not mandatory to submit, it is highly recommended that you cater for them in your solution.

```xml
<xsd:include schemaLocation="a2zn-base.xsd"/>
<xsd:element name="Price">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="SKU"/>
      <xsd:element name="StandardPrice" type="OverrideCurrencyAmount" minOccurs="0"/>
      <xsd:element name="StandardPricePoints" type="PositiveInteger" minOccurs="0"/>
      <xsd:element name="BusinessPrice" type="BasePriceCurrencyAmount" minOccurs="0"/>
      <xsd:element name="MinimumPriceType" type="QuantityPriceTypes" minOccurs="0"/>
      <xsd:element name="MaximumPrice" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

4. Surface Errors

Sellers should be able to fix autonomously as many listing creation errors as possible.

You should design your application in order to always check the quality of XMLs submitted to Amazon, especially the XML Product Feed. Errors in the Product Feed can lead to subsequent errors in other feeds, like the impossibility of associating an image to a product since the listing creation failed first.

You should always download and check the FeedProcessingReport, through the `GetFeedSubmissionResult` operation of the Feeds API. This report will provide the list of errors and error explanation on a SKU level.

Visit this page for details and a diagram to help your design.
Your application should always map the Listing Creation errors reported in the FeedProcessingReport and surface them to the Seller interface.

Below is a procedure on how to check the status of feeds that have been uploaded:

1. When uploading a feed using the SubmitFeed operation from the Feeds API, store the ‘FeedSubmissionId’ that you will get back in the response.
2. Use the ‘FeedSubmissionId’ as an input to the GetFeedSubmissionList operation to get the ‘FeedProcessingStatus’. Monitor the ‘FeedProcessingStatus’ and once it changes to “_DONE_”, you are ready to see the results.
3. Use the ‘FeedSubmissionId’ as input to the ‘GetFeedSubmissionResult’ operation to get the processing results.

You can have further information on these operations [here](#).

**Seller Experience**

If an error occurred while creating Listings, your application clearly shows the error message on the interface as well as the way to fix it. Both of these pieces of information are present in the FeedProcessingReport.

The Seller can take action, fix data and resubmit their request. In this [guide](#) you can find information on the most common feed errors.
5. Manage barcodes exception

In Order to create a new Product on Amazon, Sellers need to provide a valid EAN as StandardProductID.

You need to take into consideration three cases:

Case 1: The Seller uses the barcodes → The Seller **must** identify each product they want to list with the barcode as StandardProductID.

Case 2: The Seller sells products manufactured without Barcodes:
   a. The Seller needs to request an Amazon **Barcode exemption** for the given brand (e.g. **BRAND_ABC**).
   b. If Amazon accepts the request, the Seller will receive an email that states: "Your exemption has been accepted, use **PARAMETER_1** to identify your products instead of the StandardProductID.
   c. If Amazon does not accept the request the Seller will not be able to list products without StandardProductID.

Case 3: The Seller is the producer/brand owner:
   a. The Seller needs to register their brand to the Amazon **Brand Catalogue** (e.g. **BRAND_XYZ**).
   b. If Amazon accepts the request, the Seller will receive an email that states: "Your brand registration has been accepted, use **PARAMETER_2** to identify your products instead of the StandardProductID.
   c. If Amazon does not accept the request the Seller will not be anyhow able to list products without StandardProductID.

Case 2 and 3 Sellers are entitled to list their products without StandardProductID, therefore your application should be able to manage smoothly these cases, and accept XMLs with a slightly different structure.

In case 2.c or 3.c, your application should be able to filter these products and exclude them from the upload to Amazon.
Before uploading the Product to Amazon, your application should manage the three cases as follows:

Case 1:
Your application should always require to input the `<StandardProductID>` and check if it meets the Barcode standards.

Also consider whether to integrate an API to verify that the barcode has been properly issued by the BarCode Authority. Amazon does not provide such a functionality; you can use public Internet resources.

If there is an error, give Sellers the opportunity to correct the barcode and try again to submit the product.

Case 2:
The Seller sells products without EAN Codes and they have an exception approved.

The Seller will send you the product with:

- `<PARAMETER_1>` populated and no `<StandardProductID>` tag
- `<Brand>` populated with the Brand they specified during the Barcode exemption registration process

See the following snapshot where PARAMETER_1 is the `<MfrPartNumber>` and `<Brand>` is `BRAND_ABC`

```xml
<Message>
  <MessageID>180</MessageID>
  <OperationType>Update</OperationType>
  <Product>
    <SKU>0012946</SKU>
    <Condition>
      <ConditionType>New</ConditionType>
    </Condition>
    <DescriptionData>
      <Title>TITLE</Title>
      <Brand>BRAND_ABC</Brand>
      <Description>Product ...</Description>
      <MfrPartNumber>0012946</MfrPartNumber>
      <SearchTerms>DONNA</SearchTerms>
      <SearchTerms>CASUAL DONNA</SearchTerms>
      <SearchTerms>IGI&CO</SearchTerms>
      <SearchTerms>57712</SearchTerms>
      <SearchTerms>Scarpe donna visone</SearchTerms>
      <ItemType>shoes</ItemType>
      <RecommendedBrowseNode>700832031</RecommendedBrowseNode>
    </DescriptionData>
  </Product>
</Message>
```
Case 3: The Seller does not have the Barcodes since they are the producer/brand owner

The Seller will send you the product with

a. `<PARAMETER_2>` populated and no `<StandardProductID>` tag
b. `<Brand>` populated with the Brand they required during the Barcode exemption registration process
c. You need to add the tag `<RegisteredParameter>` with value `PrivateLabel`

See the following snapshot where PARAMETER_2 is the `<MFRPartNumber>` and `<Brand>` is `BRAND_XYZ`

```xml
<Message>
  <MessageID>2</MessageID>
  <OperationType>Update</OperationType>
  <Product>
    <SKU>609119_19447</SKU>
    <Condition>
      <ConditionType>New</ConditionType>
    </Condition>
    <DescriptionData>
      <Title>Poster Audrey</Title>
      <Brand>BRAND_XYZ</Brand>
      <Designer>Famous When Dead</Designer>
      <Description>…..Description</Description>
      <Manufacturer>POSTERLOUNGE</Manufacturer>
      <MfrPartNumber>609119_19447</MfrPartNumber>
      <IsGiftWrapAvailable>false</IsGiftWrapAvailable>
      <IsGiftMessageAvailable>false</IsGiftMessageAvailable>
      <RecommendedBrowseNode>4606042031</RecommendedBrowseNode>
      <RecommendedBrowseNode>731704031</RecommendedBrowseNode>
    </DescriptionData>
    <ProductData>
      <Home>
        <ProductType>
          <FurnitureAndDecor>
            <Material>Stampa su tela</Material>
          </FurnitureAndDecor>
        </ProductType>
      </Home>
    </ProductData>
    <RegisteredParameter>PrivateLabel</RegisteredParameter>
  </Product>
</Message>
```

To manage Case 2 and Case 3, your application should offer a "Barcode exemption/Brand Register" configuration capability to let the Seller configure for which brands they received EAN exemption/GCID. According to the configuration, your application should enable/disable the related mandatory/optional fields in the user Interface to create proper XMLs.
Seller Experience

Case 1:
The Seller should input in your interface the Barcode and get an error whenever it does not meet the standard or it is not valid.

Case 2:
The Seller should be able to configure in your application those Brands for which they have obtained Barcode exemption in the email they received from Amazon.

For these brands, the Seller will use PARAMETER_1 to identify the Product (not the StandardProductID).

Case 3:
The Seller should be able to configure in your application those Brands for which they have obtained GCID as they received in the email they received from Amazon.

For these brands, the Seller will use PARAMETER_2 to identify the Product (not the StandardProductID).
6. Check image quality

Sellers must provide only images that meet the Amazon quality criteria.

You should always check the quality of the images provided by the Sellers. We suggest you to implement a preliminary control of the images in your application to check the technical parameters (File type, Size, Dimension Resolution) and present an error when quality is not met, before submitting the Image feed to Amazon. Visit the style guide for details.

Your application should be capable of detecting these records with defective images and excluding the related SKUs also from the other XML feeds until the Seller fixes the issue.

Images must be publicly accessible on the Internet and your solution must feed Amazon systems with only their URLs.

**Seller Experience**

If the Seller creates a listing with an image that does not meet the Amazon quality criteria, they will receive an error on the interface that explains what criteria is not met and an invite to change the image and re-submit the request.
7. Manage Variations

Products within the same family must be consistent

In case of Variations (Relationship between Parent and Children element), the parent SKU must be considered only a logical element for grouping children, it should not have specific attributes like size, color, price, or quantity and is not buyable.

In order to establish a relationship of variation between two products you should:

- Define in the Product Feed which SKUs are parents and which are children, populating the <parentage> tag
- Define in the Relationship Feed which SKU is child of which parent SKU

An important element to create variations is <VariationTheme> that describes the parameter(s) by which the parent product may vary, mostly used to categorize clothing varying by Size and Color. Check the Flat File for the right valid values of variation theme, according to the Marketplace where you want to list.

<table>
<thead>
<tr>
<th>Variation theme</th>
<th>Product Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>ColorName</td>
</tr>
<tr>
<td>Size</td>
<td>SizeName</td>
</tr>
<tr>
<td>ColorSize or SizeColor</td>
<td>ColorName and Size Name</td>
</tr>
</tbody>
</table>

The value input for <VariationTheme> must be the same for both the parent and child SKUs.

**NB:** If there is an error in one of the SKUs of the family, it may prevent the creation of the entire family.
Example of a Parent Clothing Product Feed

```xml
<Message>
  <MessageID>1</MessageID>
  <OperationType>Update</OperationType>
  <Product>
    <SKU>83P60021R05835</SKU>
    <DescriptionData>
      <Title>Oltre: Trench Woman. </Title>
      <Brand>Oltre</Brand>
      <Description>Short Trench....</Description>
      <BulletPoint>Zip...</BulletPoint>
      <SearchTerms>Trench</SearchTerms>
      <RecommendedBrowseNode>2893123031</RecommendedBrowseNode>
    </DescriptionData>
    <ProductData>
      <Clothing>
        <VariationData>
          <Parentage>parent</Parentage>
          <VariationTheme>SizeColor</VariationTheme>
        </VariationData>
        <ClassificationData>
          <ClothingType>Outerwear</ClothingType>
          <Department>Woman</Department>
          <StyleKeywords>Apparel</StyleKeywords>
          <MaterialComposition>100% poliesther</MaterialComposition>
          <OuterMaterial>syntetic</OuterMaterial>
          <OccasionAndLifestyle>Casual</OccasionAndLifestyle>
          <Season>Spring/Summer 2016</Season>
          <IsAdultProduct>false</IsAdultProduct>
          <CollectionName>Pring/Summer 2016</CollectionName>
        </ClassificationData>
      </Clothing>
    </ProductData>
  </Product>
</Message>
```
Example of a Child Clothing Product Feed

<Message>
  <MessageID>2</MessageID>
  <OperationType>Update</OperationType>
  <Product>
    <SKU>83P60021R05835_02_3</SKU>
    <StandardProductID>
      <Type>EAN</Type>
      <Value>8300736338883</Value>
    </StandardProductID>
    <DescriptionData>
      <Title>Oltre: Trench Woman. Black, Size 42</Title>
      <Brand>Oltre</Brand>
      <Description>Trench corto doppio petto. Taschine chiuse con bottoni e polsini regolabili da cinturino. Fodera interna a righe.</Description>
      <BulletPoint>Chiusura doppiopetto con bottoni in tinta</BulletPoint>
      <SearchTerms>Abbigliamento donna</SearchTerms>
      <RecommendedBrowseNode>2893123031</RecommendedBrowseNode>
    </DescriptionData>
    <ProductData>
      <Clothing>
        <VariationData>
          <Parentage>child</Parentage>
          <Size>42</Size>
          <Color>beige</Color>
          <VariationTheme>SizeColor</VariationTheme>
        </VariationData>
        <ClassificationData>
          <ClothingType>Outerwear</ClothingType>
          <Department>Woman</Department>
          <StyleKeywords>Apparel</StyleKeywords>
          <ColorMap>beige</ColorMap>
          <MaterialComposition>100% poliesther</MaterialComposition>
          <OuterMaterial> sintetico</OuterMaterial>
          <OccasionAndLifestyle>Casual</OccasionAndLifestyle>
          <Season>Spring/Summer 2016</Season>
          <SizeMap>42</SizeMap>
          <CollectionName>Spring/Summer 2016</CollectionName>
        </ClassificationData>
      </Clothing>
    </ProductData>
  </Product>
</Message>
Example of their relationship feed.

Note that you should create one message for each child SKU and not one message for the entire family.

```xml
<Message>
  <MessageID>1</MessageID>
  <OperationType>Update</OperationType>
  <Relationship>
    <ParentSKU>83P60021R05835</ParentSKU>
    <Relation>
      <SKU>83P60021R05835_02_3</SKU>
      <Type>Variation</Type>
    </Relation>
  </Relationship>
</Message>

<Message>
  <MessageID>2</MessageID>
  <OperationType>Update</OperationType>
  <Relationship>
    <ParentSKU>83P60021R05835</ParentSKU>
    <Relation>
      <SKU>83P60021R05835_02_4</SKU>
      <Type>Variation</Type>
    </Relation>
  </Relationship>
</Message>
```
8. Drive data quality

High quality categorization data will benefit the product search ability

Your application should permit your Sellers to input as much information as possible to improve searchability and guarantee the best search experience for the Customer.

*Browse nodes & Refinements:*

A Browse node is a code that identifies to which node of the Amazon category tree a product is associated.

Your application should map each product to the leaf node of the Amazon catalogue tree in order to guarantee the most accurate association.

We suggest you to implement a Browse node association functionality by using the XML [BrowseTreeReport](#) downloadable via Reports API. In this way, you will be able to surface to the interface of your application the Categories and Subcategories present in the report and guide your user to Select automatically the right Browse nodes.

Alternatively, you can consider mapping the browse nodes in your application by importing the report in Excel format, known as the [Browse Tree Guide](#).

For each Browse node, Amazon suggests certain Refinements, specific category attributes that will furthermore increase the searchability of the product.

Your application should be capable to surface these fields depending on the selected Browse node and invite your users to populate those attributes.
**Titles**

It is highly recommended that product titles follow the Amazon [style guidelines](#) according to your product category. You should implement a logic in your application that creates and validates titles by concatenating the user input.

**Bullet points**

A bullet point is a brief descriptive free text, called out via a bullet point, regarding a specific aspect of the product.

You should give Sellers the possibility to include up to five bullet points into your interface to let them specify this information. Note that according to the ProductType certain bullet points can be added automatically.

**Search Terms**

A Search Term is a word or phrase that best describes the product.

This will help Amazon locate the product when customers perform searches on the site.

You should give Sellers the possibility to include up to five search terms into your interface to let your user specify this information.

It is highly recommended that product titles follow the Amazon [style guidelines](#) according to your product category.
Seller Experience

Following an example of a Seller who wants to list a bottle of Champagne on the UK Marketplace.

The application permits the Seller to select in sequence the following categories:

Grocery/Beer
   Wine & Spirits/Sparkling Wine & Champagne
      Champagne and the application

The application shows and associates the browse node 359889031.

The uses inserts:
   SKU
   Barcode
   Title
   Brand
   Description
   2 out of the 5 possible Search Terms
   1 bullet point.

But can also select two refinements from dropdown menus:
   Color (the users selects “white wine”)
   Sweetness (the user selects "extra brut")
On confirmation the Product XML should look like the following:

```xml
<Message>
  <MessageID>1</MessageID>
  <OperationType>Update</OperationType>
  <Product>
    <SKU>SKU_XXXX</SKU>
    <StandardProductID>
      <Type>EAN</Type>
      <Value>XXXXXXXXXXXXX</Value>
    </StandardProductID>
    <LaunchDate>2016-05-02T00:00:00</LaunchDate>
    <Condition>
      <ConditionType>New</ConditionType>
    </Condition>
    <DescriptionData>
      <Title>Champagne Imperial Brut Gift Boxed 75 cl</Title>
      <Brand>Imperial Brut</Brand>
      <Description>Founded in 1743, Imperial....</Description>
      <BulletPoint>A fresh crisp palate reveals white-fleshed fruits...</BulletPoint>
      <Manufacturer>Imperial Brut</Manufacturer>
      <SearchTerms>Wine</SearchTerms>
      <SearchTerms>Champagne</SearchTerms>
      <RecommendedBrowseNode>359892031</RecommendedBrowseNode>
    </DescriptionData>
    <ProductData>
      <FoodAndBeverages>
        <ProductType>
          <Food>
            <BodyDescription>white wine</BodyDescription>
            <TasteDescription>extra brut</TasteDescription>
            <DisplayVolume unitOfMeasure="liter">0.75</DisplayVolume>
          </Food>
        </ProductType>
      </FoodAndBeverages>
    </ProductData>
  </Product>
</Message>
```
9. Reports Generation
The Reports API section of the Amazon Marketplace Web Service (Amazon MWS) API lets you request various reports that are available for Sellers. To know more about report API, click here.

10. How to test submission of queries on MWS?
Amazon MWS Scratchpad is an application that enables you to easily submit query requests using any Amazon MWS API section. You can indicate the API section, the operation, and any required and optional parameters from the Amazon MWS Scratchpad user interface. The Amazon MWS Scratchpad is a very useful tool that can become very handy during your development process. It helps you simulate the requests and responses from Amazon APIs. You can have more information on the MWS Scratchpad here.

11. Throttling
Throttling is the process of limiting the number of requests you can submit to a given operation in a given amount of time. When starting your integration to Amazon MWS APIs, you have to pay special attention to throttling. There are throttling limits for each API. Please note the limits of calls you can send over time and set up your development in agreement to these limits. You can have more information on throttling limits of MWS here.

11.1 Recommended Synchronization Timing
a. Synchronize the entire catalogue (Product Feed 101, 104 Image) - Once a day
b. Synchronize quantity (stock) (Feed 106, Inventory) - Every 15 minutes
c. Price synchronization (Feed 105) – This depends on the refresh rate of your expected re-pricing strategy. If implemented a very dynamic logic (such as in case of use of the Subscription API), the update of prices could be every 5 minutes, otherwise if you do not plan to update the prices frequently, you can dilute the load at 1 time a day.