SOAP to Amazon MWS Migration Guide

Contents

SOAP to Amazon MWS Migration Guide 1

Migrating from SOAP to Amazon Marketplace Web Service (Amazon MWS) 2

- The Good News 2
- Amazon MWS Registration 2
- Authentication 3
- Endpoints 3
- Understanding Throttling in Amazon MWS 3
- Creating an Amazon MWS Request 5
- Mapping SOAP Functions to Amazon MWS Operations 5

Posting Documents Such as Inventory Feeds Using the Amazon MWS Feeds API Section 6

- Amazon MWS Feed Submission Process 6
- Document Types and MWS Feed Types 8
- Managing Inventory with the Amazon MWS Feeds API Section 8

Managing Orders with the Amazon MWS Feeds API Section 9

- Order Fulfillment 9
- Order Information 9

Creating and Retrieving Reports with Amazon MWS 10

- The Reports Process 14

Migrating from SOAP to Amazon Marketplace Web Service (Amazon MWS)

Amazon Marketplace Web Service (Amazon MWS) is Amazon's newest API for sellers. It offers extensive documentation and support, and client libraries in Java, C#, and PHP for a fast and easy way to integrate with Amazon. Amazon MWS supports seller API operations that are broader in scope and functionality than any APIs Amazon has previously offered sellers.

The Good News

Moving your code from using SOAP to Amazon MWS might not be as bad as you think. You'll be able to do more, and we are constantly adding new features. Many of the message types you used in SOAP are the same as the feed types in Amazon MWS. You won't have to rely on a WSDL to find out what operations are available, and you can still get your reports in XML.

Each Amazon MWS API section has its own client library that contains code for doing many common tasks when working with Amazon MWS. By using the code in the Amazon MWS client libraries, you save time and you know the request you send is correctly formatted.

The first two tasks you must accomplish to use Amazon MWS are to register for Amazon MWS and to identify the correct endpoint for your marketplace. From there you need to construct a valid request string, then sign that request string with your Secret Key. Finally, you create a URL with all the information required and submit it to the endpoint. These steps will be discussed in depth in this document.

Amazon MWS Registration

To use Amazon MWS, you need to register. You can register at these locations, depending on your marketplace:

CA: *http://developer.amazonservices.ca*

DE: *http://developer.amazonservices.de*

FR: *http://developer.amazonservices.fr*

JP: *http://developer.amazonservices.jp*

UK: *http://developer.amazonservices.co.uk*

US: *http://developer.amazonservices.com*

To register for Amazon MWS, you must have an Amazon MWS-eligible seller account. These accounts can include:

- A non-individual seller account
- Amazon Webstore account
- Amazon Product Ads account

- Checkout by Amazon account

You simply enter your seller account credentials and Amazon MWS returns several important values that you will use to authenticate your requests to Amazon MWS. For more information on registering for Amazon MWS, see the *Amazon MWS Developer Guide*.

Note: You must register for Amazon MWS in the same marketplace as your seller account. For example, an EU seller must register for Amazon MWS using one of the EU URLs. An EU seller cannot register for Amazon MWS using the US URL.

Authentication

Amazon MWS uses a different authentication scheme than SOAP. Instead of using a seller e-mail address and password, you pass several values to Amazon MWS that are used for authentication. These values are provided to you when you register for Amazon MWS.

Authentication occurs when you submit a request that includes a signature that you create using the Secret Key you are given when you register for Amazon MWS. How to create this signature and how to format a request is covered in the *Amazon MWS Developer Guide*.

Endpoints

To access Amazon MWS, use one of the following market-specific endpoints:

Amazon Market	MWS Endpoint
CA	https://mws.amazonservices.ca
DE	https://mws.amazonservices.de
FR	https://mws.amazonservices.fr
JP	https://mws.amazonservices.jp
UK	https://mws.amazonservices.co.uk
US	https://mws.amazonservices.com

Understanding Throttling in Amazon MWS

SOAP has “maximum request limits,” where the number of requests you can make in a given time are limited. MWS has a similar constraint, called “throttling.” To use Amazon Marketplace Web Service (Amazon MWS) effectively, you need to understand throttling. Throttling is the process of limiting the number of requests you can submit in a given amount of time. A request can be when you submit an inventory feed or when you make an order report request. Throttling protects the web service from being overwhelmed with requests and ensures all authorized developers have access to the web service.

Amazon MWS uses a variation of the leaky bucket algorithm to meter the web service and implement throttling. The algorithm is based on the analogy where a bucket has a hole in the bottom from which water leaks out at a constant rate. Water can be added to the bucket intermittently, but if too much water is added at once or if water is added at too high an average rate, the water will exceed the capacity of the bucket.

To apply this analogy to Amazon MWS, imagine that the bucket represents the maximum request quota, which is the maximum number of requests you can make at one time. The hole in the bucket represents the restore rate, which is the amount of time it takes to be able to make new requests. So, if you submit too many requests at once, then the bucket overflows and, in the case of Amazon MWS, throttling occurs. If you fill up the bucket, it takes some time before you can add more water to the bucket since the water leaks from the bucket at a steady rate. So the ability to submit more requests after you have reached the maximum request quota is governed by the restore rate, the time it takes to allow you to make new requests.

The definitions of these three values that control Amazon MWS throttling are:

• Request quota - The number of requests that you can submit at one time without throttling. The request quota decreases with each request you submit, and increases at the restore rate.

• Restore rate (also called the recovery rate) - The rate at which your request quota increases over time, up to the maximum request quota.

• Maximum request quota (also called the burst rate) - The maximum size that the request quota can reach.

While most Amazon MWS operations have a maximum request quota of 10 requests and a restore rate of one new request every minute, a few operations allow you to submit more requests initially (higher maximum request quota) but take longer to allow additional requests (a lower restore rate). For example, the `RequestReport` operation has a maximum request quota of 15, but the restore rate is one new request every two minutes.

To apply these ideas, consider this example. Say you want to use the `SubmitFeed` operation to submit 25 inventory update feeds. The `SubmitFeed` operation has a request quota of 15 and a restore rate of one new request every two minutes. If you submit all 25 feed requests at once, your requests will be throttled after 15 requests. You would then have to resubmit 10 feed requests once the request quota had been restored. Since the restore rate is one request every two minutes, it would take 20 minutes for you to be able to submit the remaining 10 feed requests. So, instead of submitting all the requests and having to resubmit the requests were throttled, you could automate your process to submit feed requests incrementally.

For example, you could submit 10 feed requests (out of your original 25 feeds), and the request quota would still have five requests left over. You could then wait 10 minutes, and the restore rate would have increased the request quota to 10 (one request every two minutes for 10 minutes gives you five new requests). You could then submit 10 more feed requests. For the remaining five feed requests, you could wait ten more minutes and then submit them. If all things go well, you would have submitted all 25 of your inventory feeds in about 20 minutes.

You should consider automating your requests and have a “back off” process where, if throttling occurs because you reached the maximum request quota or the web service experienced high traffic volumes, you could slow down the number of requests you make and resubmit requests that initially failed.

Creating an Amazon MWS Request

Amazon MWS supports query requests for calling web service actions. Query requests are simple HTTP requests, using the GET or POST method with query parameters in the URL or HTTP body, respectively. Amazon MWS requires the use of HTTPS to prevent third party eavesdropping on your communication with Amazon.

Each of the HTTP header lines must be terminated with a carriage return and a line feed. Query requests must contain an Action parameter to indicate the action to be performed. The response is an XML document.

The *Amazon MWS Developer Guide* goes into detail on how to create a valid request.

Mapping SOAP Functions to Amazon MWS Operations

The following table shows the four basic SOAP functions supported by Amazon and their equivalent actions using Amazon MWS:

Task	SOAP action	Amazon MWS action
Posting documents to Amazon.com (for example, posting a product feed containing the seller's new products)	postDocument	Use the `SubmitFeed` operation in the Feeds API section to upload a file. The operation returns a **FeedProcessingId** which you can pass to the `GetFeedSubmissionList` operation.
Getting processing status for posted documents (for example, retrieving the processing results of a posted product feed)	getDocumentProcessingStatus	Use the `GetFeedSubmissionList` operation to determine the processing status of your feed submission.
Getting documents from Amazon.com (for example, retrieving pending order reports for the last hour)	getAllPendingDocumentInfo and getDocument	Use the `RequestReport` operation in the Reports API section to request a report. With Amazon MWS, you can also use the `ManageReportSchedule` operation to schedule order reports.
Posting acknowledgements for downloaded documents (for example, acknowledging the successful download of specific order reports so they will no longer display as pending retrieval)	postDocumentDownloadAck	Use the `UpdateReportAcknowledgements` operation to acknowledge receipt of a report. You can then use the `GetReportList` operation with the **Acknowledged** parameter set to *false* so that you only retrieve report information for reports you have not acknowledged.

Posting Documents Such as Inventory Feeds Using the Amazon MWS Feeds API Section

Posting inventory feeds to Amazon MWS is similar to using the SOAP APIs. With Amazon MWS, you use the `SubmitFeed` operation to upload all feeds to Amazon MWS. The type of feed you submit, a value of the **FeedType** enumeration, determines what action is taken. Many of the message types (called a **FeedType** enumeration value in Amazon MWS) for the data you submit in a feed are the same as you currently use with SOAP.

The following table shows the action you need to take to manage your inventory, the SOAP operation you currently call, and the Amazon MWS operation you would use to accomplish the same task.

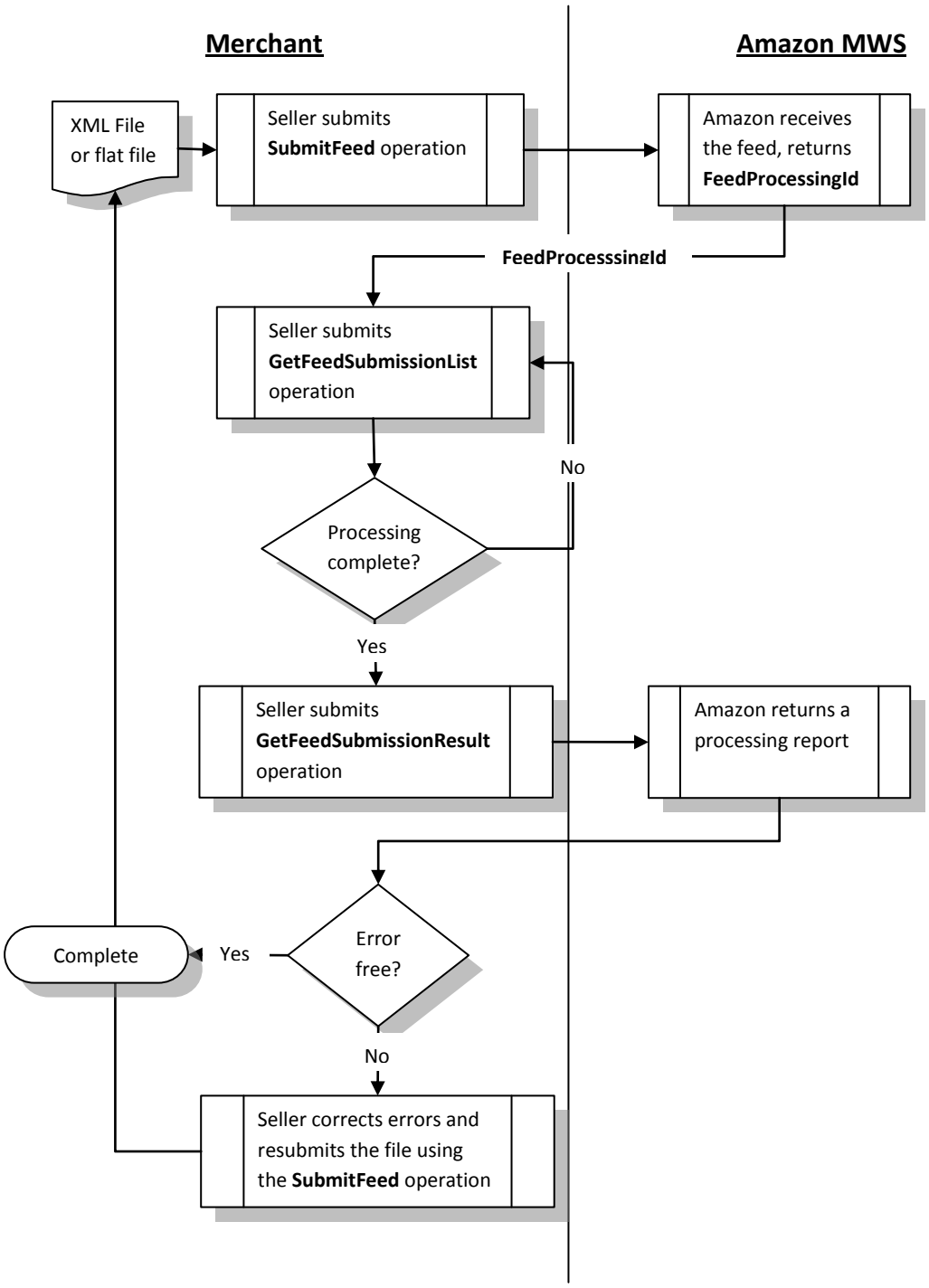
Action	SOAP operation	Amazon MWS operation
Upload Inventory Files	postDocument	`SubmitFeed`
Check Inventory Update Status	getLastNDocumentProcessingStatuses	`GetFeedSubmissionList`
Check Inventory Upload Count		`GetFeedSubmissionCount`
Retrieving Inventory Upload Error Log	getDocumentProcessingStatus	`GetFeedSubmissionResult`

Amazon MWS Feed Submission Process

The Feeds API section of the Amazon MWS API lets you upload inventory and order data to Amazon MWS. You can also use the Feeds API section to get information about the processing of feeds.
The process for submitting feeds is as follows:

1. Submit an XML or flat file using the `SubmitFeed` operation along with an encrypted header and all required metadata, including a value from the **FeedType** enumeration and a Content-MD5 header for transmission validation. As with all submissions to Amazon MWS, you must also include authentication information. The `SubmitFeed` operation returns a **FeedProcessingId**, which you use to periodically check the status of the feed.
2. Use the `GetFeedSubmissionList` operation and the **FeedProcessingId** returned from the previous step to check the status of the feed. If Amazon MWS is still processing a request the `GetFeedSubmissionList` operation returns the **FeedProcessingStatus** element with a status of *_IN_PROGRESS_*. If the processing is complete, a status of *_DONE_* is returned. If processing hasn't started, the status of *_SUBMITTED_* is returned.
3. When the feed processing is complete, you use the `GetFeedSubmissionResult` operation to receive a processing report that describes which records in the feed were successful and which records generated errors. Note that you have to set up a stream that Amazon MWS uses to write out the report when you submit the `GetFeedSubmissionResult` operation. See the Feeds API section client library code sample called *GetFeedSubmissionResultRequest* for an example of how to create the stream.
4. Analyze the processing report, correct any errors in the file or transmission, and resubmit the feed using the `SubmitFeed` operation. Repeat the process until there are no errors in the processing report. When the processing report is error free, the transmission is complete.

Flowchart of the feed submission process in Amazon MWS



Document Types and MWS Feed Types

The message types you used to submit various SOAP documents are equivalent to the Amazon MWS feed types. When you use the `SubmitFeed` operation, you use a **FeedType** to tell Amazon MWS which action you want to perform.

The following table shows the SOAP document type and the equivalent MWS feed type action for XML feeds.

SOAP Document Type Description	Valid SOAP Message Type and Amazon MWS FeedType value	Amazon MWS Feed Type description
Product data	_POST_PRODUCT_DATA_	Product Feed
Product inventory	_POST_INVENTORY_AVAILABILITY_DATA_	Inventory Feed
Product pricing	_POST_PRODUCT_PRICING_DATA_	Pricing Feed
Product relationships	_POST_PRODUCT_RELATIONSHIP_DATA_	Relationships Feed
Product images	_POST_PRODUCT_IMAGE_DATA_	Product Images Feed
Product shipping overrides	_POST_PRODUCT_OVERRIDES_DATA_	Shipping Override Feed
Order Acknowledgement	_POST_ORDER_ACKNOWLEDGEMENT_DATA_	Order Acknowledgement Feed
Order Fulfillment	_POST_ORDER_FULFILLMENT_DATA_	Order Fulfillment Feed
Order Adjustment	_POST_PAYMENT_ADJUSTMENT_DATA_	Order Adjustment Feed

Managing Inventory with the Amazon MWS Feeds API Section

The type of action that Amazon MWS performs on a feed is determined by the **FeedType** value. The **FeedType** enumeration is used to indicate how the data in the feed should be processed. The **FeedType** value must be correct for the template file submitted.

Note: Inventory feeds are NOT INCREMENTAL. Updating lead time or restock date once will not be preserved if not present the next time an update for that inventory record is submitted. All fields must be submitted every time for inventory updates. If a merchant wishes to set a lead time of, for example, three days, they must always submit this value when updating inventory.

The following table shows common seller actions and the **FeedType** value that should be passed to the `SubmitFeed` operation:

Action	Amazon MWS FeedType to pass to `SubmitFeed`
Add inventory to Amazon (XML file)	*_POST_INVENTORY_AVAILABILITY_DATA_*
Add product data to Amazon	*_POST_PRODUCT_DATA_*
Add product data to Amazon (flat file)	*_POST_FLAT_FILE_LISTINGS_DATA_*
Update price and quantity (flat file)	*_POST_FLAT_FILE_PRICEANDQUANTITYONLY_UPDATE_DATA_*

Note: if you submit flat files, use the *_POST_FLAT_FILE_PRICEANDQUANTITYONLY_UPDATE_DATA_* **FeedType** to indicate that an item is out of stock. If you submit XML files, use the *_POST_INVENTORY_AVAILABILITY_DATA* **FeedType** value to indicate that a SKU has zero inventory.

Managing Orders with the Amazon MWS Feeds API Section

Amazon MWS uses a process similar to what you have been using with SOAP to acknowledge orders and make order adjustments. With Amazon MWS, you submit feeds using the `SubmitFeed` operation with a particular **FeedType** to accomplish a given task. You can use the *_POST_ORDER_ACKNOWLEDGEMENT_DATA_* **FeedType** to acknowledge order receipt. Use the *_POST_PAYMENT_ADJUSTMENT_DATA_* to adjust payments on an order.

Order Fulfillment

With Amazon MWS, the process for fulfilling orders is the same as for submitting a feed to Amazon MWS. You use *the _POST_FLAT_FILE_FULFILLMENT_DATA_* (flat files) or the *POST_ORDER_FULFILLMENT_DATA_* (XML file) **FeedType** value to submit your order fulfillment feed using the `SubmitFeed` operation to specify the order item ids for each item you ship.

The `SubmitFeed` operation returns a **FeedProcessingId**, which you can use to periodically check the order fulfillment status using the `GetFeedSubmissionList` operation. If Amazon MWS is still processing a feed, the **FeedProcessingStatusList** parameter of the `GetFeedSubmissionList` operation returns a status of *_IN_PROGRESS_*. If the processing is complete, a status of *_DONE_* is returned.

Order Information

The new Orders API section of the Amazon MWS API provides real time order data that you can use to synchronize Amazon order data with order data in your local system. You can get order details for researching issues and answering customer questions. You can also get order status data on your Amazon orders so you can fulfill your orders using your own fulfillment system.

The order data you can retrieve using the Orders API section can also be brought into a custom application to perform business analysis. For example you could:

- Find SKUs with high or low sales volumes
- Monitor sales trends by SKU
- Find unexpected changes in sales trends by SKU
- Find SKUs that have been in a given state for too long

Note that the new Orders API section is for synchronous order querying. If you need bulk order data, be sure to use the operations in the Reports API section.

Creating and Retrieving Reports with Amazon MWS

Amazon MWS gives you more access to both on-request and scheduled reports. Amazon MWS also uses a similar process to SOAP where you request a report, query to find the status of the report, and then download the report. You can also acknowledge that a report has been received.

Using Amazon MWS, you can change the scheduling of reports for defined report types, a significant advantage over SOAP report scheduling. You can also can scheduled reports using the `ManageReportSchedule` operation, the **ReportType** of the report you want to cancel and the *_NEVER_* **Schedule** parameter. Scheduling reports can save you time and you can easily change the frequency of scheduled reports.

The following table shows the action you take to work with a report, the SOAP operation you currently call, and the Amazon MWS operation you would use to accomplish the same task.

Action	SOAP operation	Amazon MWS operation
Request a report		`RequestReport`
Check on the status of a report	getDocumentProcessingStatus	`GetReportRequestList`
Get a count of your report requests		`GetReportRequestCount`
Get Report Ids of all reports available for download	getAllPendingDocumentInfo	`GetReportList`
Download a report	getDocument	`GetReport`
Acknowledge receipt of a report	postDocumentDownloadAck	`UpdateReportAcknowledgements`

When you submit a report request using the `RequestReport` operation, you pass in a **ReportType** value that indicates what report you want created. The two SOAP report types have an equivalent report type in Amazon MWS:

Action	SOAP report type	Amazon MWS report type
Scheduled XML Order Report	_GET_ORDERS_DATA_	_GET_ORDERS_DATA_
XML Settlement Report	_GET_PAYMENT_SETTLEMENT_DATA_	_GET_PAYMENT_SETTLEMENT_DATA_ is automatically scheduled and can be found using the `GetReportList` operation.

You can request additional reports using Amazon MWS that are not available using SOAP. The following table lists some of the reports available using Amazon MWS. To see all of the reports available, review the **ReportType** enumeration in the reference documentation for the Amazon MWS Reports API section.

Name	Enumeration/API Function	Description
Open Listings Report	_GET_FLAT_FILE_OPEN_LISTINGS_DATA_ API Function: RequestReport	Tab-delimited flat file open listings report that contains the SKU, ASIN, Price, and Quantity fields. For Marketplace and Seller Central.
Merchant Listings Report	_GET_MERCHANT_LISTINGS_DATA_ API Function: RequestReport	Tab-delimited flat file detailed active listings report for up to 50,000 listings. For Marketplace and Seller Central.
Merchant Listings Lite Report	_GET_MERCHANT_LISTINGS_DATA_LITE_ API Function: RequestReport	Tab-delimited flat file active listings report that contains only the SKU, ASIN, Price, and Quantity fields for items that have a quantity greater than zero. You can use this report for more than 50,000 listings. For Marketplace and Seller Central.
Merchant Listings Liter Report	_GET_MERCHANT_LISTINGS_DATA_LITER_ API Function: RequestReport	Tab-delimited flat file active listings report that contains only the SKU and Quantity fields for items that have a quantity greater than zero. You can use this report for more than 50,000 listings. For Marketplace and Seller Central.

Name	Enumeration/API Function	Description
Canceled Listings Report	_GET_MERCHANT_CANCELLED_LISTINGS_DATA_ API Function: RequestReport	Tab-delimited flat file canceled listings report. For Marketplace sellers only.
FBA Inventory Report	_GET_AFN_INVENTORY_DATA_ API Function: RequestReport	Tab-delimited flat file FBA inventory report. For FBA sellers only. For Marketplace and Seller Central.
FBA Fulfilled Shipments Report	_GET_AMAZON_FULFILLED_SHIPMENTS_DATA_ API Function: RequestReport	Tab-delimited flat file FBA fulfilled shipments report. Contains detailed order/shipment/item information including price, address, and tracking data. For FBA sellers only. For Marketplace and Seller Central.
Unshipped Orders Report	_GET_FLAT_FILE_ACTIONABLE_ORDER_DATA_ API Function: RequestReport	Tab-delimited flat file report that contains only orders that are not confirmed as shipped. Cannot be scheduled. The information in this report is refreshed every four hours. For Marketplace and Seller Central.
Scheduled XML Order Report	_GET_ORDERS_DATA_ API Function: ManageReportSchedule	Scheduled XML order report. For Seller Central sellers only.
Flat File Order	_GET_FLAT_FILE_ORDER_REPORT_DATA_	Tab-delimited flat file order report.

Name	Enumeration/API Function	Description
Report	API Function: : RequestReport	For Seller Central sellers only.
Scheduled Flat File Order Report	_GET_FLAT_FILE_ORDERS_DATA_ API Function: RequestReport API Function: ManageReportSchedule	Tab-delimited flat file order report that can be requested and scheduled. For Seller Central sellers only.
Flat File Order Report	_GET_CONVERGED_FLAT_FILE_ORDER_REPORT_DATA_ API Function: ManageReportSchedule API Function: RequestReport	Tab-delimited flat file order report that can be both scheduled and requested. For Marketplace sellers only.
Flat File Settlement Report	_GET_FLAT_FILE_PAYMENT_SETTLEMENT_DATA_ API Function: GetReportList	Tab-delimited flat file settlement report that is automatically scheduled; it cannot be requested through `RequestReport`. For Seller Central sellers only.
XML Settlement Report	_GET_PAYMENT_SETTLEMENT_DATA_ API Function: GetReportList	XML file settlement report that is automatically scheduled; it cannot be requested through `RequestReport`. For Seller Central sellers only.

The Reports Process

The on-request report process begins by creating a report request. Next, you obtain a list of report requests which shows the report request identifier and status of each requested report. Finally, you use the report request identifier from this listing to get the actual report. The process steps are as follows:

1. Submit a report request using the `RequestReport` operation. This is a request to Amazon MWS to generate a specific report. Note that you can also schedule order and Amazon Product Ads report requests to be submitted periodically using the `ManageReportSchedule` operation.

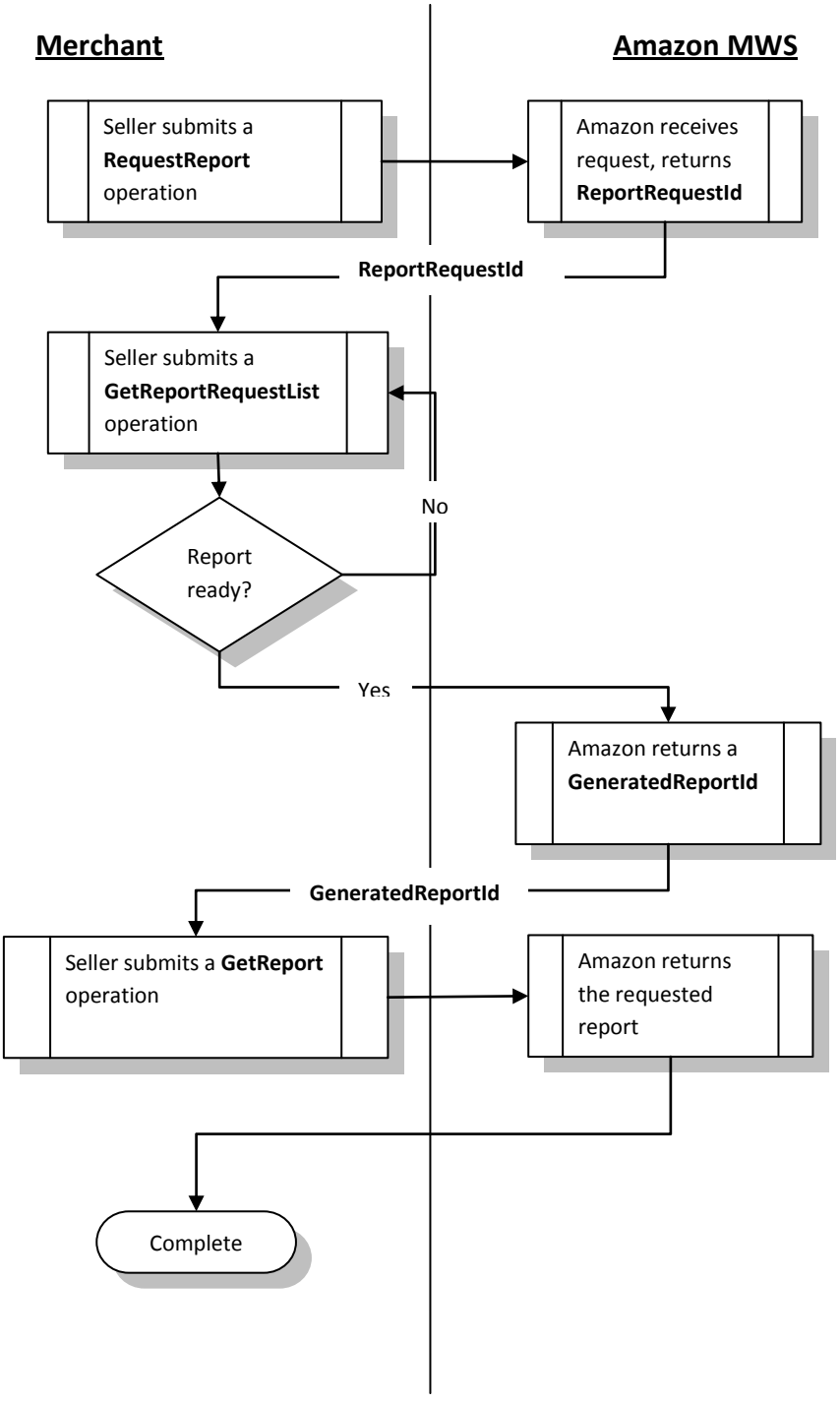
2. Submit a request using the `GetReportList` operation to get a list of report requests and the status and ID of each report request. Amazon MWS returns a **ReportRequestId** for every report requested. When Amazon MWS sets the status of a report request to *_DONE_*, you can then go on to the following step to retrieve the report. If you need to cancel a report request, use the `CancelReportRequests` operation.

3. Submit a request using the `GetReport` operation to receive a specific report. You include in the request the **ReportRequestId** of the report you want to receive and process the Content-MD5 header to confirm that the report was not corrupted during transmission. For more information on working with the Content-MD5 header, see the *Amazon MWS Developer Guide*.

You can schedule order report requests so that they are submitted periodically by using the `ManageReportSchedule` operation. The `Schedule` enumeration is used to specify the time period for submitting report requests. You can also get a list of scheduled order report requests using the `GetReportScheduleList` operation.

The following two flowcharts show the process for requesting and receiving on-request and scheduled reports:

Flowchart of a report request



Flowchart for scheduling order reports