SOFTWARE | HARDWARE | SERVICE

UTC RETAIL

Merchant Cross-Channel API Application Guide

MER5600_MCC_APPLICATION_GUIDE_V1

1 Module Overview

The Merchant Cross-Channel API (Application Programming Interface) is an extension of Merchant Store which allows 3rd-party applications to access Merchant Store's business functionality through a web service interface. This interface is designed to give other sales channels, particularly ecommerce applications but also emerging channels such as line-buster and mobile POS, the ability to interact in real-time with the Merchant system, including item querying, inventory availability, pricing, and order management.

The initial release focuses on supporting E-Commerce web applications that wants to sell merchandise that is defined and maintained in Merchant to web customers.

This purpose of this document is to describe how to use the Merchant Cross-Channel APIs to support the business functions of your web application. The manual provides overview of the functions supported through the various API's, the data structures that it supports and the contents and relationship of the data provided.

To understand the technical aspects of the Merchant Cross-Channel API's please refer to the Merchant Cross-Channel API Technical Guide.

This User Guide is organized as follows:

- **Merchant Cross-Channel Overview (User Guide Section 2)**
- **Application Developer Responsibilities (User Guide Section 3)**
- **Merchant Cross-Channel API's (User Guide Section 4)**

2 Merchant Cross-Channel Overview

The purpose of this section is to provide an overview of Merchant Cross-Channel (MCC) and the API services that it provides (MCCAPI). The section will provide information the type of data and functions available to support your application development. The initial release is designed to support web application development, but will be enhanced to support other types of applications such as mobile retail applications.

Merchant Cross-Channel uses Microsoft's Windows Communication Foundation (WCF) to support the solution and its APIs. WCF is an application programming interface (API) in the .NET Framework for building connected, service-oriented applications.

2.1 Merchant Overview

The purpose of this section is to provide an overview of the Merchant system and several concepts relative to the data that the Merchant Cross-Channel API's will provide.

The following diagram depicts the architecture of the UTC RETAIL MERCHANT solution. Built on a client/server-based architecture, the solution supports two differing types of store-corporate communication methodologies to meet the varying needs of retailers: Nightly Polling (ETL/File Router) or Real Time Messaging (RTM). Both methodologies accomplish the same objectives, simply in a different time frame.

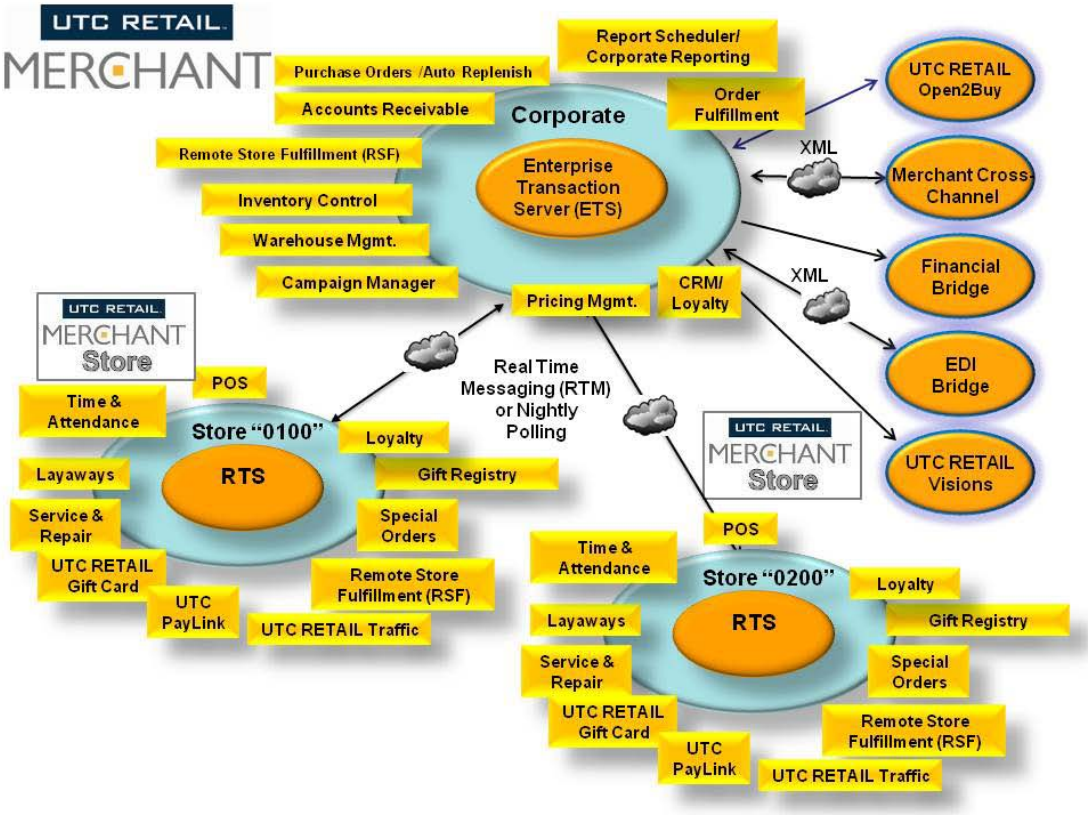

Figure 2.1 Merchant Architectural Overview

	The store-to-corporate communications methodology that your retailer has implemented may impact some of your web application design decisions related to the frequency and timing of your API System Configuration GETs especially with the Item/Catalog load function (Please refer to **Section 4.2.1 /config/catalog**). You will obviously need to understand the methodology implemented.

The Enterprise Transaction Server (ETS) is the server that is hosted at the corporate office. This server will communicate to the Retail Transaction Server (RTS) (one at each store) over DSL Internet, secure VPN, or Dial-Up Internet lines depending on the communication methodology implemented. One or more registers/access points will be connected to each RTS to support POS, receiving, purchase order generation, DSD, transfers, CRM, etc. functions.

2.1.1 Merchant and MCC API Data Structure

The following diagram illustrates the Merchant data structure that the MCC API's support.

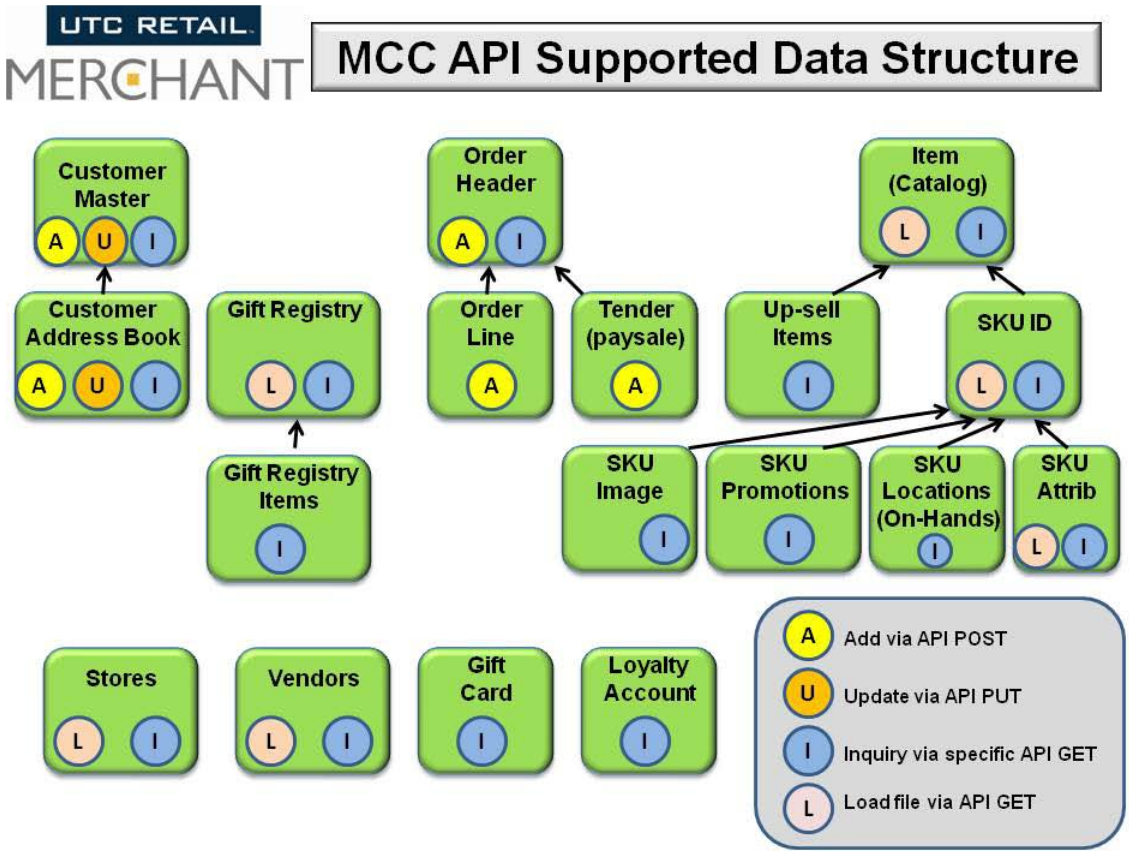

Figure 2.2 MCC API Supported Data Structure

As you review the MCC API's defined in Section 4 the above data structure will become more apparent. The following, however, provides a quick overview.

- **Customer Master and Customer Address Book** – Each customer is defined by a unique Customer ID number. The Customer ID Number and a Sequential Number are used to then define one or more Address Book entries associated with the customer.
- **Gift Registry and Gift Registry Items** – The Gift Registry is defined by a Gift Registry Number (ID) with the descriptive data describing the registry event. Gift Registry ID and a Gift Registry Line Number are combined to define the SKU's within the registry.
- **Order Header, Order Line(s) and Tender(s)(Paysale)** – When your application creates orders your POST function will create the Order Header, Order Line and Tenders. Your Order Header will include your order number. MCC will provide you with the Merchant Order Number for future reference purposes. Order Lines are controlled by the Merchant Order Number and a Sequence number while tenders or order payments are controlled by the Merchant Order Number, a sequence number and the type of tender.

2.1.2 Item-SKU-Attributes

The following diagram illustrates Merchant's Item/Style + Attribute = SKU concept. The **/config/catalog** service function (please refer to Section 4.2.1) will provide you with the web-enabled Merchant Items and their associated SKU's along with attribute descriptions for the SKU.

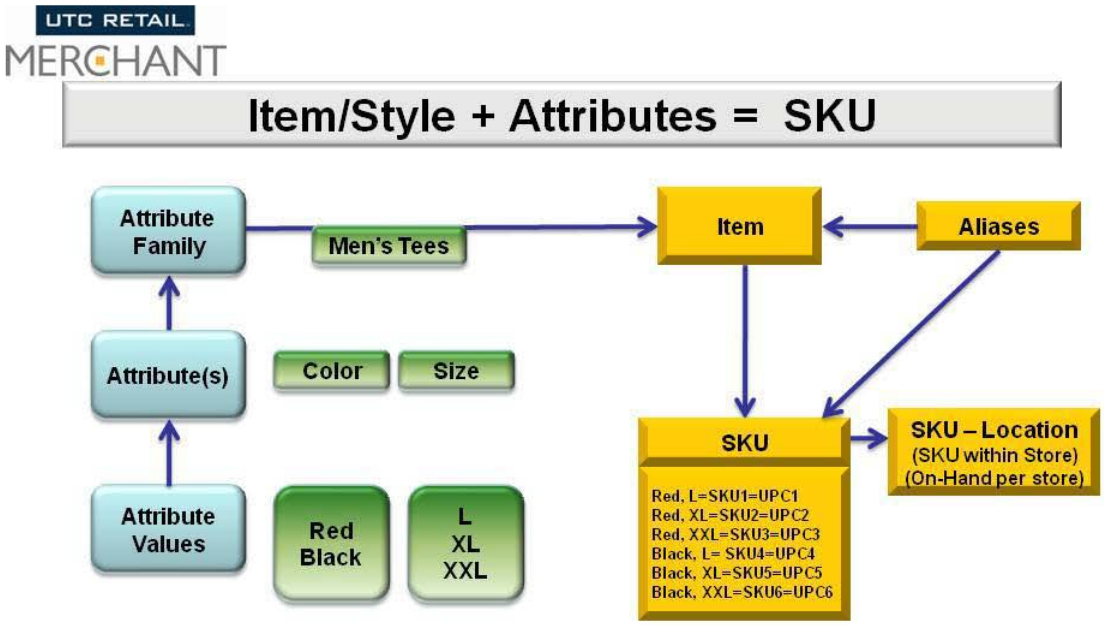

Figure 2.3 Item/Style = Attribute = SKU Concept

Merchant uses an Item or Style number to define its merchandise. Users can assign an attribute family (please refer to Section 9) to the Item/Style. The attribute family can have a defined set of attributes such as color and size. You can define as many attributes to an attribute family. Each attribute also has attribute values as illustrated in Figure 2.3. The attribute values and the Item will then have an associated SKU. An Attribute Family can also be defined to not have any

attributes. This is referred to as a “one-to-one” attribute family and would only have a single SKU associated with the Item/Style.

MMC will not provide any Aliases tied to the items or SKU’s as you will be using the SKU ID for placing your web orders.

2.1.3 Merchandise Hierarchy

Figure 2.4 below provides an overview of Merchant’s Merchandise Hierarchy. Merchant allows users to assign five levels (department, class, sub-class, product line, and group) of merchandise hierarchy to each item plus vendor, season, and source code within these levels. Your retailer may or may not have implemented all of the levels.

In addition, the Item plus its attributes (color, size, width, etc.) that are combined to create a unique SKU.

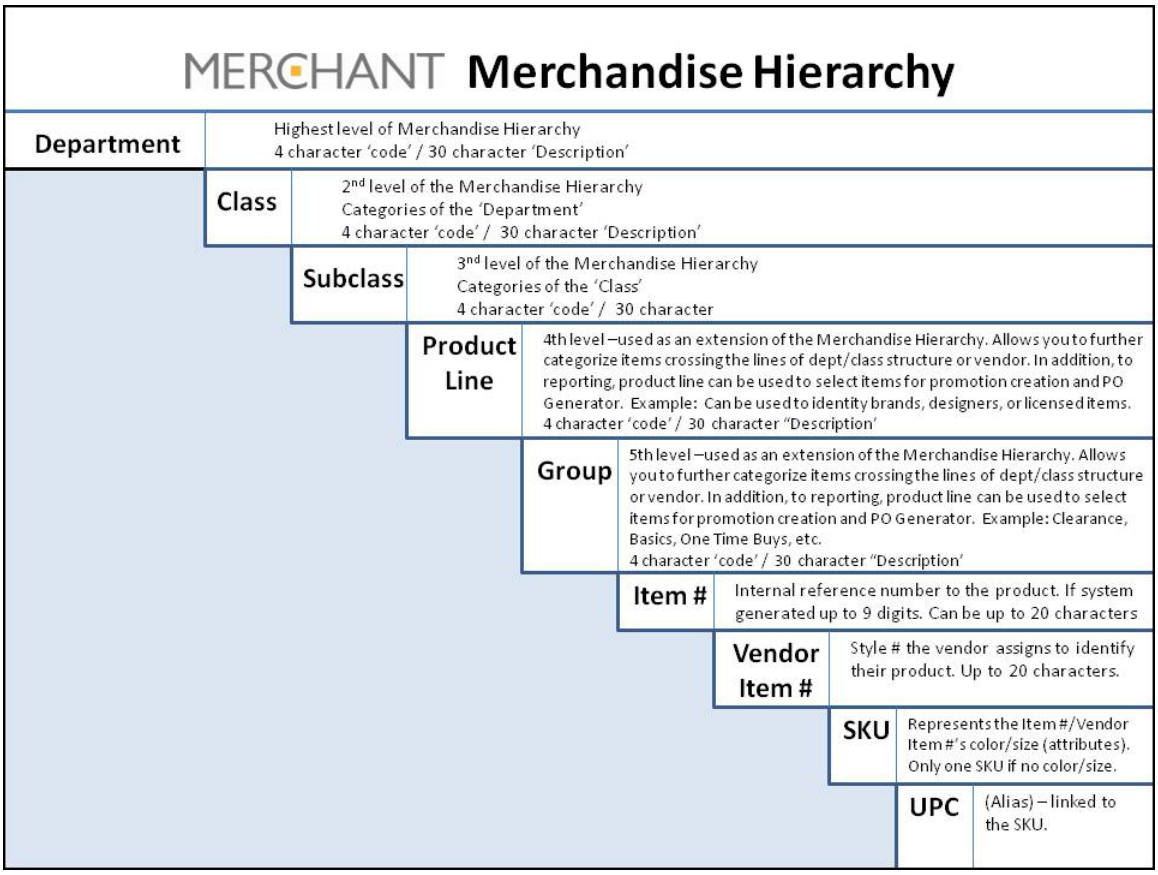

Figure 2.4 Merchant Merchandise Hierarchy

This hierarchy is available to support your web interface design for how you want to present the merchandise to the user. In addition, several other comments are relative to what the hierarchy can support:

- **Fixed versus Non-Fixed Hierarchy** - Merchant's hierarchy is a non-fixed hierarchy. This means that the retailer could have defined a class such as "Shirts" that could appear in two departments. To accomplish a fixed hierarchy structure Merchant concatenates both Department & Class to achieve a fixed structure.
- **Attributes** – Attributes are also searchable on the Attribute and their Attribute Values. For example, I want to display all "reds" or all "reds" and a size of "XL".
- **Hierarchy flexibility** - Your retailer can also use the lower hierarchy levels to define other item characteristics such as "Clearance Items", "New Arrivals", "Hot Items" or any other item characteristic that they want you present as web page grouping.

2.1.4 Web Store

When you configure Merchant Cross-Channel you will define a Merchant Store that your web application is attached to. This is your Web Store. This typically is the store or corporate location that you will be fulfilling your orders from.

As you POST new orders to Merchant they will be defined to this Web Store.

When you configure MCC, you will also provide a Register ID that has also been defined in Merchant. It recommended that you define this register as 'MCC-API' to allow you to easily differentiate this register from other registers at the Web Store. This will make the viewing of register report easier to understand.

2.1.5 SKU Locations – Inventory Availability

Merchant maintains merchandise inventory (on-hand and allocated quantities) in SKU Location records. The location is defined by the store number (ID), the region within the store, the aisle with the region and the bin within aisle. The on-hand and allocated quantities are then defined in this record. The **/skus/{SKUID}/locations** service function (please refer to Section 4.9.3 for more details) will allow your web application to retrieve these SKU Location records for a defined SKU assuming that the SKU is active and that the store is active.

	***For your purposes Inventory Availability = on-hand – allocated.*** ***To determine the SKU inventory availability for a store, you need to total the on-hands and allocated for each region/aisle/bin with a store and then subtract the two to get your Net Availability for the store.*** ***Also remember that if you are issuing the /skus/{SKUID}/locations during each customer session, then you are more likely to have real-time availability.***
	***However, you should only add SKUs to your shopping cart that has web-store SKU availability total of one or greater. Merchant will reject or "void" orders if the web store has an inventory of zero or less. Please refer to Section 4.8.3 on the /orders/status/(ORDERID}API Service function.***

The SKU Location records for all active stores have been provided in case you need to support a customer store pick-up function on your web site.

	If your retailer does not have Merchant Real-Time Messaging installed then the inventory for non web-stores will be as of the prior night.

2.1.6 Loyalty and Loyalty Card Issuance

2.1.6.1 Loyalty Points

MCC support the rewarding web customers receiving loyalty points for their web orders. Existing Loyalty numbers can be attached to an order and points will be accumulated if the order qualifies. The **/customers/{(CUSTOMERID**} Get will define whether the customer has a previously assigned loyalty number. You should check with your retailer to see if they have defined web orders as being valid for receiving loyalty points.

Please refer to Section 4.7.2 for how to determine the loyalty balance on a loyalty card.

2.1.6.2 Loyalty Points on a Web Orders

The **/loyalty/accumulation/{ORDERID}** GET provides the option to run an inquiry on a specific order to check the points accumulated on that order once the order has been shipped. Orders must be closed and completed for the loyalty points to display. Loyalty points are not calculated on orders that are not yet closed in Merchant. You can use the /order/{ORDERID} to check the order status before using this function.

Please refer to Section 4.7.1.

2.1.6.3 Web Loyalty Card Issuance

MCC does not support the direct issuance of new Loyalty cards/numbers, however your retailer must define a Loyalty SKU that can be added as part of an order if the customer wants to sign up for the Loyalty program on your web site. The Loyalty card/number will be added to the order during the Fulfillment process and mailed to the customer. Loyalty points will be accumulated for the other items on that order once an order is completed and shipped if the order and its items/SKUs qualify for points.

2.1.7 Selling Gift Cards on the Web

At this time, Gift Cards cannot be sold through the MCC-API; however they can only be used as a tender.

MCC does provide the ability to run an inquiry on the gift card to determine the gift card balance and status. When an inquiry is done on a gift card, the API will return the Gift Card Number, Current Balance, if the card has open activity, an expiration date and if it is expired if an expiration date was assigned.

2.1.8 Gift Registries

MCC provides an option to query all gift registries. All active gift registries will be returned in a summary format **(/config/giftregistries**). Every registry in Merchant is given an expiration date and that date determines whether or not the registry is active, not the actual event date. The

gift registry ID, title of the registry, customer name, occasion label, date of the occasion and the city/state will be returned for each registry.

Gift Registries can also be queried by gift registry id **(/giftregistries/{GIFTREGISTRYID})** and will return a listing of the registry with the items and SKUs attached. The description of the item, the gift registry line number, quantity requested, quantity purchased and whether the item is available to be purchased through the web site will be returned. Items on a gift registry that are not available for purchase on the web will display but cannot be purchased.

Items/SKUs associated to a gift registry must be added to an order **(/orders/new**) with the correct gift registry number and corresponding line number for the item/SKU.

Please refer to Sections 4.2.4, 4.5.1 and 4.8.2 for information on each of the above service functions.

2.1.9 Merchant Pricing

Merchant's pricing starts with an Item **Price** or Selling Price for the Item. Please refer to the Field #11 – Price in the table in Section 4.2.1. Each SKU has an **AddOnPrice** field (Field #36 in the table in Section 4.2.1) that adds (+) or subtracts (-) to the Item Price to arrive at the SKU Price. This will be the price that you will charge your customer. The **AddOnPrice** is used for several purposes:

- It is used to support up/down charging for individual SKU. For example you would set this to +$2.00 to add $2.00 for a "XXL" size.
- It is also used for permanent markdowns to reduce the SKU price from the original Item Price.

You will need to work with your retailer to make sure understand exactly how they have implemented this and how they want you to apply the SKU **AddOnPrice** for your web pricing.

This is the starting point before you start to consider promotional pricing. Please refer to Section 4.9.4 for how to handle SKU promotions.

2.1.10 Orders and the Merchant Fulfillment Process

This section describes criteria for submitting your web orders, the Merchant fulfillment process and how your web application will need to issue periodic order status checks to determine an action that you will need to take or to get information on the order that you may want to communicate with the customer.

2.1.10.1 Web Orders

Multiple Ship-To - Merchant Cross-Channel supports your web site inserting orders with multiple ship-to addresses in the order. This creates a Merchant Order with one or more Shipments.

Your application will control this capability by each of your order lines using a referenced Address Book Sequence Number on the Order Line. The Address Book Sequence Number must be a valid sequence number assigned to the customer that you are processing the order for.

Please review Section 4.3 for definitions on the Customer Operation service function and Section 4.8 on Orders service functions.

Web Store On-Hand Inventory - Your web store's on-hand inventory is the source for fulfilling your orders. If the SKU inventory does not exist in the web store then do not add the item/SKU to your shopping cart. Please also refer to Section 2.1.5 for the rules on determine how Merchant will validate and void the order if the inventory is not available. Section 4.9.3 **/skus/(SKUID}/locations** describes how to determine store on-hands.

Gift Registry - In addition, if your application is supporting the Merchant Gift Registry function, the **"/order/new"** service function will also update the Merchant Gift Registry system with the purchase activity against a specific gift registry ID once the merchandise ha shipped.

Your application can place new customer orders into the Merchant system and they will be processed through Merchant's order fulfillment function (pick, pack and ship) by using the **"/order/new"** service function. Please refer to Section 4.8.2 for further description and edit rules for your order submission.

Merchant will set the order status to "O" (Open) with an Order Method of "I" (Internet).

2.1.10.2 Merchant Fulfillment Process

Merchant fulfillment process consists of a series of steps.

- **Order Processing** – this step reviews the order and order lines and allocates inventory to your order and order lines. If inventory can be allocated to each order line, then order line will be placed in a queue to start the fulfillment process. If the web store does not have inventory that can be allocated to your order, the order will be reject or voided.

	***If Merchant cannot allocate one order line, then it will reject and void the whole order.***

- **Order Faucet**- This process creates one or more shipment headers to support your order. Picking and packing slips are created for each shipment within your order.
- **Pack & Ship** – The Pack & Ship function confirms the item/SKU's as they are placed in the shipping box/container. Shipping labels are printed and the order and shipments are held waiting for your application to authorize the shipment which also tells Merchant that you have charged the credit card.
- **Order Shipment Confirmation** – Upon your approving the shipment, Merchant then confirms the shipment and closed the order.

2.1.10.3 Order/Shipment Status

Your application will need to periodically check the status of your order to determine:

- An action that you will need to take for Merchant to continue or you to address or
- To obtain information on the order that you may want to communicate with the customer.

You will use the **/orders/status/{ORDERID} GET** API service function to obtain this status information. Section 4.8.3 describes this function, the types of status information available, and Figure 4.1 provides a table of how statuses change as your order moves through the fulfillment process.

The following diagram provides a high level overview of the Order Submittal –Status process and how it might be designed into your application.

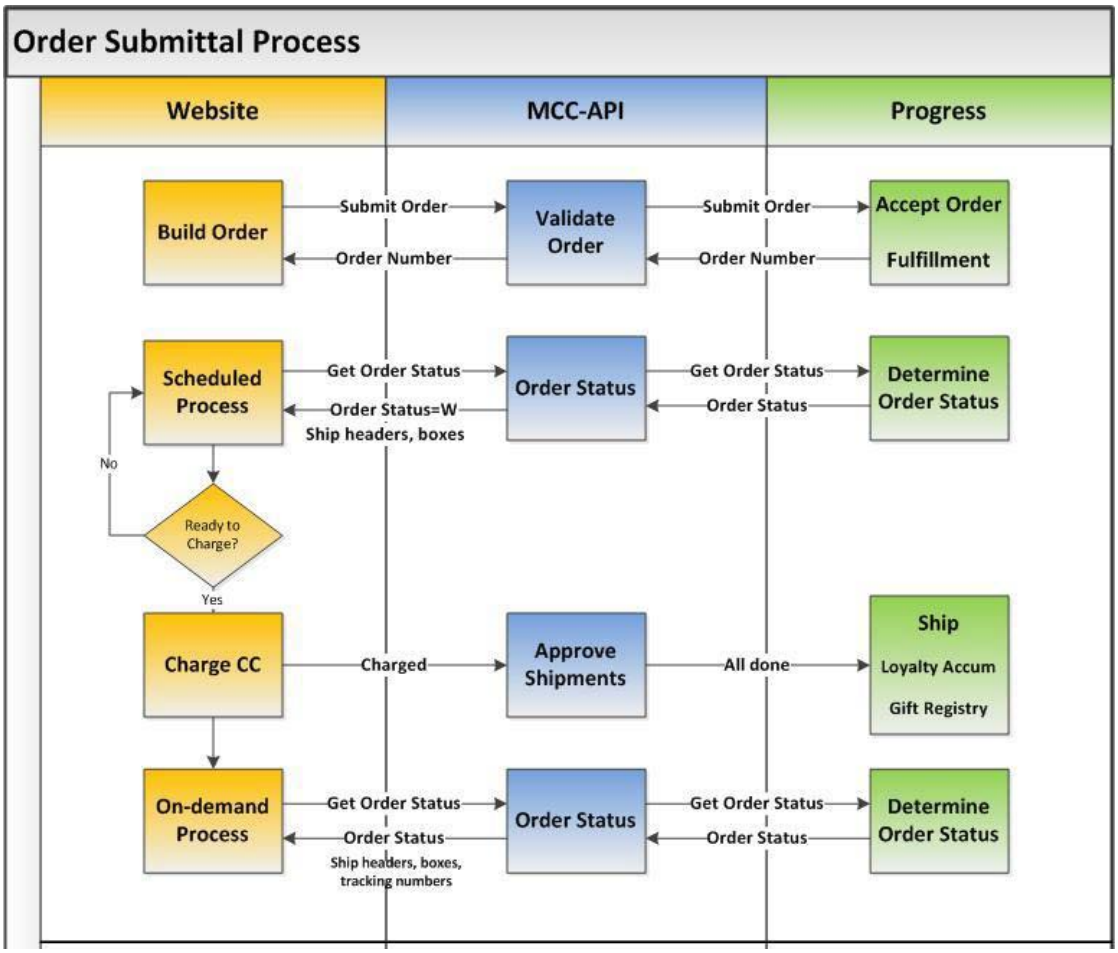

Figure 2.5 Order Submittal Process

Please refer to Section 4.8.3 for a more in-depth discussion of the MCC-API Status Codes.

2.2 MCC API Service Operations

Merchant Cross-Channel consists of API's that support standard HTTP GET, POST and PUT requests and responses to provide access to the Merchant database in real-time. The following provides a high-level overview of MCC and the organization of its MCC API Service Operation groups.

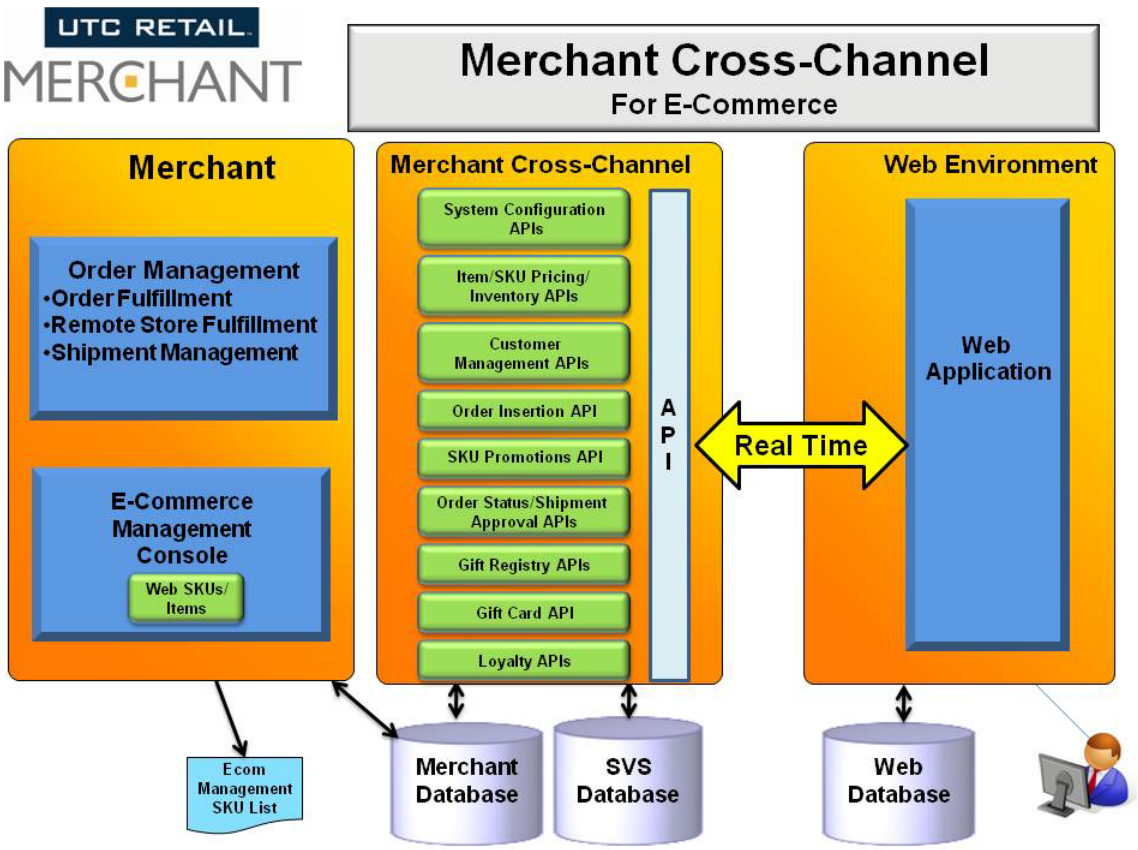

Figure 2.6 MCC API Service Functions

The following provides a detailed list of MCC APIs.

This page describes the service operations at this endpoint.

Uri	Method	Description
/config/catalog	GET	Returns all Web-Enabled Items (the Item catalog)
/config/classes	GET	Returns a list of Classes.
/config/departments	GET	Returns a list of Departments.
/config/giftregistries	GET	Returns a list of active Gift Registries
/config/groups	GET	Returns a list of Groups.
/config/productlines	GET	Returns a list of Product Lines.
/config/shippingmethods	GET	Returns a list of Shipping Methods.
/config/sources	GET	Returns a list of Sources.
/config/stores	GET	Returns a list of Stores.
/config/subclasses	GET	Returns a list of Sub-classes.
/config/vendors	GET	Returns a list of Vendors.
/customers/{customerId}	GET	Returns a Customer for a specified CustomerId.
/customers/{customerId}/addressbook	GET	Returns the Address Book for a specified CustomerId.
/customers/address/new	POST	Adds a new Address Book Entry to the specified Customer's Address Book.
/customers/address/update	PUT	Updates an existing Address Book Entry.
/customers/email/{emailAddress}	GET	Returns a list of Customers for a specified email address.
/customers/new	POST	Creates a new Customer record. Returns new Customer Number.
/customers/update	PUT	Updates an existing Customer record.
/giftcards/{giftCardNumber}	GET	Returns a GiftCard record for a specified GiftCardNumber
/giftregistries/{giftRegistryId}	GET	Returns a Gift Registry for a specified GiftRegistryId
/items/{itemId}	GET	Returns an Item for a specified ItemId.
/items/{itemId}/upsells	GET	Returns a list of Upsell Items for a specified ItemId.
/loyalty/accumulation/{orderId}	GET	Returns Loyalty points accumulation for a specified OrderId
/loyalty/inquiry/{loyaltyNumber}	GET	Returns a Loyalty inquiry record for a specified LoyaltyNumber
/orders/approveshipments	POST	Approves the shipment of either an entire Order or a specified Order Line
/orders/new	POST	Submits a new order for processing.
/orders/status/{orderId}	GET	Returns the Order status for a specified OrderId
/skus/{skuId}	GET	Returns a SKU for a specified SKU Id.
/skus/{skuId}/image	GET	Returns a jpg image for a specified SKU
/skus/{skuId}/locations	GET	Returns location information for a specified SKU
/skus/{skuId}/promotions	GET	Returns a list of Promotions, in priority order, applicable to a specified SKU
/stores/{storeId}	GET	Returns a Store for a specified StoreId.
/vendors/{vendorCode}	GET	Returns a Vendor for a specified VendorCode.

Figure 2.7 MCC API Services

Section 4 provides a detail description of how to use each service operation group and each service operation within the group.

3 Application Developer Responsibilities

Section 2 above provides an overview of the functions and types of data that MCC API supports. This section now addresses areas and functions that the MCC API will not support.

The web application developer will be responsible for certain functions that the Merchant Cross-Channel will not address. Your application will need to support the following functions if the function is a requirement for your application.

- Credit card payment activity. Merchant Cross-Channel will provide an Order Status API to define to the website when the merchandise has shipped so that you can complete the payment transaction.
- Wish List Functionality
- Web offers sign up and information maintenance
- Save Cart functions
- Handling of Sales Tax. MCC will provide for the definition of the appropriate sales taxes in each of the retailer's store locations.
- Website account creation and maintenance
- Calculating appropriate promotional pricing based on the information from MCCAPI.
- **API Initiation** – It is assumed that the your has responsibility for all API initiation and that your application will initiate the Item Retrieval **("config/catalog")** service function at least once per day after Merchants nightly polling process has been completed.
    - Website – initiates the API transaction
    - MCC – returns the requested information or an error code.
- **Address Validation: -** Merchant and the MCC-API assume that you will perform a Address Validation on any new addresses and address changes both for customer and address book POSTs and PUTs. While Merchant does address validation it does not make sense to have Merchant reject an order back to you and then you having to re-submit the order with a valid address. This is done to avoid shipping problems with the package delivery providers.

4 Merchant Cross-Channel API's

Merchant Cross-Channel provides a set of HTTP services that are defined in Figure 2.6 above. The purpose of this section is to describe the purpose of each of these service functions, recommendation of how to use the function, and relationships that may exist between the service functions and the data within them and other service functions.

Please refer to Figure 2.2 MCC API Supported Data Structure below to understand the data relationships that the service functions support. Also please refer to the Merchant Cross-Channel API Technical Guide for the details of the HTTP requests and responses.

4.1 General

4.1.1 General Edit Rules

All fields of types 'integer' and 'decimal' must have a value (cannot be left blank). If there is no applicable value to provide for a particular order request (for instance <GiftRegistryLineNumber>), then the digit '0' (zero) must be provided.

4.2 System Configuration Operations

The purpose of the System Configuration Operations GET functions is two-fold:

- To provide a series database load functions to allow you to pre-load Merchant data files into your web database during off-hour time periods. These include the following functions:
    - **/config/catalog** – This GET will download Merchant's web-enabled and "active" items, their associated web-enabled SKUs and the attributes for the SKUs.
    - **/config/giftregistries** – This GET will download Merchant's non-expired gift registry headers that that provide summary information for each active registry. Gift Registries with Registry Expiration Dates past the date of the request are not returned.

	MMC also provides a service function to get the registrants registry Items and the purchase status on each of the items/SKUs. Please refer to Section 4.5.1 for a description of the **/giftregistries/{giftregistryId}** service function.

    - **/config/vendors**- This GET will download the Merchant vendor codes and their vendor names.
- Your retailer uses a number of Merchant system tables to define how they want Merchant to function and organize their data. The System Configuration Operations are also used to provide Merchant system tables to use for code to description conversion, description sorting versus code sorting for web presentation purposes and to also use for web editing.
    - **/config/classes** – Returns a list of all Merchant Class codes and their descriptions. Each Merchant item has a defined class code. This table can be used to obtain the Class descriptions.

- **/config/departments –** Returns a list of Department codes and their descriptions. Again each Merchant item has a defined Department code.
- **/config/groups -** Returns a list of Group codes and their descriptions. Again each Merchant item has a defined Group code.
- **/config/productlines –** Returns a list of Product Line codes and their descriptions. Again each Merchant item has a assigned Product Line code.
- **/config/shippingmethods –** This returns a list of your retailer's valid carriers (FedEx, UPS, USPS, etc.), each carrier's shipping methods (next day, 2-day, ground, ground) and a description. This can be used to define the valid shipping methods that your web application will support.
- **/config/sources –** This returns a list of Merchant Sources Codes, their descriptions, and the Merchant division that the source code is tied to.
- **/config/stores –** This returns a list of active Merchant stores along with their name, the division they belong to, address phone number, and tax rate.

The following describes each of these in more detail.

4.2.1 /config/catalog

The **/config/catalog** GET service function will return all Merchants web-enabled Items (Item Catalog) and the web-enabled SKUs and their attributes within each Item ID. If there are no web-enabled SKU's within the Item then no Item will be returned.

The retailer uses a Merchant tool called E-Commerce Management Console to manage and control the Items and SKU's that they want to feed to the web site.

	If a Merchant SKU is not defined as "Active", it is not even available for web consideration.

The following describes the fields in the response that you will be returned from the **/config/catalog GET** service function.

/config/catalog			GET	
	Field Name	**Data Type**	**Format**	**Description**
Header Data				
1	**ItemCount**			The number of items extracted and included in this response.
2	**SkuCount**			The number of SKUs extracted and included in this response.
3	**CurrentAsOf**			The date and time that the extract/response was generated.
4	**BuildDuration**			

	Field Name	Data Type	Format	Description
Items				
5	**ItemId**	String	20	The Item Number
6	**PlainTextDescription**	String	30	Item Description
7	**FormattedTextDescription**	String		A HTML formatted item description.
8	**ProductInfoMessage**	String	320	
9	**DisplayProductInfoMessage**	True/ False		If "True" then display the **ProductInfoMessage**, if "False" then do not display the **ProductInfoMessage.**
10	**VendorCode**	String	6	The vendor code for where the item is normally purchased.
11	**Price**	Decimal	7,2	Selling Price for the Item
12	**SuggestedPrice**	Decimal	8,2	Suggested Retail Price for the Item
13	**TaxCode**	String	8	Special tax code for the item. If this field has a value greater than zero, then Taxable (field #28) must be "True".
14	**Weight**	Decimal	7,2	The weight of the item. Can be used to calculate shipping charges.
15	**StockUnitOfMeasurement**	String	8	The unit of measure that the item's inventory is maintained in and how it is sold.
16	**Division**	String	3	The Division Code
17	**Department**	String	4	The Item's department code
18	**Class**	String	4	The Item's class code
19	**Subclass**	String	4	The Item's Subclass code
20	**ProductLine**	String	4	The Items Product Line code
21	**Group**	String	4	The Item's Group code
22	**PriceGroup**			The Price Group Code assigned to the item.
23	**SeasonCode**	String	15	The Season Code assigned to the item
24	**CatalogPage**	String	8	The page that the item can be found in the printed catalog.
25	**AttributeFamily**	String	20	The Item attribute family
26	**IsNonMerchandise**	True/ False		If "True" then the Item is a "Non-Merchandise" item and therefore inventory is not maintained for the item. If "False" then this is a merchandise item with inventory tracked.
27	**ShippingChargesApply**	True/ False		If "True" then shipping charges should be charged for the item. If "false" then do not charge shipping charges.
28	**IsTaxable**	True/ False		If "True" then the item is taxable.
29	**IsGiftWrappable**	True/ False		If "True" the item is gift wrap able.
30	**IsPersonalizable**	True/ False		If "True" then the item can be personalized.
31	**IsDiscountable**	True/ False		If "True" then the item can be discounted
32	**DiscountGroup**	String	4	The Discount Group assigned to this item.
33	**SkuCount**	Integer	10	The number of SKU's associated with this item.

	Field Name	Data Type	Format	Description
SKUs	**More that 1 SKU per Item**			
34	**SkuId**	String	8	The SKU ID associated with the item.
35	**ItemId**	String	20	The Item Number
36	**AddOnPrice**	Decimal	7,2	The add-on price provides a way to markup or markdown a SKU from the Item Price. It is used to markup a "XXXL" size or to support permanent markdowns.
37	**VendorItemNumber**	String	20	The Vendor Item Number
38	**Length**	Integer	7	The length of the item's standard packaging for shipping charge calculations.
39	**Width**	Integer	7	The width of the item's standard packaging for shipping calculations.
40	**Depth**	Integer	7	The depth of the item's standard packaging for shipping calculations.
41	**StockUnitOfMeasure**	String	8	The unit of measure that the SKUs inventory is maintained in and how it is sold.

	Field Name	Data Type	Format	Description
SKU Attributes	**More that 1 attribute per SKU**			
42	**Name**	String	10	The attribute type tied to the SKU such as color, size, etc.
43	**Value**	String	10	The attribute value tied to the SKU such as black, red, white orange for a attribute type = color.

4.2.2 /config/classes

The **/config/classes** service function will provide your application with a list of Merchant classes and their corresponding descriptions.

This GET will provide you with the class descriptions for your class search/presentations if needed.

/config/classes			**GET**	
	Field Name	**Data Type**	**Format**	**Description**
Header Data				
1	**LevelCode**	String	4	The Class Code
2	**Description**	String	30	Class Code Description

4.2.3 /config/departments

The **/config/department** service function will provide your application with a list of Merchant departments and their corresponding department descriptions.

This GET will provide you with the Department descriptions for your department code search/presentations if needed.

/config/departments			GET	
	Field Name	**Data Type**	**Format**	**Description**
Header Data				
1	**LevelCode**	String		The Department Code
2	**Description**	String		Department Code Description

4.2.4 /config/giftregistries

The config/giftregistries GET service function will download Merchant's non-expired gift registry headers that that provide summary information for each active registry. Gift Registries with Registry Expiration Dates prior to the date of the request are not returned.

The gift registry items are not returned with this function as Merchant is dynamically updating them as purchase activity occurs. Your application will need to utilize the **/giftregistries/{giftregistryId}** service function to retrieve the gift registry items at the time of web order creation or your application supporting a web gift registry. Please refer to Section 4.5.

The following describes the fields in the response that you will be returned from the **/config/giftregistries GET** service function.

/config/giftregistries			GET	
	Field Name	**Data Type**	**Format**	**Description**
Header Data				
1	**GiftRegistryId**	String	10	The Gift Registry ID Number
2	**Title**	String	45	Gift Registry Title-the name that registrant defined to the system as heading for the occasion/event such as "Mary's Baby Shower"
3	**CustomerName**	String	30	The Gift Registrants name
4	**OccasionLabel**	String	30	The description of the occasion.
5	**OccasionDate**	Date		The occasion date and time in "YYYY-MM-DDTHH:MM:SS" format.
6	**City**	String	15	The city where the occasion event is taking place.
7	**State**	String	2	The state code where the occasion event is taking place.

4.2.5 /config/groups

The **/config/groups** service function will provide your application with a list of Merchant Group Codes and their corresponding Group Code descriptions.

This GET will provide you with the Group Code descriptions for your Group Code search/presentations if needed.

The following describes the fields in the response that you will be returned from the **/config/groups GET** service function.

/config/group			GET	
	Field Name	Data Type	Format	Description
Header Data				
1	**LevelCode**	String	4	The Group Code
2	**Description**	String	30	Group Code Description

4.2.6 /config/productlines

The **/config/productlines** service function will provide your application with a list of Merchant Group Codes and their corresponding Group Code descriptions.

This GET will provide you with the Product Line descriptions for your product line search/presentations if needed.

The following describes the fields in the response that you will be returned from the **/config/productlines GET** service function.

/config/productlines			GET	
	Field Name	Data Type	Format	Description
Header Data				
1	**LevelCode**	String	4	The Product Line Code
2	**Description**	String	30	Product Line Code Description

4.2.7 /config/shippingmethods

The **/config/shippingmethods GET** returns a list of your retailer's valid carriers (FedEx, UPS, USPS, etc.), each carrier's shipping methods (next day, 2-day, ground, and ground) and a method description. This can be used to define the valid carrier's their shipping methods when your web application is creating a new order.

The following describes the fields in the response that you will be returned from the **/config/shippingmethods GET** service function.

/config/shippingmethods			GET	
	Field Name	Data Type	Format	Description
Header Data				
1	Carrier	String	8	The Carrier Code
2	Method	String	8	The Carrier's Ship Method Code
3	Description	String	30	Ship Method Description

4.2.8 /config/sources

The **/config/source GET** service function will provide your application with a list of your retailers Merchant Source Codes and their corresponding Source Code descriptions.

This is designed to allow you to support the following:

- The retailer may have defined a promotion tied to a Source Code. In that case you will need to capture the source code in the order capture process to determine the appropriate item/SKU pricing.
- To capture the source for the customer coming to the web site if this is a requirement of your retailer.

The following describes the fields in the response that you will be returned from the **/config/sources GET** service function.

/config/sources			GET	
	Field Name	Data Type	Format	Description
Header Data				
1	Source	String	8	Source Code
2	Description	String	30	Source Code Description

4.2.9 /config/stores

The **/config/stores GET** service function will provide your application with a list of your retailer's active stores.

This service function will support the following requirements:

- Store Locator functionality.

	If Field #15 "Is Store" is false, then the store is not a selling store and therefore should not be included in the Store Locator.

- If the Customer Ship-To address for an order is in a state that the retailer has a store then you may have to charge sales tax for that state.
- If you need to support a Store Pickup function (i.e. what store has the item/SKU that I am looking for).

The following describes the fields in the response that you will be returned from the **/config/stores GET** service function.

/config/stores			GET	
	Field Name	**Data Type**	**Format**	**Description**
Header Data				
1	**StoreId**	String	4	Store ID
2	**StoreName**	String	30	Store Name
3	**Division**	String	3	The division code that the store belongs to
4	**Address**			The Store's Address
5	**StreetLines**			
6	**Street Line 1**	String	35	Street Address Line 1
7	**Street Line 2**	String	35	Street Address Line 2
8	**Street Line 3**	String	35	Street Address Line 3
9	**City**	String	15	City Name
10	**State**	String	2	State Code
11	**Zip**	Integer	10	Zip Code
12	**Country**	String	3	Country Code
13	**Phone**	String	20	The store's phone number
14	**Is Corporate**	True/ False		Define's whether the store is also a Corporate location. "True" it is, "False" it is not.
15	**Is Store**	True/ False		Define's whether the store is a selling store as opposed to a warehouse.
16	**TaxRate**	Decimal	6,4	Defines the tax rate for the store.

4.2.10 /config/subclasses

The **/config/subclasses** service function will provide your application with a list of Merchant sub-classes and their corresponding descriptions.

This GET will provide you with the sub-class descriptions for your sub-class search/presentations if needed.

The following describes the fields in the response that you will be returned from the **/config/subclasses GET** service function.

/config/subclasses			GET	
	Field Name	Data Type	Format	Description
Header Data				
1	**LevelCode**	String	4	The Sub-Class Code
2	**Description**	String	30	Sub-class Code Description

4.2.11 /config/vendors

The **/config/vendors** service function will provide your application with a list of your retailers Merchant vendors and their name.

The Item Catalog records include the vendor code that is associated with the Item. This GET will provide you with the vendor name if you need that for your search/presentations by vendor.

The following describes the fields in the response that you will be returned from the **/config/vendors GET** service function.

/config/vendors			GET	
	Field Name	Data Type	Format	Description
Header Data				
1	Code	String	6	Vendor Code
2	Name	String	35	Vendor Name

4.3 Customer Operations

Merchant supports a Customer Master table with unlimited Address Book entries per customer (please refer to Figure 2.2). The Customer Operations Service functions allows you to GET existing customer and their address books, create (POST) new customer and their address books, and to update (PUT) existing customers and their address books.

The Customer Operation Services functions support the following business functions:

- **/customers/{CUSTOMERID}** - This GET function returns the Customer when you specify the Customer ID/number.
- **/customers/{CUSTOMERID} /addressbook** – This GET function returns the address book for a specified customerID.
- **/customers/address/new –** This POST function allows you to add an address book entry to the specified Customer's address book.
- **/customers/address/update -** This PUT function allows you to update a existing address book entry for the specified Customer.
- **/customers/email/{EMAILADDRESS} –** This GET returns a list of customers for a specified email address.

- **/customers/new -** This POST function allows you to create a new Customer Master record in Merchant. Merchant will return the new Merchant Customer ID/Number to you.
- **/customers/update -** This PUT function allows you to update an existing Customer Master record.

	Merchant and the MCC-API assume that you will perform a Address Validation on any new addresses and address changes both for customer and address book POSTs and PUTs. While Merchant does address validation it does not make sense to have Merchant reject an order back to you and then you having to re-submit the order.

The following section provides detail descriptions of each of these service operations.

4.3.1 /customers/{CUSTOMERID}

The **/customers/{CUSTOMERID}** GET function returns the Customer when you specify the Customer ID/number.

MCC will return the existing customer record regardless of its status code (A, H, I). A Status of "H" indicates that the customer has been placed on "Hold". You will need to consult with your retailer as to how they want you to handle that customer.

The following describes the fields in the response that you will be returned from the **/customers/{CUSTOMERID} GET** service function.

/customers/{CUSTOMERID}				GET
	Field Name	Data Type	Format	Description
Header Data				
1	**CustomerNumber**	String	8	Customer Number
2	**CustomerFirstName**	String	30	Customer First Name
3	**CustomerMiddleInitial**	String	1	Customer Middle Initial
4	**CustomerLastName**		30	Customer last Name
5	**CompanyName**		35	Company Name
6	**Address**			
7	**StreetLines**	(May be up to four-street lines if in the database)		
8	**Street Line 1**	String	35	Street Address Line 1
9	**Street Line 2**	String	35	Street Address Line 2
10	**Street Line 3**	String	35	Street Address Line 3
11	**Street Line 4**	String	35	Street Address Line 4
12	**City**	String	15	City Name
13	**State**	String	2	State Code
14	**Zip**	Integer	10	Zip Code
15	**Country**	String	3	Country Code
16	**MailingList**	True / False		"True" indicates that the customer wants to be on your mailing list and "False" that they do not.
17	**CustomerEmail**	String	50	The customer's email address
18	**CustomerPhone**	String	20	Customer's day time phone number
19	**CustomerPhoneExtension**	String	5	Customer's day time phone number extension
20	**NightPhone**	String	20	Customer's night time phone number
21	**OtherPhone**	String	20	Other phone number
22	**MobilePhone**	String	20	Cell phone
23	**Fax**	String	20	Customer Fax Number
24	**CustomerDOB**	Numeric		Customer's date of birth (mm/dd/yyyy) format
25	**CustomerSince**	Numeric		Date that the customer first was entered into system. (mm/dd/yyyy format)
26	**LoyaltyNumber**	Integer		Customer Loyalty Number
27	**Gender**	String		Male/Female/none are valid options
28	**PriceLevel**	String	6	The price level code that has been assigned to this customer.
29	**TaxExempt**	True / False		"True" indicates that the customer is tax exempt and "False" that they are not.
30	**TaxExemptType**	String	3	Code that defines the type of tax exemption. Merchant uses Reason Code table to define these codes.
31	**StatusCode**	String	1	Customer Status Code, A=customer is active, H=customer is on-hold, and I=customer is inactive.
32	**Notes**	String		Notes related to this customer.

4.3.2 /customers/{CUSTOMERID}/addressbook

The **/customers/{CUSTOMERID} /addressbook GET** function returns the address book for a specified customer number or ID.

The following describes the fields in the response that you will be returned from the **/customers/{CUSTOMERID}/addressbook GET** service function.

/customers/{CUSTOMERID}/addressbook				GET
	Field Name	**Data Type**	**Format**	**Description**
Header Data				
1	**Id**	Integer		A Merchant generated ID number. This number is used if you need to execute a /customers/address/update.
2	**Sequence**	Integer		The address book entry sequence number within the customer number.
3	**Company**	String		The retailer's company that this customer belongs to.
4	**CompanyName**	String	30	Company Name associated with address book entry
5	**CustomerNumber**	String	8	Customer Number
6	**Title**	String	30	Title of the address book person
7	**Salutation**	String	10	Salutation for address book entry
8	**FirstName**	String	35	Address Book First Name
9	**MiddleName**	String	35	Address Book Middle Initial
10	**LastName**	String	35	Address Book last Name
11	**EmailAddress**	String	15	Address book entry email address
12	**Address**			
13	**StreetLines**			
14	**Street Line 1**	String	35	Street Line 1
15	**Street Line 2**	String	35	Street Line 2
17	**City**	String	15	City Name
18	**State**	String	2	State Code
19	**Zip**	Integer	10	Zip Code
20	**Country**	String	3	Country Code
21	**Phone Number**	String	20	Address book entry phone number
22	**Type**	String	10	Defines the type of address book entry
23	**Comments**	String	10	

4.3.3 /customers/address/new

The **/customers/address/new** POST function allows you to add an address book entry to the specified Customer's address book.

MCC assumes that the Merchant Customer Number (Field #5 below) is a valid Merchant Customer Number/ID that you had previously received via a MCC API GET or that MCC provided via a MCC API POST.

See the table below for edit rules that MCC will apply on receipt of your POST function.

/customers/address/new				POST	
	Field Name	Data Type	Format	Description	Edit Rule
Header Data					
1	**Id**	Integer		A Merchant generated ID number	Set to "0", Merchant will generate and return this number in the POST response. Be sure to use this in subsequent addressbook updates.
2	**Sequence**	Integer		The address book entry sequence number within the customer number.	Set to "0", Merchant will generate.
3	**Company**	String		The retailer's company that this customer belongs to.	
4	**CompanyName**	String	30	Company Name associated with address book entry	
5	**CustomerNumber**	String	8	Customer Number	Must be a valid Merchant customer number.
6	**Title**	String	30	Title of the address book person	
7	**Salutation**	String	10	Salutation for address book entry	
8	**FirstName**	String	35	Address Book First Name	Must not be blank.
9	**MiddleName**	String	35	Address Book Middle Initial	
10	**LastName**	String	35	Address Book last Name	Must not be blank.
11	**EmailAddress**	String	15	Address book entry email address	Must not be blank and be a valid formated email address.
12	**Address**				
13	**StreetLines**				
14	**Street Line 1**	String	35	Street Line 1	Must have at least one street line.
15	**Street Line 2**	String	35	Street Line 2	
	Street Line 3	String	35	Street Line 3	
	Street Line 4	String	35	Street Line 4	
17	**City**	String	15	City Name	Must not be blank
18	**State**	String	2	State Code	Valid 2-digit state code
19	**Zip**	Integer	10	Zip Code	Valid 5 digit zip code at minimum.
20	**Country**	String	3	Country Code	"USA" and "CAN" are valid codes unless retailer is selling beyond this.
21	**Phone Number**	String	20	Address book entry phone number	Edited phone number in (xxx)-xxx-xxxx format or may be blank.
22	**Type**	String	10	Defines the type of address book entry	
23	**Comments**	String		Comments related to the address book entry	

MCC will return the **Id** number (Field #1) if you're POST is successful. Be sure to store this with your address book entry as it will be needed if you execute a **/customers/address/update.**

4.3.4 /customers/address/update

The **/customers/address/update PUT** function allows you to update an existing Merchant address book entry to the specified Customer's address book.

MCC assumes that you will provide the Id number (Field #1) that you received from your **/customers/{CUSTOMERID}/addressbook POST**. The Merchant Customer Number (Field #5 below) must also be a valid Merchant Customer Number/ID that you had previously received via a MCC API GET or that MCC provided via a MCC API POST.

The format and edits of the **/customers/address/update** is the same as the **/customers/address/new** service function.

4.3.5 /customers/email/{EMAILADDRESS}

The **/customers/email/{EMAILADDRESS}** GET function returns a Merchant Customer based upon your specified Email Address. MCC will return the existing customer record regardless of its status code (A, H, I). A Status of "H" indicates that the customer has been placed on "Hold". You will need to consult with your retailer as to how they want you to handle that customer.

The following describes the fields in the response that you will be returned from the **/customers/email/{EMAILADDRESS} GET** service function.

/customers/email/{EMAILADDRESS}				GET
	Field Name	Data Type	Format	Description
Header Data				
1	**CustomerNumber**	String	8	Customer Number
2	**CustomerFirstName**	String	30	Customer First Name
3	**CustomerMiddleInitial**	String	1	Customer Middle Initial
4	**CustomerLastName**		30	Customer last Name
5	**CompanyName**		35	Company Name
6	**Address**			
7	**StreetLines**	(May be up to four-street lines if in the database)		
8	**Street Line 1**	String	35	Street Address Line 1
9	**Street Line 2**	String	35	Street Address Line 2
10	**Street Line 3**	String	35	Street Address Line 3
11	**Street Line 4**	String	35	Street Address Line 4
12	**City**	String	15	City Name
13	**State**	String	2	State Code
14	**Zip**	Integer	10	Zip Code
15	**Country**	String	3	Country Code
16	**MailingList**	True / False		"True" indicates that the customer wants to be on your mailing list and "False" that they do not.
17	**CustomerEmail**	String	50	The customer's email address
18	**CustomerPhone**	String	20	Customer's day time phone number
19	**CustomerPhoneExtension**	String	5	Customer's day time phone number extension
20	**NightPhone**	String	20	Customer's night time phone number
21	**OtherPhone**	String	20	Other phone number
22	**MobilePhone**	String	20	Cell phone
23	**Fax**	String	20	Customer Fax Number
24	**CustomerDOB**	Numeric		Customer's date of birth (mm/dd/yyyy) format
25	**CustomerSince**	Numeric		Date that the customer first was entered into system. (mm/dd/yyyy format)
26	**LoyaltyNumber**	Integer		Customer Loyalty Number
27	**Gender**	String	1	Male/Female/none are valid options
28	**PriceLevel**	String	6	The price level code that has been assigned to this customer.
29	**TaxExempt**	True / False		"True" indicates that the customer is tax exempt and "False" that they are not.
30	**TaxExemptType**	String	3	Code that defines the type of tax exemption. Merchant uses Reason Code table to define these codes.
31	**StatusCode**	String	1	A=customer is active, H=customer is on-hold, and I=customer is inactive.
32	**Notes**	String		Notes related to this customer.

4.3.6 /customers/new

The **/customers/new POST** function allows you to create a new Customer Master record in Merchant. Merchant will return the new Merchant Customer ID/Number to you.

The following table provides the edit rules that MCC will apply on receipt of your POST function.

/customers/new			POST		
	Field Name	**Data Type**	**Format**	**Description**	**Edits**
Header Data					
1	**CustomerNumber**	String	8	Customer Number	Merchant will generate and return this
2	**CustomerFirstName**	String	30	Customer First Name	Must not be blank.
3	**CustomerMiddleInitial**	String	1	Customer Middle Initial	
4	**CustomerLastName**		30	Customer last Name	Must not be blank.
5	**CompanyName**		35	Company Name	
6	**Address**				
7	**StreetLines**				
8	**Street Line 1**	String	35	Street Address Line 1	Must have at least one street line.
9	**Street Line 2**	String	35	Street Address Line 2	
10	**Street Line 3**	String	35	Street Address Line 3	
	Street Line 4	String	35	Street Address Line 4	
11	**City**	String	15	City Name	Must not be blank.
12	**State**	String	2	State Code	Valid 2-digit state code
13	**Zip**	Integer	10	Zip Code	Valid 5 digit zip code at minimum.
14	**Country**	String	3	Country Code	"USA" and "CAN" are valid codes unless retailer is selling beyond this.
15	**MailingList**	True / False		"True" indicates that the customer wants to be on your mailing list and "False" that they do not.	"True" or "False"
16	**CustomerEmail**	String	50	The customer's email address	Must not be blank. Valid email formatted email address.
17	**CustomerPhone**	String	20	Customer's day time phone number	Must not be blank. Edited phone number in (xxx)-xxx-xxxx format.
18	**CustomerPhoneExtension**	String	5	Customer's day time phone number extension	
19	**NightPhone**	String	20	Customer's night time phone number	Blank or edited phone number in (xxx)-xxx-xxxx format.
20	**OtherPhone**	String	20	Other phone number	Blank or edited phone number in (xxx)-xxx-xxxx format.
21	**MobilePhone**	String	20	Cell phone	Blank or edited phone number in (xxx)-xxx-xxxx format.
22	**Fax**	String	20	Customer Fax Number	Blank or edited phone number in (xxx)-xxx-xxxx format.
23	**CustomerDOB**	Numeric		Customer's date of birth (mm/dd/yyyy) format	Valid date in "mm/dd/yyyy" format
24	**CustomerSince**	Numeric		Date that the customer first was entered into system. (mm/dd/yyyy format)	Merchant will automatically create this field.
25	**LoyaltyNumber**	Integer		Customer Loyalty Number	Leave blank
26	**Gender**	String		Male/Female/none are valid options	"M", "F" or blank
27	**PriceLevel**	String	6	The price level code that has been assigned to this customer.	Leave blank
28	**TaxExempt**	True / False		"True" indicates that the customer is tax exempt and "False" that they are not.	"True" or "False"
29	**TaxExemptType**	String	3	Code that defines the type of tax exemption. Merchant uses Reason Code table to define these codes.	Leave blank
30	**StatusCode**	String	1	Customer Status Code :A=customer is active, H=customer is on-hold, and I=customer is inactive.	Automatically set to "A".
31	**Notes**	String		Notes related to this customer.	

4.3.7 /customers/update

The **/customers/update** PUT function allows you to update an existing Customer Master record.

MCC assumes that you have already gotten a Merchant Customer record and that you are now changing information in that record. It therefore assumes that you have the Merchant Customer Number.

The format and edit rules are the same as the **/customer/new** POST function.

4.4 Gift Card Operations

The Gift Card Operations service function is designed to allow you to validate a gift card for web tendering or to support a gift card balance inquiry on your web site.

- **/giftcards/{GIFTCARDNUMBER}** – This GET returns a gift card record for a gift card number.

4.4.1 /giftcards/{GIFTCARDNUMBER}

The **/giftcards/{GIFTCARDNUMBER}** GET returns a gift card record for a gift card number.

The following describes the fields in the response that you will be returned from the **/giftcards/{GIFTCARDNUMBER} GET** service function.

/giftcards/{GIFTCARDNUMBER} GET

	Field Name	Data Type	Format	Description
Header Data				
1	**Number**	Integer		Gift Card Number
2	**CurrentBalance**	Decimal		Current Gift Card Balance
3	**HasOpenActivity**	"True"/ "False"		If "True" means that the card has open issues with the card and likewise should not be honored.
4	**Expired**	"True"/ "False"		If "True" means that the card has expired and should not be honored.
5	**ExpirationDate**	Date		The Gift Cards expiration date in "YYYY-MM-DDTHH:MM:SS" format.
6	**ActiveDate**	Date		The Gift Cards expiration date in "YYYY-MM-DDTHH:MM:SS" format.
7	**LastActivity**	Date		The Gift Cards expiration date in "YYYY-MM-DDTHH:MM:SS" format.

Your retailer may be using gift cards for special promotions and they will have a value assigned to them, but will have an expiration date. Editing against the Expiration Date is important so as to not honor a card out of date.

4.5 Gift Registry Operations

The Gift Registry Operation service functions support you if you need to support the following functions:

- **Web Gift Registry** – The functions will allow you to support the following:
  - Search and select from active gift registries on a number of variables such as occasion types, registrant number, registrant name, and event date.
  - Present the items and their purchase status for a selected registry.

  - Print a Gift Registry Pick List.
- **Purchases against a Gift Registry** - Identify the items on the orders that you add to Merchant (/orders/new POST) as gift registry items.

Two service functions can be used to support the above:

- **/config/giftregistries** – see 2.1.1 above and Section 4.2.4.
- **/giftregistries/{GIFTREGISTRYID}** – This GET returns the Gift Registry Header and Gift Registry Item. Please refer to Figure 2.2 below and Section 4.5 for more details.

4.5.1 /giftregistries/{GIFTREGISTRYID}

The **/giftregistries/{GIFTREGISTRYID}** GET returns the Gift Registry Header and Gift Registry Items.

Figure 2.2 and Section 2.1.1 provides an overview of the data structure between the header and the detail registry items.

The following describes the fields in the response that you will be returned from the **/giftregistry/{GIFTREGISTRYID} GET** service function.

/giftregistries/{giftregistryId}			GET	
	Field Name	**Data Type**	**Format**	**Description**
Header Data				
1	**GiftRegistryId**	String	10	The Gift Registry Number
2	**Title**	String	45	Gift Registry Title-the name that registrant defined to the system as heading for the occasion/event such as "Mary's Baby Shower"
3	**CustomerId**	String	8	The Customer ID number for the gift registrant
4	**CustomerName**	String	30	The Gift Registrants name
5	**OccasionLabel**	String	20	The description of the occasion.
6	**OccasionDate**	Date		The occasion or event date in "YYYY-MM-DDTHH:MM:SS" format.
7	**OccasionCity**	String	15	The city where the occasion event is taking place.
8	**OccasionState**	String	2	The state code where the occasion event is taking place.
9	**AddressBookSequence**	Integer		The registrants Customer address book sequence number for where the registrant wants the items delivered.
10	**Address**			
11	**StreetLines**	(May be up to four-street lines if in the database)		
12	**Street Line 1**	String	35	Street Address Line 1
13	**Street Line 2**	String	35	Street Address Line 2
14	**Street Line 3**	String	35	Street Address Line 3
15	**Street Line 4**	String	35	Street Address Line 4
16	**City**	String	15	City Name
17	**State**	String	2	State Code
18	**Zip**	Integer	10	Zip Code
19	**Country**	String	3	Country Code

	Field Name	Data Type	Format	Description
Gift Registry Items/SKUs			**More than 1 per Regsitry ID**	
20	**GiftRegistryLine**			
21	**RecId**			
22	**GiftRegistryId**	String	10	The Gift Registry Number
23	**Line**			
24	**Item**	String	20	The Item Number
25	**Sku**	String	8	The SKU ID associated with the item.
26	**Description**	String	30	Item Description
27	**Notes**			Registrant Notes related to the item/SKU
28	**PriceEach**	Decimal	7,2	Selling Price for the Item
29	**QuantityRequested**	Integer	6	The quantity requested by the registrant
30	**QuantityPurchased**	Integer	6	Quantity purchase to date.
31	**CanPurchaseOnWeb**	True/ False		Can this item be purchase on the web (True or False). This is based upon the Item web flag.

Obviously, if the Quantity Requested = QuantityPurchased, then this item is no longer valid for sale.

You will also need to honor the CanPurchaseOnWeb field (Field #31 above) to control whether this item can be ordered on the web.

4.6 Item Operations

While the **/config/catalog** function allows you to download the Merchant web defined active item MCC API provides several service GET functions for a specific Item ID.

- **/items/{ITEMID}** – This GET will return the web enable item record along with the item's SKUs that are web-enabled and their attributes for the specified Item Id.
- **/items/(ITEMID}/upsells** – This GET returns a list of Upsell Items for the specified Item Id.

4.6.1 /items/{ITEMID}

The **/items/ITEMID}** GET will return the web enable item record along with the item's SKUs that are web-enabled and their attributes for the specified Item Id. If the item that you issued the request for is not web-enabled then you will get an error message.

The following describes the fields in the response that you will be returned from the **/items/{ITEMID} GET** service function. The response will include the item record, each of the SKU associated with the item, and the SKU attributes associate with each SKU.

Please refer to Section 2.1.2 for a description of Merchant's Item-SKU-Attributes structure.

/items/{itemId}			GET	
	Field Name	**Data Type**	**Format**	**Description**
Item				
5	**ItemId**	String	20	The Item Number
6	**PlainTextDescription**	String	30	Item Description
7	**FormattedTextDescription**	String		
8	**ProductInfoMessage**	String	320	
9	**DisplayProductInfoMessage**	True/ False		If "True" then display the **ProductInfoMessage**, if "False" then do not display the **ProductInfoMessage.**
10	**VendorCode**	String	6	The vendor code for where the item is normally purchased.
11	**Price**	Decimal	7,2	Selling Price for the Item
12	**SuggestedPrice**	Decimal	8,2	Suggested Retail Price for the Item
13	**TaxCode**	String	8	Special tax code for the item. If this field has a value greater than zero, then Taxable (field #28) must be "True".
14	**Weight**	Decimal	7,2	The weight of the item. Used to calculate shipping charges.
15	**StockUnitOfMeasurement**	String	8	The unit of measure that the item's inventory is maintained in and how it is sold.
16	**Division**	String	3	The Division Code
17	**Department**	String	4	The Item's department code
18	**Class**	String	4	The Item's class code
19	**Subclass**	String	4	The Item's Subclass code
20	**ProductLine**	String	4	The Items Product Line code
21	**Group**	String	4	The Item's Group code
22	**PriceGroup**			The Price Group Code assigned to the item.
23	**SeasonCode**	String	15	The Season Code assigned to the item
24	**CatalogPage**	String	8	The page that the item can be found in the printed catalog.
25	**AttributeFamily**	String	20	The Item attribute family
26	**IsNonMerchandise**	True/ False		If "True" then the Item is a "Non-Merchandise" item and therefore inventory is not maintained for the item. If "False" then this is a merchandise item with inventory tracked.
27	**ShippingChargesApply**	True/ False		If "True" then shipping charges should be charged for the item. If "false" then do not charge shipping charges.
28	**IsTaxable**	True/ False		If "True" then the item is taxable.
29	**IsGiftWrappable**	True/ False		If "True" the item is gift wrap able.
30	**IsPersonalizable**	True/ False		If "True" then the item can be personalized.
31	**IsDiscountable**	True/ False		If "True" then the item can be discounted
32	**DiscountGroup**	String	4	The Discount Group assigned to this item.
33	**SkuCount**	Integer	10	The number of SKU's associated with this item.

	Field Name	Data Type	Format	Description
SKUs	**More that 1 SKU per Item**			
34	**SkuId**	String	8	The SKU ID associated with the item.
35	**ItemId**	String	20	The Item Number
36	**AddOnPrice**	Decimal	7,2	The add-on price provides a way to markup or markdown a SKU from the Item Price. It is used to markup a "XXXL" size or to support permanent markdowns.
37	**VendorItemNumber**	String	20	The Vendor Item Number
38	**Length**	Integer	7	The length of the item's standard packaging for shipping charge calculations.
39	**Width**	Integer	7	The width of the item's standard packaging for shipping calculations.
40	**Depth**	Integer	7	The depth of the item's standard packaging for shipping calculations.
41	**StockUnitOfMeasure**	String	8	The unit of measure that the SKUs inventory is maintained in and how it is sold.

	Field Name	Data Type	Format	Description
SKU Attributes	**More that 1 attribute per SKU**			
42	**Name**	String	10	The attribute type tied to the SKU such as color, size, etc.
43	**Value**	String	10	The attribute value tied to the SKU such as black, red, white orange for a attribute type = color.

4.6.2 /items/{ITEMID}/upsells

The **/items/(ITEMID}/upsells** GET returns a list of Up-sell Items for the specified Item Id. This function can be used to support web requirements for providing Up-sell items as replacement items for the customer to consider.

The following describes the fields in the response that you will be returned from the **/items/{ITEMID}/upsells GET** service function. The response will include the item record, each of the SKU associated with the item, and the SKU attributes associate with each SKU.

/items/{ITEMID}/upsells		**GET**		
	Field Name	**Data Type**	**Format**	**Description**
1	**UpsellItem**	String	20	The Upsell ItemId for the requested ItemId.
2	**Sequence**	Integer		The Sequence Number of this Upsell Item with the Requested Item .
3	**Message**	String		Message for this Upsell Item related to the Requested Item.

	Field Name	Data Type	Format	Description
Upsell Item Detail				
4	**ItemId**	String	20	The Item Number
5	**PlainTextDescription**	String	30	Item Description
6	**FormattedTextDescription**	String		
7	**ProductInfoMessage**	String	320	
8	**DisplayProductInfoMessage**	True/ False		If "True" then display the **ProductInfoMessage**, if "False" then do not display the **ProductInfoMessage.**
9	**VendorCode**	String	6	The vendor code for where the item is normally purchased.
10	**Price**	Decimal	7,2	Selling Price for the Item
11	**SuggestedPrice**	Decimal	8,2	Suggested Retail Price for the Item
12	**TaxCode**	String	8	Special tax code for the item. If this field has a value greater than zero, then Taxable (field #28) must be "True".
13	**Weight**	Decimal	7,2	The weight of the item. Used to calculate shipping charges.
14	**StockUnitOfMeasurement**	String	8	The unit of measure that the item's inventory is maintained in and how it is sold.
15	**Division**	String	3	The Division Code
16	**Department**	String	4	The Item's department code
17	**Class**	String	4	The Item's class code
18	**Subclass**	String	4	The Item's Subclass code
19	**ProductLine**	String	4	The Items Product Line code
20	**Group**	String	4	The Item's Group code
21	**PriceGroup**			The Price Group Code assigned to the item.
22	**SeasonCode**	String	15	The Season Code assigned to the item
23	**CatalogPage**	String	8	The page that the item can be found in the printed catalog.
24	**AttributeFamily**	String	20	The Item attribute family
25	**IsNonMerchandise**	True/ False		If "True" then the Item is a "Non-Merchandise" item and therefore inventory is not maintained for the item. If "False" then this is a merchandise item with inventory tracked.
26	**ShippingChargesApply**	True/ False		If "True" then shipping charges should be charged for the item. If "false" then do not charge shipping charges.
27	**IsTaxable**	True/ False		If "True" then the item is taxable.
28	**IsGiftWrappable**	True/ False		If "True" the item is gift wrap able.
29	**IsPersonalizable**	True/ False		If "True" then the item can be personalized.
30	**IsDiscountable**	True/ False		If "True" then the item can be discounted
31	**DiscountGroup**	String	4	The Discount Group assigned to this item.
32	**SkuCount**	Integer	10	The number of SKU's associated with this item.

	Field Name	Data Type	Format	Description
SKUs	**More that 1 SKU per Item**			
33	**SkuId**	String	8	The SKU ID associated with the item.
34	**ItemId**	String	20	The Item Number
35	**AddOnPrice**	Decimal	7,2	The add-on price provides a way to markup or markdown a SKU from the Item Price. It is used to markup a "XXXL" size or to support permanent markdowns.
36	**VendorItemNumber**	String	20	The Vendor Item Number
37	**Length**	Integer	7	The length of the item's standard packaging for shipping charge calculations.
38	**Width**	Integer	7	The width of the item's standard packaging for shipping calculations.
39	**Depth**	Integer	7	The depth of the item's standard packaging for shipping calculations.
40	**StockUnitOfMeasure**	String	8	The unit of measure that the SKUs inventory is maintained in and how it is sold.

	Field Name	Data Type	Format	Description
SKU Attributes	**More that 1 attribute per SKU**			
41	**Name**	String	10	The attribute type tied to the SKU such as color, size, etc.
42	**Value**	String	10	The attribute value tied to the SKU such as black, red, white orange for a attribute type = color.

4.7 Loyalty Operations

The Loyalty Operation service functions support the following functions:

- **/loyalty/accumulation/{ordered}** – This GET provides the ability to determine the loyalty points associated for a specified web order that has been totally processed (i.e. Merchant Order Status = Closed). This will allow you to generate an email to the customer providing shipment information and the net impact on their loyalty card.
- **/loyalty/inquiry/{LOYALTYNUMBER}** – This GET returns the Loyalty record for the specified Loyalty Number and includes the loyalty level, level description and points balance.

4.7.1 /loyalty/accumulation/{ORDERID}

The **/loyalty/accumulation/{ORDERID}** – This GET provides the ability to determine the loyalty points associated for a specified web order that has been totally processed (i.e. Merchant Order Status = Closed). This will allow you to generate an email to the customer providing shipment information and the net impact on their loyalty card.

The following describes the fields in the response that you will be returned from the **/loyalty/accumulation/{ORDERID}/upsells GET** service function.

/loyalty/accumulation/{orderId}				GET
	Field Name	**Data Type**	**Format**	**Description**
May be more that one				
1	**OrderId**			The Merchant Order Number
2	**SequenceNumber**	Integer		Sequence Number for the returned array
3	**LoyaltyNumber**	Integer		The customers loyalty card number
4	**Date**	DATE		The date and time that the loyalty number was last updated in "YYYY-MM-DDTHH:MM:SS" format.
5	**LoyaltyLevel**	String		The Loyalty level code.
6	**LoyaltyLevelDescription**	String		Loyalty Level Description
7	**AccumulatedPoints**	Decimal		The number of points that where accumulated on this order. And sequence number.
	PointsBalance	Decimal		The balance after this order's accumulated points.

4.7.2 /loyalty/inquiry/{LOYALTYNUMBER}

The **/loyalty/inquiry/{LOYALTYNUMBER}** – This GET returns the Loyalty record for the specified Loyalty Number and includes the loyalty level, level description and points balance.

The following describes the fields in the response that you will be returned from the **/loyalty/inquiry/{LOYALTYNUMBER}/upsells GET** service function.

/loyalty/inquiry/{LOYALTYNUMBER}				GET
	Field Name	**Data Type**	**Format**	**Description**
1	**LoyaltyNumber**	Integer		The customers loyalty card number
2	**LoyaltyLevel**	String		The customers current Loyalty level code.
3	**LoyaltyLevelDescription**	String		The Loyalty level description for the customer
4	**PointsBalance**	Decimal		The balance after this order's accumulated points.

4.8 Orders

The Order API service functions support the following functions:

- **/orders/approveshipments** – This POST function approves the entire order or a specified order line for shipment.
- **/orders/new** – This POST function submits a new order for processing within Merchant.
- **/orders/status/{ORDERID}** – This GET function receives the order status for a specified order.

4.8.1 /orders/approveshipments

Merchant's fulfillment process is expecting your web application to approve an order for shipment prior to Merchant actually shipping the merchandise to the customer. Merchant will

identify to your application when an order shipment is packed and ready to ship awaiting your approval.

While your web application can approve the shipment for an order at any time the intent was for your application to successfully charge the customer's credit card and to then approval the order and/or order lines for shipment.

Merchant's fulfillment process creates one or more shipments that are tied to the order. After a shipment are picked and packed, the shipment header status is set equal to a "W" indicating that it is awaiting your approval.

The **/orders/approveshipments** POST function approves the entire order or a specified order line for shipment.

The following describes the fields and request format for you **/orders/approveshipments POST** message.

/orders/approveshipments		POST			
	Field Name	**Data Type**	**Format**	**Description**	**Edit Rules**
Order Header Data					
1	**OrderId**	String		The Merchant Order Number	Must be a valid Mercant order number
	One or more Order Lines to approve shipment				
2	**OrderLines**	Integer	4	This is a valid Merchant Order Line number, You	Must be a valid Merchant order line number

If the OrderLines node is left empty, it will approve shipping for **all** lines in the order. If you include specific OrderLines, it will approve only the lines specified:

4.8.2 /orders/new

The **"/orders/new" POST** service function allows your application to insert a new Merchant customer order into the Merchant database. The order will consist of the following:

- Order Header Data
- Order Line Request (1 or more)
- Order Tenders (1 or more)

The following defines the edits that your application will need to include to insure that you do not get HTTP response error conditions to your POST request.

/order/new		POST			
	Field Name	Data Type	Format	Description	Edit Rules
Order Header Data					
1	**WebOrderNumber**	String	20	This field represents your unique order identifier that you will generate. MCC will place this string in the Merchant Order File in the AltOrderNumber field, pre-pended with a "w".	This field must not be empty.
2	**CustomerNumber**	String	11	This number represents a valid Merchant Customer Number. This is the customer number that you previously received from MCC through a "/customer/new".	This field must not be empty.
3	**ItemAmount**	Decimal	7,2	The total amount of the orders merchandise (excluding shipping and sales tax).	
4	**TaxAmount**	Decimal	7,2	The amount of sales tax charged on the total order.	
5	**Shipping Amount**	Decimal	6,2	Shipping Amount charged to the order.	
6	**Paid Amount**	Decimal	7,2	The PaidAmount field represents total tendered amount (sum of all **OrderRequestTender / AmountReceived** values) must equal the total amount due (sum of **ItemAmount**, **TaxAmount**, **ShippingAmount**)	

For each OrderRequestLine: (must include one or more)

	Field Name	Data Type	Format	Description	Edit Rules
Order Lines					
7	**Sequence**	Integer	4	This is the Order Line number starting w	This field must be an integer greater than zero.
8	**ItemId**	String	20	Merchant Item Number	This must represent a valid Merchant Web Item Number.
9	**SkuId**	String	8	Merchant SKU ID	This must represent a valid Merchant Web Sku ID.
10	**Quantity**	Integer	6	Ordered quantity	Must be greater than zero.
11	**PriceEach**	Decimal	6,2	Price per unit	
12	**Taxable**	True/ False		True=Item is taxable, False=Item is not taxable	If True, then OrderLine TaxRate must have a value (i.e the tax rate that you want to apply to the item. If False, then OrderLine Taxrate must be zero.
13	**TaxRate**	Decimal	5,2		See #12 above.
14	**Notes**	String	320	These are notes associated with the Order Request Line. These notes will be print on shipping documents.	
15	**PromotionCode**	String	8	Promotion Code that was used for the pricing of this order line.	This must be a valid Merchant Promotion Code.
16	**AddressBookSequence**	Integer	3	This is the Customer's Address Book sequence number for the ship to address that you want to use on the order.	Must be a valid Address Book Sequence Number of the customer.
17	**CarrierCode**	String	8	Carrier Cide for the shipment	Must be a valid Merchant carrier code
18	**ShippingMethod**	String	8	The shipping method for the defined ca	Must be a valid Merchant shipping method.
19	**GiftRegistryId**	String		This defines the Gift Registry Number that you want the customer order to be applied to and	This must represent a valid Merchant Gift Registry ID number.
20	**GiftRegistryLineNumber**	Integer	4	This is the Gift Registry Line Number that this order line Item/SKU should be applied to for purchase update purposes.	Must be a valid Gift Registry Line number for the Gift Registry Number.

	**ItemId & SkuId:** Please refer to Section 4.2.1 to understand how to use the **"/config/catalog"** GET function to load valid Merchant web ItemId (item numbers) and their associated SkuId into your application database. Remember: The **ItemId** and **SkuId** must represent a valid Item/SKU combination (the SKU belongs to the Item).
	**Promotion Code:** Please refer to **Section 4.9.4** to understand how to use the **"/skus/{SKUID}/promotions"** service function to obtain the promotions defined for a given SKU ID.
	**AddressBookSequence:** Please refer to **Section 4.3.2** to understand how to use the **"/customer/{CUSTOMERID}/addressbook"** service function to obtain the address book entries for a given customer ID.
	**CarrierCode & ShippingMethod:** Please refer to **Section 4.2.7** for a description of the **"/config/shippingmethods"** service that will provide the Merchant table to support the

	validation of the "CarrierCode" and "ShippingMethod' fields.
	**GiftRegistryId:** Please refer to **Section 4.2.4** to understand how to use the **/config/giftregistries"** service function to load the active gift registries from the Merchant database.
	**GiftRegistryLineNumber: Please** refer to **Section 4.5.1** to understand how to use the **/giftregistries/{giftregistryId}"** service function to define the items/SKU ID's defined in a given Gift Registry ID. This will provide the Gift Registry Line Number.

For Each OrderRequestTender: (must include 1 or more)

	Field Name	Data Type	Format	Description	Edit Rules
Order Tenders		**(1 or more per order)**			
21	**Sequence**	Integer	4	This is the Tender Sequence number starting with a 1, then 2, etc.	This field must be an integer greater than zero.
22	**Type**	String		This is the type of tender used for the transaction.	Must be defined as one of the following: "Credit Card", "Gift Card" or "Coupon".
23	**TenderIdentifier**	String		This is used to define the card number used in the transaction for either a "Credit Card" or "Gift Card".	See Notes below for additional edits.
24	**ExpirationDate**	Numeric		This reflects the credit card expiration date.	Is required if tender="Credit card" and must be in "MMYY" where "MM" indicates the two digit month and "YY" indicates the two digit year.
25	**AmountReceived**	Decimal		This is the amount received for this tender type.	

TenderIdentifier – Edits

- **Credit Card**- If the Type equal a "Credit Card, then enter a masked credit card number.
- **Gift Card**- If the **Type** equal "Gift Card", and then enter the entire valid gift card number used in the tender.
    - You will also need to perform the following edits when you validated the Gift Card using the **"/giftcards/{GIFTCARDNUMBER}"** service function:
        - **TenderIdentifier** must be a valid gift card number
        - The gift card must not be **expired**
        - The gift card must not have **open activity**
        - The gift card must have **sufficient balance** to cover the tender line's **AmountReceived**

Please refer to **Section 4.4.1** to understand how to use the GET **/giftcards/{GIFTCARDNUMBER}"** service function to validate a gift card and to obtain the additional gift card information.

Merchant Cross-Channel will return a HTTP Status '201" (created) if your POST was successful. The Merchant Order Number will also be returned.

Upon receipt of the request, Merchant Cross-Channel will perform the above edits/validation. If the new order request is not properly formatted or missing information the "/order/new" service function will return an Http Status '400' –Bad Request with the validation errors enumerated.

	MCC does not check inventory availability during your POST function, but the Fulfillment system Order Processing function when it tries to allocate your order will "Void" the order if inventory is not available. A **/order/status/{ORDERID}** would then tell you that your order has been voided and you will need to take corrective action. It is recommended that you execute a **/skus/{SKUID}/locations** to determine inventory availability as close to your /order/new as possible to minimize this situation. At a minimum at the time the customer is adding the SKU to the shopping card or better yet just before the **order/new.**

4.8.3 /orders/status/{ORDERID}

The Merchant Cross-Channel API provides you with status and information such as shipment tracking numbers to allow you to know what is going on within Merchant with your order. You can use the **/orders/status/{ORDERID}** service function at any time to check the status of any of your orders.

Merchant Cross-Channel will provide you with three different status indicators:

- **Order Status** – this is the status of the over-all order. Valid statuses are as follows:
  - **O=Open,** this indicates that the order has been entered into the system.
  - **V=Void,** this indicates that the order has been voided for some reason. In most cases this will be due to the lack of inventory to fill the order.
  - C=Closed, this indicates that the order has been completed and shipped to the customer.
- **Fulfillment Status**– this will define to you where the order is in the Merchant fulfillment process. Valid statuses are as follows:
  - **Order Received Sent to Fulfillment** – This indicates the order has been received but that inventory has not been allocated to the order yet.
  - **Order Sent to Fulfillment** – This indicates inventory has been allocated to all of the order lines and that the order is now in the fulfillment process.
  - **Order is being Fulfilled** - This indicates that your order is in the fulfillment process. Please refer to Figure 4.1 below for the events that occur in this process.
  - **Order Completed & Shipping** – This indicates that the order is completed and has been shipped to your customer.

- **Shipment Header Status -** As described above, Merchant will create shipment headers for each ship-to address in your order. This status indicates the status of each shipment with your order.
    - **P –** This indicates that a picking and packing list has been printed for the shipment and that the shipment is in process of being picked.
    - **W –** This indicates that the shipment has been packed into boxes, shipping labels printed and Merchant is now awaiting your approval to ship.
    - **A –** This indicates that Merchant has received your approval but has not yet completed the process of confirming that it has been shipped.
    - **C –** This indicates that the shipment is completed and shipped to your customer.

The following table defines how these statuses would appear as your order goes through the process and is tied to the events that occur in the process.

MCC/Merchant Fulfillment Order Status Flow

Event	Description	/orders/status/{ORDERID}		
		Order Status	Fulfillment Status	Shipment Header Status
Immediately after Order Insertion	Web/MCC has just processed the /order/new POST	**O**	Order Received Sent to Fulfillment	**NA**
Merchant Order Processing	Order Processing reviews the orders and allocates merchandise that is at the web store. If merchandise is not in the web store then the order is voided.	**O** **or** **V** with line number and reason	Order Received Sent to Fulfillment or Order has been voided	**NA**
Merchant Order Faucet	Orders are processed and shipment headers and lines are created. The picking/packing list is printed for each shipment within the order.	**O**	Order is being fulfilled	**P**
Merchant Pack & Ship	Merchandise is picked and packed in boxes.	**O**	Order is being fulfilled	**W**
	The order/shipment is now waiting for approval to ship from the web site.			
Web/MCC approves for shipment	Web/MCC checks Order/Shipment Status and sees the Shipment Header Status = W			
	Web charges credit card			
	Web/MCC issues /orders/approveshipments POST function.	**O**	Order is being shipped	**A**
Merchant Order Shipment Confirmation	Retailer/Merchant confirms order shipment	**C**	Order is complete and shipped	**C**

Figure 4.1 Order Status Flow

The "**/order/status/{ORDERID}"** service function provides the ability to check the status of an existing order that you had created with the "order/new" service function. The "order/new"

service function had returned the Merchant Order Number to you and it is assumed that you had stored that in your applications database.

You will use this service function to determine when to apply the final credit card charges. You can also use the function to obtain information regarding each shipment tracking numbers.

To link the Merchant shipments to your order lines you will need to compare on the addresses in the Shipment Headers.

The following describes the fields that you will receive in the GET response:

/order/status/{ORDERID}			GET	
	Field Name	**Data Type**	**Format**	**Description**
Order Header Data				
1	**OrderId**	String		
2	**Status Code**	String		
3	**StatusDate**			This is the date and time that Merchant updated the Order or Order Line Status.
4	**FulfillmentStatus**	String		
5	**Lines ApprovedForShipment**	Integer		Multiples identifying the Order Lines that have been approved for shipment.
	Field Name	**Data Type**	**Format**	**Description**
Shipments (Header)		**One of More**		
6	**OrderNumber**	Integer		The Merchant Order ID/number
7	**FirstName**	String	30	Ship-To Customer First Name
8	**LastName**	String	30	Ship-To Customer Last Name
9	**ShipmentNumber**	Integer		Merchant Shipment Number
10	**StatusCode**	String		Shipment Status Code
11	**ShipDate**	Date		The Shipment Dateand time in "YYYY-MM-DDTHH:MM:SS" format.
12	**BoxCount**	Integer		Number of boxes in the shipment
13	**ShipCarrier**	String	8	The Shipment Carrier Code
14	**ShipMethod**	String	8	The Carrier's Ship Method
15	**ShipAddress**			
16	**StreetLines**			
17	**Street Line 1**	String	35	May have from 1-4 Address Lines
18	**Street Line 2**	String	35	
19	**Street Line 3**	String	35	
20	**Street Line 4**	String	35	
21	**City**	String	15	City Name
22	**State**	String	2	State Code
23	**Zip**	Integer	10	Zip Code
24	**Country**	String	3	Country Code

	Field Name	Data Type	Format	Description
Boxes within the Shipment		**One of More**		
25	**BoxNumber**	String		The Box Number within the shipment
26	**TrackingNumber**	String		The Shipment Tracking Number received from the carrier.
27	**ShippingCarrier**	String	8	The Shipment Carrier Code
28	**ShippingMethod**	String	8	The Carrier's Ship Method

4.9 SKUs Operations

The Order API service functions support the following functions:

- **/skus/{SKUID}** – This GET function returns a SKU for a specific SKUID.
- **/skus/{SKUID}/image** – This GET returns a JPG image for a specified SKUID.
- **/skus/SKUID}/locations** – This GET returns the SKU location records for a specified SKUID.
- **/skus/{SKUID}/promotions –** This GET returns a list of active promotions, in priority order, applicable to the specified SKUID.

These service functions will only return the information if the SKUID is web-enabled and active.

The following sections describe each of these service functions in more detail.

4.9.1 /skus/{SKUID}

The **/skus/{SKUID}** GET function returns a SKU for a specific SKUID.

The function is designed to support situations where your application may need information for a specific SKU.

The following describes the fields that you will receive in the response to your GET.

/skus/{SKUID}			GET	
	Field Name	Data Type	Format	Description
SKUs	More that 1 SKU per Item			
34	SkuId	String	8	The SKU ID associated with the item.
35	ItemId	String	20	The Item Number
36	AddOnPrice	Decimal	7,2	The add-on price provides a way to markup or markdown a SKU from the Item Price. It is used to markup a "XXXL" size or to support permanent markdowns.
37	VendorItemNumber	String	20	The Vendor Item Number
38	Length	Integer	7	The length of the item's standard packaging for shipping charge calculations.
39	Width	Integer	7	The width of the item's standard packaging for shipping calculations.
40	Depth	Integer	7	The depth of the item's standard packaging for shipping calculations.
41	StockUnitOfMeasure	String	8	The unit of measure that the SKUs inventory is maintained in and how it is sold.

	Field Name	Data Type	Format	Description
SKU Attributes	More that 1 attribute per SKU			
42	Name	String	10	The attribute type tied to the SKU such as color, size, etc.
43	Value	String	10	The attribute value tied to the SKU such as black, red, white orange for a attribute type = color.

4.9.2 /skus/{SKUID}/image

The **skus/{SKUID}/image** GET returns a JPG image for a specified SKUID.

Merchant supports defining an image to the item and to each SKU. If the SKU does not have an image assign to it, then the GET will return the Item image.

Again the SKU must be web-enabled and active.

If the SKU is not web enabled or there is no image on a web enabled SKU, the API will return an http '404' (not found) response. The image is returned in JPG format and as it exists on disk; it is not reformatted or resized.

4.9.3 /skus/{SKUID}/locations

The **/skus/SKUID}/locations** GET returns the SKU location records for a specified SKUID.

The SKU Location records define what the on-hand and allocated quantities are at each store location. Please refer to Section 2.1.5 for a description of SKU Location records and its inclusion of the region/aisle/bin location of the on-hand quantity within each store.

The SKU Location records are to be used for the following applications:

- Determining the SKU Available Quantity to sell in your web-store.

- Providing inventory availability at all store locations to support a customer pick-up function.

The following describes the fields that you will receive in the response to your GET.

/skus/{SKUID}/location			GET	
	Field Name	**Data Type**	**Format**	**Description**
1	**SKU**	String	8	The SKU ID that you requested
2	**ReservedAmount**	Integer	7	The reserved quantity for this SKU across all locations.
Stores Locations		**An arrray of all locations that have inventory for the requested SKU ID**		
3	**Store**	String	4	The store that the inventory is located in.
4	**Region**	String	1	The region within the store that the inventory is located in.
5	**Aisle**	String	2	The Aisle location within the region within the store that the inventory is located in
6	**Bin**	String	4	The Bin location within the Aisle within the region within the store that the inventory is located in
7	**OnHand**	Integer	7	The on-hand quantity at this location.
8	**Allocated**	Integer	7	The allocated quantity at his location
	Available Qty = OnHand - Allocated			

4.9.4 /skus/{SKUID}/promotions

The **/skus/{SKUID}/promotions** GET returns a list of active promotions, in priority order, applicable to the specified SKUID and for your web store.

MCC will pull only those promotions that are active on the date of your request. That is the request date is between the Effective Date (Field # 7 below) and the Expire Date (Field #8).

The following describes the fields that you will receive in the response to your GET.

/skus/{SKUID}/promotions			GET	
	Field Name	Data Type	Format	Description
Stores Locations		An arrray of all locations that have inventory for the requested SKU ID		
1	**Priority**	Decimal	5,2	The store that the inventory is located in.
2	**PromotionCode**	String	15	The Promotion Code
3	**SkuId**	String	8	The SKU ID for the promotion
4	**NewPrice**	Decimal	7,2	The new price for this promotion code
5	**OriginalPrice**	Decimal	7,2	The original price for the SKU ID
6	**Description**	String		The description of the promotion
7	**EffectiveDate**	Numeric		The date and time that the promotion starts in "YYYY-MM-DDTHH:MM:SS" format.
8	**ExpireDate**			The date and time that the promotion ends in "YYYY-MM-DDTHH:MM:SS" format.
9	**IsTaxExempt**	True/ False		Identifies this promotion is being used to define the SKU ID as tax exempt during this effective and expiration dates.
10	**OrderTypeInclusion**	String		"Include" or "Exclude" - defines whether or not the following Order Type codes are to be included or excluded from the promotion.
11	**OrderType**	String		Order Type Codes (1 or more) that are to be included or excluded.
12	**CustomerTypeInclusion**	String		"Include" or "Exclude" - defines whether or not the following Customer Type codes are to be included or excluded from the promotion.
13	**CustomerType**	String		Customer Type Codes (1 or more) that are to be included or excluded.
14	**SourceCodeInclusion**	String		"Include" or "Exclude" - defines whether or not the following Source Codes are to be included or excluded from the promotion.
15	**SourceCode**	String		Source Codes (1 or more) that are to be included or excluded.
16	**PriceLevelInclusion**	String		"Include" or "Exclude" - defines whether or not the following Price Levels codes are to be included or excluded from the promotion.
17	**PriceLevel**	String		Price Level Codes (1 or more) that are to be included or excluded.
18	**IsCouponPromotion**	True/ False		"True" indicates that this promotion is a coupon promotion and requires a promotion code. "False" it is not a coupon promotion.
19	**CouponCode**	String		The Coupon Code to be used to validate the promotion against.
20	**LoyaltyLevel**	String		The loyalty level that the promotion is valid against
21	**IsEligibleForLoyaltyPoints**	True/ False		"True" indicates that the items/SKUs in this promotion are elibible to receive loyalty points, "False" defines that the Items/SKUs in this promotion are not eligible to receive loyalty points.

The promotion list needs to be evaluated before applying to the SKUID for the customer as follows:

- MCC is giving these promotions in the sequence of their priority. The first promotion that meets the evaluation criteria below can be applied to your transaction.
- The promotion includes a set of criteria that may or may not exist in a particular promotion record. If nothing is defined, then apply that particular promotion to your transaction and end your evaluation of the list.
- **IsTaxExempt** – This promotion is designed to support tax exempt merchandise during a date range. If True then do not charge sales tax.
- **Order Type Inclusion** – This defines if the following Order Types are to be "included" or "Excluded". The Order Type for a **web order is "ORDR"** so apply that to the "Inclusion" or "Exclusion" logic.
- **Order Type** – this is a list of Order Types that are to be "Included" or Excluded" from the promotion.
- **Customer Type Inclusion:** -Ignore these promotion criteria as MCC does provide you with the Customer Type in your GETs, POSTs, or PUTs at this time.
- **Customer Type:** - Ignore
- **Source Code Inclusion: -** This defines if the following Source Codes are to be "Included" or "Excluded". The Source Code will be the Source Code that you captured if you are supporting this function.

You also have the list of valid Source Codes. See Section 4.2.8.

- **Source Code: -** this is a list of Source Codes that are to be "Included" or Excluded" from the promotion.
- **Price Level Inclusions:** The Price Level that this is referencing is found in the Customer record. This will not have been set for a "New" customer but may have been updated by the retailer for existing customer. A /customers/{CUSTOMERID} would return the current Price Level assigned to the customer. This defines if the following Price Levels are to be "included" or "Excluded" from the promotion.
- **Price Level: -** – this is a list of Price Levels that are to be "Included" or Excluded" from the promotion.
- **Is Coupon Promotion: -** This "True" or "False" defines whether this promotion is a coupon promotion and if so then the Coupon Code field will define the coupon code that you will have to prompt and edit for.
- **Coupon Code:** - The coupon code or number that you will need to edit against.
- **Loyalty Level:** - This identifies the loyalty level that this promotion is to be applied to. Use the customer loyalty number in the customer record to issue the **/loyalty/inquiry/{LOYALTYNUMBER}** to obtain the customers loyalty level.
- **Is Eligible for Loyalty Points:** - This "True" or "False" will identify whether this promotion is eligible for loyalty points. It is provided so that the customer can be warned as to not to expect loyalty points.

Buy x Get Y Promotions are not supported.

4.10 Store Operations

4.10.1 /stores/{STOREID}

The Stores Operations service function **(/stores/{STOREID})** returns a Store record for a specific StoreId.

Please also refer to Section 4.2.9 if you need to pull a list of all the stores.

The following describes the fields that you will receive in the response to your GET.

/stores/{STOREID}			GET	
	Field Name	Data Type	Format	Description
Header Data				
1	**StoreId**	String	4	Store ID
2	**StoreName**	String	30	Store Name
3	**Division**	String	3	The division code that the store belongs to
4	**Address**			The Store's Address
5	**StreetLines**			
6	**Street Line 1**	String	35	Street Address Line 1
7	**Street Line 2**	String	35	Street Address Line 2
8	**Street Line 3**	String	35	Street Address Line 3
9	**City**	String	15	City Name
10	**State**	String	2	State Code
11	**Zip**	Integer	10	Zip Code
12	**Country**	String	3	Country Code
13	**Phone**	String	20	The store's phone number
14	**Is Corporate**	True/ False		Define's whether the store is also a Corporate location. "True" it is, "False" it is not.
15	**Is Store**	True/ False		Define's whether the store is a selling store as opposed to a warehouse.
16	**TaxRate**	Decimal	6,4	Defines the tax rate for the store.

4.11 Vendor Operations

4.11.1 /vendors/{VENDORCODE}

The vendor operation service function **(/vendors/{VENDORCODE})** provides the ability to inquire on the Vendor Code and to get back the vendor name.

Please also refer to Section 4.2.11 if you need to pull a list of all the vendors.

The following describes the fields that you will receive in the response to your GET.

/vendors/{VENDORCODE}			GET	
	Field Name	**Data Type**	**Format**	**Description**
Header Data				
1	Code	String	6	Vendor Code
2	Name	String	35	Vendor Name

List of Figures