



Reference Guide
Web2PS - Fulfilment API
Web Service Descriptions and Examples
Add On
V21.1.0200

Copyright © 1997 - 2021 by Electronics for Imaging, Inc. All Rights Reserved.
EFI PrintStream | V21.1.0200 Web2PS Fulfillment API Reference Guide

This publication is protected by copyright, and all rights are reserved. No part of it may be reproduced or transmitted in any form or by any means for any purpose without express prior written consent from Electronics for Imaging, Inc. Information in this document is subject to change without notice and does not represent a commitment on the part of Electronics for Imaging, Inc.

Patents

This product may be covered by one or more of the following U.S. Patents: 4,716,978, 4,828,056, 4,917,488, 4,941,038, 5,109,241, 5,170,182, 5,212,546, 5,260,878, 5,276,490, 5,278,599, 5,335,040, 5,343,311, 5,398,107, 5,424,754, 5,442,429, 5,459,560, 5,467,446, 5,506,946, 5,517,334, 5,537,516, 5,543,940, 5,553,200, 5,563,689, 5,565,960, 5,583,623, 5,596,416, 5,615,314, 5,619,624, 5,625,712, 5,640,228, 5,666,436, 5,745,657, 5,760,913, 5,799,232, 5,818,645, 5,835,788, 5,859,711, 5,867,179, 5,940,186, 5,959,867, 5,970,174, 5,982,937, 5,995,724, 6,002,795, 6,025,922, 6,035,103, 6,041,200, 6,065,041, 6,112,665, 6,116,707, 6,122,407, 6,134,018, 6,141,120, 6,166,821, 6,173,286, 6,185,335, 6,201,614, 6,215,562, 6,219,155, 6,219,659, 6,222,641, 6,224,048, 6,225,974, 6,226,419, 6,238,105, 6,239,895, 6,256,108, 6,269,190, 6,271,937, 6,278,901, 6,279,009, 6,289,122, 6,292,270, 6,299,063, 6,310,697, 6,321,133, 6,327,047, 6,327,050, 6,327,052, 6,330,071, 6,330,363, 6,331,899, 6,340,975, 6,341,017, 6,341,018, 6,341,307, 6,347,256, 6,348,978, 6,356,359, 6,366,918, 6,369,895, 6,381,036, 6,400,443, 6,429,949, 6,449,393, 6,476,927, 6,490,696, 6,501,565, 6,519,053, 6,539,323, 6,543,871, 6,546,364, 6,549,294, 6,549,300, 6,550,991, 6,552,815, 6,559,958, 6,572,293, 6,590,676, 6,606,165, 6,633,396, 6,636,326, 6,643,317, 6,647,149, 6,657,741, 6,662,199, 6,678,068, 6,707,563, 6,741,262, 6,748,471, 6,753,845, 6,757,436, 6,757,440, 6,778,700, 6,781,596, 6,816,276, 6,825,943, 6,832,865, 6,836,342, RE33,973, RE36,947, D341,131, D406,117, D416,550, D417,864, D419,185, D426,206, D439,851, D444,793.

Trademarks

The APPS logo, AutoCal, Auto-Count, Balance, Best, the Best logo, BESTColor, BioVu, BioWare, ColorPASS, Colorproof, ColorWise, Command WorkStation, CopyNet, Cretachrom, Cretaprint, the Cretaprint logo, Cretaprinter, Cretaroller, DockNet, Digital StoreFront, DocBuilder, DocBuilder Pro, DocStream, DSFdesign Studio, Dynamic Wedge, EDOX, EFI, the EFI logo, Electronics For Imaging, Entrac, EPCount, EPPhoto, EPRegister, EPStatus, Estimate, ExpressPay, Fabrivu, Fast-4, Fiery, the Fiery logo, Fiery Driven, the Fiery Driven logo, Fiery JobFlow, Fiery JobMaster, Fiery Link, Fiery Prints, the Fiery Prints logo, Fiery Spark, FreeForm, Hagen, Inktenity, Inkware, Jetrion, the Jetrion logo, LapNet, Logic, MiniNet, Monarch, MicroPress, OneFlow, Pace, PhotoXposure, PressVu, Printcafe, PrinterSite, PrintFlow, PrintMe, the PrintMe logo, PrintSmith, PrintSmith Site, PrintStream, Print to Win, Prograph, PSI, PSI Flexo, Radius, Rastek, the Rastek logo, Remoteproof, RIPChips, RIP-While-Print, Screenproof, SendMe, Sincolor, Splash, Spot-On, TrackNet, UltraPress, UltraTex, UltraVu, UV Series 50, VisualCal, VUTEk, the VUTEk logo, and WebTools are trademarks of Electronics For Imaging, Inc. and/or its wholly owned subsidiaries in the U.S. and/or certain other countries.

All other terms and product names may be trademarks or registered trademarks of their respective owners and are hereby acknowledged.

Table of Contents

Introduction	13
Overview	13
Contact Information	13
Glossary of Terms.....	14
Workflow - Sample Use Case	18
Add Contact.....	22
Narrative	22
Request Transaction	22
Sample Request Transaction	22
Response Transaction	24
Add Digital Asset.....	25
Narrative	25
Request Transaction	25
Sample Request Transaction	26
Response Transaction	26
Authenticate	27
Narrative	27
Request Transaction	27
Cancel Pick	29
Narrative	29
Request Transaction	29
Sample Reason Code	29
Sample Request Transaction	30
Response Transaction	30
Delete Contacts	31
Narrative	31
Request Transaction	31
Sample Request Transaction	32
Response Transaction	32
Encrypt Password.....	33
Narrative	33
Request Transaction	33
Sample Request Transaction	34
Response Transaction	34
Generate Corporate Structure.....	35
Narrative	35
Request Transaction	35

Sample Request Transaction	36
Response Transaction	36
Get Active Profile	37
Narrative	37
Request Transaction	37
Sample Request Transaction	38
Response Transaction	38
Get Additional Information	39
Narrative	39
Request Transaction	39
Sample Request Transaction	39
Response Transaction	40
Get Back Orders for Recipients	41
Narrative	41
Request Transaction	41
Sample Request Transaction	42
Response Transaction	42
Get Bill To Ship To	43
Narrative	43
Request Transaction	43
Sample Request Transaction	44
Response Transaction	44
Get Calc Method	45
Narrative	45
Request Transaction	45
Sample Request Transaction	46
Response Transaction	46
Get Contact List	47
Narrative	47
Request Transaction	47
Sample Request Transaction	48
Response Transaction	48
Get Contact List Detail	49
Narrative	49
Request Transaction	49
Sample Request Transaction	50
Response Transaction	51
Get Country Detail	52
Narrative	52

Request Transaction	52
Sample Request Transaction	53
Response Transaction	53
Get Country List	54
Narrative	54
Request Transaction	54
Sample Request Transaction	55
Response Transaction	55
Get CustID by ACNO	56
Narrative	56
Request Transaction	56
Sample Request Transaction	57
Response Transaction	57
Get Credit Card List	58
Narrative	58
Request Transaction	58
Sample Request Transaction	58
Response Transaction	59
Get Credit Card Processing Fee	60
Narrative	60
Request Transaction	60
Sample Request Transaction	61
Response Transaction	61
Get Credit Card Types	62
Narrative	62
Request Transaction	62
Sample Request Transaction	63
Response Transaction	63
Get Customer Attributes	64
Narrative	64
Request Transaction	64
Schema	64
Sample Request Transaction	64
Response Transaction	65
Get Customer Batches	69
Narrative	69
Request Transaction	69
Sample Request Transaction	69
Response Transaction	71

Get Customer Details	72
Narrative	72
Request Transaction	72
Sample Request Transaction	73
Response Transaction	73
Get FF Transaction List	74
Narrative	74
Request Transaction	74
Sample Request Transaction	74
Response Transaction	75
Get Help and Feedback Email Address	76
Request Transaction	76
Sample Request Transaction	77
Response Transaction	77
Get Inventory Stock Level	78
Narrative	78
Request Transaction	78
Sample Request Transaction	78
Response Transaction	78
Get Item Detail	79
Narrative	79
Request Transaction	79
Sample Request Transaction	79
Response Transaction	80
Get Item Discount.....	83
Narrative	83
Request Transaction	83
Sample Request Transaction	84
Response Transaction	84
Get Item No	85
Narrative	85
Request Transaction	85
Sample Request Transaction	86
Response Transaction	86
Get Item Notes	87
Narrative	87
Request Transaction	87
Sample Request Transaction	88
Response Transaction	88

Get Item Price	89
Narrative	89
Request Transaction	89
Sample Request Transaction	89
Response Transaction	90
Get Items By Job	91
Narrative	91
Request Transaction	91
Sample Request Transaction	92
Response Transaction	92
Get Nom Group Lists	93
Narrative	93
Request Transaction	93
Sample Request Transaction	93
Response Transaction	94
Get Nomination List	95
Narrative	95
Request Transaction	95
Sample Request Transaction	96
Response Transaction	96
Get Order Address Info.....	98
Narrative	98
Request Transaction	98
Sample Request Transaction	99
Response Transaction	99
Get Order Details.....	100
Narrative	100
Request Transaction	100
Sample Request Transaction	101
Response Transaction	101
Get Orders By Job.....	104
Narrative	104
Request Transaction	104
Sample Request Transaction	105
Response Transaction	105
Get Package Pricing.....	107
Narrative	107
Request Transaction	107
Sample Request Transaction	107
Response Transaction	108

Get PriceGroup Name	110
Narrative	110
Request Transaction	110
Sample Request Transaction	111
Response Transaction	111
Get Project Setting	112
Narrative	112
Request Transaction	112
Sample Request Transaction	112
Response Transaction	113
Get Projects	114
Narrative	114
Request Transaction	114
Sample Request Transaction	114
Response Transaction	115
Get PSItem Via Number	116
Narrative	116
Request Transaction	116
Sample Request Transaction	117
Response Transaction	117
Get Reason Code Description	118
Narrative	118
Request Transaction	118
Sample Request Transaction	119
Response Transaction	119
Get Recipient Attributes	120
Narrative	120
Request Transaction	120
Schema	120
Sample Request Transaction	121
Response Transaction	121
Get Recipient Information	123
Narrative	123
Request Transaction	123
Sample Request Transaction	124
Response Transaction	125
Get Results	126
Narrative	126
Request Transaction	126

Sample Request Transaction	127
Response Transaction	128
Get Sales Tax.....	130
Narrative	130
Request Transaction	130
Sample Request Transaction	131
Response Transaction	131
Get Shipping Preferences	132
Narrative	132
Request Transaction	132
Sample Request Transaction	133
Response Transaction	134
Get SO List.....	136
Narrative	136
Request Transaction	136
Sample Request Transaction	137
Response Transaction	138
Get SYS Param	139
Narrative	139
Request Transaction	139
Sample Request Transaction	140
Response Transaction	140
Directory of System Parameters	141
Get Token.....	144
Narrative	144
Request Transaction	144
Sample Request Transaction	145
Response Transaction	145
Get Unreceived PO Receipts	146
Narrative	146
Request Transaction	146
Sample Transaction Request	146
Response Transaction	147
Get Up Sell and Cross Sell Items	150
Narrative	150
Request Transaction	150
Sample Request Transaction	150
Response Transaction	151
Get Web Description	154

Narrative	154
Request Transaction	154
Sample Request Transaction	155
Response Transaction	155
Get Web User	156
Narrative	156
Request Transaction	156
Sample Request Transaction	156
Response Transaction	157
Hello World	158
Narrative	158
Request Transaction	158
Sample Request Transaction	158
Response Transaction	158
Inventory ASN	159
Narrative	159
Request Transaction	159
Sample Request Transaction	159
Response Transaction	160
Inventory Create	161
Narrative	161
Request Transaction	161
Sample Request Transaction	164
Response Transaction	167
Inventory Create by Batch	168
Narrative	168
Request Transaction	168
Sample Request Transaction	168
Response Transaction	174
Inventory Update	176
Narrative	176
Request Transaction	176
Sample Request Transaction	178
Response Transaction	180
Inventory Update by Batch	181
Narrative	181
Request Transaction	181
Sample Request Transaction	181
Response Transaction	187
Is Recipient Taxable	188

Narrative	188
Request Transaction	188
Sample Request Transaction	188
Response Transaction	189
Login	190
Narrative	190
Request Transaction	190
Sample Request Transaction	190
Response Transaction	191
Main Interface	192
Narrative	192
Request Transaction	192
Sample Request Transaction	192
Response Transaction	193
Save Nom Group Item	196
Narrative	196
Request Transaction	196
Sample Request Transaction	197
Response Transaction	198
Save Product	199
Narrative	199
Request Transaction	199
Sample Request Transaction	199
Response Transaction	201
Service Test	202
Narrative	202
Request Transaction	202
Submit Order	204
Narrative	204
Request Transaction	204
Sample Request Transaction	204
Sample Request Transaction with External Payment Information.....	207
Update Recipient Attributes	209
Narrative	209
Request Transaction	209
Schema	209
Sample Request Transaction	210
Response Transaction	211
Validate Cost Center	212

Narrative	212
Request Transaction	212
Sample Request Transaction	212
Response Transaction	213
Write Item Notes	214
Narrative	214
Request Transaction	214
Sample Request Transaction	214
Response Transaction	215
Web2PS Login Test Setup	216
Adding a Security Event	216
Create a Security User	216
Map Security User	217
Create Recipient	217
Determine the Customer Account Number	218
Run Login Test	218
How To Attach Files to Order Line Items	220
Programming Examples	220
Sample Code	220
VB.Net example (Microsoft)	220
VB.Net example (PrintStream)	221
C# Example (Microsoft)	221
C# Example (Printstream)	222
Utilize the PrintStream Dll to Encode a File	223
How to Setup Item Notes XML:	223
Canceling an Order	226
Updating/Editing an Order	227
Working with Kits	228
Adding Order Variable Fields	230

Introduction

Overview

This is a reference guide which lists the functions used in the Web2PS Fulfillment API. This guide is meant for the experienced website administrator who is familiar with implementing APIs.

Contact Information

EFI Support

US Phone:	855.334.4457 (first select option 3, then press option 8, then press option 1)
US Fax:	415.233.4157
US E-mail:	PrintStream-support@efi.com
Regular Service Desk hours are 8:00 AM to 7:00 PM Central Time, Monday – Friday. Outside of these hours, you may leave a voice mail message and an on-call support representative will be paged. Response time is based on the severity of the issue.	
Note	For problems involving infrastructure (i.e., computers, networks, operating systems, backup software, printers, third-party software, etc.), contact the appropriate vendor. EFI cannot support these types of issues.

EFI Professional Services

US Phone:	651.365.5321
US Fax:	651.365.5334
E-Mail:	ProfessionalServicesOperations@efi.com
EFI Professional Services can help you perform EFI software installations, upgrades, and updates. This group can also help you implement, customize, and optimize your EFI software plus offer a range of training options.	

Glossary of Terms

The following list provides the meaning of specific terms used throughout this document.

Term	Description	Synonyms / Examples
Item	Items are what people order on the web and place into their shopping cart. Items are tied to the inventory system to track on hand quantity, order commitments and history. Items can also be marked as "Print on Demand" and if so, are linked to a "Master Quote Template" which contains the specifications and processes that are used to manufacture the item on demand. Items can also be marked as "Kits", which in turn contain a bill of materials which identifies all the individual items that make up the kit.	Products Product Codes SKU's Part Numbers
Item Attributes	PrintStream provides the ability to add user defined "attributes" to any item. These attributes are typically used during the ordering process to provide further information about an item. A user typically responds to these attributes by choosing from a multiple-choice list box or entering freeform text, the responses are then used for reporting purposes or for analyzing ordering trends or patterns. They may also provide necessary information such as Dates or coupon codes or other data that needs to be passed to a print on demand variable data application.	Variable Fields e.g. ... Please enter the coupon code that should print on the card. Event Date on Letter Billing Code
Item Options	Item options are typically used to allow someone placing an order to "customize" the product in some way. Item Options are only applicable to items that are printed on demand. Options will generally also contain pricing information so as the user selects the option it potentially adds additional charges to the order.	Shrink Wrap Gloss Coating Drill Holes Print on Back of Card Send me a Color Proof
Item Nomination Profile / Nomination Group	Item Nomination Profiles are basically a collection of items. Rather than showing each user a complete list of all items – it is often necessary to filter in some way the items that a user is allowed to order. Once a Nomination Profile has been created – items are then added to the profile. An item can appear in multiple profiles (catalogs) and that item can then have specific ordering rules based on the profile it is in. Rules include changing price, max order qty, descriptions, approvals etc. Item Nomination Profiles – can also be configured with user defined "Groups" (sections). This provides an easy way for a user to navigate through different areas of the catalog (profile) to quickly find items they wish to order.	Catalogs Main Catalog Summer Catalog Retail Catalog Nomination Groups. E.g.... Stationery Manuals Marketing Collateral Premium Items
Company Code	PrintStream provides the ability to configure the system with multiple companies (financial entities) on the same database. When a fulfillment order is placed, PrintStream must know which Company the order is being placed with. All the web services require that Company Code be passed to the system as part of the authentication process. *the StkHeader section describes sharing items across companies.	01 = ABC Printing 02 = Target Mailing Co. 03 = Better Fulfillment Inc.
Plant Code	PrintStream provides the ability to configure the system with multiple physical plants (locations) on the same database. When a fulfillment order is placed, PrintStream must know which Plant will fulfill the order. All the web services require that Plant Code be passed to the system as part of the authentication process. *the StkHeader section describes sharing items across plants.	0110 = ABC – West Coast Div 0120 = ABC – Mid West Div 0130 = ABC – Northeast Div 0220 = Target – Illinois Plant

Term	Description	Synonyms / Examples
StkHeader	In situations where multiple companies and plants exist in the same database, PrintStream provides a method to associate which items are common to each other across each of the companies and plants. When an item is held in inventory or printed on demand at a particular plant or company – the item costs, stocking rules and manufacturing methods may change dramatically from another plant or company. So, it is necessary to hold this item information at a plant and company level. However, it would not be appropriate to have the same item appear multiple times on the web for a customer – just because there are multiple plants or companies. Each time an item is created at a plant, PrintStream asks the user to enter the product code that the customer wishes to use for this item. PrintStream immediately creates a record in the StkHeader table that identifies a unique inventory code for that customer. Each time a different plant needs to carry or manufacture this same item (with the same inventory code) – PrintStream will create a cross reference between all plants that identifies which item numbers in each plant correspond to the same inventory code. When Item Nomination Profiles (catalogs) are created, PrintStream always builds these from items in the StkHeader table.	STKHEADER Table Customer = ABC Company Inventory Code = ABC-1234 StkHeader ID = 1001 <i>Item Nominations (Catalogs) are created from items in this table</i> PAPSIZE Table Plant 01 Customer = ABC Company Inventory Code = ABC 1234 Item # = 3456 StkHeader ID = 1001 Plant 02 Customer = ABC Company Inventory Code = ABC 1234 Item # = 4567 StkHeader ID = 1001
Recipient	A recipient is the person who places the order and or receives the products ordered, and or is billed for the products being ordered. Every person placing an order will be stored in the RECIPIENT table in PrintStream and assigned a unique Recipient ID number.	User ID Login Name Consumer Store Location
Customer	A Customer represents the Business or Individual that the items belong to that are being fulfilled. In a Business to Business scenario – there will typically be one Customer with multiple recipients tied to that customer. Recipients are the people or office locations that place orders from that business. In a Business to Consumer scenario the Customer may represent the company selling the product to the end consumer – and each recipient would be an end consumer. PrintStream also provides the ability to “elevate” a recipient to customer level – so you can track outstanding invoices and receivables balances if you are billing the end consumer for the items shipped to them. PrintStream also supports the ability to accept orders from individual recipients but bill the customer for the items instead of the end consumer.	Customer ABC Financial Services Recipients Chicago Sales Office San Francisco Broker Office New York Funding Center Customer Big Cookie Company Recipients Chris Anderson David Williams Alicia Keystone
Account No	Every Customer has a unique Account Number. This account number is used throughout the system to link items, projects recipients, profiles etc to the correct customer. When a fulfillment order is placed, PrintStream must know which Account No the order is for.	10002 = ABC Financial Servcs 26638 = Big Cookie Company
Project (Fulfillment Project)	All fulfillment orders must be associated with a specific Fulfillment Project. Projects are configured with Item Nomination Profiles (Catalogs) and order processing rules such as back-order rules, packing slip format, effective dates, in many cases billing is based on project as well. Most customers may only need one fulfillment project. If however, a customer wanted to have very specific rules for handling orders that go to end consumer's vs.	Consumer Product Literature Internal Corporate Materials New Customer Acquisition

Term	Description	Synonyms / Examples
	the products that are ordered by its internal staff and offices it may be appropriate to setup separate projects for the two different target audiences.	
Sales Order	A Sales order basically represents a request for one or more items to be fulfilled. Sales orders can be entered on the web, they can be manually entered by a call center, they can be imported from a file, or they can be streamed into PrintStream via a web service.	Order Shopping Cart Session
Sales Order Line Item	Each Sales Order will contain one or more Sales Order Line Items. Every Sales Order Line Item must be linked to an "Item".	Line Items Order Items Products Ordered
Order By	Every Sales Order must contain a valid Recipient ID that represents the recipient placing the order.	100987 = Order By Recipient
Ship To	Every Sales Order must contain a valid Recipient ID that represents the recipient receiving the products	100987 = Ship To Recipient
Bill To	Every Sales Order must contain a valid Recipient ID that represents the recipient that the items are being shipped to.	202514 = Bill To Recipient
Master Template / Master Quote / Quote Template	A Master Template (or Master Quote) is simply an estimate that has been entered into PrintStream that contains all the manufacturing instructions for producing an item on demand.	Recipe Book Collection of Bills of Material
Package (in a Master Quote)	A Package is a unique combination of components and processes (a recipe) that determines how the item being ordered is manufactured on demand. Packages are built inside the PrintStream Estimating and Order Entry program within a quote number. This quote number is then marked as a "Master Template". Each Master Template Quote can hold up to 999 different packages. Typically, items that are somewhat similar to each other or are constructed from similar components or processes are entered as packages in the same Master Quote. Each Package is then tied to a specific item number – so when the order is placed, PrintStream can find the correct Master Quote Package (recipe) to manufacture the item being ordered.	Recipe Bill of Materials
Order Release	Once an order has been placed and saved in the fulfillment system it then needs to be released into production for picking or manufacturing. This "Release" process can happen automatically on an order by order basis in real time – or orders can be queued up for batch release.	Order Processing Release Orders for Picking Print Pick Pack Tickets Create Manufacturing Jobs
Pick ID	Once an order has been Released – PrintStream generates a unique Pick ID that contains a list of all items that need to be manufactured or picked to fulfill the order. Based on back order rules and on hand qty's – the system may not necessarily include all the items on the original order on the Pick ID. A subsequent Pick ID with its own unique number may be created at a later stage as items come off back order to fulfill the remainder of the original order. In this case there would be multiple Pick ID's for that order. The Pick ID is also included on the Pick Pack ticket as a unique bar code. This is the bar code that will be scanned into the shipping screen for shipping the order.	Order #123456 – 02/20/11 100 of Item 234 100 of item 345 100 of item 456 Pick ID #5678 – 02/20/11 100 of item 234 60 of item 345 Pick ID #6789 – 03/05/11 40 of item 345 100 of item 456
Shipping Package	When a user scans the Pick ID barcode on the Pick Pack ticket – they then confirm that they are shipping all the items shown. Upon confirmation of this – the system then generates a unique "Package ID" so a	Pick ID #6789 – 03/05/11 40 of item 345 100 of item 456



Term	Description	Synonyms / Examples
	record can be made of what items actually shipped on that Shipping Package. If all items cannot be shipped at once – then a subsequent shipping process would create a second unique Shipping Package ID. I.e. Multiple Shipping Packages per Pick ID.	Ship Package #789 03/05/11 40 of item 345 20 of item 456 Ship Package #891 03/07/11 80 of item 456
System Parameters	PrintStream includes a table that holds a series of system parameters can control the default behavior of the software or contain links to external systems or applications.	Default Settings External Linkages

Workflow - Sample Use Case

The table below describes some typical day to day workflows and sample use cases found in an external website that would utilize the functions in the Web2PS fulfillment API.

Function	Workflow / Sample Use Case
Hello World	Establish Connection Initial test to ensure connection and communication with web service
Main Interface	Authentication Establish correct authentication as standard procedure for all functions
Encrypt Password	Password Encryption User wishes to change password – so ask PrintStream to Encrypt
Get Web User	Establish Security / Privileges User logs into website – identify their user role for security purposes and to determine what options are available to them as the page is built for them.
Get SYS Param	Determine System Behavior and Links Establish any default system behaviors and / or initialize any in memory variables for connection to external programs or services such as credit card processing, web to print integration etc.
Get Active Profile	Make Sure User has Active Status Is the person entering the site set as Active or Inactive
Get Recipient Information	Get User Information Get personal contact information about user (recipient)
Get Projects	Find Fulfillment Projects for User Determine which fulfillment projects this user has security clearance to use. Show the list of “projects” to the user so they can start the shopping or inquiry process.
Get Project Setting	Establish Default Project Settings Once the user selects a project – establish any default behaviors for the project regarding, shipping and handling, how to display items – with images or without – in a grid or just a list, enable credit card processing, etc.
Get Bill To Ship To	Get Default Ship to and Bill to Info for User Get the users default Bill to and Ship to ID for shipping and billing screens. Now that you have the default ShipTo and BillToID – use the Get Recipient Information function again for both the ShipToID and the BillToID (if they are different from the OrderBy ID) to get detailed address data etc.
Get Nomination List	Get Catalogs Find which catalogs this user can shop from and display these on the web page for them to select from
Get Nom Group List	Display Items in selected Catalog When the user clicks on a specific catalog – retrieve the list of items that are available in that catalog.
Get Item No	Get Plant or Company Specific Inventory Item # Items in the catalog are at StkHeader level – use this function to drill down and get the specific item number for the item at a specific company and plant.
GetPSItemViaNumber	Get Inventory Information at Company or Plant Level If you need to show min, reorder, level, sell price, unit of measure etc for an item at plant level.
Get Item Details	Get Detailed Inventory Information at Company or Plant Level for lots of Items Show an inventory summary of all items, where you can control the number of items per page and how many pages to show. A more efficient method for retrieving multiple items instead of having to do round trip calls for each item.
Get Additional Information	Get Additional Info for Item (User Defined Attributes) Show on any specific user defined attributes for an item.

Get Web Description	Establish any Links to Additional Information for Item You may want to place random links to additional information or documents for download.
Get Item Notes	Display any Notes for this Item This can be helpful for creating clickable links that allow the user to see HTML text with detailed description about the item or to click on a link to pull up an image – e.g. JPEG, GIF, PDF document etc.
Get Item Price	Get the Price for the Item Show the user the price for the item. – Complex business rules in PrintStream will interrogate pricing tables based on user, item nomination (catalog), order qty, etc.
Get Item Discount	Show any Discounts Display (or factor into price displayed) any discounts applicable to this item
Get Package Pricing	Show any Options available for this Item Display any “Options” where the user can customize the product they are ordering. E.g. Option to Shrink Wrap or Drill or Laminate etc.
Get PriceGroup Name	Get Descriptions for Option “Groups” Display user friendly language for the options and groups of options for the item
Get Calc Method	Determine Calculation Method for Options Determine if the calculation for the option is a flat charge or based on the qty ordered.
Validate Cost Center	Validate Cost Center if Necessary. If user needs to enter their cost center or billing code during the checkout process – validate if necessary.
Get Reason Code Description	Display Reason Code Description User defined reason codes can be established at several different levels when certain actions are taken – e.g. Cancel Order etc. This function allows you to retrieve the user-friendly description for this reason code.
Get Contact List	Show User Defined Distribution Lists (Contact Lists) Show all the previously entered distributions lists if user would like to ship this order to a list of people.
Get Contact List Detail	Show Individual Contacts in a Distribution List As user selects a distribution list – show them all the people in that list.
Add Contact	Allow User to Add or Edit a Contact in the Distribution List User can then add new contacts to a list on the fly.
Delete Contacts	Allow User to Remove a Contact from the Distribution List User can remove contacts from a list on the fly.
Get Back Orders for Recipient	Show any orders in Back Order Status Get a list of all items that are on back order for this user. Then use the GetPSItemViaNumber function to retrieve detailed item information for each item on back order.
Get Country List	Get Standard Country List Used to pre-load a drop-down list of country names – when adding addresses.
Get Country Detail	Get Detailed Country Information For retrieving Country code, full name etc.
Get Credit Card Types	Get a List of Standard Credit Card Types E.g. Visa, Mastercard etc
Get Credit Card List	Retrieve a list of all Credit Cards on Profile for this user ** Please see special comments about security for this function **
Get Credit Card Processing Fee	Get any Standard Credit Card Processing Fees You may choose to factor these into the sell price on the checkout page.
Get Customer Batches	Show the Status of Orders A convenient and efficient function to quickly retrieve a list of all order that meet a specific criteria – e.g. Status, Date Range. Etc.

Get FF Transaction List	Show Fulfillment Activity Summary An efficient way to quickly show fulfillment statistics to the customer. E.g. Sales Orders, Line Items, Shipments, Cancelled Orders etc for a period.
Get Help and Feedback Email Address	Get Email Address for Help & Feedback If you provide a tool for the customer to click on a link and send an email regarding help or feedback. This function returns the email address currently assigned to receive these emails.
Get Items By Job	Show Print on Demand Master Template Package Info You may wish to retrieve information about which print on demand items are contained in a particular job.
Get Order Address Info	Display Order By and Ship To Information Quick way to just return basic address information for an order.
Get Order Details	Show Order Status and/or Details Very detailed information about any sales order including all line items on order and all shipments.
Get Orders by Job	Show Orders by Job or Project This provides an easy way to retrieve all sales order associated with a specific fulfillment job or project.
Get SO List	Show Orders by Multiple Criteria. This provides a way to quickly retrieve all orders based on multiple criteria – e.g. Date Range, Status, Item#, Project, etc.
Is Recipient Taxable	Find out if Recipient is Marked as Taxable You may encounter situations where taxable or non-taxable status needs to be defined at the recipient level.
Save Nom Group Item	User needs to add item to a Catalog (Nomination Profile) You may wish to provide functionality for users to add items to one or more catalogs
Submit Order	Add / Edit Sales Order This function allows you to insert a fulfillment and/or print on demand sales order into PrintStream
Generate Corporate Structure	Return Corporate Structure information for a Recipient Retrieves the Corporate Structure settings for a specific Recipient
Get CustID by ACNO	Return internal Customer ID by Account Number Retrieves the internal customer ID by account number.
Get Customer Details	Returns basic customer information by Customer ID Retrieves basic customer information for a specific customer.
Get Sales Tax	Return tax amount Calculate the tax amount for a specific dollar amount based on the Postal Code, City and State
Get Shipping Preferences	Return shipping preferences Retrieves all available shipping preferences for a shopping cart based on Project and Recipient settings.
Get UpSell and CrossSell	Return Up Sell and Cross Sell item list Retrieves list of Up Sell and Cross Sell items for a specific nomination item.
Inventory ASN	Add Advanced Shipping Notification Create an Advanced Shipping Notification for an inventory item.
Inventory Create	Create Inventory Item Add a new inventory item for a customer.
Inventory Update	Update Inventory Item Update an existing inventory item.
Login	Validate Login Credentials This function will validate login credentials for a user in PrintStream.
Process Data For Order	This function is only used with the DSF Interface
Set SYS Param	This function is only used with the DSF Interface

Write Items Notes	Create Item Notes for a specific line item in an order This function creates notes, including file attachments, to a line item within an order in PrintStream.
Get Invoice List	This function is only used with the DSF Interface
Save Product	Save Products with Features
Add Digital Asset	Create Catalog Digital Assets

Please Note: The Ecommerce Web Site cannot be used for testing Web2PS calls. This programming was used in older VB code and not available in .Net Web2PS. Most customers will write code for calls or use an available downloadable application such as Fiddler

Add Contact

Narrative

This function will add a new contact to the database. Contacts are used to allow a user to create a standard distribution list of people to whom orders should be shipped to. Each time a contact is added the user will indicate which "Contact List" you wish to add the contact to.

Request Transaction

```
POST /Web2PS WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/AddContact"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <AddContact xmlns="http://Printstream.org/">
      <ListId>int</ListId>
      <CustomerId>int</CustomerId>
      <UserId>string</UserId>
      <Password>string</Password>
      <FirstName>string</FirstName>
      <LastName>string</LastName>
      <Company>string</Company>
      <WPhone>string</WPhone>
      <EPhone>string</EPhone>
      <Fax>string</Fax>
      <Mobile>string</Mobile>
      <Email>string</Email>
      <SecUserID>string</SecUserID>
      <Address1>string</Address1>
      <Address2>string</Address2>
      <Address3>string</Address3>
      <CountryName>string</CountryName>
      <City>string</City>
      <State>string</State>
      <Zip>string</Zip>
    </AddContact>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
```

```

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Header>

    <Authenticate xmlns="http://Printstream.org/">

      <RecipientID>JJWEB</RecipientID>

      <Password>ok</Password>

      <CompanyCode>01</CompanyCode>

      <PlantCode>0100</PlantCode>

      <AccountNo>JJD</AccountNo>

    </Authenticate>

  </soap:Header>

  <soap:Body>

    <AddContact xmlns="http://Printstream.org/">

      <ListId>12</ListId>
      <CustomerId>13</CustomerId>
      <UserId>TestAddCont14</UserId>
      <Password />
      <FirstName>Chris</FirstName>
      <LastName>Doe</LastName>
      <Company> Test Corporation</Company>
      <WPhone>111-111-1111</WPhone>
      <EPhone>222-111-1111</EPhone>
      <Fax>333-111-1111</Fax>
      <Mobile>444-111-1111</Mobile>
      <Email>testcont@test.com</Email>
      <SecUserID />
      <Address1>100 West 22nd Street</Address1>
      <Address2> </Address2>
      <Address3></Address3>
      <CountryName>US</CountryName>
      <City>aNYTOWN</City>
      <State>CA</State>
      <Zip>90000-1234</Zip>

    </AddContact>

  </soap:Body>

</soap:Envelope>

```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Contacts xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Deleted>0</Deleted>
  <Added>1</Added>
  <ListID>12</ListID>
  <RecipientId>119024</RecipientId>
  <FirstName>Chris</FirstName>
  <LastName>Doe</LastName>
  <Company>Test Corporation</Company>
  <Address>100 West 22nd Street</Address>
  <City>Anytown</City>
  <StateCode>CA</StateCode>
  <ZipCode>90000-1234</ZipCode>
</Contacts>
```


Add Digital Asset

Narrative

This API allows you to add/update digital assets to an existing catalog /customer

Request Transaction

```
POST /Web2PSWCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/SaveProduct"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <AddDigitalAsset xmlns="http://Printstream.org/">
      <ProductXML>string</ProductXML>
    </AddDigitalAsset>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>

<Catalog xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" CatalogName="House">

  <DigitalAsset DocumentID="House-Da1" DocumentName="House-Da1" AssetFormat=""
    AssetType="" Pages="0" Status="" ScannedBy="Un" FileSizeInBytes="0"
    xmlns="printstream:efi:com" />

  <DigitalAsset DocumentID="House-DA2" DocumentName="House-DA2" AssetFormat=""
    AssetType="" Pages="12" Status="" ScannedBy="Scanned By" RevisionLabel="Rev Label"
    FileSizeInBytes="0" xmlns="printstream:efi:com" />

  <DigitalAsset DocumentID="House-DA3" DocumentName="House-DA3" AssetFormat=""
    AssetType="" AssetLocation="Asset Location" Pages="1555" Status=""
    RevisionLabel="121212" FileSizeInBytes="1212" xmlns="printstream:efi:com" />

  <DigitalAsset DocumentID="House-DA4" DocumentName="House-DA4" AssetFormat=""
    AssetType="" AssetLocation="Asset Location" Pages="1666" Status=""
    RevisionLabel="121212" FileSizeInBytes="1212" xmlns="printstream:efi:com" />

  <DigitalAsset DocumentID="House-DA7" DocumentName="House-DA7" AssetFormat=""
    AssetType="" AssetLocation="Asset Location" Pages="1777" Status=""
    RevisionLabel="121212" FileSizeInBytes="1212" xmlns="printstream:efi:com" />

  <DigitalAsset DocumentID="House-DA5" DocumentName="House-DA5" AssetFormat=""
    AssetType="" AssetLocation="Asset Location" Pages="1888" Status=""
    RevisionLabel="121212" FileSizeInBytes="1212" xmlns="printstream:efi:com" />

</Catalog>
```

Response Transaction

```
<pst:CatalogDigitalAsset ID="1" Status="Success" Message="House saved succesfully"
  xmlns:pst="printstream:efi:com">
  <pst:DigitalAsset ID="6" Status="Success" Message="House-Da1 saved succesfully" />
  <pst:DigitalAsset ID="7" Status="Success" Message="House-DA2 saved succesfully" />
  <pst:DigitalAsset ID="8" Status="Success" Message="House-DA3 saved succesfully" />
  <pst:DigitalAsset ID="9" Status="Success" Message="House-DA4 saved succesfully" />
  <pst:DigitalAsset ID="10" Status="Success" Message="House-DA7 saved succesfully" />
  <pst:DigitalAsset ID="11" Status="Success" Message="House-DA5 saved succesfully" />
</pst:CatalogDigitalAsset>
```

Authenticate

Narrative

All web service calls are preceded with a series of tags that are used to authenticate the request as well as validate security roles for the Recipient. Note – most all functions require an Authenticate node within the request. This may be the same for all your request transactions or could be different should you want to secure the transaction more highly than it already is.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/MainInterface"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <MainInterface xmlns="http://Printstream.org/">
      <strRequest>string</strRequest>
    </MainInterface>
  </soap:Body>
</soap:Envelope>
```

RecipientID

ID of entity making request. Every web service request must identify the “person” (recipient) that is making the request. This ID must match an active recipient in the Recipient table. If the “recipient” making the request represents a “virtual” person (i.e. an external system making a request as opposed to an actual person on a website) then a “System” recipient should be established in the recipient table.

Password

Password for the Recipient making the request

CompanyCode

Company Code for entity making request. PrintStream supports the ability to configure multiple companies (financial entities) on a single database. The Company Code should represent the code of the company that houses the customer account number that the request is intended for.

PlantCode

Plant Code for entity making request. PrintStream supports the ability to configure multiple plants or production locations within a “Company” on a single database. The Plant Code should represent the code of the Plant that houses the information the request is intended for.

AccountNo

Customer Account Number specific to the request. Most information returned by the web services is constrained to date that is specific to a particular “Customer”. Thereby ensuring that the information returned in the web service only shows data specific to a customer. The Account No must represent an active valid account number for the customer.

Cancel Pick

Narrative

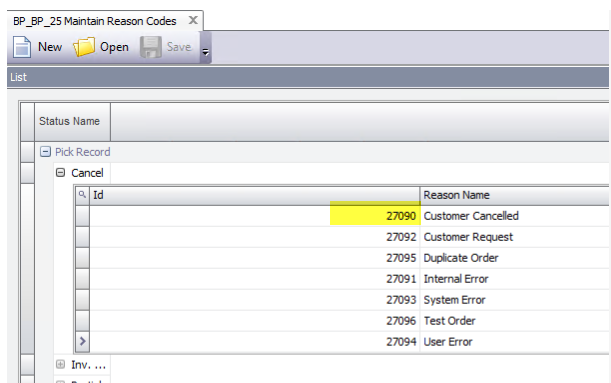
This function will cancel a pick from the database. You will need to supply the numeric table value for the cancel reason code (see screen shot below).

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/CancelPick"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <CancelPick xmlns="http://Printstream.org/">
      <PickID>int</PickID>
      <ReasonCode>int</ReasonCode>
    </CancelPick>
  </soap:Body>
</soap:Envelope>
```

Sample Reason Code



Id	Reason Name
27090	Customer Cancelled
27092	Customer Request
27095	Duplicate Order
27091	Internal Error
27093	System Error
27096	Test Order
27094	User Error

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <CancelPick xmlns="http://Printstream.org/">
      <PickID>270950</PickID>
      <ReasonCode>27090</ReasonCode>
    </CancelPick>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<CancelPickDTO xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <PickID>270950</PickID >
  <ReasonCode>27090</ReasonCode >
  <Result>Sucess</Result >
  <Message>${PickId} cancelled successfully</Message >
</CancelPickDTO>
```

Delete Contacts

Narrative

This function will delete a contact from the database. Contacts are used to allow a user to create a standard distribution list of people to whom orders should be shipped to. 1 = True, 0 = False.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/DeleteContacts"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <DeleteContacts xmlns="http://Printstream.org/">
      <ListId>int</ListId>
      <CustomerId>int</CustomerId>
      <RecipientId>int</RecipientId>
    </DeleteContacts>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <DeleteContacts xmlns="http://Printstream.org/">
      <ListId>12</ListId>
      <CustomerId>13</CustomerId>
      <RecipientId>119024</RecipientId>
    </DeleteContacts>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Contacts xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Deleted>1</Deleted >
  <Added>0</Added >
</Contacts>
```


Encrypt Password

Narrative

Used if adding a new password to an existing Login (Recipient). The function will accept an un-encrypted password, generate an encrypted result and pass this back in the response.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/EncryptPassword"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <EncryptPassword xmlns="http://Printstream.org/">
      <MyString>string</MyString>
    </EncryptPassword>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <EncryptPassword xmlns="http://Printstream.org/">
      <MyString>ok</MyString>
    </EncryptPassword>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Misc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <EncryptedString>D9B39C7F87BB1833</sEncryptedString>
</Misc>
```

Generate Corporate Structure

Narrative

This function will return the Corporate Structure setting for a specific recipient user.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GenerateCorporateStructure"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GenerateCorporateStructure xmlns="http://Printstream.org/">
      <CustomerId>int</CustomerId>
      <RecipientID>int</RecipientID>
    </GenerateCorporateStructure>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GenerateCorporateStructure xmlns="http://Printstream.org/">
      <CustomerId>42</CustomerId>
      <RecipientID>118445</RecipientID>
    </GenerateCorporateStructure>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfRecipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Recipient>
    <CorporateStructID>46</CorporateStructID>
    <Level>4</Level>
    <Description>Lower</Description>
    <ParentID>45</ParentID>
    <URL>&Level1=34&Level2=35&Level3=45&Level4=46</URL>
  </Recipient>
</ArrayOfRecipient>
```

Get Active Profile

Narrative

This function will indicate whether a recipient is “Active” or not. 1 = Active, 0 = Inactive.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetActiveProfile"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetActiveProfile xmlns="http://Printstream.org/">
      <RecipientId>int</RecipientId>
    </GetActiveProfile>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetActiveProfile xmlns="http://Printstream.org/">
      <RecipientId>11771</RecipientId>
    </GetActiveProfile>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMisc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Misc>
    <CCValidated>1</CCValidated>
  </Misc>
</ArrayOfMisc>

```

Get Additional Information

Narrative

The function returns a list of all additional information for an item. These are implemented as User defined "Attributes" that can be configured and associated with items.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetAdditionalInformation"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetAdditionalInformation xmlns="http://Printstream.org/">
      <ItemNumber>string</ItemNumber>
      <CustomerId>int</CustomerId>
      <ProjectId>int</ProjectId>
      <NominationId>int</NominationId>
      <AddlInfoType>string</AddlInfoType>
    </GetAdditionalInformation>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
```

```

    <AccountNo>JJD</AccountNo>

  </Authenticate>

</soap:Header>

<soap:Body>
  <GetAdditionalInformation xmlns="http://Printstream.org/">
    <ItemNumber>245</ItemNumber>
    <CustomerId>13</CustomerId>
    <ProjectId>29</ProjectId>
    <NominationId>24</NominationId>
    <AddlInfoType>L</AddlInfoType>
  </GetAdditionalInformation>
</soap:Body>

</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>

<ArrayOfItemAdditionalInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ItemAdditionalInfo>
    <FieldID>78</FieldID>
    <FieldName>COLOR</FieldName>
    <Mandatory>false</Mandatory>
    <FieldType>200</FieldType>
    <DefaultValue />
    <FieldWidth>10</FieldWidth>
    <AddlInfoType>L</AddlInfoType>
    <FieldValues>Red;White;Blue;</FieldValues>
    <VariableFieldNumber>1</VariableFieldNumber>
    <FieldOrder>1</FieldOrder>
    <FieldDescription>Color</FieldDescription>
    <NominationId>24</NominationId>
  </ItemAdditionalInfo>
  ... <ItemAdditionalInfo> repeats N times for each defined attribute for an item
</ArrayOfItemAdditionalInfo>

```


Get Back Orders for Recipients

Narrative

This function will return a list of all items that currently have back orders for a recipient.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetBackOrdersForRecipient"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetBackOrdersForRecipient xmlns="http://Printstream.org/">
      <RecipientId>int</RecipientId>
    </GetBackOrdersForRecipient>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetBackOrdersForRecipient xmlns="http://Printstream.org/">
      <RecipientId>11771</RecipientId>
    </GetBackOrdersForRecipient>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfBasketItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <BasketItem>
    <ItemNumber>320</ItemNumber>
    <BOQuantity>1</BOQuantity>
  </BasketItem>
  ... <BasketItems> repeats N times for each item on Back Order
</ArrayOfBasketItem>
```

Get Bill To Ship To

Narrative

This function returns default Bill To and Ship To information for a Recipient.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetBillToShipTo"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetBillToShipTo xmlns="http://Printstream.org/">
      <SalesOrderId>int</SalesOrderId>
    </GetBillToShipTo>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetBillToShipTo xmlns="http://Printstream.org/">
      <SalesOrderId>150560</SalesOrderId>
    </GetBillToShipTo>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Recipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <RecipientID>0</RecipientID>
  <IsTaxable>0</IsTaxable>
  <BillTo>118588</BillTo>
  <ShipTo>118588</ShipTo>
  <ISUserLogged>>false</ISUserLogged>
  <ISRecipLogin>>false</ISRecipLogin>
  <IsRecipLogged>>false</IsRecipLogged>
</Recipient>
```

Get Calc Method

Narrative

The function returns the “Calculation Method” for each Option that can be selected on an item.

Flat = Flat Charge (one fixed amount regardless of order qty)

Variable = Charge is calculated based on the order qty.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetCalcMethod"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCalcMethod xmlns="http://Printstream.org/">
      <PriceGroupId>int</PriceGroupId>
    </GetCalcMethod>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCalcMethod xmlns="http://Printstream.org/">
      <PriceGroupId>139</PriceGroupId>
    </GetCalcMethod>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<Misc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <CalcMethod>Flat</CalcMethod>
</Misc>

```

Get Contact List

Narrative

This function will return a list of "Contact Lists". Contact Lists are used to allow a user to create a standard distribution list of people to whom orders should be shipped to.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetContactList"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetContactList xmlns="http://Printstream.org/">
      <CustomerId>int</CustomerId>
      <RecipientID>int</RecipientID>
    </GetContactList>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetContactList xmlns="http://Printstream.org/">
      <CustomerId>13</CustomerId>
      <RecipientID>11771</RecipientID>
    </GetContactList>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfContacts xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Contacts>
    <Deleted>0</Deleted>
    <Added>0</Added>
    <ListID>12</ListID>
    <ListName>My Contacts</ListName>
  </Contacts>
  ... <Contacts> repeats N times for each List of Contacts for the customer
</ArrayOfContacts>
```


Get Contact List Detail

Narrative

This function will return a list of all “Contacts” within the specified “Contact List”. Contact Lists are used to allow a user to create a standard distribution list of people to whom orders should be shipped to.

Page Size – is used to elect the number of contact names to display per page

PageNum – is used to set the page number to return the contacts for that page

The two fields described above are helpful for passing data in page sized chunks.

The other fields such as LastName, Company, City, ZipCode can be used to filter the responses returned based on these criteria (e.g. Where LastName LIKE %anderson%)

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetContactListDetail"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetContactListDetail xmlns="http://Printstream.org/">
      <ListId>int</ListId>
      <CustomerId>int</CustomerId>
      <PageSize>long</PageSize>
      <PageNum>long</PageNum>
      <RecipientId>string</RecipientId>
      <Lastname>string</Lastname>
      <Company>string</Company>
      <City>string</City>
      <ZipCode>string</ZipCode>
    </GetContactListDetail>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetContactListDetail xmlns="http://Printstream.org/">
      <ListId>12</ListId>
      <CustomerId>13</CustomerId>
      <PageSize>100</PageSize>
      <PageNum>1</PageNum>
      <RecipientId>0</RecipientId>
      <Lastname />
      <Company />
      <City />
      <ZipCode />
    </GetContactListDetail>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<ArrayOfContacts xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
  <Contacts>
```

```
    <Deleted>0</Deleted >
```

```
  <Added>0</Added >
```

```
  <RecipientId>119024</RecipientId>
```

```
  <CustomerRecipientId>119024</CustomerRecipientId>
```

```
  <FirstName>Chris</FirstName>
```

```
  <LastName>Doe</LastName>
```

```
  <Company>Test Corporation</Company>
```

```
  <Address>100 West 22nd Street</Address>
```

```
  <City>Anytown</City >
```

```
  <StateCode>CA</StateCode>
```

```
  <ZipCode>90000-1234</ZipCode>
```

```
  <NumItems>32</NumItems>
```

```
  <NumPages>1</NumPages>
```

```
  <PageSize>100</PageSize>
```

```
</Contacts>
```

... <Contacts> repeats N times for each Contact in the Contact List

```
</ArrayOfContacts>
```

Get Country Detail

Narrative

This function will return detailed information about each Country. This is typically used to validate or display additional information about the Country for a ship to or bill to address.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetCountryDetail"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCountryDetail xmlns="http://Printstream.org/">
      <CountryNumber>int</CountryNumber>
    </GetCountryDetail>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCountryDetail xmlns="http://Printstream.org/">
      <CountryNumber>840</CountryNumber>
    </GetCountryDetail>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Misc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Active>1</Active>
  <CountryName>UNITED STATES</CountryName>
  <ZipRequired>1</ZipRequired>
  <StateRequired>1</StateRequired>
  <CountryCode2>US</CountryCode2>
  <CountryCode3>USA</CountryCode3>
  <CountryId>840</CountryId>
</Misc>
```

Get Country List

Narrative

This function will return a list of all Countries.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetCountryList"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCountryList xmlns="http://Printstream.org/" />
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCountryList xmlns="http://Printstream.org/" />
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMisc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Misc>
    <ErrorNumber>0</ErrorNumber>
    <Active>1</Active>
    <CountryCode>1</CountryCode>
    <CountryName>UNITED STATES</CountryName>
  </Misc>
  ... <Misc> repeats N times for each Country
</ArrayOfMisc>
```

Get CustID by ACNO

Narrative

This function will return the internal customer id (CUSTID) using the customer account number (ACNO).

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetCustIDByACNO"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCustIDByACNO xmlns="http://Printstream.org/">
      <AccountNumber>string</AccountNumber>
    </GetCustIDByACNO>
  </soap:Body>
</soap:Envelope>
```


Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCustIDByACNO xmlns="http://Printstream.org/">
      <AccountNumber>JJD</AccountNumber>
    </GetCustIDByACNO>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<Recipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <CustomerID>13</CustomerID>
</Recipient>

```

Get Credit Card List

Narrative

This function returns a list of all credit cards held on profile for a particular recipient.

Important! Credit Card numbers are encrypted in PrintStream. However, card numbers are returned in this web service in an unencrypted format. It is your responsibility, if using this function, to appropriately protect the visibility and access of these credit card numbers within your application.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetCreditCardList"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCreditCardList xmlns="http://Printstream.org/">
      <RecipientID>int</RecipientID>
      <CustomerID>string</CustomerID>
    </GetCreditCardList>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
```

```

    <AccountNo>JJD</AccountNo>

  </Authenticate>

</soap:Header>

<soap:Body>

  <GetCreditCardList xmlns="http://Printstream.org/">
    <RecipientID>11771</RecipientID>
    <CustomerID>13</CustomerID>
  </GetCreditCardList>
</soap:Body>

</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>

<ArrayOfCreditCard xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <CreditCard>

    <RecipientID>11771</RecipientID>

    <CreditCardID>103</CreditCardID>

    <CreditCardTypeID>4</CreditCardTypeID>

    <CreditCardTypeDescription>Visa</CreditCardTypeDescription>

    <CreditCardNumber>5105105105105100</CreditCardNumber>

    <CreditCardExpiryDate>2/28/2014</CreditCardExpiryDate>

    <OwnerName>Dan Doe</OwnerName>

    <CreditCardEmail />

    <SecurityID />

    <Validated>0</Validated>

    <Comments />

    <Active>0</Active>

  </CreditCard>

  ... <Credit Card> repeats N times for each credit card on profile for recipient

</ArrayOfCreditCard>

```

Get Credit Card Processing Fee

Narrative

This function will return any defined credit card processing fees for an order based on the total order amount passed in the request and the charge type.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetCreditCardProcessingFee"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCreditCardProcessingFee xmlns="http://Printstream.org/">
      <ProjectId>int</ProjectId>
      <LineItemAmount>double</LineItemAmount>
      <ChargeType>int</ChargeType>
    </GetCreditCardProcessingFee>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCreditCardProcessingFee xmlns="http://Printstream.org/">
      <ProjectId>29</ProjectId>
      <LineItemAmount>10</LineItemAmount>
      <Chargetype>1</Chargetype>
    </GetCreditCardProcessingFee>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<CreditCardProcessingFee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ErrorNumber>0</ErrorNumber>
  <OrderAmount>10</OrderAmount>
  <ProcessingFee>4</ProcessingFee>
</CreditCardProcessingFee>

```

Get Credit Card Types

Narrative

The function will return a list of all Credit Card “Types”. E.g. Visa, MasterCard etc

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetCreditCardTypes"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCreditCardTypes xmlns="http://Printstream.org/" />
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCreditCardTypes xmlns="http://Printstream.org/" />
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfCreditCard xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <CreditCard>
    <RecipientID>0</RecipientID>
    <CreditCardID>0</CreditCardID>
    <CreditCardTypeID>2</CreditCardTypeID>
    <CCTypeName>Master Card</CCTypeName>
    <CreditCardTypeDescription>Master Card</CreditCardTypeDescription>
    <Validated>0</Validated>
    <Active>1</Active>
  </CreditCard>
  ... <Credit Card> repeats N times for each type of credit card
</ArrayOfCreditCard>
```

Get Customer Attributes

Narrative

This function will return a list of Campaign Attributes for a customer

ACNO - this is the customer's account number

Request Transaction

```
POST /Web2PSWCF.WebService.Setup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetAttributes"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetAttributes xmlns="http://Printstream.org/">
      <xmlString>string</xmlString>
    </GetAttributes>
  </soap:Body>
</soap:Envelope>
```

Schema

GetAttributes.xsd

Sample Request Transaction

```
<?xml version="1.0" ?>
<pbs:GetAttributes xmlns="http://Printstream.org/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema"
xmlns:ns0="printstream:efi.com">
  <pbs:CustomerInfo ACNO="VAL002" />
</pbs:GetAttributes>
```


Response Transaction

```

<?xml version="1.0" encoding="utf-16" ?>

<pbs:MasterAttributeList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="printstream:efi.com"
  xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">

<pbs:CustomerList>

<pbs:Customer CustID="18" ACNO="VAL002">

<pbs:AttributeList>

  <pbs:Attribute AttributeName="Type" Description="Type" IsActive="Y" Sequence="60"
    Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Dist" Description="Dist" IsActive="Y" Sequence="61" Quantity="1"
    Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Reg" Description="Reg" IsActive="Y" Sequence="62" Quantity="1"
    Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Time Zone" Description="Time Zone" IsActive="Y" Sequence="63"
    Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="WAVE" Description="WAVE" IsActive="Y" Sequence="64"
    Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="PULL Day" Description="PULL Day" IsActive="Y" Sequence="65"
    Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Mail Day" Description="Mail Day" IsActive="Y" Sequence="66"
    Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Size" Description="Size" IsActive="Y" Sequence="67" Quantity="1"
    Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Rack" Description="Rack" IsActive="Y" Sequence="68"
    Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Edu /Demos" Description="Edu /Demos" IsActive="Y"
    Sequence="69" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Custom Framing" Description="Custom Framing" IsActive="Y"
    Sequence="70" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Viking" Description="Viking" IsActive="Y" Sequence="71"
    Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="American Greetings" Description="American Greetings"
    IsActive="Y" Sequence="72" Quantity="1" Type="G" FieldType="S" MaxLength="0"
    Decimals="0" />

  <pbs:Attribute AttributeName="Inspiration" Description="Inspiration" IsActive="Y"
    Sequence="73" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="IBN" Description="IBN" IsActive="Y" Sequence="74" Quantity="1"
    Type="G" FieldType="S" MaxLength="0" Decimals="0" />

  <pbs:Attribute AttributeName="Lifestyle Focal Size" Description="Lifestyle Focal Size"
    IsActive="Y" Sequence="75" Quantity="1" Type="G" FieldType="S" MaxLength="0"
    Decimals="0" />

```

```

<pbs:Attribute AttributeName="vestibule sign Qty" Description="vestibule sign Qty" IsActive="Y"
  Sequence="76" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Seasonal Store Layout" Description="Seasonal Store Layout"
  IsActive="Y" Sequence="77" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />

<pbs:Attribute AttributeName="Seasonal Feature Fixture Style" Description="Seasonal Feature
  Fixture Style" IsActive="Y" Sequence="78" Quantity="1" Type="G" FieldType="S"
  MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Lowboy" Description="Lowboy" IsActive="Y" Sequence="79"
  Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="SF version / LF version HOLIDAY Store Hours" Description="SF
  version / LF version HOLIDAY Store Hours" IsActive="Y" Sequence="80" Quantity="1"
  Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Right Door Store Hours" Description="Right Door Store Hours"
  IsActive="Y" Sequence="81" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />

<pbs:Attribute AttributeName="Left Door Version" Description="Left Door Version" IsActive="Y"
  Sequence="82" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="# of entrances" Description="# of entrances" IsActive="Y"
  Sequence="83" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Max door sign size" Description="Max door sign size"
  IsActive="Y" Sequence="84" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />

<pbs:Attribute AttributeName="floral Tiered fixture" Description="floral Tiered fixture"
  IsActive="Y" Sequence="85" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />

<pbs:Attribute AttributeName="Promo windows" Description="Promo windows" IsActive="Y"
  Sequence="86" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Price Checkers" Description="Price Checkers" IsActive="Y"
  Sequence="87" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Take a ticket" Description="Take a ticket" IsActive="Y"
  Sequence="88" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Project Yr" Description="Project Yr" IsActive="Y" Sequence="89"
  Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Project Month" Description="Project Month" IsActive="Y"
  Sequence="90" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Project type" Description="Project type" IsActive="Y"
  Sequence="91" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

<pbs:Attribute AttributeName="Soft Open / Complete date" Description="Soft Open / Complete
  date" IsActive="Y" Sequence="92" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />

<pbs:Attribute AttributeName="Shopping Cart" Description="Shopping Cart" IsActive="Y"
  Sequence="93" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />

```

```

<pbs:Attribute AttributeName="No Smoking" Description="No Smoking" IsActive="Y"
  Sequence="94" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Phone System" Description="Phone System" IsActive="Y"
  Sequence="95" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Paging" Description="Paging" IsActive="Y" Sequence="96"
  Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Music" Description="Music" IsActive="Y" Sequence="97"
  Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Endcap Assortments" Description="Endcap Assortments"
  IsActive="Y" Sequence="98" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />
<pbs:Attribute AttributeName="Sign Kit Sizes" Description="Sign Kit Sizes" IsActive="Y"
  Sequence="99" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Behind the Register" Description="Behind the Register"
  IsActive="Y" Sequence="100" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />
<pbs:Attribute AttributeName="Promo Window Qty" Description="Promo Window Qty"
  IsActive="Y" Sequence="101" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />
<pbs:Attribute AttributeName="Backer Paper" Description="Backer Paper" IsActive="Y"
  Sequence="102" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Home Dec Tan" Description="Home Dec Tan" IsActive="Y"
  Sequence="103" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Crafts Purple" Description="Crafts Purple" IsActive="Y"
  Sequence="104" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Crafts Purple 24x66" Description="Crafts Purple 24x66"
  IsActive="Y" Sequence="105" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />
<pbs:Attribute AttributeName="Brand Green" Description="Brand Green" IsActive="Y"
  Sequence="106" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Brand Green_30" Description="Brand Green_30" IsActive="Y"
  Sequence="107" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Brand Green_36" Description="Brand Green_36" IsActive="Y"
  Sequence="108" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Sewing Blue" Description="Sewing Blue" IsActive="Y"
  Sequence="109" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Black" Description="Black" IsActive="Y" Sequence="110"
  Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="White Bridal" Description="White Bridal" IsActive="Y"
  Sequence="111" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Foodcrafting White" Description="Foodcrafting White"
  IsActive="Y" Sequence="112" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />

```



```
<pbs:Attribute AttributeName="Kids Turquoise" Description="Kids Turquoise" IsActive="Y"
  Sequence="113" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Window Graphics" Description="Window Graphics" IsActive="Y"
  Sequence="114" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Window Shades" Description="Window Shades" IsActive="Y"
  Sequence="115" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Triple Tiered Stanchion" Description="Triple Tiered Stanchion"
  IsActive="Y" Sequence="116" Quantity="1" Type="G" FieldType="S" MaxLength="0"
  Decimals="0" />
<pbs:Attribute AttributeName="Ceiling Graphics" Description="Ceiling Graphics" IsActive="Y"
  Sequence="117" Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
<pbs:Attribute AttributeName="Group" Description="Group" IsActive="Y" Sequence="118"
  Quantity="1" Type="G" FieldType="S" MaxLength="0" Decimals="0" />
</pbs:AttributeList>
</pbs:Customer>
</pbs:CustomerList>
</pbs:MasterAttributeList>
```

Get Customer Batches

Narrative

This function will return a list of all sales orders that are associated with a specific customer Batch Date or Batch ID passed in the request.

Status ID – Can be passed to filter orders based on a specific Status

Page Size – is used to elect the number of sales orders to display per page

PageNum – is used to set the page number to return the sales orders for that page

The two fields described above are helpful for passing data in page sized chunks.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetCustomerBatches"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCustomerBatches xmlns="http://Printstream.org/">
      <BatchDate>string</BatchDate>
      <CustBatchId>string</CustBatchId>
      <ProjectId>int</ProjectId>
      <CustId>int</CustId>
      <StatusId>int</StatusId>
      <PageSize>long</PageSize>
      <PageNum>long</PageNum>
    </GetCustomerBatches>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
```



```
<RecipientID>JJWEB</RecipientID>

<Password>ok</Password>

<CompanyCode>01</CompanyCode>

<PlantCode>0100</PlantCode>

<AccountNo>JJD</AccountNo>

</Authenticate>

</soap:Header>

<soap:Body>

  <GetCustomerBatches xmlns="http://Printstream.org/">
    <BatchDate>02/21/12</BatchDate>
    <CustBatchId>237720</CustBatchId>
    <ProjectId>29</ProjectId>
    <CustId>13</CustId>
    <StatusId>0</StatusId>
    <PageSize>99</PageSize>
    <PageNum>1</PageNum>
  </GetCustomerBatches>
</soap:Body>

</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<ArrayOfSalesOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
  <SalesOrder>
```

```
    <ErrorNumber>0</ErrorNumber>
```

```
    <NoOfOrders>2</NoOfOrders>
```

```
    <OrderNo>150388</OrderNo>
```

```
    <BillTo>11776</BillTo>
```

```
    <BatchNo>8806</BatchNo>
```

```
    <Status>Open</Status>
```

```
    <ShipTo>11776</ShipTo>
```

```
    <OrderBy>11776</OrderBy>
```

```
    <DateCreated>09/09/2010</DateCreated>
```

```
    <CustomerId>0</CustomerId>
```

```
    <ShippingHandling>0</ShippingHandling>
```

```
    <Discount>0</Discount>
```

```
    <SalesTax>0</SalesTax>
```

```
    <Total>0</Total>
```

```
  </SalesOrder>
```

```
</SalesOrder>
```

... <SalesOrder > repeats N times for each sales order in the batch range

```
</ArrayOfSalesOrder>
```

Get Customer Details

Narrative

This function will return basic customer information using the customer account number (AccountNo).

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetCustomerDetails"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCustomerDetails xmlns="http://Printstream.org/">
      <CustomerID>string</CustomerID>
    </GetCustomerDetails>
  </soap:Body>
</soap:Envelope>
```


Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetCustomerDetails xmlns="http://Printstream.org/">
      <CustomerID>42</CustomerID>
    </GetCustomerDetails>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Recipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <CustomerID>183</CustomerID>
  <AccNo>50017</AccNo>
  <Name>Test Company</Name>
  <Contact>BP</Contact>
  <Phone>000-000-0000</Phone>
  <Fax />
  <CSR>Claude</CSR>
  <SalesRepresentative>Brian Doe</SalesRepresentative>
  <Address1>1 Main Street</Address1>
  <Address2>PAWTUCKET,RI 02861</Address2>
  <Address3 />
  <DSFProjID>0</DSFProjID>
</Recipient>
```

Get FF Transaction List

Narrative

This function will scan the fulfillment system and compute rolling totals for all fulfillment transactions for the project specified in the request.

This function is particularly suitable for reporting fulfillment activity for such activities as:

Number of Orders Created, Line Items Created, Orders Cancelled etc.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetFFTransactionList"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetFFTransactionList xmlns="http://Printstream.org/">
      <CustomerId>int</CustomerId>
      <SalesOrderId>int</SalesOrderId>
      <ProjectId>int</ProjectId>
    </GetFFTransactionList>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
```

```

    <AccountNo>JJD</AccountNo>

  </Authenticate>

</soap:Header>

<soap:Body>
  <GetFFTtransactionList xmlns="http://Printstream.org/">
    <CustomerId>42</CustomerId>
    <SalesOrderId>161777</SalesOrderId>
    <ProjectId>107</ProjectId>
  </GetFFTtransactionList>
</soap:Body>

</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>

<ArrayOfMisc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Misc>
    <ActivityPPU>0</ActivityPPU>
    <ActivityQty>1</ActivityQty>
    <ActivityName>Save Order</ActivityName>
    <ActivityCode>104000</ActivityCode>
    <ActivityId>1</ActivityId>
  </Misc>

  ... <Misc > repeats N times for each Activity Type for the Project

</ArrayOfMisc>

```

Get Help and Feedback Email Address

This function will return the system defined email address for sending messages to the appropriate person regarding request for Help and Feedback on the web.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetHAFEmailAddress"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetHAFEmailAddress xmlns="http://Printstream.org/" />
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetHAFEmailAddress xmlns="http://Printstream.org/" />
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Misc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <HAFEmailAddress>HelpDesk@foo.com</HAFEmailAddress>
</Misc>
```

Get Inventory Stock Level

Narrative

This API is for OPS Integration only. This feature accepts a request from OPS to update the stock level (available on hand quantity in PrintStream) for an item shared between OPS and PrintStream.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetInvStockLevel"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <GetInvStockLevel xmlns="http://Printstream.org/">
      <OPSXML>string</OPSXML>
    </GetInvStockLevel>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" ?>
<StockLevelLookup>
  <SharedSecret>JJWEB123</SharedSecret>
  <SupplierPartID>245-ay3</SupplierPartID>
</StockLevelLookup>
```

Response Transaction

```
<?xml version="1.0" encoding="UTF-8" ?>
<StockLevelLookup>
  <SupplierPartID>245-ay3</SupplierPartID>
  <StockLevel>1092244</StockLevel>
</StockLevelLookup>
```

Get Item Detail

Narrative

This function will return a list of all inventory items and status and balances for a specific Project and Catalog for a specific customer.

The **PageSize** value passes the number of items that you wish to show per page

The **PageNumber** value passes the "Page Number" you wish to return results for.

The above mechanism provides a simple way for an external application to receive data in "Page Size chunks" – to reduce the amount of data being transmitted when necessary.

Simply use -1 to return all results at once.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetItemDetails"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemDetails xmlns="http://Printstream.org/">
      <CustomerId>int</CustomerId>
      <NominationId>int</NominationId>
      <ProjectId>int</ProjectId>
      <RecipientId>int</RecipientId>
      <PageSize>int</PageSize>
      <PageNumber>int</PageNumber>
      <InvCode>string</InvCode>
    </GetItemDetails>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Header>
```

```

    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemDetails xmlns="http://Printstream.org/">
      <CustomerId>13</CustomerId>
      <NominationId>24</NominationId>
      <ProjectId>29</ProjectId>
      <RecipientId>11771</RecipientId>
      <PageSize>-1</PageSize>
      <PageNumber>-1</PageNumber>
      <InvCode>3109</InvCode>
    </GetItemDetails>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Item>
    <PAPSizeId>269</PAPSizeId>
    <StkHeaderId>245</StkHeaderId>
    <ItemCode>245</ItemCode>
    <ItemNumber>314</ItemNumber>
    <InventoryCode>314</InventoryCode>
    <Description>Angler Fly Rod Outfit</Description>
    <Description2 />
    <Description3 />
    <OnHand>1101302</OnHand>
    <Allocated>8321</Allocated>
    <Price>10</Price>
  </Item>

```




```
<MinimumLimit>0</MinimumLimit>
<OrderLimit>0</OrderLimit>
<OnHold>false</OnHold>
<IsBackOrder>false</IsBackOrder>
<ForceMax>false</ForceMax>
<ForceMin>false</ForceMin>
<NewItem>false</NewItem>
<Notes>True</Notes>
<RevisionCode />
<HoldReason>0</HoldReason>
<MinReason>0</MinReason>
<MaxReason>0</MaxReason>
<POP />
<Customizable>N</Customizable>
<UnitOfIssue>3</UnitOfIssue>
<UnitIssueDescription>yard</UnitIssueDescription>
<BORule>4</BORule>
<OrderQuantityOf>0</OrderQuantityOf>
<ProductType>CAT</ProductType>
<ReorderQuantity>300</ReorderQuantity>
<SubstituteBasis>0</SubstituteBasis>
<SubstituteItem>0</SubstituteItem>
<SubstituteEffectDate>2005-04-18T09:00:55</SubstituteEffectDate>
<SubExpireDate>2006-07-07T10:29:04</SubExpireDate>
<Available>1092981</Available>
<ExternalItemNumber />
<ExternalInvCode />
<SalesTax>0</SalesTax>
<Active>true</Active>
<MasterNumber />
<Commitments>
  <Commitments>
```



```
<StockReqId>370274</StockReqId>
<JobNumber>461</JobNumber>
<QuantityCommitted>4</QuantityCommitted>
<QuantityUsed>0</QuantityUsed>
<Balance>4</Balance>
<DateReserved>2010-06-29T00:00:00</DateReserved>
<DateRequired>0001-01-01T00:00:00</DateRequired>
</Commitments>
... <Commitments> repeats N times for each Commitment for the item
</Commitments>
</Item>
... <Item> repeats N times for each Item within the requested Page.
</ArrayOfItem>
```

Get Item Discount

Narrative

This function returns any Price Discount rules / information for an item, relevant to the order qty passed in the request.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetItemDiscount"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemDiscount xmlns="http://Printstream.org/">
      <ItemNumber>string</ItemNumber>
      <CustomerId>int</CustomerId>
      <NominationId>int</NominationId>
      <RecipientId>int</RecipientId>
      <OrderQuantity>double</OrderQuantity>
      <Price>double</Price>
    </GetItemDiscount>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemDiscount xmlns="http://Printstream.org/">
      <ItemNumber>314</ItemNumber>
      <CustomerId>13</CustomerId>
      <NominationId>24</NominationId>
      <RecipientId>17711</RecipientId>
      <OrderQuantity>10</OrderQuantity>
      <Price>10</Price>
    </GetItemDiscount>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<Discount xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <TotalDiscount>2</TotalDiscount>
  <HasDiscount>true</HasDiscount>
  <DiscountText>$2 off, if you buy from 5 to 99999 Items.</DiscountText>
  <IsFreeItem>false</IsFreeItem>
  <TCDCCode>Discount 1%</TCDCCode>
</Discount>

```

Get Item No

Narrative

This function will return the specific internal "Item No" for an item at the specified Company and Plant Level. Because PrintStream supports the ability to configure multiple financial entities and physical plant locations on a single database – we maintain a structure that holds item information at a "consolidated" level and then pointers to the same item number at company and plant level. When a customer is placing an order on the web, they would only wish to see the item listed once. Then when the order is placed some decision is made to determine which physical company and plant to fulfill their order from. This function provides the ability to feed the STKHEADER.ID into the system and then have PrintStream return the unique "ItemNo" (Code) for that item at the specified company and plant.

Request Transaction

```
POST /Web2PS WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetItemNo"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemNo xmlns="http://Printstream.org/">
      <StkHeaderId>int</StkHeaderId>
    </GetItemNo>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemNo xmlns="http://Printstream.org/">
      <StkHeaderId>0</StkHeaderId>
      <InvCode>316</InvCode>
    </GetItemNo>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<Misc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Code>314</Code>
</Misc>

```

Get Item Notes

Narrative

This function returns a list of all miscellaneous notes that have been associated with an item.

In future releases you will be able to pass different Context and SubContext ID's to retrieve notes from different entities and levels within fulfillment. E.g. Sales Order Header, Sales Order Line etc.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetItemNotes"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemNotes xmlns="http://Printstream.org/">
      <CustomerId>int</CustomerId>
      <ContextId>int</ContextId>
      <SubContextId>int</SubContextId>
      <NoteType>int</NoteType>
    </GetItemNotes>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemNotes xmlns="http://Printstream.org/">
      <CustomerId>13</CustomerId>
      <ContextId>141581</ContextId>
      <SubContextId>0</SubContextId>
      <NoteType>3</NoteType>
    </GetItemNotes>
  </soap:Body>
</soap:Envelope>

```

ContextID = Sales Order Id

SubContextID = 0 is order level, otherwise would be a line item code

NoteType = 3 (order notes)

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfNotes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Notes>
    <ThreadId>1</ThreadId>
    <SequenceId>1</SequenceId>
    <Title>Item Revision Warning </Title>
    <NotesText>This item is scheduled to be revised in June</NotesText>
    <LineItemNumber>2</LineItemNumber>
  </Notes>
  ... <Notes> repeats N times for each note for the item
</ArrayOfNotes>

```


Get Item Price

Narrative

This function returns the sell price for the item. This is based on complex rules as to whether pricing was configured at item level or nomination level etc and is also based on the qty breaks in a price list relevant to the order qty passed in the request.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswbl2.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetItemPrice"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemPrice xmlns="http://Printstream.org/">
      <ItemNumber>string</ItemNumber>
      <ProjectId>int</ProjectId>
      <CustomerId>int</CustomerId>
      <NominationId>int</NominationId>
      <RecipientId>int</RecipientId>
      <OrderQuantity>double</OrderQuantity>
    </GetItemPrice>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
```

```
<AccountNo>JJD</AccountNo>
</Authenticate>
</soap:Header>
<soap:Body>
  <GetItemPrice xmlns="http://Printstream.org/">
    <ItemNumber>314</ItemNumber>
    <ProjectId>29</ProjectId>
    <CustomerId>13</CustomerId>
    <NominationId>24</NominationId>
    <RecipientId>11771</RecipientId>
    <OrderQuantity>10</OrderQuantity>
  </GetItemPrice>
</soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMisc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Misc>
    <ItemPrice>99.5</ItemPrice>
  </Misc>
</ArrayOfMisc>
```

Get Items By Job

Narrative

This function will return information about a job template setup in PrintStream that is configured for Print on Demand manufacturing. Each Job template can contain links to multiple orderable item numbers. This function is typically used to display all item numbers linked to a specific job template (QuoteID).

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetItemsByJob"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemsByJob xmlns="http://Printstream.org/">
      <QuoteId>string</QuoteId>
    </GetItemsByJob>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetItemsByJob xmlns="http://Printstream.org/">
      <QuoteId>670.99</QuoteId>
    </GetItemsByJob>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMisc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Misc>
    <Description>Package1</Description>
    <InventoryCode />
    <ItemNo />
    <Type>P</Type>
    <ComponentQty>0</ComponentQty>
    <PackageQty>1</PackageQty>
  </Misc>
  ... <Misc > repeats N times for each Item (Package) in the Job Template
</ArrayOfMisc>
```

Get Nom Group Lists

Narrative

This function will return a list of all Item Nomination Groups (Catalogs) that a specific Item is included in, for the specific customer and fulfillment project.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetNomGroupList"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetNomGroupList xmlns="http://Printstream.org/">
      <ProjectId>int</ProjectId>
      <STKHeaderId>int</STKHeaderId>
    </GetNomGroupList>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
```

```

</soap:Header>
<soap:Body>
  <GetNomGroupList xmlns="http://Printstream.org/">
    <ProjectId>29</ProjectId>
    <StkHeaderId>245</StkHeaderId>
  </GetNomGroupList>
</soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfNomGroupItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <NomGroupItem>
    <Featured>false</Featured>
    <Level1>1</Level1>
    <Level2>0</Level2>
    <Level3>0</Level3>
    <Level4>0</Level4>
    <NomDescription>Fishing 2005 Catalog</NomDescription>
    <LongDescription>Fly Rod * Reel Outfits</LongDescription>
    <NomGroupId>526</NomGroupId>
    <NomGroupIdForItem>10198</NomGroupIdForItem>
    <CustomerId>13</CustomerId>
    <NominationId>24</NominationId>
    <NominationName>Fishing 2005</NominationName>
    <STKHeaderId>245</STKHeaderId>
    <Checked>true</Checked>
    <ProjectId>0</ProjectId>
  </NomGroupItem>
  ... The <NomGroupItem> section repeats N times for each Catalog this item is in.
</ArrayOfNomGroupItem>

```

Get Nomination List

Narrative

This function will return a list of all Item Nomination Groups (Catalogs) and the items in each catalog for the specific customer and fulfillment project.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetNominationList"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetNominationList xmlns="http://Printstream.org/">
      <ProjectId>int</ProjectId>
      <CustomerId>int</CustomerId>
      <RecipientId>int</RecipientId>
    </GetNominationList>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetNominationList xmlns="http://Printstream.org/">
      <ProjectId>29</ProjectId>
      <CustomerId>13</CustomerId>
      <RecipientId>11771</RecipientId>
    </GetNominationList>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfNomination xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Nomination>
    <NominationId>24</NominationId>
    <NominationName>Fishing 2005</NominationName>
    <NominationDescription>Fishing 2005 Catalog</NominationDescription>
    <NewDays>33</NewDays>
    <WebGroupImage>1</WebGroupImage>
    <WebGroupName>1</WebGroupName>
    <WebGroupDescription>1</WebGroupDescription>
    <WebGroupNumUp>3</WebGroupNumUp>
  </Nomination>

```



```

<NomGroupItem>
  <Featured>false</Featured>
  <Level1>1</Level1>
  <Level2>0</Level2>
  <Level3>0</Level3>
  <Level4>0</Level4>
  <NomDescription>Fly Rod * Reel Outfits</NomDescription>
  <Image>Fish.gif</Image>
  <LongDescription />
  <NomGroupId>0</NomGroupId>
  <NomGroupIdForItem>0</NomGroupIdForItem>
  <CustomerId>0</CustomerId>
  <NominationId>0</NominationId>
  <STKHeaderId>0</STKHeaderId>
  <Checked>false</Checked>
  <ProjectId>0</ProjectId>
</NomGroupItem>

... The <NomGroupItem> section repeats N times for each item in the Catalog
</Nomination>

... The <Nomination> section repeats N times for each Catalog for the Project
</ArrayOfNomination>

```

Get Order Address Info

Narrative

This function will return order header address information for a sales order.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetOrderAddressInfo"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetOrderAddressInfo xmlns="http://Printstream.org/">
      <OrderId>int</OrderId>
    </GetOrderAddressInfo>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetOrderAddressInfo xmlns="http://Printstream.org/">
      <OrderId>149904</OrderId>
    </GetOrderAddressInfo>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Recipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <RecipientID>0</RecipientID>
  <IsTaxable>0</IsTaxable>
  <ISUserLogged>false</ISUserLogged>
  <ISRecipLogin>false</ISRecipLogin>
  <IsRecipLogged>false</IsRecipLogged>
  <OrderByAddress>Dan Doe<br>JJD<br>123 New South Rd<br>Sometown, CT
    06123<br>UNITED STATES<br>phone: 8606429999<br>Email:
    jjweb4@foo.com</OrderByAddress>
  <ShipToAddress>Jimmy Doe<br>Radio One<br>1234 Airline trail<br>Sometown, CT
    06123<br>UNITED STATES<br>Email: jj@ro.com</ShipToAddress>
</Recipient>
```

Get Order Details

Narrative

This function will return detailed order line item and shipping information for a sales order.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetOrderDetails"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetOrderDetails xmlns="http://Printstream.org/">
      <OrderId>int</OrderId>
    </GetOrderDetails>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetOrderAddressInfo xmlns="http://Printstream.org/">
      <OrderId>150420</OrderId>
    </GetOrderAddressInfo>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<SalesOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ErrorNumber>0</ErrorNumber>
  <OrderNo />
  <LocalAccount>29</LocalAccount>
  <PSOrderId>150420</PSOrderId>
  <NeedDate>9/23/2010</NeedDate>
  <SOTypeName />
  <OrderOrigin />
  <Status>Sales Order Complete</Status>
  <ShipTo>11771</ShipTo>
  <CustomerId>13</CustomerId>
```

```

<ShippingHandling>10.53</ShippingHandling>
<Discount>5.25</Discount>
<SalesTax>0.61</SalesTax>
<Total>14.09</Total>
<Items>
  <LineItem>
    <ItemNumber>316</ItemNumber>
    <OrderQuantity>1</OrderQuantity>
    <ItemDescription>Quest Fly Rod</ItemDescription>
    <StockHeaderRecNumber>247</StockHeaderRecNumber>
    <LineItemStatus>Shipped</LineItemStatus>
    <PartNumber>316</PartNumber>
    <BackOrderQuantity>0</BackOrderQuantity>
    <CarrierCode>UPS GROUND</CarrierCode>
    <PONumber />
    <Notes />
    <NominationId>24</NominationId>
    <Discount>5</Discount>
    <FFPriced>1</FFPriced>
    <HoldStatus>0</HoldStatus>
    <BOStatus>0</BOStatus>
    <DocumentId />
    <UOM />
    <GrossQuantity>1</GrossQuantity>
    <SubstituteItem>0</SubstituteItem>
    <CreditCardProcessingFee>0</CreditCardProcessingFee>
    <ItemTotal>7.45</ItemTotal>
  </LineItem>
  ... <LineItem> repeats N times for each Line Item on sales order
</Items>
<PACKAGES>
  <Package>

```

```

    <ShipmentID>5992</ShipmentID>
    <TrackingNumber>z1231313133</TrackingNumber>
    <ShippingCharges>0</ShippingCharges>
    <ShipmentDate>2010-09-23T13:11:01</ShipmentDate>
    <ShippedBy />
    <ShipMode>U11</ShipMode>
    <Address1>JJD</Address1>
    <Address2>JJD</Address2>
    <Address3>327 NEW SOUTH ROAD</Address3>
    <Country>UNITED STATES</Country>
    <Zip>6249</Zip>
    <StatusID>0</StatusID>
    <StatusName>Shipped</StatusName>
  </Package>
  ... <Package> repeats N times for each shipment on a sales order
</PACKAGES>
  <OtherCharges>
    <LineItemOtherCharges>
      <ChargeDescription>Order Charge</ChargeDescription>
      <ChargeQuantity>0</ChargeQuantity>
      <ChargeUnitPrice>0</ChargeUnitPrice>
      <ChargeDiscount>0</ChargeDiscount>
      <ChargeTotal>1.00</ChargeTotal>
      <ChargeTax>0</ChargeTax>
      <ChargeInventoryCode>Save Order</ChargeInventoryCode>
      <ChargeLineNumber>0</ChargeLineNumber>
    </LineItemOtherCharges>
    ... <LineItemOtherCharges>repeats N times for each misc order charge
  </OtherCharges>
</SalesOrder>

```

Get Orders By Job

Narrative

This function returns a list of all sales orders associated with the fulfillment project job number passed in the request. This is typically used to report on sales order activity or status for a monthly fulfillment job number.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetOrdersByJob"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetOrdersByJob xmlns="http://Printstream.org/">
      <JobId>string</JobId>
    </GetOrdersByJob>
  </soap:Body>
</soap:Envelope>
```


Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetOrdersByJob xmlns="http://Printstream.org/">
      <JobId>432</JobId>
    </GetOrdersByJob>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfSalesOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SalesOrder>
    <ErrorNumber>0</ErrorNumber>
    <OrderNo>13350</OrderNo>
    <BatchNo>232132</BatchNo>
    <Status>Open</Status>
    <ShipTo>Nick Doe SOMETOWN CT</ShipTo>
    <OrderBy>Nick Doe SOMETOWN CT</OrderBy>
    <OrderType />
    <DateCreated>4/27/2005 10:10:57 AM</DateCreated>
    <CustOrderNo>23231</CustOrderNo>
```



```
<CustomerId>0</CustomerId>
<ShippingHandling>0</ShippingHandling>
<Discount>0</Discount>
<SalesTax>0</SalesTax>
<Total>0</Total>
</SalesOrder>
... <SalesOrder> repeats N times for each sales order in the job specified
</ArrayOfSalesOrder>
```

Get Package Pricing

Narrative

This function returns a list of all Package Pricing Options for an Item. When an item is configured with Package Pricing – the recipient will be able to select optional features when placing an order for the item.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetPackagePricing"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetPackagePricing xmlns="http://Printstream.org/">
      <ItemNumber>int</ItemNumber>
      <MasterNumber>int</MasterNumber>
      <PriceType>string</PriceType>
      <PriceGroupId>int</PriceGroupId>
      <PlantCode>string</PlantCode>
      <CompanyCode>string</CompanyCode>
    </GetPackagePricing>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
```

```

    <PlantCode>0100</PlantCode>
    <AccountNo>JJD</AccountNo>
  </Authenticate>
</soap:Header>
<soap:Body>
  <GetPackagePricing xmlns="http://Printstream.org/">
    <ItemNumber>362</ItemNumber>
    <MasterNumber>551</MasterNumber>
    <PriceType>0</PriceType>
    <PriceGroupId>0</PriceGroupId>
    <PlantCode>0100</PlantCode>
    <CompanyCode>01</CompanyCode>
  </GetPackagePricing>
</soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfPackage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Package>
    <PackageOptions>
      <PackageOption>
        <PriceGroupId>139</PriceGroupId>
        <PriceGroupName>Additional Pages</PriceGroupName>
        <BillingGroup>ADDL PGS</BillingGroup>
        <POD>N</POD>
        <Simplex>0</Simplex>
        <Duplex>0</Duplex>
        <Blanks>0</Blanks>
        <Option>Y</Option>
        <BillUOM>1</BillUOM>
        <OrderQuantity>0</OrderQuantity>
        <SellPrice>0.3</SellPrice>
        <AdditionalListPrice>0.2</AdditionalListPrice>
        <BillUOMDescription>Ea</BillUOMDescription>
        <UseOrderQuantity>true</UseOrderQuantity>
      </PackageOption>
    </PackageOptions>
  </Package>
</ArrayOfPackage>

```



```
<AdditionalSellPrice>0.3</AdditionalSellPrice>
<ListPrice>0.2</ListPrice>
<PackageTiers>
  <PackageTier>
    <PriceTierId>118</PriceTierId>
    <TierNumber>1</TierNumber>
    <PackageId>21016</PackageId>
    <TierQuantity>99999999</TierQuantity>
    <TierPrice>0</TierPrice>
    <QuoteNumber>551</QuoteNumber>
    <CalculationMethod>Flat</CalculationMethod>
  </PackageTier>
  ... <PackageTier> repeats N times for each Qty Break
</PackageTiers>
</PackageOption>
... <PackageOption> repeats N times for each Option for the item
</PackageOptions>
</Package>
</ArrayOfPackage>
```

Get PriceGroup Name

Narrative

This function returns information about each “Option” that was selected for a specific sales order.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetPriceGroupName"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetPriceGroupName xmlns="http://Printstream.org/">
      <SalesOrderId>int</SalesOrderId>
    </GetPriceGroupName>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetPriceGroupName xmlns="http://Printstream.org/">
      <SalesOrderId>160907</SalesOrderId>
    </GetPriceGroupName>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMisc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Misc>
    <LineItemno>1</LineItemno>
    <PriceGroupName>Additional Pages</PriceGroupName>
    <OrderQuantity>100</OrderQuantity>
    <SellPrice>0.3</SellPrice>
    <PriceGroupId>139</PriceGroupId>
  </Misc>
  ... <Misc> repeats N times for each Option selected for this sales order
</ArrayOfMisc>
```

Get Project Setting

Narrative

This function returns a list of all detailed Fulfillment Project settings for a specific Project.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetProjectSetting"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetProjectSetting xmlns="http://Printstream.org/">
      <ProjectId>int</ProjectId>
    </GetProjectSetting>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
```



```

<soap:Body>
  <GetProjectSetting xmlns="http://Printstream.org/">
    <projectId>29</projectId>
  </GetProjectSetting>
</soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<Project xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <PricedFF>0</PricedFF>
  <ProjectID>0</ProjectID>
  <SourceID>0</SourceID>
  <AllowCC>0</AllowCC>
  <RealTimeSale>0</RealTimeSale>
  <CustomerBatchRequired>0</CustomerBatchRequired>
  <CustomerOrderIDRequired>0</CustomerOrderIDRequired>
  <ShipHandleCharge>0</ShipHandleCharge>
  <AllowInvoiced>0</AllowInvoiced>
  <DefaultPlantCode>0</DefaultPlantCode>
  <WebDisplayMode>0</WebDisplayMode>
  <WebNumUp>0</WebNumUp>
  <DocMethod>0</DocMethod>
  <WebSortKey>0</WebSortKey>
  <EnableCCFees>1</EnableCCFees>
  <EmbedCCFeesInSH>1</EmbedCCFeesInSH>
  <PodMethod>1</PodMethod>
  <ShowGrossQty>1</ShowGrossQty>
</Project>

```

Get Projects

Narrative

This function will return a list of all Fulfillment "Projects" for a specific customer and the default settings for each of the projects.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetProjects"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetProjects xmlns="http://Printstream.org/">
      <CustomerId>int</CustomerId>
    </GetProjects>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
```

```

</soap:Header>
<soap:Body>
  <GetProjects xmlns="http://Printstream.org/">
    <CustomerId>24</CustomerId>
  </GetProjects>
</soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfProject xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

```

```

- <Project>
  <PricedFF>0</PricedFF>
  <ProjectID>174</ProjectID>
  <ProjectName>Reg Pick & Pack</ProjectName>
  <SourceID>0</SourceID>
  <AllowCC>0</AllowCC>
  <RealTimeSale>0</RealTimeSale>
  <CustomerBatchRequired>0</CustomerBatchRequired>
  <CustomerOrderIDRequired>0</CustomerOrderIDRequired>
  <ShipHandleCharge>0</ShipHandleCharge>
  <AllowInvoiced>0</AllowInvoiced>
  <DefaultPlantCode>100</DefaultPlantCode>
  <WebDisplayMode>0</WebDisplayMode>
  <WebNumUp>2</WebNumUp>
  <DocMethod>0</DocMethod>
  <WebSortKey>0</WebSortKey>
</Project>

```

... The **<Project>** section repeats N times for each project for this customer

```

</ArrayOfProject>

```

Get PSItem Via Number

Narrative

This function returns summary information about an item.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetPSItemviaNumber"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetPSItemviaNumber xmlns="http://Printstream.org/">
      <ItemNumber>string</ItemNumber>
    </GetPSItemviaNumber>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetPSItemviaNumber xmlns="http://Printstream.org/">
      <ItemNumber>314</ItemNumber>
    </GetPSItemviaNumber>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<PSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ReorderQuantity>300</ReorderQuantity>
  <MinimumLevel>100</MinimumLevel>
  <SellPrice>9</SellPrice>
  <Active>Y</Active>
  <StockType>C</StockType>
  <ExpiryDate>12/31/2099</ExpiryDate>
  <Weight>12</Weight>
  <InventoryCode>314</InventoryCode>
  <Description>Angler Fly Rod Outfit</Description>
  <Description2 />
  <Description3 />
  <CustAccountNumber />
  <ItemNumber />
  <UOMConversion>3</UOMConversion>
  <UOM>yard</UOM>
</PSItem>
```

Get Reason Code Description

Narrative

This function will return the “Name/Description” of a “Reason Code” stored against an order.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetReasonCodeDesc"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetReasonCodeDesc xmlns="http://Printstream.org/">
      <ReasonCode>int</ReasonCode>
    </GetReasonCodeDesc>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetReasonCodeDesc xmlns="http://Printstream.org/">
      <ReasonCode>2</ReasonCode>
    </GetReasonCodeDesc>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<Misc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ReasonCodeDesc>Customer Request</ReasonCodeDesc>
</Misc>

```

Get Recipient Attributes

Narrative

This function will return a list of Attributes for a recipient

ACNO - this is the customer's account number

CustRecipID - this is the customer recipient id (if left blank, all recipients with attributes, for the customer ACNO, will be returned).

Request Transaction

```
POST /Web2PSWCF.WebService.Setup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetRecipientAttributes"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetRecipientAttributes xmlns="http://Printstream.org/">
      <xmlString>string</xmlString>
    </GetRecipientAttributes>
  </soap:Body>
</soap:Envelope>
```

Schema

GetRecipientAttributes.xsd

Sample Request Transaction

```
<?xml version="1.0" ?>
<pbs:GetRecipientAttributes xmlns="http://Printstream.org/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema"
  xmlns:ns0="printstream:efi:com">
  <pbs:RecipientInfo ACNO="VAL002" CustRecipID="0102" />
</pbs:GetRecipientAttributes>
```

Response Transaction

```
<pbs:RecipientAttributeList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="printstream:efi:com"
  xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">
  <pbs:RecipList>
    <pbs:Recipient ACNO="VAL002" CustRecipID="0102">
      <pbs:AttributeList>
        <pbs:Attribute Name="# of entrances" Value="1" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="American Greetings" Value="yes" Type="S" MaxLength="0"
Decimals="0" />
        <pbs:Attribute Name="Backer Paper" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Behind the Register" Value="11" Type="S" MaxLength="0" Decimals="0"
/>
        <pbs:Attribute Name="Black" Value="yes" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Brand Green" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Brand Green_30" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Brand Green_36" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Ceiling Graphics" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Crafts Purple" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Crafts Purple 24x66" Value="no" Type="S" MaxLength="0"
Decimals="0" />
        <pbs:Attribute Name="Custom Framing" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Edu /Demos" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Endcap Assortments" Value="10" Type="S" MaxLength="0"
Decimals="0" />
        <pbs:Attribute Name="floral Tiered fixture" Value="no" Type="S" MaxLength="0" Decimals="0"
/>
        <pbs:Attribute Name="Foodcrafting White" Value="no" Type="S" MaxLength="0" Decimals="0"
/>
        <pbs:Attribute Name="Home Dec Tan" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="IBN" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Inspiration" Value="yes" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Kids Turquoise" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Left Door Version" Value="141-167C" Type="S" MaxLength="0"
Decimals="0" />
        <pbs:Attribute Name="Lifestyle Focal Size" Value="NA" Type="S" MaxLength="0" Decimals="0"
/>
        <pbs:Attribute Name="Lowboy" Value="NA" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Mail Day" Value="Monday" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Max door sign size" Value="26.5" Type="S" MaxLength="0"
Decimals="0" />
        <pbs:Attribute Name="Music" Value="no" Type="S" MaxLength="0" Decimals="0" />
```

```

    <pbs:Attribute Name="No Smoking" Value="no" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Paging" Value="yes" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Phone System" Value="Comdial" Type="S" MaxLength="0"
Decimals="0" />
    <pbs:Attribute Name="Price Checkers" Value="no" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Project Month" Value="February" Type="S" MaxLength="0"
Decimals="0" />
    <pbs:Attribute Name="Project type" Value="Remodel" Type="S" MaxLength="0" Decimals="0"
/>
    <pbs:Attribute Name="Project Yr" Value="FY13" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Promo Window Qty" Value="2" Type="S" MaxLength="0" Decimals="0"
/>
    <pbs:Attribute Name="PULL Day" Value="Friday" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Rack" Value="84" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Right Door Store Hours" Value="32" Type="S" MaxLength="0"
Decimals="0" />
    <pbs:Attribute Name="Seasonal Feature Fixture Style" Value="NA" Type="S" MaxLength="0"
Decimals="0" />
    <pbs:Attribute Name="Seasonal Store Layout" Value="C00-72" Type="S" MaxLength="0"
Decimals="0" />
    <pbs:Attribute Name="Sewing Blue" Value="no" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="SF version / LF version HOLIDAY Store Hours" Value="SF in LF Market"
Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Shopping Cart" Value="no" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Sign Kit Sizes" Value="L" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Take a ticket" Value="Microframe Corp" Type="S" MaxLength="0"
Decimals="0" />
    <pbs:Attribute Name="Triple Tiered Stanchion" Value="yes" Type="S" MaxLength="0"
Decimals="0" />
    <pbs:Attribute Name="Type" Value="Small" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="vestibule sign Qty" Value="no" Type="S" MaxLength="0" Decimals="0"
/>
    <pbs:Attribute Name="Viking" Value="no" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="White Bridal" Value="no" Type="S" MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Window Graphics" Value="yes - green with graphic" Type="S"
MaxLength="0" Decimals="0" />
    <pbs:Attribute Name="Window Shades" Value="yes" Type="S" MaxLength="0" Decimals="0"
/>
  </pbs:AttributeList>
</pbs:Recipient>
</pbs:RecipList>
</pbs:RecipientAttributeList>

```

Get Recipient Information

Narrative

This function will return basic contact information about a recipient.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetRecipientInformation"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetRecipientInformation xmlns="http://Printstream.org/">
      <CustomerId>int</CustomerId>
      <CustomerRecipientId>string</CustomerRecipientId>
    </GetRecipientInformation>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetRecipientInformation xmlns="http://Printstream.org/">
      <CustomerId>13</CustomerId>
      <CustomerRecipientId>JJWEB</CustomerRecipientId>
    </GetRecipientInformation>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>

<ArrayOfRecipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <Recipient>

    <RecipientID>11771</RecipientID>

    <FirstName>Dan</FirstName>

    <LastName>Doe</LastName>

    <Password>ok</Password>

    <CompanyName>JJD, LLC.</CompanyName>

    <Address1>123 New South Rd</Address1>

    <Address2 />

    <Address3 />

    <CountryID>USA</CountryID>

    <CountryDescription>UNITED STATES</CountryDescription>

    <City>Sometown</City>

    <StateCode>CT</StateCode>

    <ZipCode>06123</ZipCode>

    <DayPhone>8606667777</DayPhone>

    <Email>testguy@foo.com</Email>

    <SecurityUserID>WEBADMIN</SecurityUserID>

    <IsTaxable>0</IsTaxable>

    <ISUserLogged>>false</ISUserLogged>

    <ISRecipLogin>>false</ISRecipLogin>

    <IsRecipLogged>>false</IsRecipLogged>

  </Recipient>

</ArrayOfRecipient>
```

Get Results

Narrative

This function returns a result set based on a SQL query.

The contents of xmlString must be wrapped in CDATA.

The xmlString "GetResults" contains the attribute "key". This is the encrypted token returned by GetToken. If it is invalid, the request will be rejected.

In the Cmd element, the attribute "str" is your SQL Query.

The only queries allowed must begin with a SELECT.

The query may not contain a semi-colon.

The query may not contain any of these Reserved Words:

INSERT, DELETE, MERGE, UPDATE, SET, UPDATETEXT, WRITE, TRUNCATE, EXECUTE, EXEC, ALTER, DROP, DUMP, LOAD, KILL, RESTORE, BACKUP, SAVE, SHUTDOWN, TRANSACTION, DISABLE, READTEXT, TRIGGER, REPLACE

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetResults"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetResults xmlns="http://Printstream.org/">
      <xmlString>string</xmlString>
    </GetResults>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Header>

    <Authenticate xmlns="http://Printstream.org/">

      <RecipientID>JJWEB</RecipientID>

      <Password>ok</Password>

      <CompanyCode>01</CompanyCode>

      <PlantCode>0100</PlantCode>

      <AccountNo>JJD</AccountNo>

    </Authenticate>

  </soap:Header>

  <soap:Body>

    <GetResults xmlns='http://Printstream.org/'>
      <xmlString><![CDATA[<pbs:GetResults
xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema"
key="7334C1DAF1B3FCE7">
  <pbs:Cmd str="SELECT cust_recip_id, company_name, first_name, last_name, email from
RECIPIENT where RECIP_ID = 11771 " />
</pbs:GetResults>]]></xmlString>
    </GetResults>

  </soap:Body>

</soap:Envelope>
```

Within the Response element you will find two things.

First, the schema representing the results of your SQL query.

Second, the result itself, repeated for as many lines returned by the query.

For a successful transaction, ErrorNumber will be zero (0), while ErrorMessage will be "Success". If an error condition is met, ErrorNumber will be one (1), the Response element will be empty, while ErrorMessage will contain the relevant error message.

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>

<Results xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ErrorNumber>0</ErrorNumber>
  <ErrorMessage>Success</ErrorMessage>
  <Response>
    <NewDataSet>

      <xs:schema id="NewDataSet" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata">
        <xs:element name="NewDataSet" msdata:IsDataSet="true"
msdata:MainDataTable="Result" msdata:UseCurrentLocale="true">
          <xs:complexType>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
              <xs:element name="Result">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="cust_recip_id">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="100" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="company_name">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="40" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="first_name" minOccurs="0">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="25" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="last_name">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="50" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="email" minOccurs="0">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="129" />
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:choice>
          </xs:complexType>
        </xs:schema>
      </NewDataSet>
    </Response>
  </Results>

```




```
    </xs:element>
  </xs:choice>
</xs:complexType>
</xs:element>
</xs:schema>
```

```
  <Result>
    <cust_recip_id>JJWEB</cust_recip_id>
    <company_name>JJ Dean</company_name>
    <first_name>Dean</first_name>
    <last_name>Bosse</last_name>
    <email>dean.bosse@efi.com</ email >
  </Result>

</NewDataSet>
</Response>
</Results>
```

Get Sales Tax

Narrative

This function calculates the tax amount for a specific dollar amount based on the Postal Code (Zip), City, and State.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetSalesTax"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetSalesTax xmlns="http://Printstream.org/">
      <Zip>string</Zip>
      <City>string</City>
      <State>string</State>
      <TaxAmount>double</TaxAmount>
    </GetSalesTax>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetSalesTax xmlns="http://Printstream.org/">
      <Zip>06001</Zip>
      <City>Avon</City>
      <State>CT</State>
      <TaxAmount>100</TaxAmount>
    </GetSalesTax>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<Item xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SalesTax>6.12</SalesTax>
</Item>

```

Get Shipping Preferences

Narrative

This function retrieves all available shipping preferences for a shopping cart based on Project and Recipient settings.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetShippingPreferences"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetShippingPreferences xmlns="http://Printstream.org/">
      <ItemsDTO>
        <BasketItem>
          <ItemCode>string</ItemCode>
          <Quantity>string</Quantity>
          <Price>string</Price>
          <DiscountPrice>string</DiscountPrice>
          <ItemNumber>int</ItemNumber>
          <BOQuantity>int</BOQuantity>
        </BasketItem>
        <BasketItem>
          <ItemCode>string</ItemCode>
          <Quantity>string</Quantity>
          <Price>string</Price>
          <DiscountPrice>string</DiscountPrice>
          <ItemNumber>int</ItemNumber>
          <BOQuantity>int</BOQuantity>
        </BasketItem>
      </ItemsDTO>
      <ProjectId>int</ProjectId>
      <RecipId>int</RecipId>
    </GetShippingPreferences>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetShippingPreferences xmlns="http://Printstream.org/">
      <ItemsDTO>
        <BasketItem>
          <ItemCode>314</ItemCode>
          <Quantity>1</Quantity>
          <Price>10.00</Price>
          <DiscountPrice>0</DiscountPrice>
          <ItemNumber>0</ItemNumber>
          <BOQuantity>0</BOQuantity>
        </BasketItem>
      </ItemsDTO>
      <ProjectId>29</ProjectId>
      <RecipId>11771</RecipId>
    </GetShippingPreferences>
  </soap:Body>
</soap:Envelope>

```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<ArrayOfMisc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<Misc>
```

```
  <ErrorNumber>0</ErrorNumber>
  <ProfileId>51</ProfileId>
  <ShipModeId>1</ShipModeId>
  <ShipModeDesc>UPS Ground</ShipModeDesc>
  <BillNo>mark</BillNo>
  <BillName>mark</BillName>
  <SatDelivery>N</SatDelivery>
  <DeliveryAck>N</DeliveryAck>
  <Insurance>N</Insurance>
  <InsAmount>15</InsAmount>
  <Defaults>N</Defaults>
  <ShipCharge>15</ShipCharge>
  <NumPackages>0</NumPackages>
  <ShowOnWeb>Y</ShowOnWeb>
  <ShipBillId>0</ShipBillId>
  <AddressId>23502</AddressId>
  <Active>Y</Active>
```

```
</Misc>
```

```
<Misc>
```

```
  <ErrorNumber>0</ErrorNumber>
  <ProfileId>60</ProfileId>
  <ShipModeId>9</ShipModeId>
  <ShipModeDesc>USMAIL</ShipModeDesc>
  <BillNo />
  <BillName />
  <SatDelivery>N</SatDelivery>
  <DeliveryAck>N</DeliveryAck>
```

```

    <Insurance>N</Insurance>
    <InsAmount>0</InsAmount>
    <Defaults>N</Defaults>
    <ShipCharge>15</ShipCharge>
    <NumPackages>0</NumPackages>
    <ShowOnWeb>Y</ShowOnWeb>
    <ShipBillId>0</ShipBillId>
    <AddressId>0</AddressId>
    <Active>Y</Active>
  </Misc>
<Misc>
  <ErrorNumber>0</ErrorNumber>
  <ProfileId>540</ProfileId>
  <ShipModeId>255</ShipModeId>
  <ShipModeDesc>UPS 3 Day Select</ShipModeDesc>
  <BillNo />
  <BillName />
  <SatDelivery>N</SatDelivery>
  <DeliveryAck>N</DeliveryAck>
  <Insurance>N</Insurance>
  <InsAmount>0</InsAmount>
  <Defaults>Y</Defaults>
  <ShipCharge>0</ShipCharge>
  <NumPackages>0</NumPackages>
  <ShowOnWeb>Y</ShowOnWeb>
  <ShipBillId>0</ShipBillId>
  <AddressId>0</AddressId>
  <Active>Y</Active>
</Misc>
</ArrayOfMisc>

```

Get SO List

Narrative

This function returns a list of all sales orders that meet the filter criteria passed in the request.

FromDate = All orders where Sales Order Creation Date >= Date specified

ToDate = All orders where Sales Order Creation Date >= Date specified

Status = All orders whose status = Status specified

Project ID = All orders placed for Project = Project ID specified

Job No = All orders processed under Job Number = Job No specified

StkHeaderID = All orders where any line item ordered = StkHeaderID specified

BOStatus = All orders where Back Order Status = BOStatus specified

Request Transaction

```
POST /Web2PS WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetSOList"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetSOList xmlns="http://Printstream.org/">
      <FromDate>string</FromDate>
      <ToDate>string</ToDate>
      <StatusID>int</StatusID>
      <ProjectID>int</ProjectID>
      <JobNO>int</JobNO>
      <StkHeaderId>int</StkHeaderId>
      <BOStatus>int</BOStatus>
    </GetSOList>
  </soap:Body>
</soap:Envelope>
```


Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetSOList xmlns="http://Printstream.org/">
      <FromDate>03/26/2012</FromDate>
      <ToDate>03/29/2012</ToDate>
      <StatusID>3</StatusID>
      <ProjectID>29</ProjectID>
      <JobNO>461</JobNO>
      <StkHeaderId>0</StkHeaderId>
      <BOStatus>0</BOStatus>
    </GetSOList>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<ArrayOfSalesOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
  <SalesOrder>
```

```
    <ErrorNumber>0</ErrorNumber>
```

```
    <OrderNo>150279</OrderNo>
```

```
    <BillTo>11776</BillTo>
```

```
    <Status>Open</Status>
```

```
    <ShipTo>11776</ShipTo>
```

```
    <OrderBy>11776</OrderBy>
```

```
    <CustomerId>13</CustomerId>
```

```
    <ModifiedDate>06/04/2010</ModifiedDate>
```

```
    <ShippingHandling>0</ShippingHandling>
```

```
    <Discount>0</Discount>
```

```
    <SalesTax>0</SalesTax>
```

```
    <Total>0</Total>
```

```
  </SalesOrder>
```

... **<SalesOrder>** repeats N times for each sales order within range requested

```
</ArrayOfSalesOrder>
```

Get SYS Param

Narrative

System parameters can be queried so an external system can apply business logic based on parameters (behaviors) held in PrintStream. In the example below the system is returning the URL string for a connection to an external Web to Print site.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswbl2.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetSYSPARAM"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetSYSPARAM xmlns="http://Printstream.org/">
      <ParamName>string</ParamName>
    </GetSYSPARAM>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetSYSPARAM xmlns="http://Printstream.org/">
      <ParamName>HAF_EMAIL</ParamName>
    </GetSYSPARAM>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Misc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <SYS_PARAM>help@foo.com</SYS_PARAM>
</Misc>
```

Directory of System Parameters

The following table describes each of the system parameters that can be accessed through the above web service function.

PARAM_NAME indicates the value that should be placed in the ParamName tag in the Request

PARAM_GROUP indicates the kind of parameter based on system defined "Groups".

PARAM_DESCRIPTION describes the intent of the parameter and how it is applied.

PARAM_VALUE describes the value that will be returned in the SYS_PARAM response.

PARAM_NAME	PARAM_GROUP	PARAM_DESCRIPTION	PARAM_VALUE (example)
ANZEgate_URL	ANZ Egate Settings	ANZEgate URL Australian electronic banking portal.	https://migs.mastercard.com.au/vpcdps
ANZEgate_Version	ANZ Egate Settings	ANZEgate Version Australian electronic banking portal	1
EXT_SEARCH_DISPNAME	External Search	External Search Program Caption	Advanced Search
EXT_SEARCH_PROGNAM	External Search	External Search Program Name	D:\stream\advsearch.exe
FF_CUSTDEF	Fulfillment Settings	Customer Mode Allow fulfillment interface to default all functions to just a single customer.	0 = False, 1 = True
FF_DEFAULT_PRINTER	Fulfillment Settings	Default Printer	\\dbserver\Samsung CLP-500 Series
FF_DISPLAY	Fulfillment Settings	Display Settings	0 = False, 1 = True
FF_PHONE_FORMAT	Fulfillment Settings	Phone Number Format	###-#####
FF_SHOW_CUST_TREE	Fulfillment Settings	Show Customer Tree	0 = False, 1 = True
FF_SHOW_LISTBAR	Fulfillment Settings	Show List Bar	0 = False, 1 = True
FF_SHOW_STATUSBAR	Fulfillment Settings	Show Status Bar	0 = False, 1 = True
FF_SHOW_TOOLBAR	Fulfillment Settings	Show Tool Bar	0 = False, 1 = True
FF_TOOLBAR_SIZE	Fulfillment Settings	Toolbar Size	0 = False, 1 = True
FF_TREEVIEW_FONTNAME	Fulfillment Settings	Treeview Font Name	MS Sans Serif
FF_TREEVIEW_FONTSIZE	Fulfillment Settings	Treeview Font Size	8
HAF_EMAIL	Fulfillment Settings	Web Help and Feedback Email	HelpDesk@foo.com
SO_FOCUS	Fulfillment Settings	Sales Order Focus Field. Which field has primary focus when sales order entry screen opens	Sales Order #
ICVERIFY_LICENSE_COUNT	ICVerify Settings	ICVerify Credit Card License Count	5
Pageflex_URL	Pageflex Interface	Pageflex URL	http://PS.pageflex/storefront/
PSWebNet_URL	PS Web Net Interface	PS Web Net URL	http://demo.foo.com/
PSWeb_AutoComplete	PS Web Settings	PS Web Use Auto Complete	0 = False, 1 = True
PSInventoryWS_Mode	PSInventoryWS Interface	PSInventory Web Service Mode	0 = False, 1 = True

PARAM_NAME	PARAM_GROUP	PARAM_DESCRIPTION	PARAM_VALUE (example)
PSInventoryWS_URL	PSInventoryWS Interface	PSInventory Web Service URL	http://127.0.0.1/Streamline.Webservice.Inventory/Inventory.asmx
PSShippingWS_URL	PSShippingWS Interface	PSShipping Web Service URL	
SN_HOST_ADDRESS	Sequence Service	Sequence Services URL	
SN_HOST_PORT	Sequence Service	Sequence Service Host Port	
SN_PASSWORD	Sequence Service	Sequence Service End Point Authentication Password	
SN_PROXY_ADDRESS	Sequence Service	Sequence Proxy Address	
SN_PROXY_LOGIN	Sequence Service	Sequence Proxy Login	
SN_PROXY_PASSWORD	Sequence Service	Sequence Proxy Password	
SN_PROXY_PORT	Sequence Service	Sequence Proxy Port	
SN_TIMEOUT	Sequence Service	Sequence Service Timeout	
SN_USER	Sequence Service	Sequence Service End Point Authentication User	
SN_WSDL_LOCATION	Sequence Service	Sequence WSDL File Location	
SPE_HOST_ADDRESS	Storage Pricing Interface	Storage Pricing Engine URL	
SPE_TIMEOUT	Storage Pricing Interface	Storage Pricing Engine Timeout	
SPE_WSDL_LOCATION	Storage Pricing Interface	Storage Pricing Engine WSDL	
STKHEADER_ON	System Settings	Multi-Plant Inventory Takes Effect On Setting this Flag	0 = False, 1 = True
SYS_RUN_MODE	System Settings	System Interface Settings	0 = False, 1 = True
TAX_HOST_ADDRESS	Tax Service	Tax Services URL	
TAX_HOST_PORT	Tax Service	Tax Service Host Port	
TAX_PASSWORD	Tax Service	Tax Service End Point Authentication Password	
TAX_PROXY_ADDRESSES	Tax Service	Tax Proxy Address	
TAX_PROXY_LOGIN	Tax Service	Tax Proxy Login	
TAX_PROXY_PASSWORD	Tax Service	Tax Proxy Password	
TAX_PROXY_PORT	Tax Service	Tax Proxy Port	
TAX_TIMEOUT	Tax Service	Tax Service Timeout	
TAX_USER	Tax Service	Tax Service End Point Authentication User	
TAX_WSDL_LOCATION	Tax Service	Tax WSDL File Location	
VS_HOST_ADDRESS	VeriSign Settings	VeriSign Host Address	test-payflow.verisign.com
VS_HOST_PORT	VeriSign Settings	VeriSign Host Port	443
VS_PAYFLOWPRO_URL	VeriSign Settings	VeriSign PayflowPro URL	
VS_PROXY_ADDRESS	VeriSign Settings	VeriSign Proxy Address	
VS_PROXY_LOGIN	VeriSign Settings	VeriSign Proxy Login	
VS_PROXY_PASSWORD	VeriSign Settings	VeriSign Proxy Password	



PARAM_NAME	PARAM_GROUP	PARAM_DESCRIPTION	PARAM_VALUE (example)
VS_PROXY_PORT	VeriSign Settings	VeriSign Proxy Port	0
VS_TIMEOUT	VeriSign Settings	VeriSign Timeout	30
FILEUP_URL	Web File Upload	FileUpEE Target URL	http://127.0.0.1/FileUpEE/UploadSamples/Intermediate/ServerToServer/AS
FILEUP_URL_BR	Web File Upload	FileUpEE URL (BR)	
Web2Print_URL	Web2Print Interface	Web2Print URL	http://demo.foo.com/W2PStudio
Web2Print_WS	Web2Print Interface	Web2Print Web Service	http://demo.foo.com/W2PStudio/WebService/WebService.asmx
XMPie_URL	XMPie Interface	XMPie_URL	http://up2.xmpie.net/ustorewsapi/

Get Token

Narrative

The Token returned in the Response Transaction is used by the GetResults function as part of the authentication logic.

The unencrypted token is stored in the System Parameters table.

Web2PS Service			
Parameter Name	User can over-ride	Value	
Web2PS Token	☐	test123	

To retrieve the encrypted token, simply passed the unencrypted token in the Request Transaction. If the incoming token matches the value in the System Parameters table, the encrypted token will be returned. Otherwise, an error will be returned.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetToken"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetToken xmlns="http://Printstream.org/">
      <ParamValue>string</ParamValue>
    </GetToken>
  </soap:Body>
</soap:Envelope>
```


Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetToken xmlns="http://Printstream.org/">
      <ParamValue>test123</ParamValue>
    </GetToken>
  </soap:Body>
</soap:Envelope>
```

For a successful transaction, ErrorNumber will be zero (0), while ErrorMessage will be "Success". If an error condition is met, ErrorNumber will be one (1), the Token element will be empty, while ErrorMessage will contain the relevant error message.

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Results xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <ErrorNumber>0</ErrorNumber>
  <ErrorMessage>Success</ErrorMessage>
  <Token>7334C1DAF1B3FCE7</Token>
</Results>
```

Get Unreceived PO Receipts

Narrative

This function returns a list of unreceived purchase order receipts generated from an integrated system (Pace or Monarch)

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetUnReceivedPOReceipts"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetUnReceivedPOReceipts xmlns="http://Printstream.org/">
      <ACNO>string</ACNO>
      <FromDate>string</FromDate>
      <ToDate>string</ToDate>
      <CompanyCode>string</CompanyCode>
    </GetUnReceivedPOReceipts>
  </soap:Body>
</soap:Envelope>
```

Sample Transaction Request

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
```

```

    <AccountNo>JJD</AccountNo>
  </Authenticate>
</soap:Header>
<soap:Body>
  <GetUnReceivedPOReceipts xmlns="http://Printstream.org/">
    <ACNO>NEL123</ACNO>
    <FromDate>11/22/13 00:00:00</FromDate>
    <ToDate>11/25/13 23:59:59</ToDate>
    <CompanyCode>01</CompanyCode>
  </GetUnReceivedPOReceipts>
</soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<UnreceivedFGARN Message="Success" xmlns="http://tempuri.org/UnreceivedFGARN.xsd">
  <FGARNByDate>
    <DueDate>2013-11-24T13:49:27-05:00</DueDate>
    <CustomerAcNo>NEL123</CustomerAcNo>
    <CustomerName>Nelsen Fulfillment Orders Service</CustomerName>
    <ItemCode>4297</ItemCode>
    <InventoryCode>ALLFIELDS-POD-1</InventoryCode>
    <Description1>All Fields Testing POD - 1</Description1>
    <Description2 />
    <Description3 />
    <QtyToReceive>87</QtyToReceive>
    <PSPOOrderLineNO>1</PSPOOrderLineNO>
    <PSPOOrderNO>8575</PSPOOrderNO>
    <InvMISJobNo>14462</InvMISJobNo>
    <PartFinal xml:space="preserve"></PartFinal>
    <DeliveryDate>1900-01-01T00:00:00-05:00</DeliveryDate>
    <DeliveryFlag xml:space="preserve"></DeliveryFlag>
    <OrderType>J</OrderType>
    <EFIPOHdrID>14462</EFIPOHdrID>
    <EFIPOLineID>Prod:37562</EFIPOLineID>
  </FGARNByDate>
</UnreceivedFGARN>

```



```
<RequestNumber>7432</RequestNumber>
</FGARNByDate>
<FGARNByDate>
  <DueDate>2013-11-24T15:12:27-05:00</DueDate>
  <CustomerAcNo>NEL123</CustomerAcNo>
  <CustomerName>Nelsen Fulfillment Orders Service</CustomerName>
  <ItemCode>6842</ItemCode>
  <InventoryCode>NOV22-POD-1</InventoryCode>
  <Description1>NOV22-POD-1</Description1>
  <Description2 />
  <Description3 />
  <QtyToReceive>76</QtyToReceive>
  <PSPOOrderLineNO>2</PSPOOrderLineNO>
  <PSPOOrderNO>8575</PSPOOrderNO>
  <InvMISJobNo>14462</InvMISJobNo>
  <PartFinal xml:space="preserve"></PartFinal>
  <DeliveryDate>1900-01-01T00:00:00-05:00</DeliveryDate>
  <DeliveryFlag xml:space="preserve"></DeliveryFlag>
  <OrderType>J</OrderType>
  <EFIPOHdrID>14462</EFIPOHdrID>
  <EFIPOLineID>Prod:37563</EFIPOLineID>
  <RequestNumber>7433</RequestNumber>
</FGARNByDate>
<FGARNByDate>
  <DueDate>2013-11-25T21:11:15-05:00</DueDate>
  <CustomerAcNo>NEL123</CustomerAcNo>
  <CustomerName>Nelsen Fulfillment Orders Service</CustomerName>
  <ItemCode>4412</ItemCode>
  <InventoryCode>REGINA BOF NO POD</InventoryCode>
  <Description1>REGINA BOF NO POD</Description1>
  <Description2 />
  <Description3 />
```



```
<QtyToReceive>100</QtyToReceive>
<PSPOOrderLineNO>1</PSPOOrderLineNO>
<PSPOOrderNO>8587</PSPOOrderNO>
<InvMISJobNo>14482</InvMISJobNo>
<PartFinal xml:space="preserve"></PartFinal>
<DeliveryDate>1900-01-01T00:00:00-05:00</DeliveryDate>
<DeliveryFlag xml:space="preserve"></DeliveryFlag>
<OrderType>J</OrderType>
<EFIPOHdrID>14482</EFIPOHdrID>
<EFIPOLineID>Prod:37644</EFIPOLineID>
<RequestNumber>7452</RequestNumber>
</FGARNByDate>
</UnreceivedFGARN>
```

Note The root node **UnreceivedFGARN** contains an attribute entitled "**Message**". Should an error occur, look in this attribute for a meaningful message. The message "**Success**" means a successful response was generated from the xml request.

Get Up Sell and Cross Sell Items

Narrative

This function returns a list of Up Sell and Cross Sell items for a specific nomination item.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetUpSellAndCrossSell"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetUpSellAndCrossSell xmlns="http://Printstream.org/">
      <STKHeaderId>int</STKHeaderId>
      <ProjectId>int</ProjectId>
      <CustomerId>int</CustomerId>
      <NominationId>int</NominationId>
      <RecipientId>int</RecipientId>
    </GetUpSellAndCrossSell>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
```

```

        <AccountNo>JJD</AccountNo>

    </Authenticate>

</soap:Header>

<soap:Body>

    <GetUpSellAndCrossSell xmlns="http://Printstream.org">
        <STKHeaderId>245</STKHeaderId>
        <ProjectId>29</ProjectId>
        <CustomerId>13</CustomerId>
        <NominationId>24</NominationId>
        <RecipientId>11771</RecipientId>
    </GetUpSellAndCrossSell>
</soap:Body>

</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>

<ArrayOfItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<Item>

    <PAPSizeId>270</PAPSizeId>

    <STKHeaderCode>246</STKHeaderCode>

    <StkHeaderId>0</StkHeaderId>

    <ItemCode>246</ItemCode>

    <ItemNumber>315</ItemNumber>

    <InventoryCode>315</InventoryCode>

    <Description>Classic Fly Rod Outfit</Description>

    <Description2>2012 Version</Description2>

    <Description3 />

    <OnHand>2712490</OnHand>

    <Allocated>-2689414</Allocated>

    <Price>10</Price>

    <MinimumLimit>0</MinimumLimit>

    <OrderLimit>0</OrderLimit>

    <OnHold>false</OnHold>

    <IsBackOrder>false</IsBackOrder>

    <ForceMax>false</ForceMax>

```

```

<ForceMin>false</ForceMin>
<NewItem>false</NewItem>
<Notes>False</Notes>
<UpCross>1</UpCross>
<HoldReason>0</HoldReason>
<MinReason>0</MinReason>
<MaxReason>0</MaxReason>
<POP />
<Customizable>N</Customizable>
<UnitOfIssue>1</UnitOfIssue>
<UnitIssueDescription>each</UnitIssueDescription>
<BORule>0</BORule>
<OrderQuantityOf>0</OrderQuantityOf>
<ReorderQuantity>0</ReorderQuantity>
<SubstituteBasis>0</SubstituteBasis>
<SubstituteItem>0</SubstituteItem>
<SubstituteEffectDate>2005-04-18T09:00:55</SubstituteEffectDate>
<SubExpireDate>2006-07-03T08:49:52</SubExpireDate>
<Available>0</Available>
<SalesTax>0</SalesTax>
<Active>true</Active>
  </Item>
<Item>
  <PAPSizeId>271</PAPSizeId>
  <STKHeaderCode>247</STKHeaderCode>
  <StkHeaderId>0</StkHeaderId>
  <ItemCode>247</ItemCode>
  <ItemNumber>316</ItemNumber>
  <InventoryCode>316</InventoryCode>
  <Description>Quest Fly Rod</Description>
  <Description2>ok</Description2>
  <Description3 />

```



```

<OnHand>1201943</OnHand>
  <Allocated>12489</Allocated>
  <Price>10</Price>
  <MinimumLimit>0</MinimumLimit>
  <OrderLimit>0</OrderLimit>
  <OnHold>false</OnHold>
  <IsBackOrder>false</IsBackOrder>
  <ForceMax>false</ForceMax>
  <ForceMin>false</ForceMin>
  <NewItem>false</NewItem>
  <Notes>False</Notes>
  <UpCross>2</UpCross>
  <HoldReason>0</HoldReason>
  <MinReason>0</MinReason>
  <MaxReason>0</MaxReason>
  <POP />
  <Customizable>N</Customizable>
  <UnitOfIssue>0</UnitOfIssue>
  <UnitIssueDescription>0</UnitIssueDescription>
  <BORule>0</BORule>
  <OrderQuantityOf>0</OrderQuantityOf>
  <ReorderQuantity>0</ReorderQuantity>
  <SubstituteBasis>0</SubstituteBasis>
  <SubstituteItem>0</SubstituteItem>
  <SubstituteEffectDate>2005-04-18T09:00:55</SubstituteEffectDate>
  <SubExpireDate>2099-12-31T00:00:00</SubExpireDate>
  <Available>0</Available>
  <SalesTax>0</SalesTax>
  <Active>true</Active>
</Item>
</ArrayOfItem>

```

Get Web Description

Narrative

The function returns a list of all miscellaneous options that should be displayed in the “Help and Download” area of the website for the customer.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: pswb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetWebDescription"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetWebDescription xmlns="http://Printstream.org/">
      <AccNo>string</AccNo>
    </GetWebDescription>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetWebDescription xmlns="http://Printstream.org/">
      <AccountNumber>JJD</AccountNumber>
    </GetWebDescription>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMisc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Misc>
    <WebFormDescription>Help Document Download</WebFormDescription>
    <Description1>Helpdoc.pdf</Description1>
  </Misc>
  ... <Misc> repeats N times for each user defined web option for the customer
</ArrayOfMisc>
```

Get Web User

Narrative

This function will return the "Name" of the user's (recipients) security role. This could then be used to subsequently request specific security privileges tied to that role.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/GetWebUser"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <GetWebUser xmlns="http://Printstream.org/" />
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
```



```
<GetWebUser xmlns="http://Printstream.org/" />
</soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<ArrayOfMisc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Misc>
    <Name>WEBADMIN</Name>
  </Misc>
</ArrayOfMisc>
```

Hello World

Narrative

Verify that Web Service is running and responding.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/HelloWorld"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <HelloWorld xmlns="http://Printstream.org/" />
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <HelloWorld xmlns="http://Printstream.org/">
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Misc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <string xmlns="http://Printstream.org/">Hello World 2/11/2011 11:20:05
AM</string>
</Misc>
```

Inventory ASN

Narrative

This function creates an Advanced Shipping Notification for an inventory item.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/InventoryASN"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <InventoryASN xmlns="http://Printstream.org/">
      <CustACNO>string</CustACNO>
      <StkItemNum>string</StkItemNum>
      <InvCode>string</InvCode>
      <EstQty>double</EstQty>
      <Comments>string</Comments>
      <SchedDate>dateTime</SchedDate>
      <JobNum>double</JobNum>
      <Priority>int</Priority>
    </InventoryASN>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
```

```

    <CompanyCode>01</CompanyCode>

    <PlantCode>0100</PlantCode>

    <AccountNo>JJD</AccountNo>

  </Authenticate>

</soap:Header>

<soap:Body>
  <InventoryASN xmlns="http://Printstream.org/">
    <CustACNO>JJD</CustACNO>
    <StkItemNum />
    <InvCode>BW SC-98-04</InvCode>
    <EstQty>5000</EstQty>
    <Comments>Test the service</Comments>
    <SchedDate>11/27/2012</SchedDate>
    <JobNum>0</JobNum>
    <Priority>0</Priority>
  </InventoryASN>
</soap:Body>

</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>

<UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <STKItemId>0</STKItemId>
  <ItemNumber>0</ItemNumber>
  <Message>Success</Message>
</UpdatedPSItem>

```


Inventory Create

Narrative

This function will create a new inventory item for a customer.

Message will be "Success" when transaction is good as well along with the new ItemNumber, otherwise an appropriate error will display (ItemNumber will be zero in an error condition).

Request Transaction

```
POST /Web2PSWCF.WebService.Setup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap12:Header>
  <soap12:Body>
    <InventoryCreate xmlns="http://Printstream.org/">
      <CustACNO>string</CustACNO>
      <InvCode>string</InvCode>
      <RevCode>string</RevCode>
      <StockType>string</StockType>
      <Description1>string</Description1>
      <Description2>string</Description2>
      <Description3>string</Description3>
      <TrackSkids>string</TrackSkids>
      <MinLevel>double</MinLevel>
      <ReorderQty>double</ReorderQty>
      <ActivityCode>double</ActivityCode>
      <UnitOfCost>double</UnitOfCost>
      <UOMCostDesc>string</UOMCostDesc>
      <AverageCost>double</AverageCost>
      <Location>string</Location>
      <Skids>string</Skids>
      <Cartons>string</Cartons>
      <Pcspercarton>int</Pcspercarton>
      <LoosePieces>int</LoosePieces>
      <LotTracking>string</LotTracking>
      <VendorLotNo>string</VendorLotNo>
      <TrackingType>string</TrackingType>
      <CreatedDate>dateTime</CreatedDate>
      <Group>string</Group>
      <Family>string</Family>
      <Category>string</Category>
      <DeclaredValue>double</DeclaredValue>
      <ExpirationDate>dateTime</ExpirationDate>
      <DefaultPickLocation>string</DefaultPickLocation>
```

```

<AllowWebView>string</AllowWebView>
<ProductCode>string</ProductCode>
<StandardSellPrice>double</StandardSellPrice>
<SkidNotes>string</SkidNotes>
<Classification>string</Classification>
<AllowRevision>string</AllowRevision>
<UOM>string</UOM>
<UOMDescription>string</UOMDescription>
<ChargeMethod>string</ChargeMethod>
<SellMethod>string</SellMethod>
<FixedMarkup>string</FixedMarkup>
<StandardCost>double</StandardCost>
<WebLevel1>string</WebLevel1>
<WebLevel2>string</WebLevel2>
<WebLevel3>string</WebLevel3>
<WebLevel4>string</WebLevel4>
<IsGenericGFC>string</IsGenericGFC>
<Numberup>string</Numberup>
<Weight>double</Weight>
<Width>string</Width>
<Length>string</Length>
<Pages>string</Pages>
<PagesBack>string</PagesBack>
<PlannerItem>string</PlannerItem>
<PackageType>string</PackageType>
<ExternalItemNo>string</ExternalItemNo>
<IsActive>string</IsActive>
<PartialPOD>string</PartialPOD>
<EFICtgID>string</EFICtgID>
<EFIDptID>string</EFIDptID>
<MISTemplateID>string</MISTemplateID>
<DSFShared>string</DSFShared>
<CombineInCarton>string</CombineInCarton>
<Height>string</Height>
<VendorCode>string</VendorCode>
<DefPOTolerance>string</DefPOTolerance>
<ExtInvCode>string</ExtInvCode>
<WebFormDesc>string</WebFormDesc>
<CountryOfOrigin>string</CountryOfOrigin>
<LayoutAcross>int</LayoutAcross>
<LayoutAround>int</LayoutAround>
<CSRFullName>string</CSRFullName>
<SalesRepFullName>string</SalesRepFullName>
<ChargeForStorage>string</ChargeForStorage>
<Customizable>string</Customizable>
<MaxUseQty>int</MaxUseQty>
<MaxUsePercent>double</MaxUsePercent>
<PurchaseUOM>double</PurchaseUOM>
<PurchaseUOMDesc>string</PurchaseUOMDesc>
<WebForm>string</WebForm>
<CubeHeight>double</CubeHeight>
<CubeLength>double</CubeLength>
<CubeWidth>double</CubeWidth>
<CubeQuantity>double</CubeQuantity>
<CubeStorageCost>double</CubeStorageCost>
<SortField1>string</SortField1>
<SortField2>string</SortField2>
<ReplacementCode>string</ReplacementCode>
<ReviseDate>dateTime</ReviseDate>
<MinimumDays>int</MinimumDays>
<ReorderDays>int</ReorderDays>
<InternationalType>string</InternationalType>

```



```
<OneTimeUse>string</OneTimeUse>
<Notes>string</Notes>
<EmailReplenish>string</EmailReplenish>
<EmailRequired>string</EmailRequired>
<PackMaterialUse>string</PackMaterialUse>
<TurnTime>double</TurnTime>
<ForestTypeID>string</ForestTypeID>
<Manager>string</Manager>
<Taxable>string</Taxable>
</InventoryCreate>
</soap12:Body>
</soap12:Envelope>
```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Header>

    <Authenticate xmlns="http://Printstream.org/">

      <RecipientID>JJWEB</RecipientID>

      <Password>ok</Password>

      <CompanyCode>01</CompanyCode>

      <PlantCode>0100</PlantCode>

      <AccountNo>JJD</AccountNo>

    </Authenticate>

  </soap:Header>

  <soap:Body>

    <InventoryCreate xmlns="http://Printstream.org/">
      <CustACNO>JJD</CustACNO>
      <InvCode>TESTWS-041912-32</InvCode>
      <RevCode />
      <StockType>C</StockType>
      <Description1>Test WS on Wed</Description1>
      <Description2 />
      <Description3 />
      <TrackSkids>Y</TrackSkids>
      <MinLevel>0</MinLevel>
      <ReorderQty>0</ReorderQty>
      <ActivityCode>0</ActivityCode>
      <UnitOfCost>1</UnitOfCost>
      <AverageCost>0</AverageCost>
      <Location />
      <Skids />
      <Cartons />
      <Pcspercarton>0</Pcspercarton>
      <LoosePieces>0</LoosePieces>
      <LotTracking />
      <VendorLotNo />
      <TrackingType />
      <CreatedDate>03/28/2012</CreatedDate>
      <Group>env</Group>
      <Family>fam</Family>
      <Category>cat</Category>
      <DeclaredValue>0</DeclaredValue>
      <ExpirationDate>12/25/2999</ExpirationDate>
      <DefaultPickLocation />
      <AllowWebView>N</AllowWebView>
    </InventoryCreate>
  </soap:Body>
</soap:Envelope>

```

```

<ProductCode />
<StandardSellPrice>0</StandardSellPrice>
  <SkidNotes />
  <Classification />
  <AllowRevision />
  <UOM />
  <UOMDescription />
  <ChargeMethod />
  <SellMethod />
  <FixedMarkup />
  <StandardCost>0</StandardCost>
  <WebLevel1 />
  <WebLevel2 />
  <WebLevel3 />
  <WebLevel4 />
  <IsGenericGFC>N</IsGenericGFC>
  <Numberup />
  <Weight>4.22</Weight>
  <Width></Width>
  <Length></Length>
  <Pages></Pages>
  <PagesBack></PagesBack>
  <PlannerItem></PlannerItem>
  <PackageType>P</PackageType>
  <ExternalItemNo></ExternalItemNo>
  <IsActive>Y</IsActive>
    <PartialPOD></PartialPOD>
  <EFICtgID>1669</EFICtgID>
  <EFIDptID>1779</EFIDptID>
  <MISTemplateID></MISTemplateID>
  <DSFShared>N</DSFShared>
  <CombineInCarton></CombineInCarton>
  <Height>10</Height>
  <VendorCode></VendorCode>
  <DefPOTolerance>0</ DefPOTolerance >
  <ExtInvCode></ExtInvCode>
  <WebFormDesc></WebFormDesc>
  <CountryOfOrigin>Canada</CountryOfOrigin>
  <LayoutAcross>0</LayoutAcross>
  <LayoutAround>0</LayoutAround>
  <CSRFullName></CSRFullName>
  <SalesRepFullName></SalesRepFullName>
  <ChargeForStorage>N</ChargeForStorage>
  <Customizable>N</Customizable>
  <MaxUseQty>0</MaxUseQty>
  <MaxUsePercent>0.0</MaxUsePercent>
  <PurchaseUOM>0</PurchaseUOM>
  <PurchaseUOMDesc></PurchaseUOMDesc>
  <WebForm>N</WebForm>
  <CubeHeight>0.0</CubeHeight>
  <CubeLength>0.0</CubeLength>
  <CubeWidth>0.0</CubeWidth>
  <CubeQuantity>0.0</CubeQuantity>
  <CubeStorageCost>0</CubeStorageCost>
  <SortField1>sort1</SortField1>
  <SortField2>sort2</SortField2>

```



```
<ReplacementCode>rep code 1</ReplacementCode>
<ReviseDate>2019-02-25T00:00:00</ReviseDate>
<MinimumDays>33</MinimumDays>
<ReorderDays>55</ReorderDays>
<InternationalType></InternationalType>
<OneTimeUse></OneTimeUse>
<Notes>these are the item notes</Notes>
<EmailReplenish></EmailReplenish>
<EmailRequired>N</EmailRequired>
<PackMaterialUse>N</PackMaterialUse>
<TurnTime>0</TurnTime>
<ForestTypeID>FT-2-Dean</ForestTypeID>
<Manager>Jim</Manager>
<Taxable>Y</Taxable>
</InventoryCreate>
</soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <STKItemId>0</STKItemId>
  <ItemNumber>78935</ItemNumber>
  <Message>Success</Message>
</UpdatedPSItem>
```

Inventory Create by Batch

Narrative

This function will create new inventory items for a customer, from a batch xml.

Message will be "Success" when each transaction is good as well along with the new ItemNumber, otherwise an appropriate error will display (ItemNumber will be zero in an error condition).

Transactions will be validated against the InventoryCreate schema.

Request Transaction

```
POST /Web2PSWCF.WebService.Setup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/InventoryCreateByBatch"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <InventoryCreateByBatch xmlns="http://Printstream.org/">
      <xmlString>string</xmlString>
    </InventoryCreateByBatch>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><soap:Header><Authenticate
xmlns="http://Printstream.org/"><RecipientID>JJWEB</RecipientID><Password>ok</Password><CompanyCo
de>01</CompanyCode><PlantCode>0100</PlantCode><AccountNo>JJD</AccountNo></Authenticate></soap:
Header><soap:Body><InventoryCreateByBatch xmlns="http://Printstream.org/"><xmlString>
```

```
<![CDATA[<pbs:InventoryCreate xmlns="http://Printstream.org/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema"
xmlns:ns0="printstream:efi.com"> <pbs:ItemList> <pbs:Item> <pbs:CustACNO>JJD</pbs:CustACNO>
<pbs:InvCode>TESTWS-040815 (3)</pbs:InvCode> <pbs:RevCode/> <pbs:StockType>C</pbs:StockType>
<pbs:Description1>Test DP dll for (3)</pbs:Description1> <pbs:Description2>(12190)</pbs:Description2>
<pbs:Description3/> <pbs:TrackSkids>Y</pbs:TrackSkids> <pbs:MinLevel>0</pbs:MinLevel>
<pbs:ReorderQty>0</pbs:ReorderQty> <pbs:ActivityCode>0</pbs:ActivityCode>
<pbs:UnitOfCost>1000</pbs:UnitOfCost> <pbs:UOMCostDesc>Thousand</pbs:UOMCostDesc>
<pbs:AverageCost>0</pbs:AverageCost> <pbs:Location/> <pbs:Skids>0</pbs:Skids>
```



```

<pbs:Cartons>0</pbs:Cartons> <pbs:Pcspercarton>0</pbs:Pcspercarton>
<pbs:LoosePieces>0</pbs:LoosePieces> <pbs:LotTracking/> <pbs:VendorLotNo/> <pbs:TrackingType/>
<pbs:CreateDate>2015-03-26T00:00:00</pbs:CreateDate> <pbs:Group>env</pbs:Group>
<pbs:Family>fam</pbs:Family> <pbs:Category>cat</pbs:Category> <pbs:DeclaredValue>0</pbs:DeclaredValue>
<pbs:ExpirationDate>2099-12-25T00:00:00</pbs:ExpirationDate> <pbs:DefaultPickLocation/>
<pbs:AllowWebView>N</pbs:AllowWebView> <pbs:ProductCode>0</pbs:ProductCode>
<pbs:StandardSellPrice>0</pbs:StandardSellPrice> <pbs:SkidNotes/> <pbs:Classification/> <pbs:AllowRevision/>
<pbs:UOM>0</pbs:UOM> <pbs:UOMDescription/> <pbs:ChargeMethod/> <pbs:SellMethod/>
<pbs:FixedMarkup/> <pbs:StandardCost>0</pbs:StandardCost> <pbs:WebLevel1/> <pbs:WebLevel2/>
<pbs:WebLevel3/> <pbs:WebLevel4/> <pbs:IsGenericGFC>N</pbs:IsGenericGFC>
<pbs:Numberup>0</pbs:Numberup> <pbs:Weight>6.29</pbs:Weight> <pbs:Width>100</pbs:Width>
<pbs:Length>1</pbs:Length> <pbs:Pages>0</pbs:Pages> <pbs:PagesBack>0</pbs:PagesBack>
<pbs:PlannerItem/> <pbs:PackageType/> <pbs:ExternalItemNo/> <pbs:IsActive>Y</pbs:IsActive>
<pbs:PartialPOD/> <pbs:EFICtgID>1669</pbs:EFICtgID> <pbs:EFIDptID>1779</pbs:EFIDptID>
<pbs:MISTemplateID/> <pbs:DSFShared>N</pbs:DSFShared> <pbs:CombineInCarton/>
<pbs:Height>10</pbs:Height> <pbs:VendorCode/> <pbs:DefPOTolerance>0</pbs:DefPOTolerance>
<pbs:ExtInvCode>Ext Inv Code1</pbs:ExtInvCode> <pbs:WebFormDesc>Web Form
Desc1</pbs:WebFormDesc> <pbs:CountryOfOrigin>Canada</pbs:CountryOfOrigin>
<pbs:LayoutAcross>0</pbs:LayoutAcross> <pbs:LayoutAround>0</pbs:LayoutAround> <pbs:CSRFullName/>
<pbs:SalesRepFullName/> <pbs:ChargeForStorage>0</pbs:ChargeForStorage>
<pbs:Customizable>N</pbs:Customizable> <pbs:MaxUseQty>0</pbs:MaxUseQty>
<pbs:MaxUsePercent>0.0</pbs:MaxUsePercent> <pbs:PurchaseUOM>0</pbs:PurchaseUOM>
<pbs:PurchaseUOMDesc/> <pbs:WebForm>0</pbs:WebForm> <pbs:CubeHeight>0.0</pbs:CubeHeight>
<pbs:CubeLength>0.0</pbs:CubeLength> <pbs:CubeWidth>0.0</pbs:CubeWidth>
<pbs:CubeQuantity>0.0</pbs:CubeQuantity> <pbs:CubeStorageCost>0</pbs:CubeStorageCost>
<pbs:SortField1>sort1</pbs:SortField1> <pbs:SortField2>sort2</pbs:SortField2> <pbs:ReplacementCode>rep
code 1</pbs:ReplacementCode> <pbs:ReviseDate>2019-02-25T00:00:00</pbs:ReviseDate>
<pbs:MinimumDays>33</pbs:MinimumDays> <pbs:ReorderDays>55</pbs:ReorderDays>
<pbs:InternationalType/> <pbs:OneTimeUse/> <pbs:Notes>these are the item notes</pbs:Notes>
<pbs:EmailReplenish/> <pbs:EmailRequired>N</pbs:EmailRequired>
<pbs:PackMaterialUse>N</pbs:PackMaterialUse> <pbs:TurnTime>0</pbs:TurnTime> <pbs:ForestTypeID>FT-2-
Dean</pbs:ForestTypeID> <pbs:Manager>Jim</pbs:Manager> <pbs:Taxable>N</pbs:Taxable> </pbs:Item>
<pbs:Item> <pbs:CustACNO>JJD</pbs:CustACNO> <pbs:InvCode>TESTWS-040815 (4)</pbs:InvCode>
<pbs:RevCode/> <pbs:StockType>C</pbs:StockType> <pbs>Description1>Test DP dll (4)</pbs>Description1>
<pbs>Description2>(12190)</pbs>Description2> <pbs>Description3/> <pbs:TrackSkids>Y</pbs:TrackSkids>
<pbs:MinLevel>0</pbs:MinLevel> <pbs:ReorderQty>0</pbs:ReorderQty>
<pbs:ActivityCode>0</pbs:ActivityCode> <pbs:UnitOfCost>1000</pbs:UnitOfCost>
<pbs:UOMCostDesc>Thousand</pbs:UOMCostDesc> <pbs:AverageCost>0</pbs:AverageCost> <pbs:Location/>
<pbs:Skids>0</pbs:Skids> <pbs:Cartons>0</pbs:Cartons> <pbs:Pcspercarton>0</pbs:Pcspercarton>
<pbs:LoosePieces>0</pbs:LoosePieces> <pbs:LotTracking/> <pbs:VendorLotNo/> <pbs:TrackingType/>
<pbs:CreateDate>2015-03-26T00:00:00</pbs:CreateDate> <pbs:OldInvCode/> <pbs:Group>env</pbs:Group>
<pbs:Family>fam</pbs:Family> <pbs:Category>cat</pbs:Category> <pbs:DeclaredValue>0</pbs:DeclaredValue>
<pbs:ExpirationDate>2099-12-25T00:00:00</pbs:ExpirationDate> <pbs:DefaultPickLocation/>
<pbs:AllowWebView>N</pbs:AllowWebView> <pbs:ProductCode>0</pbs:ProductCode>
<pbs:StandardSellPrice>0</pbs:StandardSellPrice> <pbs:ItemNumber/> <pbs:SkidNotes/> <pbs:Classification/>
<pbs:AllowRevision/> <pbs:UOM>0</pbs:UOM> <pbs:UOMDescription/> <pbs:ChargeMethod/>
<pbs:SellMethod/> <pbs:FixedMarkup/> <pbs:StandardCost>0</pbs:StandardCost> <pbs:WebLevel1/>
<pbs:WebLevel2/> <pbs:WebLevel3/> <pbs:WebLevel4/> <pbs:IsGenericGFC>N</pbs:IsGenericGFC>
<pbs:Numberup>0</pbs:Numberup> <pbs:Weight>6.29</pbs:Weight> <pbs:Width>100</pbs:Width>
<pbs:Length>1</pbs:Length> <pbs:Pages>0</pbs:Pages> <pbs:PagesBack>0</pbs:PagesBack>
<pbs:PlannerItem/> <pbs:PackageType/> <pbs:PickMethod/> <pbs:ExternalItemNo/>
<pbs:IsActive>Y</pbs:IsActive> <pbs:PartialPOD/> <pbs:EFICtgID>1669</pbs:EFICtgID>
<pbs:EFIDptID>1779</pbs:EFIDptID> <pbs:MISTemplateID/> <pbs:DSFShared>N</pbs:DSFShared>
<pbs:CombineInCarton/> <pbs:Height>10</pbs:Height> <pbs:VendorCode/>
<pbs:DefPOTolerance>0</pbs:DefPOTolerance> <pbs:ExtInvCode>Ext Inv Code1</pbs:ExtInvCode>
<pbs:WebFormDesc>Web Form Desc1</pbs:WebFormDesc> <pbs:CountryOfOrigin>United
States</pbs:CountryOfOrigin> <pbs:LayoutAcross>0</pbs:LayoutAcross>
<pbs:LayoutAround>0</pbs:LayoutAround> <pbs:CSRFullName/> <pbs:SalesRepFullName/>
<pbs:ChargeForStorage>0</pbs:ChargeForStorage> <pbs:Customizable>N</pbs:Customizable>
<pbs:MaxUseQty>0</pbs:MaxUseQty> <pbs:MaxUsePercent>0.0</pbs:MaxUsePercent>
<pbs:PurchaseUOM>0</pbs:PurchaseUOM> <pbs:PurchaseUOMDesc/> <pbs:WebForm>0</pbs:WebForm>

```

```

<pbs:CubeHeight>0.0</pbs:CubeHeight> <pbs:CubeLength>0.0</pbs:CubeLength>
<pbs:CubeWidth>0.0</pbs:CubeWidth> <pbs:CubeQuantity>0.0</pbs:CubeQuantity>
<pbs:CubeStorageCost>0</pbs:CubeStorageCost> <pbs:SortField1>sort1</pbs:SortField1>
<pbs:SortField2>sort2</pbs:SortField2> <pbs:ReplacementCode>rep code 1</pbs:ReplacementCode>
<pbs:ReviseDate>2019-02-25T00:00:00</pbs:ReviseDate> <pbs:MinimumDays>33</pbs:MinimumDays>
<pbs:ReorderDays>55</pbs:ReorderDays> <pbs:InternationalType/> <pbs:OneTimeUse/> <pbs:Notes>these are
the item notes</pbs:Notes> <pbs:EmailReplenish/> <pbs:EmailRequired>N</pbs:EmailRequired>
<pbs:PackMaterialUse>N</pbs:PackMaterialUse> <pbs:TurnTime>0</pbs:TurnTime> <pbs:ForestTypeID>FT-2-
Dean</pbs:ForestTypeID><pbs:Manager>Joe</pbs:Manager> <pbs:Taxable>N</pbs:Taxable </pbs:Item>
</pbs:ItemList>< /pbs:InventoryCreate>]]>

```

```
</xmlString></InventoryCreateByBatch></soap:Body></soap:Envelope>
```

This is the Inventory Create xml, separated from the CDATA section

```

<?xml version="1.0" encoding="UTF-8"?>
<pbs:InventoryCreate xmlns="http://Printstream.org/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema" xmlns:ns0="printstream:efi:com">
  <pbs:ItemList>
    <pbs:Item>
      <pbs:CustACNO>JJD</pbs:CustACNO>
      <pbs:InvCode>TESTWS-040815 (3)</pbs:InvCode>
      <pbs:RevCode/>
      <pbs:StockType>C</pbs:StockType>
      <pbs:Description1>Test DP dll for (3)</pbs:Description1>
      <pbs:Description2>(12190)</pbs:Description2>
      <pbs:Description3/>
      <pbs:TrackSkids>Y</pbs:TrackSkids>
      <pbs:MinLevel>0</pbs:MinLevel>
      <pbs:ReorderQty>0</pbs:ReorderQty>
      <pbs:ActivityCode>0</pbs:ActivityCode>
      <pbs:UnitOfCost>1000</pbs:UnitOfCost>
      <pbs:UOMCostDesc>Thousand</pbs:UOMCostDesc>
      <pbs:AverageCost>0</pbs:AverageCost>
      <pbs:Location/>
      <pbs:Skids>0</pbs:Skids>
      <pbs:Cartons>0</pbs:Cartons>
      <pbs:Pcspercarton>0</pbs:Pcspercarton>
      <pbs:LoosePieces>0</pbs:LoosePieces>
      <pbs:LotTracking/>
      <pbs:VendorLotNo/>
      <pbs:TrackingType/>
      <pbs:CreatedDate>2015-03-26T00:00:00</pbs:CreatedDate>
      <pbs:Group>env</pbs:Group>
      <pbs:Family>fam</pbs:Family>
      <pbs:Category>cat</pbs:Category>
      <pbs:DeclaredValue>0</pbs:DeclaredValue>
      <pbs:ExpirationDate>2099-12-25T00:00:00</pbs:ExpirationDate>
      <pbs:DefaultPickLocation/>
      <pbs:AllowWebView>N</pbs:AllowWebView>
      <pbs:ProductCode>0</pbs:ProductCode>
      <pbs:StandardSellPrice>0</pbs:StandardSellPrice>
      <pbs:SkidNotes/>
      <pbs:Classification/>
      <pbs:AllowRevision/>
      <pbs:UOM>0</pbs:UOM>
      <pbs:UOMDescription/>
      <pbs:ChargeMethod/>
      <pbs:SellMethod/>
      <pbs:FixedMarkup/>
    </pbs:Item>
  </pbs:ItemList>
</pbs:InventoryCreate>

```

```

<pbs:StandardCost>0</pbs:StandardCost>
<pbs:WebLevel1/>
<pbs:WebLevel2/>
<pbs:WebLevel3/>
<pbs:WebLevel4/>
<pbs:IsGenericGFC>N</pbs:IsGenericGFC>
<pbs:Numberup>0</pbs:Numberup>
<pbs:Weight>6.29</pbs:Weight>
<pbs:Width>100</pbs:Width>
<pbs:Length>1</pbs:Length>
<pbs:Pages>0</pbs:Pages>
<pbs:PagesBack>0</pbs:PagesBack>
<pbs:PlannerItem/>
<pbs:PackageType/>
<pbs:ExternalItemNo/>
<pbs:IsActive>Y</pbs:IsActive>
<pbs:PartialPOD/>
<pbs:EFICtgID>1669</pbs:EFICtgID>
<pbs:EFIDptID>1779</pbs:EFIDptID>
<pbs:MISTemplateID/>
<pbs:DSFShared>N</pbs:DSFShared>
<pbs:CombineInCarton/>
<pbs:Height>10</pbs:Height>
<pbs:VendorCode/>
<pbs:DefPOTolerance>0</pbs:DefPOTolerance>
<pbs:ExtInvCode>Ext Inv Code1</pbs:ExtInvCode>
<pbs:WebFormDesc>Web Form Desc1</pbs:WebFormDesc>
<pbs:CountryOfOrigin>Canada</pbs:CountryOfOrigin>
<LayoutAcross>0</LayoutAcross>
<LayoutAround>0</LayoutAround>
<CSRFullName/>
<SalesRepFullName/>
<ChargeForStorage>N</ChargeForStorage>
<Customizable>N</Customizable>
<MaxUseQty>0</MaxUseQty>
<MaxUsePercent>0.0</MaxUsePercent>
<PurchaseUOM>0</PurchaseUOM>
<PurchaseUOMDesc/>
<WebForm>N</WebForm>
<CubeHeight>0.0</CubeHeight>
<CubeLength>0.0</CubeLength>
<CubeWidth>0.0</CubeWidth>
<CubeQuantity>0.0</CubeQuantity>
<CubeStorageCost>0</CubeStorageCost>
<SortField1>sort1</SortField1>
<SortField2>sort2</SortField2>
<ReplacementCode>rep code 1</ReplacementCode>
<ReviseDate>2019-02-25T00:00:00</ReviseDate>
<MinimumDays>33</MinimumDays>
<ReorderDays>55</ReorderDays>
<InternationalType/>
<OneTimeUse/>
<Notes>these are the item notes</Notes>
<EmailReplenish/>
<EmailRequired>N</EmailRequired>
<PackMaterialUse>N</PackMaterialUse>
<TurnTime>0</TurnTime>
<ForestTypeID>FT-2-Dean</ForestTypeID>
<pbs:Manager>Jim</pbs:Manager>
<pbs:Taxable>N</pbs:Taxable>

```

```

</pbs:Item>
<pbs:Item>
  <pbs:CustACNO>JJD</pbs:CustACNO>
  <pbs:InvCode>TESTWS-040815 (4)</pbs:InvCode>
  <pbs:RevCode/>
  <pbs:StockType>C</pbs:StockType>
  <pbs:Description1>Test DP dll (4)</pbs:Description1>
  <pbs:Description2>(12190)</pbs:Description2>
  <pbs:Description3/>
  <pbs:TrackSkids>Y</pbs:TrackSkids>
  <pbs:MinLevel>0</pbs:MinLevel>
  <pbs:ReorderQty>0</pbs:ReorderQty>
  <pbs:ActivityCode>0</pbs:ActivityCode>
  <pbs:UnitOfCost>1000</pbs:UnitOfCost>
  <pbs:UOMCostDesc>Thousand</pbs:UOMCostDesc>
  <pbs:AverageCost>0</pbs:AverageCost>
  <pbs:Location/>
  <pbs:Skids>0</pbs:Skids>
  <pbs:Cartons>0</pbs:Cartons>
  <pbs:Pcspercarton>0</pbs:Pcspercarton>
  <pbs:LoosePieces>0</pbs:LoosePieces>
  <pbs:LotTracking/>
  <pbs:VendorLotNo/>
  <pbs:TrackingType/>
  <pbs:CreatedDate>2015-03-26T00:00:00</pbs:CreatedDate>
  <pbs:Group>env</pbs:Group>
  <pbs:Family>fam</pbs:Family>
  <pbs:Category>cat</pbs:Category>
  <pbs:DeclaredValue>0</pbs:DeclaredValue>
  <pbs:ExpirationDate>2099-12-25T00:00:00</pbs:ExpirationDate>
  <pbs:DefaultPickLocation/>
  <pbs:AllowWebView>N</pbs:AllowWebView>
  <pbs:ProductCode>0</pbs:ProductCode>
  <pbs:StandardSellPrice>0</pbs:StandardSellPrice>
  <pbs:SkidNotes/>
  <pbs:Classification/>
  <pbs:AllowRevision/>
  <pbs:UOM>0</pbs:UOM>
  <pbs:UOMDescription/>
  <pbs:ChargeMethod/>
  <pbs:SellMethod/>
  <pbs:FixedMarkup/>
  <pbs:StandardCost>0</pbs:StandardCost>
  <pbs:WebLevel1/>
  <pbs:WebLevel2/>
  <pbs:WebLevel3/>
  <pbs:WebLevel4/>
  <pbs:IsGenericGFC>N</pbs:IsGenericGFC>
  <pbs:Numberup>0</pbs:Numberup>
  <pbs:Weight>6.29</pbs:Weight>
  <pbs:Width>100</pbs:Width>
  <pbs:Length>1</pbs:Length>
  <pbs:Pages>0</pbs:Pages>
  <pbs:PagesBack>0</pbs:PagesBack>
  <pbs:PlannerItem/>
  <pbs:PackageType/>
  <pbs:ExternalItemNo/>
  <pbs:IsActive>Y</pbs:IsActive>
  <pbs:PartialPOD/>
  <pbs:EFICtgID>1669</pbs:EFICtgID>
  <pbs:EFIDptID>1779</pbs:EFIDptID>

```

```

    <pbs:MISTemplateID/>
    <pbs:DSFShared>N</pbs:DSFShared>
    <pbs:CombineInCarton/>
    <pbs:Height>10</pbs:Height>
    <pbs:VendorCode/>
    <pbs:DefPOTolerance>0</pbs:DefPOTolerance>
    <pbs:ExtInvCode>Ext Inv Code1</pbs:ExtInvCode>
    <pbs:WebFormDesc>Web Form Desc1</pbs:WebFormDesc>
    <pbs:CountryOfOrigin>United States</pbs:CountryOfOrigin>
    <pbs:LayoutAcross>0</pbs:LayoutAcross>
    <pbs:LayoutAround>0</pbs:LayoutAround>
    <pbs:CSRFullName/>
    <pbs:SalesRepFullName/>
    <pbs:ChargeForStorage>N</pbs:ChargeForStorage>
    <pbs:Customizable>N</pbs:Customizable>
    <pbs:MaxUseQty>0</pbs:MaxUseQty>
    <pbs:MaxUsePercent>0.0</pbs:MaxUsePercent>
    <pbs:PurchaseUOM>0</pbs:PurchaseUOM>
    <pbs:PurchaseUOMDesc/>
    <pbs:WebForm>N</pbs:WebForm>
    <pbs:CubeHeight>0.0</pbs:CubeHeight>
    <pbs:CubeLength>0.0</pbs:CubeLength>
    <pbs:CubeWidth>0.0</pbs:CubeWidth>
    <pbs:CubeQuantity>0.0</pbs:CubeQuantity>
    <pbs:CubeStorageCost>0</pbs:CubeStorageCost>
    <pbs:SortField1>sort1</pbs:SortField1>
    <pbs:SortField2>sort2</pbs:SortField2>
    <pbs:ReplacementCode>rep code 1</pbs:ReplacementCode>
    <pbs:ReviseDate>2019-02-25T00:00:00</pbs:ReviseDate>
    <pbs:MinimumDays>33</pbs:MinimumDays>
    <pbs:ReorderDays>55</pbs:ReorderDays>
    <pbs:InternationalType/>
    <pbs:OneTimeUse/>
    <pbs:Notes>these are the item notes</pbs:Notes>
    <pbs:EmailReplenish/>
    <pbs:EmailRequired>N</pbs:EmailRequired>
    <pbs:PackMaterialUse>N</pbs:PackMaterialUse>
    <pbs:TurnTime>0</pbs:TurnTime>
    <pbs:ForestTypeID>FT-2-Dean</pbs:ForestTypeID>
    <pbs:Manager>Larry</pbs:Manager>
    <pbs:Taxable>Y</pbs:Taxable>
  </pbs:Item>
</pbs:ItemList>
</pbs:InventoryCreate>

```

Response Transaction

This is a successful transaction response (two items).

```
<?xml version="1.0" encoding="utf-8" ?>
<InventoryCreateResponse>
  <ItemList>
    <UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <STKItemId>0</STKItemId>
      <ItemNumber>79131</ItemNumber>
      <Message>Success</Message>
      <InvCode>TESTWS-040815 (3)</InvCode>
    </UpdatedPSItem>
    <UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <STKItemId>0</STKItemId>
      <ItemNumber>79132</ItemNumber>
      <Message>Success</Message>
      <InvCode>TESTWS-040815 (4)</InvCode>
    </UpdatedPSItem>
  </ItemList>
</InventoryCreateResponse>
```

This is an unsuccessful transaction response (two items). Note - ItemNumber is zero for an error condition. The Inventory Code is included for validation purposes.

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<InventoryCreateResponse>
```

```
<ItemList>
```

```
<UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<STKItemId>0</STKItemId>
```

```
<ItemNumber>0</ItemNumber>
```

```
<Message>Error: Item # 79131 already exists., #:1, in StkitemMaintenance -
CheckDuplicateItemNo</Message>
```

```
<InvCode>TESTWS-040815 (3)</InvCode>
```

```
</UpdatedPSItem>
```

```
<UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```
<STKItemId>0</STKItemId>
```

```
<ItemNumber>0</ItemNumber>
```

```
<Message>Error: Item # 79132 already exists., #:1, in StkitemMaintenance -
CheckDuplicateItemNo</Message>
```

```
<InvCode>TESTWS-040815 (4)</InvCode>
```

```
</UpdatedPSItem>
```

```
</ItemList>
```

```
</InventoryCreateResponse>
```

Inventory Update

Narrative

This function updates an existing customer item.

Message will be "Success" when transaction is good, otherwise an appropriate error will display.

12/3/15 - "Location" removed from Update XML

Request Transaction

```
POST /Web2PSWCF.WebService.Setup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/InventoryUpdate"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <InventoryUpdate xmlns="http://Printstream.org/">
      <ItemNumber>string</ItemNumber>
      <InvCode>string</InvCode>
      <RevCode>string</RevCode>
      <Description1>string</Description1>
      <Description2>string</Description2>
      <Description3>string</Description3>
      <IsActive>string</IsActive>
      <MinLevel>double</MinLevel>
      <Group>string</Group>
      <Family>string</Family>
      <Category>string</Category>
      <DeclaredValue>double</DeclaredValue>
      <IsGenericGFC>string</IsGenericGFC>
      <Width>string</Width>
      <Length>string</Length>
      <Pages>string</Pages>
      <PagesBack>string</PagesBack>
      <PlannerItem>string</PlannerItem>
      <PackageType>string</PackageType>
      <EFICtgID>string</EFICtgID>
      <EFIDptID>string</EFIDptID>
      <StockType>string</StockType>
      <ReorderQty>string</ReorderQty>
      <ActivityCode>string</ActivityCode>
      <TrackingType>string</TrackingType>
      <LotTracking>string</LotTracking>
      <ExpirationDate>string</ExpirationDate>
    </InventoryUpdate>
  </soap:Body>
</soap:Envelope>
```



```

<DefaultPickLocation>string</DefaultPickLocation>
<AllowWebView>string</AllowWebView>
<ProductCode>string</ProductCode>
<StandardSellPrice>double</StandardSellPrice>
<StandardCost>double</StandardCost>
<Classification>string</Classification>
<UOM>string</UOM>
<UOMDescription>string</UOMDescription>
<UnitOfCost>string</UnitOfCost>
<UOMCostDesc>string</UOMCostDesc>
<SellMethod>string</SellMethod>
<FixedMarkup>string</FixedMarkup>
<WebLevel1>string</WebLevel1>
<WebLevel2>string</WebLevel2>
<WebLevel3>string</WebLevel3>
<WebLevel4>string</WebLevel4>
<NumberUp>string</NumberUp>
<PartialPOD>string</PartialPOD>
<Weight>double</Weight>
<ExternalItemNo>string</ExternalItemNo>
<MISTemplateID>string</MISTemplateID>
<DSFShared>string</DSFShared>
<CombineInCarton>string</CombineInCarton>
<AverageCost>double</AverageCost>
<AllowRevision>string</AllowRevision>
<Height>string</Height>
<VendorCode>string</VendorCode>
<DefPOTolerance>string</DefPOTolerance>
<ExtInvCode>string</ExtInvCode>
<WebFormDesc>string</WebFormDesc>
<CountryOfOrigin>string</CountryOfOrigin>
<LayoutAcross>int</LayoutAcross>
<LayoutAround>int</LayoutAround>
<CSRFullName>string</CSRFullName>
<SalesRepFullName>string</SalesRepFullName>
<ChargeForStorage>string</ChargeForStorage>
<Customizable>string</Customizable>
<MaxUseQty>int</MaxUseQty>
<MaxUsePercent>double</MaxUsePercent>
<PurchaseUOM>double</PurchaseUOM>
<PurchaseUOMDesc>string</PurchaseUOMDesc>
<WebForm>string</WebForm>
<CubeHeight>double</CubeHeight>
<CubeLength>double</CubeLength>
<CubeWidth>double</CubeWidth>
<CubeQuantity>double</CubeQuantity>
<CubeStorageCost>double</CubeStorageCost>
<SortField1>string</SortField1>
<SortField2>string</SortField2>
<ReplacementCode>string</ReplacementCode>
<ReviseDate>dateTime</ReviseDate>
<MinimumDays>int</MinimumDays>
<ReorderDays>int</ReorderDays>
<InternationalType>string</InternationalType>
<OneTimeUse>string</OneTimeUse>
<Notes>string</Notes>
<EmailReplenish>string</EmailReplenish>
<EmailRequired>string</EmailRequired>
<PackMaterialUse>string</PackMaterialUse>
<TurnTime>double</TurnTime>
<ForestTypeID>string</ForestTypeID>

```

```

    <Manager>string</Manager>
    <Taxable>string</Taxable>
  </InventoryUpdate>
</soap:Body>
</soap:Envelope>

```

Sample Request Transaction

```

<?xml version="1.0" encoding="utf-8" ?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Header>

    <Authenticate xmlns="http://Printstream.org/">

      <RecipientID>JJWEB</RecipientID>

      <Password>ok</Password>

      <CompanyCode>01</CompanyCode>

      <PlantCode>0100</PlantCode>

      <AccountNo>JJD</AccountNo>

    </Authenticate>

  </soap:Header>

  <soap:Body>

    <InventoryUpdate xmlns="http://Printstream.org/">
      <ItemNumber>78935</ItemNumber>
      <InvCode> TESTWS-041912-32</InvCode>
      <RevCode />
      <Description1> Test WS on Wed</Description1>
      <Description2>CREATED FROM WEB2PS</Description2>
      <Description3> Description3>
      <IsActive>Y</IsActive>
      <MinLevel>100</MinLevel>
      <Group />
      <Family />
      <Category />
      <DeclaredValue>0</DeclaredValue>
      <IsGenericGFC>N</IsGenericGFC>
      <Width>25</Width>
      <Length>40</Length>
      <Pages>250</Pages>
      <PagesBack>15</PagesBack>
      <PlannerItem />
      <PackageType>P</PackageType>
      <EFICtgID>555</EFICtgID>
      <EFIDptID>444</EFIDptID>
      <StockType></StockType>
    </InventoryUpdate>
  </soap:Body>
</soap:Envelope>

```



```

<ReorderQty>500</ReorderQty>
<ActivityCode></ActivityCode>
<TrackingType></TrackingType>
<LotTracking></LotTracking>
<ExpirationDate></ExpirationDate>
<DefaultPickLocation></DefaultPickLocation>
<AllowWebView>Y</AllowWebView>
<ProductCode></ProductCode>
<StandardSellPrice>0</StandardSellPrice>
<StandardCost>0</StandardCost>
<Classification></Classification>
<UOM></UOM>
<UOMDescription></UOMDescription>
<UnitOfCost></UnitOfCost>
<UOMCostDesc></UOMCostDesc>
<SellMethod></SellMethod>
<FixedMarkup></FixedMarkup>
<WebLevel1></WebLevel1>
<WebLevel2></WebLevel2>
<WebLevel3></WebLevel3>
<WebLevel4></WebLevel4>
<NumberUp></NumberUp>
<PartialPOD></PartialPOD>
<Weight>0</Weight>
<ExternalItemNo></ExternalItemNo>
<MISTemplateID></MISTemplateID>
<DSFShared>N</DSFShared>
<CombineInCarton></CombineInCarton>
<AverageCost>0</AverageCost>
<AllowRevision></AllowRevision>
<Height>10</Height>
<VendorCode></VendorCode>
<DefPOTolerance>0</ DefPOTolerance >
<ExtInvCode></ExtInvCode>
<WebFormDesc></WebFormDesc>
<CountryOfOrigin>United States</CountryOfOrigin>
<LayoutAcross>0</LayoutAcross>
<LayoutAround>0</LayoutAround>
<CSRFullName></CSRFullName>
<SalesRepFullName></SalesRepFullName>
<ChargeForStorage>N</ChargeForStorage>
<Customizable>N</Customizable>
<MaxUseQty>0</MaxUseQty>
<MaxUsePercent>0.0</MaxUsePercent>
<PurchaseUOM>0</PurchaseUOM>
<PurchaseUOMDesc></PurchaseUOMDesc>
<WebForm>N</WebForm>
<CubeHeight>0.0</CubeHeight>
<CubeLength>0.0</CubeLength>
<CubeWidth>0.0</CubeWidth>
<CubeQuantity>0.0</CubeQuantity>
<CubeStorageCost>0</CubeStorageCost>
<SortField1>sort1</SortField1>
<SortField2>sort2</SortField2>
<ReplacementCode>rep code 1</ReplacementCode>
<ReviseDate>2019-02-25T00:00:00</ReviseDate>
<MinimumDays>33</MinimumDays>

```

```

    <ReorderDays>55</ReorderDays>
    <InternationalType></InternationalType>
    <OneTimeUse></OneTimeUse>
    <Notes>these are the item notes</Notes>
    <EmailReplenish></EmailReplenish>
    <EmailRequired>N</EmailRequired>
    <PackMaterialUse>N</PackMaterialUse>
    <TurnTime>0</TurnTime>
    <ForestTypeID>FT-1-Dean</ForestTypeID>
    <Manager>Larry</Manager>
    <Taxable>N</Taxable>
  </InventoryUpdate>
</soap:Body>

</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <STKItemId>0</STKItemId>
  <ItemNumber>0</ItemNumber>
  <Message>Success</Message>
</UpdatedPSItem>

```

Inventory Update by Batch

Narrative

This function will update inventory items for a customer, from a batch xml.

Message will be "Success" when each transaction is good as well along with the new ItemNumber, otherwise an appropriate error will display (ItemNumber will be zero in an error condition).

Transactions will be validated against the InventoryUpdate schema.

12/3/15 - "Location" removed from Update XM and Schema

Request Transaction

```
POST /Web2PSWCF.WebService.Setup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/InventoryUpdateByBatch"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <InventoryUpdateByBatch xmlns="http://Printstream.org/">
      <xmlString>string</xmlString>
    </InventoryUpdateByBatch>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="UTF-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"><soap:Header><Authenticate
xmlns="http://Printstream.org/"><RecipientID>JJWEB</RecipientID><Password>ok</Password><CompanyCo
de>01</CompanyCode><PlantCode>0100</PlantCode><AccountNo>JJJ</AccountNo></Authenticate></soap:
Header><soap:Body><InventoryUpdateByBatch xmlns="http://Printstream.org/"><xmlString>

<![CDATA[<pbs:InventoryUpdate xmlns="http://Printstream.org/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema"
xmlns:ns0="printstream:efi:com"> <pbs:ItemList> <pbs:Item> <pbs:ItemNumber>791310</pbs:ItemNumber>
<pbs:InvCode>TESTWS-040815 (3)</pbs:InvCode><pbs:RevCode></pbs:RevCode>
<pbs:StockType>C</pbs:StockType> <pbs:Description1>Test DP dll for (3)</pbs:Description1>
<pbs:Description2>(12190-UPDATE-3)</pbs:Description2> <pbs:Description3/>
<pbs:MinLevel>0</pbs:MinLevel> <pbs:ReorderQty>0</pbs:ReorderQty>
```

```

<pbs:ActivityCode>90000</pbs:ActivityCode> <pbs:UnitOfCost>1000</pbs:UnitOfCost>
<pbs:UOMCostDesc>Thousand</pbs:UOMCostDesc> <pbs:AverageCost>0</pbs:AverageCost>
<pbs:LotTracking/> <pbs:TrackingType/> <pbs:Group>env</pbs:Group> <pbs:Family>fam</pbs:Family>
<pbs:Category>cat</pbs:Category> <pbs:DeclaredValue>0</pbs:DeclaredValue> <pbs:ExpirationDate>2099-
12-25T00:00:00</pbs:ExpirationDate> <pbs:DefaultPickLocation/>
<pbs:AllowWebView>N</pbs:AllowWebView> <pbs:ProductCode>0</pbs:ProductCode>
<pbs:StandardSellPrice>0</pbs:StandardSellPrice> <pbs:Classification/> <pbs:AllowRevision/>
<pbs:UOM>0</pbs:UOM> <pbs:UOMDescription/> <pbs:SellMethod/> <pbs:FixedMarkup/>
<pbs:StandardCost>0</pbs:StandardCost> <pbs:WebLevel1/> <pbs:WebLevel2/> <pbs:WebLevel3/>
<pbs:WebLevel4/> <pbs:IsGenericGFC>N</pbs:IsGenericGFC> <pbs:Numberup>0</pbs:Numberup>
<pbs:Weight>6.29</pbs:Weight> <pbs:Width>100</pbs:Width> <pbs:Length>1</pbs:Length>
<pbs:Pages>0</pbs:Pages> <pbs:PagesBack>0</pbs:PagesBack> <pbs:PlannerItem/> <pbs:PackageType/>
<pbs:ExternalItemNo/> <pbs:IsActive>Y</pbs:IsActive> <pbs:PartialPOD/>
<pbs:EFICtgID>1669</pbs:EFICtgID> <pbs:EFIDptID>1779</pbs:EFIDptID> <pbs:MISTemplateID/>
<pbs:DSFShared>N</pbs:DSFShared> <pbs:CombineInCarton/> <pbs:Height>10</pbs:Height>
<pbs:VendorCode/> <pbs:DefPOTolerance>0</pbs:DefPOTolerance> <pbs:ExtInvCode>Ext Inv
Code1</pbs:ExtInvCode> <pbs:WebFormDesc>Web Form Desc1</pbs:WebFormDesc>
<pbs:CountryOfOrigin>Mexico</pbs:CountryOfOrigin> <pbs:LayoutAcross>0</pbs:LayoutAcross>
<pbs:LayoutAround>0</pbs:LayoutAround> <pbs:CSRFFullName/> <pbs:SalesRepFullName/>
<pbs:ChargeForStorage>0</pbs:ChargeForStorage> <pbs:Customizable>N</pbs:Customizable>
<pbs:MaxUseQty>0</pbs:MaxUseQty> <pbs:MaxUsePercent>0.0</pbs:MaxUsePercent>
<pbs:PurchaseUOM>0</pbs:PurchaseUOM> <pbs:PurchaseUOMDesc/> <pbs:WebForm>0</pbs:WebForm>
<pbs:CubeHeight>0.0</pbs:CubeHeight> <pbs:CubeLength>0.0</pbs:CubeLength>
<pbs:CubeWidth>0.0</pbs:CubeWidth> <pbs:CubeQuantity>0.0</pbs:CubeQuantity>
<pbs:CubeStorageCost>0</pbs:CubeStorageCost> <pbs:SortField1>sort1</pbs:SortField1>
<pbs:SortField2>sort2</pbs:SortField2> <pbs:ReplacementCode>rep code 1</pbs:ReplacementCode>
<pbs:ReviseDate>2019-02-25T00:00:00</pbs:ReviseDate> <pbs:MinimumDays>33</pbs:MinimumDays>
<pbs:ReorderDays>55</pbs:ReorderDays> <pbs:InternationalType/> <pbs:OneTimeUse/> <pbs:Notes>these
are the item notes</pbs:Notes> <pbs:EmailReplenish/> <pbs:EmailRequired>N</pbs:EmailRequired>
<pbs:PackMaterialUse>N</pbs:PackMaterialUse> <pbs:TurnTime>0</pbs:TurnTime> <pbs:ForestTypeID>FT-2-
Dean</pbs:ForestTypeID> <pbs:Manager>Larry</pbs:Manager> <pbs:Taxable>Y</pbs:Taxable> </pbs:Item>
<pbs:Item> <pbs:ItemNumber>79132</pbs:ItemNumber> <pbs:InvCode>TESTWS-040815 (4)</pbs:InvCode>
<pbs:RevCode></pbs:RevCode> <pbs:StockType>C</pbs:StockType> <pbs:Description1>Test DP dll
(4)</pbs:Description1> <pbs:Description2>(12190-16 UPDATE-3)</pbs:Description2> <pbs:Description3/>
<pbs:MinLevel>0</pbs:MinLevel> <pbs:ReorderQty>0</pbs:ReorderQty>
<pbs:ActivityCode>90000</pbs:ActivityCode> <pbs:UnitOfCost>1000</pbs:UnitOfCost>
<pbs:UOMCostDesc>Thousand</pbs:UOMCostDesc> <pbs:AverageCost>0</pbs:AverageCost>
<pbs:LotTracking/> <pbs:TrackingType/> <pbs:Group>env</pbs:Group> <pbs:Family>fam</pbs:Family>
<pbs:Category>cat</pbs:Category> <pbs:DeclaredValue>0</pbs:DeclaredValue> <pbs:ExpirationDate>2099-
12-25T00:00:00</pbs:ExpirationDate> <pbs:DefaultPickLocation/>
<pbs:AllowWebView>N</pbs:AllowWebView> <pbs:ProductCode>0</pbs:ProductCode>
<pbs:StandardSellPrice>0</pbs:StandardSellPrice> <pbs:Classification/> <pbs:AllowRevision/>
<pbs:UOM>0</pbs:UOM> <pbs:UOMDescription/> <pbs:SellMethod/> <pbs:FixedMarkup/>
<pbs:StandardCost>0</pbs:StandardCost> <pbs:WebLevel1/> <pbs:WebLevel2/> <pbs:WebLevel3/>
<pbs:WebLevel4/> <pbs:IsGenericGFC>N</pbs:IsGenericGFC> <pbs:Numberup>0</pbs:Numberup>
<pbs:Weight>6.29</pbs:Weight> <pbs:Width>100</pbs:Width> <pbs:Length>1</pbs:Length>
<pbs:Pages>0</pbs:Pages> <pbs:PagesBack>0</pbs:PagesBack> <pbs:PlannerItem/> <pbs:PackageType/>
<pbs:ExternalItemNo/> <pbs:IsActive>Y</pbs:IsActive> <pbs:PartialPOD/>
<pbs:EFICtgID>1669</pbs:EFICtgID> <pbs:EFIDptID>1779</pbs:EFIDptID> <pbs:MISTemplateID/>
<pbs:DSFShared>N</pbs:DSFShared> <pbs:CombineInCarton/> <pbs:Height>10</pbs:Height>
<pbs:VendorCode/> <pbs:DefPOTolerance>0</pbs:DefPOTolerance> <pbs:ExtInvCode>Ext Inv
Code1</pbs:ExtInvCode> <pbs:WebFormDesc>Web Form Desc1</pbs:WebFormDesc>
<pbs:CountryOfOrigin>United States</pbs:CountryOfOrigin> <pbs:LayoutAcross>0</pbs:LayoutAcross>
<pbs:LayoutAround>0</pbs:LayoutAround> <pbs:CSRFFullName/> <pbs:SalesRepFullName/>
<pbs:ChargeForStorage>0</pbs:ChargeForStorage> <pbs:Customizable>N</pbs:Customizable>
<pbs:MaxUseQty>0</pbs:MaxUseQty> <pbs:MaxUsePercent>0.0</pbs:MaxUsePercent>
<pbs:PurchaseUOM>0</pbs:PurchaseUOM> <pbs:PurchaseUOMDesc/> <pbs:WebForm>0</pbs:WebForm>
<pbs:CubeHeight>0.0</pbs:CubeHeight> <pbs:CubeLength>0.0</pbs:CubeLength>
<pbs:CubeWidth>0.0</pbs:CubeWidth> <pbs:CubeQuantity>0.0</pbs:CubeQuantity>
<pbs:CubeStorageCost>0</pbs:CubeStorageCost> <pbs:SortField1>sort1</pbs:SortField1>
<pbs:SortField2>sort2</pbs:SortField2> <pbs:ReplacementCode>rep code 1</pbs:ReplacementCode>

```

```

<pbs:ReviseDate>2019-02-25T00:00:00</pbs:ReviseDate> <pbs:MinimumDays>33</pbs:MinimumDays>
<pbs:ReorderDays>55</pbs:ReorderDays> <pbs:InternationalType/> <pbs:OneTimeUse/> <pbs:Notes>these
are the item notes</pbs:Notes> <pbs:EmailReplenish/> <pbs:EmailRequired>N</pbs:EmailRequired>
<pbs:PackMaterialUse>N</pbs:PackMaterialUse> <pbs:TurnTime>0</pbs:TurnTime> <pbs:ForestTypeID>FT-2-
Dean</pbs:ForestTypeID> <pbs:Manager>Lucy</pbs:Manager> <pbs:Taxable>N</pbs:Taxable> </pbs:Item>
</pbs:ItemList> </pbs:InventoryUpdate>]]>

```

```
</xmlString></InventoryUpdateByBatch></soap:Body></soap:Envelope>
```

This is the Inventory Update xml, separated from the CDATA section

```

<?xml version="1.0" encoding="UTF-8"?>
<pbs:InventoryUpdate xmlns="http://Printstream.org/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema" xmlns:ns0="printstream:efi.com">
  <pbs:ItemList>
    <pbs:Item>
      <pbs:ItemNumber>791310</pbs:ItemNumber>
      <pbs:InvCode>TESTWS-040815 (3)</pbs:InvCode>
      <pbs:RevCode></pbs:RevCode>
      <pbs:StockType>C</pbs:StockType>
      <pbs:Description1>Test DP dll for (3)</pbs:Description1>
      <pbs:Description2>(12190-UPDATE-3)</pbs:Description2>
      <pbs:Description3/>
      <pbs:MinLevel>0</pbs:MinLevel>
      <pbs:ReorderQty>0</pbs:ReorderQty>
      <pbs:ActivityCode>90000</pbs:ActivityCode>
      <pbs:UnitOfCost>1000</pbs:UnitOfCost>
      <pbs:UOMCostDesc>Thousand</pbs:UOMCostDesc>
      <pbs:AverageCost>0</pbs:AverageCost>
      <pbs:LotTracking/>
      <pbs:TrackingType/>
      <pbs:Group>env</pbs:Group>
      <pbs:Family>fam</pbs:Family>
      <pbs:Category>cat</pbs:Category>
      <pbs:DeclaredValue>0</pbs:DeclaredValue>
      <pbs:ExpirationDate>2099-12-25T00:00:00</pbs:ExpirationDate>
      <pbs:DefaultPickLocation/>
      <pbs:AllowWebView>N</pbs:AllowWebView>
      <pbs:ProductCode>0</pbs:ProductCode>
      <pbs:StandardSellPrice>0</pbs:StandardSellPrice>
      <pbs:Classification/>
      <pbs:AllowRevision/>
      <pbs:UOM>0</pbs:UOM>
      <pbs:UOMDescription/>
      <pbs:SellMethod/>
      <pbs:FixedMarkup/>
      <pbs:StandardCost>0</pbs:StandardCost>
      <pbs:WebLevel1/>
      <pbs:WebLevel2/>
      <pbs:WebLevel3/>
      <pbs:WebLevel4/>
      <pbs:IsGenericGFC>N</pbs:IsGenericGFC>
      <pbs:Numberup>0</pbs:Numberup>
      <pbs:Weight>6.29</pbs:Weight>
      <pbs:Width>100</pbs:Width>
      <pbs:Length>1</pbs:Length>
      <pbs:Pages>0</pbs:Pages>
      <pbs:PagesBack>0</pbs:PagesBack>
    </pbs:Item>
  </pbs:ItemList>
</pbs:InventoryUpdate>

```



```

<pbs:PlannerItem/>
<pbs:PackageType/>
<pbs:ExternalItemNo/>
<pbs:IsActive>Y</pbs:IsActive>
<pbs:PartialPOD/>
<pbs:EFICtgID>1669</pbs:EFICtgID>
<pbs:EFIDptID>1779</pbs:EFIDptID>
<pbs:MISTemplateID/>
<pbs:DSFShared>N</pbs:DSFShared>
<pbs:CombineInCarton/>
<pbs:Height>10</pbs:Height>
<pbs:VendorCode/>
<pbs:DefPOTolerance>0</pbs:DefPOTolerance>
<pbs:ExtInvCode>Ext Inv Code1</pbs:ExtInvCode>
<pbs:WebFormDesc>Web Form Desc1</pbs:WebFormDesc>
<pbs:CountryOfOrigin>Mexico</pbs:CountryOfOrigin>
<pbs:LayoutAcross>0</pbs:LayoutAcross>
<pbs:LayoutAround>0</pbs:LayoutAround>
<pbs:CSRFullName/>
<pbs:SalesRepFullName/>
<pbs:ChargeForStorage>0</pbs:ChargeForStorage>
<pbs:Customizable>N</pbs:Customizable>
<pbs:MaxUseQty>0</pbs:MaxUseQty>
<pbs:MaxUsePercent>0.0</pbs:MaxUsePercent>
<pbs:PurchaseUOM>0</pbs:PurchaseUOM>
<pbs:PurchaseUOMDesc/>
<pbs:WebForm>0</pbs:WebForm>
<pbs:CubeHeight>0.0</pbs:CubeHeight>
<pbs:CubeLength>0.0</pbs:CubeLength>
<pbs:CubeWidth>0.0</pbs:CubeWidth>
<pbs:CubeQuantity>0.0</pbs:CubeQuantity>
<pbs:CubeStorageCost>0</pbs:CubeStorageCost>
<pbs:SortField1>sort1</pbs:SortField1>
<pbs:SortField2>sort2</pbs:SortField2>
<pbs:ReplacementCode>rep code 1</pbs:ReplacementCode>
<pbs:ReviseDate>2019-02-25T00:00:00</pbs:ReviseDate>
<pbs:MinimumDays>33</pbs:MinimumDays>
<pbs:ReorderDays>55</pbs:ReorderDays>
<pbs:InternationalType/>
<pbs:OneTimeUse/>
<pbs:Notes>these are the item notes</pbs:Notes>
<pbs:EmailReplenish/>
<pbs:EmailRequired>N</pbs:EmailRequired>
<pbs:PackMaterialUse>N</pbs:PackMaterialUse>
<pbs:TurnTime>0</pbs:TurnTime>
<pbs:ForestTypeID>FT-2-Dean</pbs:ForestTypeID>
<pbs:Manager>Jim</pbs:Manager>
<pbs:Taxable>Y</pbs:Taxable>
</pbs:Item>
<pbs:Item>
  <pbs:ItemNumber>79132</pbs:ItemNumber>
  <pbs:InvCode>TESTWS-040815 (4)</pbs:InvCode>
  <pbs:RevCode></pbs:RevCode>
  <pbs:StockType>C</pbs:StockType>
  <pbs:Description1>Test DP dll (4)</pbs:Description1>
  <pbs:Description2>(12190-16 UPDATE-3)</pbs:Description2>
  <pbs:Description3/>
  <pbs:MinLevel>0</pbs:MinLevel>
  <pbs:ReorderQty>0</pbs:ReorderQty>
  <pbs:ActivityCode>90000</pbs:ActivityCode>
  <pbs:UnitOfCost>1000</pbs:UnitOfCost>

```



```

<pbs:UOMCostDesc>Thousand</pbs:UOMCostDesc>
<pbs:AverageCost>0</pbs:AverageCost>
<pbs:LotTracking/>
<pbs:TrackingType/>
<pbs:Group>env</pbs:Group>
<pbs:Family>fam</pbs:Family>
<pbs:Category>cat</pbs:Category>
<pbs:DeclaredValue>0</pbs:DeclaredValue>
<pbs:ExpirationDate>2099-12-25T00:00:00</pbs:ExpirationDate>
<pbs:DefaultPickLocation/>
<pbs:AllowWebView>N</pbs:AllowWebView>
<pbs:ProductCode>0</pbs:ProductCode>
<pbs:StandardSellPrice>0</pbs:StandardSellPrice>
<pbs:Classification/>
<pbs:AllowRevision/>
<pbs:UOM>0</pbs:UOM>
<pbs:UOMDescription/>
<pbs:SellMethod/>
<pbs:FixedMarkup/>
<pbs:StandardCost>0</pbs:StandardCost>
<pbs:WebLevel1/>
<pbs:WebLevel2/>
<pbs:WebLevel3/>
<pbs:WebLevel4/>
<pbs:IsGenericGFC>N</pbs:IsGenericGFC>
<pbs:Numberup>0</pbs:Numberup>
<pbs:Weight>6.29</pbs:Weight>
<pbs:Width>100</pbs:Width>
<pbs:Length>1</pbs:Length>
<pbs:Pages>0</pbs:Pages>
<pbs:PagesBack>0</pbs:PagesBack>
<pbs:PlannerItem/>
<pbs:PackageType/>
<pbs:ExternallItemNo/>
<pbs:IsActive>Y</pbs:IsActive>
<pbs:PartialPOD/>
<pbs:EFICtgID>1669</pbs:EFICtgID>
<pbs:EFIDptID>1779</pbs:EFIDptID>
<pbs:MISTemplateID/>
<pbs:DSFShared>N</pbs:DSFShared>
<pbs:CombineInCarton/>
<pbs:Height>10</pbs:Height>
<pbs:VendorCode/>
<pbs:DefPOTolerance>0</pbs:DefPOTolerance>
<pbs:ExtInvCode>Ext Inv Code1</pbs:ExtInvCode>
<pbs:WebFormDesc>Web Form Desc1</pbs:WebFormDesc>
<pbs:CountryOfOrigin>United States</pbs:CountryOfOrigin>
<pbs:LayoutAcross>0</pbs:LayoutAcross>
<pbs:LayoutAround>0</pbs:LayoutAround>
<pbs:CSRFullName/>
<pbs:SalesRepFullName/>
<pbs:ChargeForStorage>0</pbs:ChargeForStorage>
<pbs:Customizable>N</pbs:Customizable>
<pbs:MaxUseQty>0</pbs:MaxUseQty>
<pbs:MaxUsePercent>0.0</pbs:MaxUsePercent>
<pbs:PurchaseUOM>0</pbs:PurchaseUOM>
<pbs:PurchaseUOMDesc/>
<pbs:WebForm>0</pbs:WebForm>
<pbs:CubeHeight>0.0</pbs:CubeHeight>
<pbs:CubeLength>0.0</pbs:CubeLength>
<pbs:CubeWidth>0.0</pbs:CubeWidth>

```



```
<pbs:CubeQuantity>0.0</pbs:CubeQuantity>
<pbs:CubeStorageCost>0</pbs:CubeStorageCost>
<pbs:SortField1>sort1</pbs:SortField1>
<pbs:SortField2>sort2</pbs:SortField2>
<pbs:ReplacementCode>rep code 1</pbs:ReplacementCode>
<pbs:ReviseDate>2019-02-25T00:00:00</pbs:ReviseDate>
<pbs:MinimumDays>33</pbs:MinimumDays>
<pbs:ReorderDays>55</pbs:ReorderDays>
<pbs:InternationalType/>
<pbs:OneTimeUse/>
<pbs:Notes>these are the item notes</pbs:Notes>
<pbs:EmailReplenish/>
<pbs:EmailRequired>N</pbs:EmailRequired>
<pbs:PackMaterialUse>N</pbs:PackMaterialUse>
<pbs:TurnTime>0</pbs:TurnTime>
<pbs:ForestTypeID>FT-2-Dean</pbs:ForestTypeID>
<pbs:Manager>Larry</pbs:Manager>
<pbs:Taxable>N</pbs:Taxable>
</pbs:Item>
</pbs:ItemList>
</pbs:InventoryUpdate>
```

Response Transaction

This is a successful transaction response (two items).

```
<?xml version="1.0" encoding="utf-8" ?>
<InventoryUpdateResponse>
  <ItemList>
    <UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <STKItemId>0</STKItemId>
      <ItemNumber>79131</ItemNumber>
      <Message>Success</Message>
    </UpdatedPSItem>
    <UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <STKItemId>0</STKItemId>
      <ItemNumber>79132</ItemNumber>
      <Message>Success</Message>
    </UpdatedPSItem>
  </ItemList>
</InventoryUpdateResponse>
```

This is an unsuccessful transaction response (two items). Note - a successful update will have a Message of "Success", otherwise the error message will be displayed.

```
<?xml version="1.0" encoding="utf-8" ?>
<InventoryUpdateResponse>
  <ItemList>
    <UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <STKItemId>0</STKItemId>
      <ItemNumber>791310</ItemNumber>
      <Message>Item Number: 791310 does not exist.</Message>
    </UpdatedPSItem>
    <UpdatedPSItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <STKItemId>0</STKItemId>
      <ItemNumber>79132</ItemNumber>
      <Message>Success</Message>
    </UpdatedPSItem>
  </ItemList>
</InventoryUpdateResponse>
```

Is Recipient Taxable

Narrative

This function will return the default taxable status of a recipient within an order.

1 = Taxable, 0 = Non Taxable

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/IsRecipientTaxable"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <IsRecipientTaxable xmlns="http://Printstream.org/">
      <CustomerId>int</CustomerId>
      <TypeID>int</TypeID>
      <SOID>int</SOID>
      <ActivityCode>string</ActivityCode>
      <ItemNo>long</ItemNo>
      <RecipientID>long</RecipientID>
      <ProjID>long</ProjID>
    </IsRecipientTaxable>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
```

```

    <PlantCode>0100</PlantCode>
    <AccountNo>JJD</AccountNo>
  </Authenticate>
</soap:Header>
<soap:Body>
  <IsRecipientTaxable xmlns="http://Printstream.org/">
    <CustomerId>13</CustomerId>
    <TypeID>1</TypeID>
    <SOID>161583</SOID>
    <ActivityCode />
    <ItemNo>0</ItemNo>
    <RecipientID>0</RecipientID>
    <ProjID>0</ProjID>
  </IsRecipientTaxable>
</soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<Recipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <IsTaxable>1</IsTaxable>
</Recipient>

```

Login

Narrative

This function validates login credentials for a user in Printstream.

ISUserLogged will be "true" if credentials are valid.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/Login"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Login xmlns="http://Printstream.org/">
      <AccountNo>string</AccountNo>
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
    </Login>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Login xmlns="http://Printstream.org/">
      <AccountNo>jjd</AccountNo>
      <RecipientID>jjweb</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
    </Login>
  </soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Recipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <RecipientID>11771</RecipientID>
  <IsTaxable>0</IsTaxable>
  <UserID>WEBADMIN</UserID>
  <ISUserLogged>true</ISUserLogged>
  <ISRecipLogin>false</ISRecipLogin>
  <CustRecipientID>JJWEB</CustRecipientID>
  <IsRecipLogged>false</IsRecipLogged>
</Recipient>
```

Main Interface

Narrative

MainInterface supports all legacy requests used by the original version of Web2PS.

See the Web2PS Field Map for details regarding legacy xml requests.

Note, when using legacy xml requests, they must be wrapped using the Soap Envelope structure as shown below. The Sample Request Transaction shows a legacy Inventory Status request. Nothing in the request has changed. It is simply inserted as part of the strRequest.

The CDATA tags are necessary so the legacy xml is treated as a string.

Request Transaction

```
POST /Web2PSWCF/ECommerceWeb.asmx HTTP/1.1
Host: pswEB12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/MainInterface"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <MainInterface xmlns="http://Printstream.org/">
      <strRequest>string</strRequest>
    </MainInterface>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
```



```

    <PlantCode>0100</PlantCode>

    <AccountNo>JJD</AccountNo>

</Authenticate>

</soap:Header>

<soap:Body>

<MainInterface xmlns="http://Printstream.org/">

    <strRequest>

        <![CDATA[ <WEB2PS>
            <INFO>
                <PLANTCODE>0100</PLANTCODE>
                <CUSTOMER>13</CUSTOMER>
                <ACTION>InvStat</ACTION>
                <SYSDB>D:\Stream\SQL_DB\Generic_14_Multi\dosrun</SYSDB>
                <COMPCODE>01</COMPCODE>
            </INFO>
            <INV_REQ>
                <PROJ_ID>29</PROJ_ID>
                <INV_TRANS>1</INV_TRANS>
                <INVENTORY_CODE>326</INVENTORY_CODE>
                <CLEAN_DATA>0</CLEAN_DATA>
            </INV_REQ>
        </WEB2PS> ]]>

    </strRequest>

</MainInterface>

</soap:Body>

</soap:Envelope>

```

Response Transaction

```

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xmlns:xsd='http://www.w3.org/2001/XMLSchema'
xmlns:soap='http://schemas.xmlsoap.org/soap/envelope/'>
  <soap:Header>
    <Authenticate xmlns='http://Printstream.org/'>
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <MainInterface xmlns='http://Printstream.org/'>
      <strRequest><![CDATA[<WEB2PS>
        <INFO>

```

```
<ACTION>OrdImp</ACTION>
<SYSDB>D:\Stream\SQL_DB\Generic_14_Multi\dosrun</SYSDB>
<COMPCODE>01</COMPCODE>
<PLANTCODE>0100</PLANTCODE>
<CUSTOMER>13</CUSTOMER>
<REJECT_ENTIRE_ORDER>1</REJECT_ENTIRE_ORDER>
</INFO>
<ORDER>
  <CUST_SO_ID>G-Calvin-00001186</CUST_SO_ID>
  <CUST_BATCH_ID>
  <PROJECT_ID>29</PROJECT_ID>
  <ORDER_TYPE>Normal</ORDER_TYPE>
  <RECIPIENTS>
    <RECIPIENT>
      <CUST_RECIP_ID>DELMUR</CUST_RECIP_ID>
      <RECIP_TYPE>OB</RECIP_TYPE>
      <FIRST_NAME>Delvin</FIRST_NAME>
      <LAST_NAME>Murray</LAST_NAME>
      <COMPANY_NAME>SL</COMPANY_NAME>
      <DAY_PHONE></DAY_PHONE>
      <EMAIL>dmurray@test.com</EMAIL>
      <ADDR1>2000 Oak Industrial Dr.</ADDR1>
      <ADDR2></ADDR2>
      <ADDR3></ADDR3>
      <CITY>Grand Rapids</CITY>
      <STATE_CODE>MI</STATE_CODE>
      <ZIP_CODE>49505</ZIP_CODE>
      <COUNTRY_CODE>USA</COUNTRY_CODE>
    </RECIPIENT>
    <RECIPIENT>
      <CUST_RECIP_ID>DELMUR</CUST_RECIP_ID>
      <RECIP_TYPE>ST</RECIP_TYPE>
      <FIRST_NAME>Delvin</FIRST_NAME>
      <LAST_NAME>Murray</LAST_NAME>
      <COMPANY_NAME>SL</COMPANY_NAME>
      <DAY_PHONE></DAY_PHONE>
      <EMAIL></EMAIL>
      <ADDR1>2000 Oak Industrial Dr.</ADDR1>
      <ADDR2></ADDR2>
      <ADDR3></ADDR3>
      <CITY>Grand Rapids</CITY>
      <STATE_CODE>MI</STATE_CODE>
      <ZIP_CODE>49505</ZIP_CODE>
      <COUNTRY_CODE>USA</COUNTRY_CODE>
    </RECIPIENT>
```



```

<RECIPIENT>
  <CUST_RECIP_ID>DELMUR</CUST_RECIP_ID>
  <RECIP_TYPE>BT</RECIP_TYPE>
  <FIRST_NAME>Delvin</FIRST_NAME>
  <LAST_NAME>Murray</LAST_NAME>
  <COMPANY_NAME>SL</COMPANY_NAME>
  <DAY_PHONE></DAY_PHONE>
  <EMAIL></EMAIL>
  <ADDR1>2000 Oak Industrial Dr.</ADDR1>
  <ADDR2></ADDR2>
  <ADDR3></ADDR3>
  <CITY>Grand Rapids</CITY>
  <STATE_CODE>MI</STATE_CODE>
  <ZIP_CODE>49505</ZIP_CODE>
  <COUNTRY_CODE>USA</COUNTRY_CODE>
</RECIPIENT>
</RECIPIENTS>
<ITEMS>
  <ITEM>
    <INVENTORY_CODE>314</INVENTORY_CODE>
    <ORDER_QTY>500</ORDER_QTY>
    <UNIT_PRICE>0.25</UNIT_PRICE>
    <SHIPPING_HANDLING>0</SHIPPING_HANDLING>
    <SHIP_MODE_NAME>A09</SHIP_MODE_NAME>
    <DOC_ID>D-Calvin-00001385</DOC_ID>
  </ITEM>
</ITEMS>
</ORDER>
</WEB2PS>]]</strRequest>
  </MainInterface>
</soap:Body>
</soap:Envelope>

```

Save Nom Group Item

Narrative

This function provides the ability to add an item to an inventory nomination group (Catalog)

Response: 1 = Saved, 0 = Not Saved

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/SaveNomGroupItems"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <SaveNomGroupItems xmlns="http://Printstream.org/">
      <NomGroupList>
        <NomGroupItem>
          <Featured>boolean</Featured>
          <Level1>int</Level1>
          <Level2>int</Level2>
          <Level3>int</Level3>
          <Level4>int</Level4>
          <NomDescription>string</NomDescription>
          <Image>string</Image>
          <LongDescription>string</LongDescription>
          <NomGroupItemId>int</NomGroupItemId>
          <NomGroupIdForItem>int</NomGroupIdForItem>
          <CustomerId>int</CustomerId>
          <NominationId>int</NominationId>
          <NominationName>string</NominationName>
          <STKHeaderId>int</STKHeaderId>
          <Checked>boolean</Checked>
          <ProjectId>int</ProjectId>
          <PlantCode>string</PlantCode>
          <CompanyCode>string</CompanyCode>
        </NomGroupItem>
        <NomGroupItem>
          <Featured>boolean</Featured>
          <Level1>int</Level1>
          <Level2>int</Level2>
          <Level3>int</Level3>
          <Level4>int</Level4>
          <NomDescription>string</NomDescription>
          <Image>string</Image>
```

```

        <LongDescription>string</LongDescription>
        <NomGroupItemId>int</NomGroupItemId>
        <NomGroupIdForItem>int</NomGroupIdForItem>
        <CustomerId>int</CustomerId>
        <NominationId>int</NominationId>
        <NominationName>string</NominationName>
        <STKHeaderId>int</STKHeaderId>
        <Checked>boolean</Checked>
        <ProjectId>int</ProjectId>
        <PlantCode>string</PlantCode>
        <CompanyCode>string</CompanyCode>
    </NomGroupItem>
</NomGroupList>
</SaveNomGroupItems>
</soap:Body>
</soap:Envelope>

```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
```

```

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

```

```
<soap:Header>
```

```
  <Authenticate xmlns="http://Printstream.org/">
```

```
    <RecipientID>JJWEB</RecipientID>
```

```
    <Password>ok</Password>
```

```
    <CompanyCode>01</CompanyCode>
```

```
    <PlantCode>0100</PlantCode>
```

```
    <AccountNo>JJD</AccountNo>
```

```
  </Authenticate>
```

```
</soap:Header>
```

```
<soap:Body>
```

```
  <NomGroupList>
```

```
    <NomGroupItem>
```

```
      <Featured>false</Featured>
```

```
      <Level1>2</Level1>
```

```
      <Level2>0</Level2>
```

```
      <Level3>0</Level3>
```

```
      <Level4>0</Level4>
```

```
      <NomDescription>Fishing 2005 Catalog</NomDescription>
```

```

    <Image />
    <LongDescription>Fly Rods</LongDescription>
    <NomGroupId>527</NomGroupId>
    <NomGroupIdForItem>10124</NomGroupIdForItem>
    <CustomerId>13</CustomerId>
    <NominationId>24</NominationId>
    <NominationName>Fishing 2005</NominationName>
    <STKHeaderId>245</STKHeaderId>
    <Checked>true</Checked>
    <ProjectId>29</ProjectId>
    <PlantCode>0100</PlantCode>
    <CompanyCode>01</CompanyCode>
  </NomGroupItem>
</NomGroupList>
</soap:Body>
</soap:Envelope>

```

Response Transaction

```

<?xml version="1.0" encoding="utf-8" ?>
<SaveNomGroupItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <IsSaved>1</IsSaved>
</SaveNomGroupItem>

```

Save Product

Narrative

This API allows you to create or update Products , their components, associated specs and digital assets.

Request Transaction

```
POST /Web2PSWCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/SaveProduct"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <SaveProduct xmlns="http://Printstream.org/">
      <ProductXML>string</ProductXML>
    </SaveProduct>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>

<Product xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" AccountDistributionCode="441"
CategoryCode="CC1" Comments="ret88 dsf dsf sef esfesfes" MarketingCode="MC2"
ProductCatalog="Test Co Items" ProductDescription="This is for description"
ProductSource="PS" PublicationType="44" RootPartNumber="BPSD-For my use only"
SecondaryDescription="erte4 sec description" Status="Ready" Version="1"
Action="ADD_OR_UPDATE" xmlns="printstream:efi:com">

  <Component ComponentCategory="" ComponentName="SD110612FG1 FG Item 1"
ComponentType="Insert" IncludeInComposite="false" CompositeSequence="0" PricingUnit=""
NumberOfUnits="100" FulfillmentType="FinishedGoods" VendorCode=""
ShortDescription="SD110612FG1" Action="ADD_OR_UPDATE" InventoryItem="false" />

  <ProductFormat BodyMatch="BM1" CalculatedProductWeight="20" CanRelease="true"
CanShip="true" CompositeDocumentID="" CompositeBuilderStatus="Ready" Comments="test
comments ewrewr" CoverMatch="CM1234" CustomsValue="2" ExtendedTurntimeUsed="true"
FinalSizeLength="0" FinalSizeWidth="0" Format="CD" FulfillmentType="FinishedGoods"
GenerateComposite="true" GenerateXjT="true" MediaType="Book" MinOrderQty="0"
OriginCountry="UKR" OverrideCalculatedWeight="false" PrepInstructions="34543543"
```

```

PrintWorkType="Label" QCInstructions="erttrt" QtyPerBox="100" ScheduleBCode="SBC"
ShortDescription="CD3333" TemplateInstructions="4e54e54e" VendorCode="VC"
VerifyWhenOrdered="true">

<Component ComponentCategory="" ComponentName="CD Comp" ComponentType="Multi Sheet"
  IncludeInComposite="true" CompositeSequence="1" PricingUnit="23" NumberOfUnits="100"
  FulfillmentType="FinishedGoods" VendorCode="3243242" ShortDescription="CD Comp"
  Action="ADD_OR_UPDATE" InventoryItem="false">

  <ProductSpec SpecName="Digital print 4 color" Qty="100" Pages="2" Images="4" />

  <ProductSpec SpecName="3 hole punch" Qty="200" Pages="2" Images="3" />

</Component>

<Component ComponentCategory="" ComponentName="test 123" ComponentType="Insert"
  IncludeInComposite="false" CompositeSequence="2" PricingUnit="4" NumberOfUnits="100"
  FulfillmentType="FinishedGoods" VendorCode="ewrewr" ShortDescription="123"
  Action="ADD_OR_UPDATE" InventoryItem="false">

  <ProductSpec SpecName="Digital print 4 color" Qty="100" Pages="2" Images="3" />

</Component>

<Component ComponentCategory="" ComponentName="New Component2eq"
  ComponentType="Insert" IncludeInComposite="true" CompositeSequence="2" PricingUnit="2"
  NumberOfUnits="1" FulfillmentType="FinishedGoods" VendorCode="eewrewrw3"
  ShortDescription="2e2q" Action="ADD_OR_UPDATE" InventoryItem="false">

  <ProductSpec SpecName="Digital print 4 color" Qty="100" Pages="2" Images="5" />

</Component>

<Component ComponentCategory="A" ComponentName="t332423" ComponentType="Insert"
  IncludeInComposite="true" CompositeSequence="4" PricingUnit="5" NumberOfUnits="100"
  FulfillmentType="FinishedGoods" VendorCode="tytrytrytr" ShortDescription="wr3r3w"
  Action="ADD_OR_UPDATE" InventoryItem="false">

  <ProductSpec SpecName="Collate" Qty="100" Pages="5" Images="3" />

</Component>

</ProductFormat>

</Product>

```


Response Transaction

```
<pst:Product ID="50" Status="Success" Message="BPSD-For my use only-CD saved successfully"
  xmlns:pst="printstream:efi:com">
  <pst:Component ID="101" Status="Success" Message="CD Comp saved successfully">
    <pst:ProductSpec ID="38" Status="Success" Message="38 saved successfully" />
    <pst:ProductSpec ID="39" Status="Success" Message="39 saved successfully" />
  </pst:Component>
  <pst:Component ID="103" Status="Success" Message="test 123 saved successfully">
    <pst:ProductSpec ID="40" Status="Success" Message="40 saved successfully" />
  </pst:Component>
  <pst:Component ID="109" Status="Success" Message="New Component2eq saved successfully">
    <pst:ProductSpec ID="44" Status="Success" Message="44 saved successfully" />
  </pst:Component>
  <pst:Component ID="110" Status="Success" Message="t332423 saved successfully">
    <pst:ProductSpec ID="45" Status="Success" Message="45 saved successfully" />
  </pst:Component>
</pst:Product>
```

Service Test

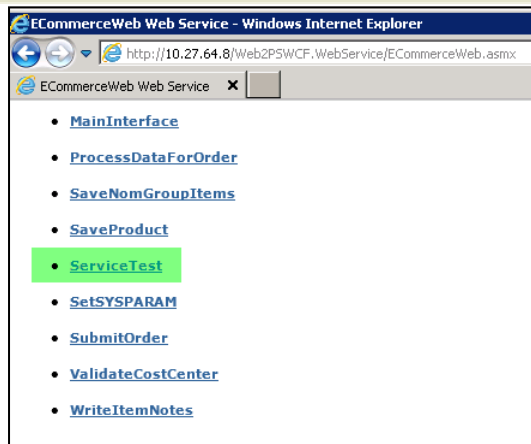
Narrative

This function sends a GetSYSPARAM request for the parameter Sales Order Focus Field. The purpose is to provide a simple test from a web browser to determine if the service is communicating with the database.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/ServiceTest"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ServiceTest xmlns="http://Printstream.org/" />
  </soap:Body>
</soap:Envelope>
```



No xml is required. Simply click on the Invoke button.

ServiceTest

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Good Result

```
<?xml: xsi="http://www.w3.org/2001/XMLSchema-instance"
er> <SYS_PARAM>Sales Order #</SYS_PARAM> </Misc></string>
```

Bad Result (Incorrect database path in config file)

```
Error in ServiceTest: An error occurred when verifying security for the message.<
```

Bad Result (Invalid authentication credentials in web.config)

```
">Error in ServiceTest: Unknown AccountNumber, or incorrect Password</st
```

Bad Result (WCF Service is not running)

```
System.ServiceModel.EndpointNotFoundException: Could not connect to http://localhost:8002/WCFHost/Web2PS.svc/Miscellaenous.
TCP error code 10061: No connection could be made because the target machine actively refused it 127.0.0.1:8002.
Could not connect to http://localhost:8002/WCFHost/Web2PS.svc/Miscellaenous.
```

Submit Order

Narrative

This function provides the ability to add a new sales order to the system.

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/SubmitOrder"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <SubmitOrder xmlns="http://Printstream.org/">
      <InputXML>string</InputXML>
    </SubmitOrder>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<pbs:OrderList
  xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">
  <pbs:Order CustomerID="13" ProjectID="29" CustomerOrderID="Sep04TST1"
    CustomerBatchID="Sep04TB1" ChargeType="0" ShipToAttention="Attention: Office Manager
    Sally Simpson" BillToAttention="Attention: Billing Manager Tommy Walker">
    <pbs:SOType Name="Normal" LookupName="true" />
    <pbs:OrderBy CustomerRecipientID="KDOE1123e" CustomerID="13" FirstName="Kathy"
      LastName="Doe" DayPhone="860-889-1234" Email="test@foo.com" CompanyName="SL"
      LookupName="true">
      <pbs:Address City="Hartford" State="CT" Zip="06001" AddressLine1="327 Main St"
        AddressLine2="Suite 201">
        <pbs:Country Code="US" LookupName="true" />
      </pbs:Address>
    </pbs:OrderBy>
```

```

<pbs:ShipTo CustomerRecipientID=" KDOE1123e " CustomerID="13" FirstName="Kathy"
  LastName=" Doe " DayPhone="860-889-1234"
  Email=" test@foo.com " CompanyName="SL" LookupName="true">
<pbs:Address City=" Hartford " State="CT" Zip="06001" AddressLine1="327 Main St"
  AddressLine2="Suite 201">
  <pbs:Country Code="US" LookupName="true" />
</pbs:Address>
</pbs:ShipTo>

<pbs:BillTo CustomerRecipientID=" KDOE1123e " CustomerID="13" FirstName="Kathy" LastName="
  Doe " DayPhone="860-889-1234"
  Email=" test@foo.com " CompanyName="SL" LookupName="true">
<pbs:Address City=" Hartford " State="CT" Zip="06001" AddressLine1="327 Main St"
  AddressLine2="Suite 201">
  <pbs:Country Code="US" LookupName="true" />
</pbs:Address>
</pbs:BillTo>

<pbs:OrderItemList>
<pbs:OrderItem Quantity="1" CustomerOrderID="Sep04TST1" PackageNotes="these are my
  package notes">
  <pbs:Product InventoryCode="314" ItemID="314" />
  <pbs:ShippingMode Name="U21" LookupName="true" />
</pbs:OrderItem>

<pbs:OrderItem Quantity="1" CustomerOrderID="Sep04TST1">
  <pbs:Product InventoryCode="315" ItemID="315" />
  <pbs:ShippingMode Name="U21" LookupName="true" />
</pbs:OrderItem>
</pbs:OrderItemList>
</pbs:Order></pbs:OrderList>

Note: Like MainInterface, since we pass an xml request as a string, you must use a CDATA tag
around the xml.

<?xml version="1.0" encoding="utf-8" ?>

<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
  <Authenticate xmlns="http://Printstream.org/">

```



```
<RecipientID>JJWEB</RecipientID>
<Password>ok</Password>
<CompanyCode>01</CompanyCode>
<PlantCode>0100</PlantCode>
<AccountNo>JJD</AccountNo>
</Authenticate>
</soap:Header>
<soap:Body>
<SubmitOrder xmlns="http://Printstream.org/">
  <InputXML>
    <![CDATA[
      ... Insert XML string here
    ]]>
  </ InputXML >
</ SubmitOrder >
</soap:Body>
</soap:Envelope>
```

Sample Request Transaction with External Payment Information

```
<?xml version="1.0" encoding="UTF-8"?>
<pbs:OrderList
xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">
  <pbs:Order CustomerID="13" ProjectID="29" CustomerOrderID="Sep04TST5a"
CustomerBatchID="Sep04TB1" ChargeType="0" ShipToAttention="Attention: Office
Manager Sally Simpson" BillToAttention="Attention: Billing Manager Tommy Walker"
PNREF="A71A6DAFE29F" OrderTransactionID="A71A6DAFE29F"
OrderToken="XXXXXXXXXXXX5100" OrderAuthorizeAmount="26.00000000"
OrderAuthorizeDateTime="2014-08-18T09:08:57" ExternalCardType="MC"
ExternalCardExpiration="2016-12-01T00:00:00" ExternalPaymentMethod="Credit Card">
    <pbs:SOType Name="Normal" LookupName="true" />
    <pbs:OrderBy CustomerRecipientID="KDIP1123f" CustomerID="13" FirstName="Kathy"
LastName="DiPaolo" DayPhone="860-889-1234" Email="dean@printstream.com"
CompanyName="SL" LookupName="true">
      <pbs:Address City="Bozrah" State="CT" Zip="06334" AddressLine1="327 Main St"
AddressLine2="Suite 201">
        <pbs:Country Code="US" LookupName="true" />
      </pbs:Address>
    </pbs:OrderBy>
    <pbs:ShipTo CustomerRecipientID="KDIP1123f" CustomerID="13" FirstName="Kathy"
LastName="DiPaolo" DayPhone="860-889-1234" Email="dean@printstream.com"
CompanyName="SL" LookupName="true">
      <pbs:Address City="Bozrah" State="CT" Zip="06334" AddressLine1="327 Main St"
AddressLine2="Suite 201">
        <pbs:Country Code="US" LookupName="true" />
      </pbs:Address>
    </pbs:ShipTo>
    <pbs:BillTo CustomerRecipientID="KDIP1123Credit" CustomerID="13"
FirstName="Kathy" LastName="DiPaolo" DayPhone="860-889-1234"
Email="dean@printstream.com" CompanyName="KD" LookupName="true">
      <pbs:Address City="Norwich" State="CT" Zip="06360" AddressLine1="123 Credit
Ave" AddressLine2="Suite 610">
        <pbs:Country Code="US" LookupName="true" />
      </pbs:Address>
    </pbs:BillTo>
    <pbs:OrderItemList>
      <pbs:OrderItem Quantity="1" CustomerOrderID="Sep04TST5a" PackageNotes="these
are my package notes" FileNotes="the are my file notes" >
        <pbs:Product InventoryCode="314" ItemID="314" />
        <pbs:ShippingMode Name="U21" LookupName="true"/>
      </pbs:OrderItem>
      <pbs:OrderItem Quantity="1" CustomerOrderID="Sep04TST5a" >
        <pbs:Product InventoryCode="315" ItemID="315" />
        <pbs:ShippingMode Name="U21" LookupName="true"/>
      </pbs:OrderItem>
    </pbs:OrderItemList>
  </pbs:Order>
</pbs:OrderList>
```

External Payment information is displayed from the Sales Order Maintenance screen, by clicking External Payment Info.

The screenshot shows a software application window with a menu bar and a dialog box titled "External Payment".

Menu Bar:

- Finance: Invoice, Credit Card, Credit, Add Discount Charge, Audit, Transactions, Billing
- Shipping: Shipping, Estimated Shipping, Save, Ac...

External Payment Dialog Box:

External Payment Information

Payment Method:

Transaction ID:

Order Token:

PN Reference:

Authorize Amount: Authorize Date:

External Card Type: Expiration Date:

Billing Entity PO #:

Billing Entity A/c #:

Free Form A/c #:

Cost Center #:

Update Recipient Attributes

Narrative

This function will update Campaign Attributes by Recipient

Name - this is the name of the attribute to be updated

Value - this is the value of attribute

- If the Value is blank, the attribute will be removed for the recipient.
- The Value will be validated against the Options for the Attribute. If the Value is invalid for the attribute, an error will be returned in the response xml.
- If the attribute is not attached to the recipient, it will be added as new.
- Only the attributes to be changed need be included in the Request Transaction.

Request Transaction

```
POST /Web2PSWCF.WebService.Setup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/UpdateRecipientAttributes"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <UpdateRecipientAttributes xmlns="http://Printstream.org/">
      <xmlString>string</xmlString>
    </UpdateRecipientAttributes>
  </soap:Body>
</soap:Envelope>
```

Schema

RecipientAttributeList.xsd

Sample Request Transaction

```

<?xml version="1.0" ?>

<pbs:RecipientAttributeList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns0="printstream:efi:com"
xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">
  <pbs:RecipList xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">
    <pbs:Recipient ACNO="VAL002" CustRecipID="0102">
      <pbs:AttributeList>
        <pbs:Attribute Name="American Greetings" Value="yes" Type="S" MaxLength="0"
Decimals="0" />
        <pbs:Attribute Name="# of entrances" Value="1" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Backer Paper" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Black" Value="yes" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Behind the Register" Value="11" Type="S" MaxLength="0" Decimals="0"
/>
        <pbs:Attribute Name="Dist" Value="" Type="S" MaxLength="0" Decimals="0" />
      </pbs:AttributeList>
    </pbs:Recipient>
    <pbs:Recipient ACNO="VAL002" CustRecipID="0106">
      <pbs:AttributeList>
        <pbs:Attribute Name="American Greetings" Value="yes" Type="S" MaxLength="0"
Decimals="0" />
        <pbs:Attribute Name="# of entrances" Value="1111111" Type="S" MaxLength="0"
Decimals="0" />
        <pbs:Attribute Name="Backer Paper" Value="no" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Black" Value="yes" Type="S" MaxLength="0" Decimals="0" />
        <pbs:Attribute Name="Behind the Register" Value="2" Type="S" MaxLength="0" Decimals="0"
/>
        <pbs:Attribute Name="Dist" Value="26" Type="S" MaxLength="0" Decimals="0" />
      </pbs:AttributeList>
    </pbs:Recipient>
  </pbs:RecipList>
</pbs:RecipientAttributeList>

```

Response Transaction

```

<pbs:RecipientAttributeList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns0="printstream:efi.com"
  xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">
  <pbs:RecipList
    xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">
    <pbs:Recipient ACNO="VAL002" CustRecipID="0102">
    <pbs:AttributeList>
      <pbs:Attribute Name="American Greetings" Value="yes" Type="S" MaxLength="0" Decimals="0"
        Error="" />
      <pbs:Attribute Name="# of entrances" Value="1" Type="S" MaxLength="0" Decimals="0" Error=""
        />
      <pbs:Attribute Name="Backer Paper" Value="no" Type="S" MaxLength="0" Decimals="0" Error=""
        />
      <pbs:Attribute Name="Black" Value="yes" Type="S" MaxLength="0" Decimals="0" Error="" />
      <pbs:Attribute Name="Behind the Register" Value="11" Type="S" MaxLength="0" Decimals="0"
        Error="" />
      <pbs:Attribute Name="Dist" Value="" Type="S" MaxLength="0" Decimals="0" Error="" />
    </pbs:AttributeList>
    </pbs:Recipient>
  </pbs:RecipList>
</pbs:RecipientAttributeList>
= <pbs:Recipient ACNO="VAL002" CustRecipID="0106">
= <pbs:AttributeList>
  <pbs:Attribute Name="American Greetings" Value="yes" Type="S" MaxLength="0" Decimals="0"
    Error="" />
  <pbs:Attribute Name="# of entrances" Value="111111" Type="S" MaxLength="0" Decimals="0"
    Error="111111 is not a valid option value" />
  <pbs:Attribute Name="Backer Paper" Value="no" Type="S" MaxLength="0" Decimals="0" Error=""
    />
  <pbs:Attribute Name="Black" Value="yes" Type="S" MaxLength="0" Decimals="0" Error="" />
  <pbs:Attribute Name="Behind the Register" Value="2" Type="S" MaxLength="0" Decimals="0"
    Error="" />
  <pbs:Attribute Name="Dist" Value="26" Type="S" MaxLength="0" Decimals="0" Error="" />
</pbs:AttributeList>
</pbs:Recipient>
</pbs:RecipList>
</pbs:RecipientAttributeList>

```

Validate Cost Center

Narrative

This function will confirm if a "Cost Center" is Valid. PrintStream provides the ability to configure user defined lists of cost centers for a customer. These cost centers are entered on orders and can be used for reporting or billing purposes. True = Valid, False = Invalid

Request Transaction

```
POST /Web2PS_WCF/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/ValidateCostCenter"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <ValidateCostCenter xmlns="http://Printstream.org/">
      <RecipientId>int</RecipientId>
      <CostCenterId>string</CostCenterId>
    </ValidateCostCenter>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>JJWEB</RecipientID>
      <Password>ok</Password>
      <CompanyCode>01</CompanyCode>
      <PlantCode>0100</PlantCode>
      <AccountNo>JJD</AccountNo>
```

```
</Authenticate>
</soap:Header>
<soap:Body>
  <ValidateCostCenter xmlns="http://Printstream.org/">
    <RecipientId>84650</RecipientId>
    <CostCenterId>CC34534</CostCenterId>
  </ValidateCostCenter>
</soap:Body>
</soap:Envelope>
```

Response Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<Recipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <IsValidate>True </IsValidate>
</Recipient>
```

Write Item Notes

Narrative

This function creates notes, including file attachments, to a line item within an order in Printstream.

Order level notes (and attachments) may be sent by setting SubContextId and LineItemNumber to 0.

Note EncodedString must be a Base64 encoding string of the file. See the document "How To Attach Files to Order Line Items" for details on how to implement this function.

Request Transaction

```
POST /Streamline.Ecommerce.WebSetup/ECommerceWeb.asmx HTTP/1.1
Host: psweb12.printstream.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/WriteItemNotes"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <WriteItemNotes xmlns="http://Printstream.org/">
      <CustomerId>int</CustomerId>
      <Title>string</Title>
      <NotesText>string</NotesText>
      <SalesOrderId>int</SalesOrderId>
      <SubContextId>int</SubContextId>
      <ThreadId>int</ThreadId>
      <LineItemNumber>int</LineItemNumber>
      <EncodedString>string</EncodedString>
      <FileName>string</FileName>
    </WriteItemNotes>
  </soap:Body>
</soap:Envelope>
```

Sample Request Transaction

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
```



```

    <RecipientID>JJWEB</RecipientID>

    <Password>ok</Password>

    <CompanyCode>01</CompanyCode>

    <PlantCode>0100</PlantCode>

    <AccountNo>JJD</AccountNo>

  </Authenticate>

</soap:Header>

<soap:Body>
  <WriteItemNotes xmlns="http://Printstream.org/">
    <CustomerId>13</CustomerId>
    <Title>Attachment Test 1</Title>
    <NotesText>Test attaching order note on line item 2</NotesText>
    <SalesOrderId>1000000012</SalesOrderId>
    <SubContextId>0</SubContextId>
    <ThreadId>1</ThreadId>
    <LineItemNumber>0</LineItemNumber>
    <EncodedString />
    <FileName>MyOrderFile.xml</FileName>
  </WriteItemNotes>
</soap:Body>

</soap:Envelope>

```

Response Transaction

Write Item Notes is an asynchronous function. (No response returned).

Web2PS Login Test Setup

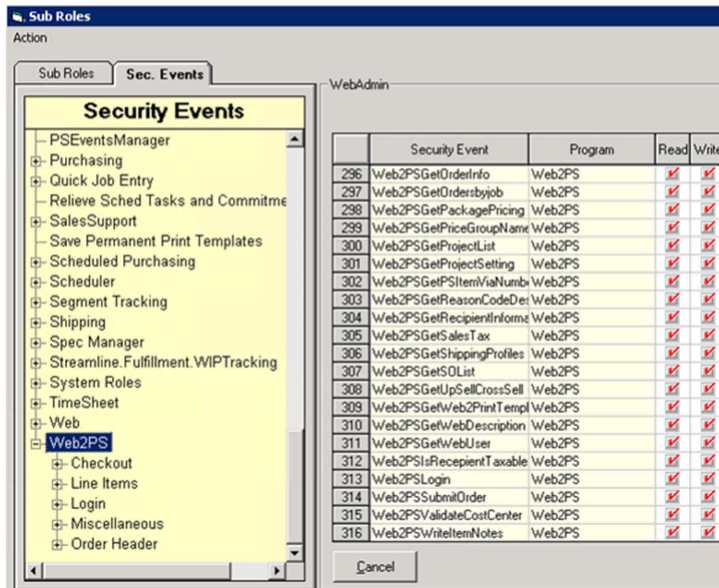
This will outline how to setup a Web2PS Login user by adding security and roles, creating a new recipient and testing the login process.

Adding a Security Event

This setup is illustrated in the document *Web2PS Installation and Setup* (page 13, step 6). Create a new Sub Role or add Web2PS security events to an existing Sub Role. Create a new Role, or use an existing Role, which utilizes the Web2ps Security Events.

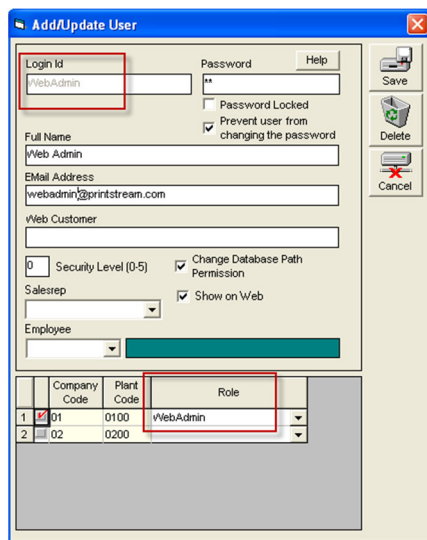
Add Security Events (Application Server where it can connect to PSMB)

- Security events are available for Web2PS in "Printstream Administration" section.
- Assign security events to particular role (say Web Admin).
- Assign the above role to a recipient to use in the `web.config` file of the web service.
(You will do this in Step 9, below.)



Create a Security User

Create a new PrintStream User, or utilize an existing User, which is mapped to the Role updated in Step 1.



Add/Update User

Login Id: WebAdmin Password: [password field] Help [icon]

Full Name: Web Admin

E-Mail Address: webadmin@printstream.com

Web Customer: [empty field]

Security Level (0-5): 0 ☒ Change Database Path Permission

Salesrep: [dropdown] ☒ Show on Web

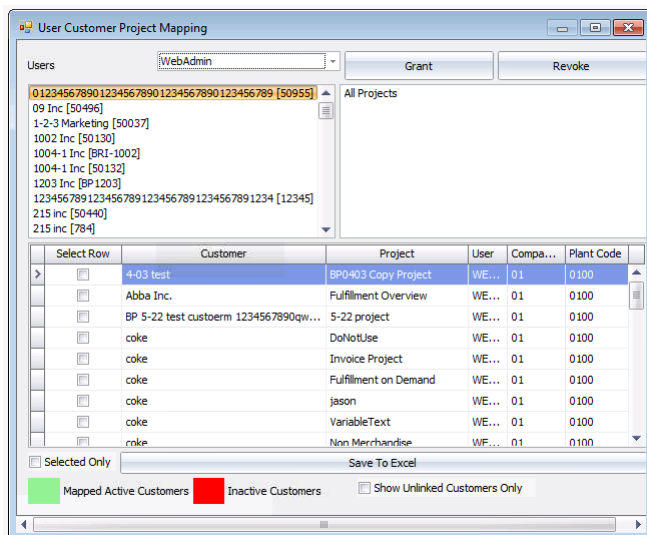
Employee: [dropdown]

Buttons: Save, Delete, Cancel

	Company Code	Plant Code	Role
1	01	0100	WebAdmin
2	02	0200	[dropdown]

Map Security User

Map the PrintStream User created in Step 2 to the Customer/Project.



User Customer Project Mapping

Users: WebAdmin Grant Revoke

Users list: 012345678901234567890123456789 [50955], 09 Inc [50496], 1-2-3 Marketing [50037], 1002 Inc [50130], 1004-1 Inc [BRI-1002], 1004-1 Inc [50132], 1203 Inc [BP 1203], 1234567891234567891234567891234 [12345], 215 inc [50440], 215 inc [784]

All Projects

Select Row	Customer	Project	User	Compa...	Plant Code
>	4-03 test	BP0403 Copy Project	WE...	01	0100
	Abba Inc.	Fulfillment Overview	WE...	01	0100
	BP 5-22 test custoerm 1234567890qw...	5-22 project	WE...	01	0100
	coke	DoNotUse	WE...	01	0100
	coke	Invoice Project	WE...	01	0100
	coke	Fulfillment on Demand	WE...	01	0100
	coke	Jason	WE...	01	0100
	coke	VariableText	WE...	01	0100
	coke	Non Merchandise	WF...	01	0100

Selected Only Save To Excel

Mapped Active Customers Inactive Customers Show Unlinked Customers Only

Create Recipient

Create a new recipient in Fulfillment, or utilize an existing recipient, where the Security User is mapped to the PrintStream User account created in Step 2. Note: The User account must have a Cust Recipient Id and a Password.

Determine the Customer Account Number

From Customer Maintenance, determine the Customer Account Number (Code) for the customer which has the recipient created in Step 4.

Run Login Test

Run Web2PS from a web browser. Find and click on Login.

ECommerceWeb

Click [here](#) for a complete list of operations.

Login

Test

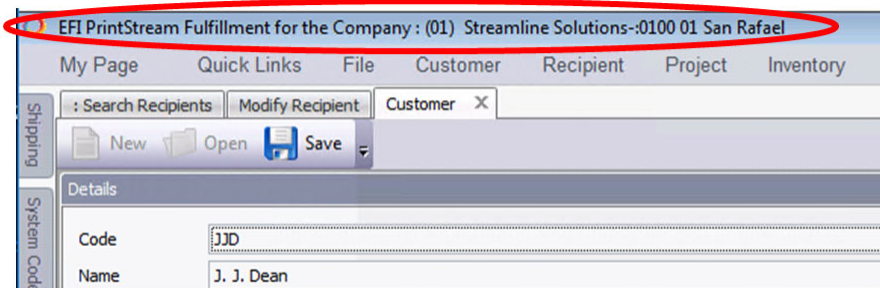
To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
AccountNo:	jjd
RecipientID:	jjweb
Password:	ok
CompanyCode:	01
PlantCode:	0100
Invoke	

Fill out the test form as follows.

- Account No - the customer code from Step 5.
- RecipientID - the customer recipient id from Step 4.
- Password - the recipient password from Step 4.
- CompanyCode - see below
- PlantCode - see below

If you are not sure of your company code, it is displayed with the Plant Code at the top of the Fulfillment screen as shown here:



Press **Invoke**. The response should look similar to this.

```
<?xml version="1.0" encoding="utf-8" ?>
<string xmlns="http://Printstream.org/"><?xml version="1.0" encoding="utf-8"?> <Recipient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <RecipientID>11771</RecipientID> <UserID>WEBADMIN</UserID> <ISUserLogged>True</ISUserLogged>
<CustRecipientID>JJWEB</CustRecipientID> </Recipient></string>
```

How To Attach Files to Order Line Items

In order to attach files (images, pdf's, xml, etc.) to a PrintStream order using Web2PS, the images must first be converted to a string using Base64 encoding. PrintStream will decode the string before writing the file to the database sub folder for the Note attachment.

In the *Programming Examples* section, you have the option of writing your own encoding routine (.Net examples provided) or by using a PrintStream dll.

The *How To Set Up Item Notes XML* section will show you how to construct an XML request which creates a Note for the order line item and attach a file (or files) to the Note.

Programming Examples

You have two options when programming your Web2PS interface to utilize file attachment option of the *WriteItemNotes* function in Web2PS. You may write your own encoding function or call a Printstream dll to encode the file for you. Both options are explained here.

Sample Code

There are plenty of code samples available on the web. This code sample is from Microsoft and was used as the basis for the Encoding and Decoding functions utilized by Web2PS.

While there are many ways to encode files as strings, it is important to use Base64 encoding logic since this is the standard across programming languages and is used by PrintStream to decode the string.

The following code samples were taken from the following Microsoft MSDN page.

VB.Net example (Microsoft)

```
Public Sub EncodeWithString()
    Dim inFile As System.IO.FileStream
    Dim binaryData() As Byte
    Try
        inFile = New System.IO.FileStream(inputFileName, System.IO.FileMode.Open,
System.IO.FileAccess.Read)
        ReDim binaryData(inFile.Length)
        Dim bytesRead As Long = inFile.Read(binaryData, 0, inFile.Length)
        inFile.Close()
    Catch exp As System.Exception
        ' Error creating stream or reading from it.
        System.Console.WriteLine("{0}", exp.Message)
        Return
    End Try

    ' Convert the binary input into Base64 UUEncoded output.
    Dim base64String As String
    Try
        base64String = System.Convert.ToBase64String(binaryData, 0, binaryData.Length)
    Catch exp As System.ArgumentNullException
        System.Console.WriteLine("Binary data array is null.")
        Return
    End Try

    ' Write the UUEncoded version to the output file.
    Dim outFile As System.IO.StreamWriter
```

```

Try
    outFile = New System.IO.StreamWriter(outputFileName, False, System.Text.Encoding.ASCII)
    outFile.Write(base64String)
    outFile.Close()
Catch exp As System.Exception
    ' Error creating stream or writing to it.
    System.Console.WriteLine("{0}", exp.Message)
End Try
End Sub

```

VB.Net example (PrintStream)

```

Public Function EncodeWithString(ByVal FilePath As String, ByRef ReturnString As String) As Boolean
    Dim inFile As System.IO.FileStream
    Dim binaryData() As Byte

    Try
        inFile = New System.IO.FileStream(FilePath, System.IO.FileMode.Open,
System.IO.FileAccess.Read)
        ReDim binaryData(inFile.Length)
        Dim bytesRead As Long = inFile.Read(binaryData, 0, inFile.Length)
        inFile.Close()
    Catch exp As System.Exception
        ' Error creating stream or reading from it.
        ReturnString = "Error: " & exp.Message
        Return False
    End Try

    ' Convert the binary input into Base64 UUEncoded output.
    Dim base64String As String
    Try
        base64String = System.Convert.ToBase64String(binaryData, 0, binaryData.Length)
    Catch exp As System.ArgumentNullException
        ReturnString = "Binary data array is null. Error: " & exp.Message
        Return False
    End Try

    ReturnString = base64String
    Return True

    '' Write the UUEncoded version to the output file.
    'Dim outFile As System.IO.StreamWriter
    'Try
    '    outFile = New System.IO.StreamWriter("c:\testEncode.txt", False, System.Text.Encoding.ASCII)
    '    outFile.Write(base64String)
    '    outFile.Close()
    'Catch exp As System.Exception
    '    ' Error creating stream or writing to it.
    '    System.Console.WriteLine("{0}", exp.Message)
    '    MsgBox(exp.Message)
    'End Try
End Function

```

C# Example (Microsoft)

```

public void EncodeWithString()
{
    System.IO.FileStream inFile; byte[] binaryData;
    try

```

```

    {
        inFile = new System.IO.FileStream(inputFileName, System.IO.FileMode.Open,
System.IO.FileAccess.Read);
        binaryData = new Byte[inFile.Length];
        long bytesRead = inFile.Read(binaryData, 0, (int)inFile.Length);
        inFile.Close();
    }
    catch (System.Exception exp)
    {
        // Error creating stream or reading from it.
        System.Console.WriteLine("{0}", exp.Message);
        return;
    }

    // Convert the binary input into Base64 UUEncoded output.
    string base64String;
    try
    {
        base64String = System.Convert.ToBase64String(binaryData, 0, binaryData.Length);
    }
    catch (System.ArgumentNullException)
    {
        System.Console.WriteLine("Binary data array is null.");
        return;
    }

    // Write the UUEncoded version to the output file.
    System.IO.StreamWriter outFile;
    try
    {
        outFile = new System.IO.StreamWriter(outputFileName, false, System.Text.Encoding.ASCII);
        outFile.Write(base64String);
        outFile.Close();
    }
    catch (System.Exception exp)
    {
        // Error creating stream or writing to it.
        System.Console.WriteLine("{0}", exp.Message);
    }
}
}

```

C# Example (Printstream)

```

public bool EncodeWithString(string FilePath, ref string ReturnString)
{
    System.IO.FileStream inFile = null;
    byte[] binaryData = null;

    try
    {
        inFile = new System.IO.FileStream(FilePath, System.IO.FileMode.Open,
System.IO.FileAccess.Read);
        binaryData = new byte[inFile.Length + 1];
        long bytesRead = inFile.Read(binaryData, 0, inFile.Length);
        inFile.Close();
    }
    catch (System.Exception exp)
    {
        // Error creating stream or reading from it.
        ReturnString = "Error: " + exp.Message;
        return false;
    }
}

```

```

    }

    // Convert the binary input into Base64 UUEncoded output.
    string base64String = null;
    try
    {
        base64String = System.Convert.ToBase64String(binaryData, 0, binaryData.Length);
    }
    catch (System.ArgumentNullException exp)
    {
        ReturnString = "Binary data array is null. Error: " + exp.Message;
        return false;
    }

    ReturnString = base64String;
    return true;

    //' Write the UUEncoded version to the output file.
    //Dim outFile As System.IO.StreamWriter
    //Try
    //    outFile = New System.IO.StreamWriter("c:\testEncode.txt", False, System.Text.Encoding.ASCII)
    //    outFile.Write(base64String)
    //    outFile.Close()
    //Catch exp As System.Exception
    //    ' Error creating stream or writing to it.
    //    'System.Console.WriteLine("{0}", exp.Message)
    //    MsgBox(exp.Message)
    //End Try

}/,
556 }

```

Utilize the PrintStream DLL to Encode a File

In the Printstream Winexe, you will find a dll call *Streamline.Ecommerce.Utility*

Among other functions, this dll contains a function to encode a file as string using Base64. The following example is in VB.Net. From your code, set a reference to the dll and create an object for the Common class.

```
Dim Encoder = New Streamline.ECommerce.Utility.Common
```

Within the Common class is a function called *EncodeWithString*.

```

Encoder.EncodeWithString(FilePath & FileName, EncodedString)
End Public Function EncodeWithString(FilePath As String, ByRef ReturnString As String) As Boolean

```

Simply pass the *FilePath* (i.e. C:\temp\MyDoc.xml). If the result is *True*, then ReturnString will be the encoded string. If the result is *False*, then ReturnString will contain the error message explaining the *False* result.

In the section *How To Set Up Item Notes XML* we explain how to pass the encoded string.

How to Setup Item Notes XML:

This section explains the work flow which ends by creating Notes with files attached. The first step is to submit an order to Printstream using Web2PS. The return xml will identify the internal order id (PSOrderID).

```

<?xml version="1.0" ?>
<?xml-stylesheet type='text/xsl'?>
<!-- Order Status Document -->
<ORDER_STATUS>
- <ORDERS>
- <ORDER>
  <CustOrderID>Jul02TST1</CustOrderID>
  <PSOrderID>1000000012</PSOrderID>
  <CustomerBatchID>Jan30TB1</CustomerBatchID>
  <BatchID>13712</BatchID>
  <StatusId>2</StatusId>
  <Status>Sales Order Open</Status>
</ORDER>
</ORDERS>
<ERRORS />
</ORDER_STATUS>

```

The next step is to call the Web2PS function *GetOrderDetails*. The return XML will contain additional information used in *WriteItemNotes*.

```

- <SalesOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema-ins
  <ErrorNumber>0</ErrorNumber>
  <OrderNo>Jul02TST1</OrderNo>
  <LocalAccount>29</LocalAccount>
  <PSOrderID>1000000012</PSOrderID>
  <NeedDate>7/5/2012 12:24:31 PM</NeedDate>
  <SOTypeName>Normal</SOTypeName>
  <OrderOrigin />
  <Status>Sales Order Open</Status>
  <ShipTo>127590</ShipTo>
  <CustomerId>13</CustomerId>
  <ShippingHandling>0</ShippingHandling>
  <Discount>0</Discount>
  <SalesTax>1.32</SalesTax>
  <Total>23.32</Total>
- <Items>
- <LineItem>
  <ItemNumber>314</ItemNumber>
  <LineItemNumber>1</LineItemNumber>
  <OrderQuantity>1</OrderQuantity>
  <ItemDescription>Angler Fly Rod Outfit!</ItemDescription>
  <StockHeaderRecNumber>245</StockHeaderRecNumber>
  <LineItemStatus>Committed</LineItemStatus>
  <PartNumber>314</PartNumber>
  <BackOrderQuantity>0</BackOrderQuantity>
  <CarrierCode>UPS 3 Day Select</CarrierCode>
  <PONumber />
  <Notes />
  <NominationId>24</NominationId>
  <Discount>0</Discount>
  <FFPriced>1</FFPriced>
  <HoldStatus>0</HoldStatus>
  <BOStatus>0</BOStatus>
  <DocumentId />
  <UOM />
  <GrossQuantity>1</GrossQuantity>
  <SubstituteItem>0</SubstituteItem>
  <CreditCardProcessingFee>0</CreditCardProcessingFee>
  <ItemTotal>10</ItemTotal>
</LineItem>
- <LineItem>
  <ItemNumber>315</ItemNumber>
  <LineItemNumber>2</LineItemNumber>
  <OrderQuantity>1</OrderQuantity>
  <ItemDescription>Classic Fly Rod Outfit</ItemDescription>
  <StockHeaderRecNumber>246</StockHeaderRecNumber>
  <LineItemStatus>Committed</LineItemStatus>
  <PartNumber>315</PartNumber>

```


In this example, the order has two items. When constructing the `WriteItemNotes` xml, you will need to extract the `LineItemNumber` as well as the `StockHeaderRecNumber` to identify the line item you are attaching a file to.

Other information, `CustomerId` and `PSOrderId` will be used as well.

Here is a sample `WriteItemNotes` request xml.

```

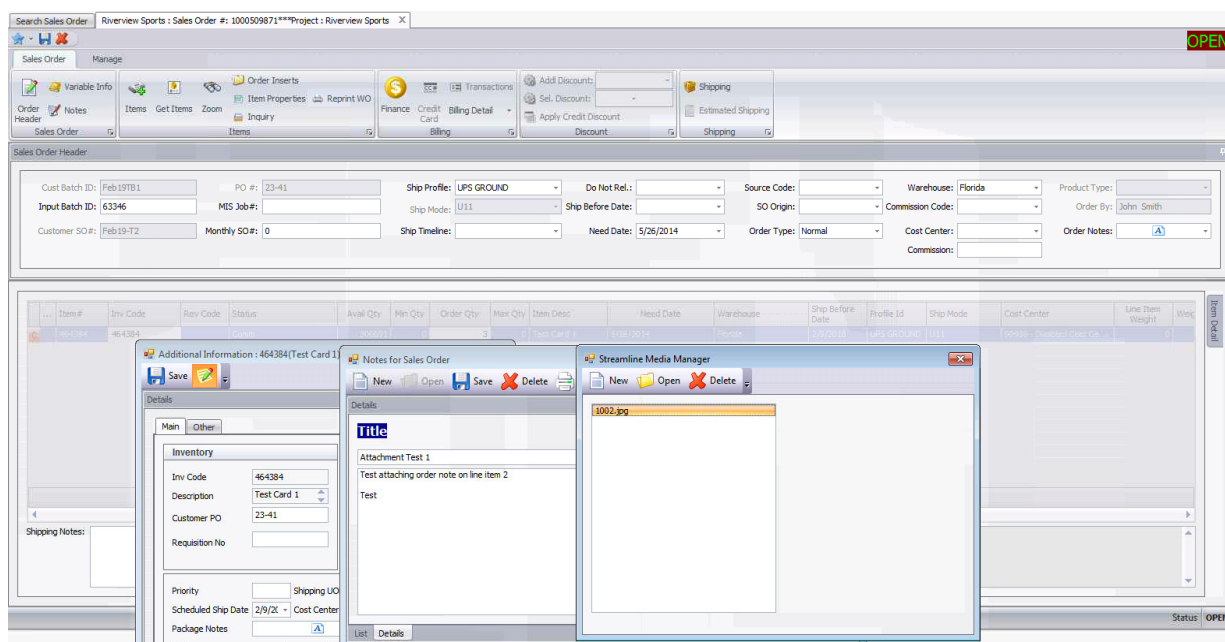
1  <?xml version="1.0"?>
2  <WriteItemNotes xmlns="http://Printstream.org/">
3    <CustomerId>13</CustomerId>
4    <Title>Attachment Test 1</Title>
5    <NotesText>Test attaching order note on line item 2</NotesText>
6    <SalesOrderId>1000000012</SalesOrderId>
7    <SubContextId>245</SubContextId>
8    <ThreadId>1</ThreadId>
9    <LineItemNumber>1</LineItemNumber>
10   <EncodedString>R01GOD1hYAFAAOYAAP////j5/Pb4+vH0+O71Ky+1Ky7zKa81aS2zJy20Zm0...</EncodedString>
11   <FileName>MySample.xml</FileName>
12 </WriteItemNotes>

```

CustomerId	- Customer Id for the order.
Title	- Notes title.
NotesText	- Detailed Notes text.
SalesOrderId	- This is the PSOrderId.
SubContextId	- For an order note, this is the StockHeaderRecNumber.
ThreadId	- For the first note (attachment), start with 1. If there are additional files to attach, the next will be 2, then 3, etc.
LineItemNumber	- This is the LineItemNumber for this item (Must match StockHeaderRecNumber).
EncodedString	- This is the encoded base64 string of the file. (Will be a MUCH large string than displayed in sample).

This request must be repeated for each attachment to each line item.

The end result will be found in the Notes and Attachments for each line item in the order.



Canceling an Order

Order Cancel has an OrderID (internal PrintStream Order Id), an action of "Cancel" and a ReasonCodeDesc, which must be a valid PrintStream Reason Code description for this customer.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <pbs:OrderList xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">
- <pbs:Order OrderID="1000000052" Action="Cancel" ReasonCodeDesc="CSR" CustomerID="13" ProjectID="29" CustomerOrderID="Sep04TST1" Cu
  Sally Simpson" BillToAttention="Attention: Billing Manager Tommy Walker">
  <pbs:SOType Name="Normal" LookupName="true" />
- <pbs:OrderBy CustomerRecipientID="KDIP1123e" CustomerID="13" FirstName="Kathy" LastName="DiPaolo" DayPhone="860-889-1234" Email="c
- <pbs:Address City="Bozrah" State="CT" Zip="06334" AddressLine1="327 Main St" AddressLine2="Suite 201">
  <pbs:Country Code="US" LookupName="true" />
</pbs:Address>
</pbs:OrderBy>
- <pbs:ShipTo CustomerRecipientID="KDIP1123e" CustomerID="13" FirstName="Kathy" LastName="DiPaolo" DayPhone="860-889-1234" Email="c
- <pbs:Address City="Bozrah" State="CT" Zip="06334" AddressLine1="327 Main St" AddressLine2="Suite 201">
  <pbs:Country Code="US" LookupName="true" />
</pbs:Address>
</pbs:ShipTo>
- <pbs:BillTo CustomerRecipientID="KDIP1123e" CustomerID="13" FirstName="Kathy" LastName="DiPaolo" DayPhone="860-889-1234" Email="de
- <pbs:Address City="Bozrah" State="CT" Zip="06334" AddressLine1="327 Main St" AddressLine2="Suite 201">
  <pbs:Country Code="US" LookupName="true" />
</pbs:Address>
</pbs:BillTo>
- <pbs:OrderItemList>
- <pbs:OrderItem Quantity="1" CustomerOrderID="Sep04TST1" PackageNotes="these are my package notes">
  <pbs:Product InventoryCode="314" ItemID="314" />
  <pbs:ShippingMode Name="U21" LookupName="true" />
</pbs:OrderItem>
- <pbs:OrderItem Quantity="1" CustomerOrderID="Sep04TST1">
  <pbs:Product InventoryCode="315" ItemID="315" />
  <pbs:ShippingMode Name="U21" LookupName="true" />
</pbs:OrderItem>
</pbs:OrderItemList>
</pbs:Order>
</pbs:OrderList>
```

Updating/Editing an Order

To Update/Edit an Order Line Item you need an OrderID (internal PrintStream Order Id) in the header, an action of "Edit". For the line item, the Action is "Edit" the ReasonCodeDesc for the Line Items, which must be a valid PrintStream Reason Code and the corresponding Line number in PrintStream. To Cancel a Line Number, you need all the above but an Action of "Cancel" instead of "Edit"

```
<?xml version="1.0" encoding="UTF-8" ?>
- <pbs:OrderList xmlns:pbs="www.streamlinesolutions.com/SchemaDef/OrderEntry/BaseSchema">
-   <pbs:Order CustomerID="13" ProjectID="29" CustomerOrderID="MAY30Test1" OrderID="1000001515" Action="Edit" CustomerBatchID="Apr29TB1" NeedDate="
Office Manager Sally Simpson" BillToAttention="Attention: Billing Manager Tommy Walker">
     <pbs:SOType Name="Normal" LookupName="true" />
-   <pbs:OrderBy CustomerRecipientID="KDIP1123f3" CustomerID="13" FirstName="Kathy" LastName="DiPaolo" DayPhone="860-889-1234" Email="dean@pri
     <pbs:Address City="Bozrah" State="CT" Zip="06334" AddressLine1="1107 Main St" AddressLine2="Cnr North and Main">
       <pbs:Country Code="US" LookupName="true" />
     </pbs:Address>
     </pbs:OrderBy>
-   <pbs:ShipTo CustomerRecipientID="KDIP1123f3" CustomerID="13" FirstName="Kathy" LastName="DiPaolo" DayPhone="860-889-1234" Email="dean@prin
     <pbs:Address City="Bozrah" State="CT" Zip="06334" AddressLine1="1107 Main St" AddressLine2="Cnr North and Main">
       <pbs:Country Code="US" LookupName="true" />
     </pbs:Address>
     </pbs:ShipTo>
-   <pbs:BillTo CustomerRecipientID="KDIP1123f3" CustomerID="13" FirstName="Kathy" LastName="DiPaolo" DayPhone="860-889-1234" Email="dean@prints
     <pbs:Address City="Bozrah" State="CT" Zip="06334" AddressLine1="1107 Main St" AddressLine2="Cnr North and Main">
       <pbs:Country Code="US" LookupName="true" />
     </pbs:Address>
     </pbs:BillTo>
-   <pbs:OrderItemList>
-     <pbs:OrderItem Action="Edit" ReasonCodeDesc="Test Order" LineNo="1" Quantity="2" CustomerOrderID="MAY30Test1" UnitPrice="27" SalesTax="3" Ship
       <pbs:Product InventoryCode="315" />
       <pbs:ShippingMode Name="U11" LookupName="true" />
     </pbs:OrderItem>
     </pbs:OrderItemList>
   </pbs:Order>
</pbs:OrderList>
```

J. J. Dean Maintain Reason Codes

New Open Save

Details

Status Type: SO Line Item / Cancel

Reason Name: Test Order

☒ Active

Working with Kits

When you want to update an existing kit item through Web2PS you need to know the internal PrintStream kit id number. You will also need to know a list of the existing internal PrintStream inventory codes. If you want to add a New Kit you just need to make the KitId = 0 as this will create a new kit.

```

<WEB2PS>
<INFO>
  <PLANTCODE>0100</PLANTCODE>
  <CUSTOMER>13</CUSTOMER>
  <ACTION>KITIMPORT</ACTION>
  <SYSDB>D:\Stream\SQL_DB\Generic_14_Multi\dosrun</SYSDB>
  <COMPCODE>01</COMPCODE>
</INFO>
<KITS>
  <Kit KitId="0">
    <CustomerId>13</CustomerId>
    <KitCode>Kit Import Nov 23 #1</KitCode>
    <KitType>Manufacture</KitType>
    <KitName></KitName>
    <KitDesc>New Test Kit from Web2PS Tester Nov 23 #1</KitDesc>
    <KitNotedId>0</KitNotedId>
    <KitRevisionNo>0</KitRevisionNo>
    <CSR>Claude</CSR>
  </Kit>
</KITS>

```

For an update – add the KitId

```
<?xml version="1.0" ?>
<!-- Kit Information Document -->
- <KITS>
- <KIT>
  <KITCODE>Kit Import Nov 23 #2</KITCODE>
  <KITID>509</KITID>
  <ERROR>Success</ERROR>
  - <KITITEM>
    <InventoryCode>314</InventoryCode>
    <InventoryCode>317</InventoryCode>
  </KITITEM>
</KIT>
</KITS>
```

Then run the xml through the "MainInterface" option on the Web2PS ECommerceWeb site as shown here:

- [InventoryUpdateByBatch](#)
- [IsRecipientTaxable](#)
- [Login](#)
- [MainInterface](#) 
- [ProcessDataForOrder](#)
- [SaveNomGroupItems](#)

ECommerceWeb

Click [here](#) for a complete list of operations.

MainInterface

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
strRequest:	

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be re

```
POST /Web2PSWCF.WebService.Setup/ECommerceWeb.asmx HTTP/1.1
Host: 127.0.0.1
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://Printstream.org/MainInterface"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <soap:Header>
    <Authenticate xmlns="http://Printstream.org/">
      <RecipientID>string</RecipientID>
      <Password>string</Password>
      <CompanyCode>string</CompanyCode>
      <PlantCode>string</PlantCode>
      <AccountNo>string</AccountNo>
    </Authenticate>
  </soap:Header>
  <soap:Body>
    <MainInterface xmlns="http://Printstream.org/">
      <strRequest>string</strRequest>
    </MainInterface>
  </soap:Body>
</soap:Envelope>
```

Adding Order Variable Fields

To add an Order Variable field as part of the Web2PS xml, add the following fields.

Note: The Sequential field is the key to identify the Order Variable in the Project List. It's simply based on the sequence as listed in the Project Variable Screen.

```

</pbs:Address>
</pbs:BillTo>
- <pbs:OrderItemList>
- <pbs:OrderItem Quantity="1" CustomerOrderID="June24TST1" PackageNotes="these are my package notes" ItemMetaData="<testxml><something>tada</something></testxml>">
  <pbs:Product InventoryCode="314" ItemID="314" />
  <pbs:ShippingMode Name="U21" LookupName="true" />
  <pbs:OrderItemVariableFields>
    <pbs:VariableField Sequence="6" FieldName="ACCOUNTNUM" Value="AC816" />
  </pbs:OrderItemVariableFields>
</pbs:OrderItem>
</pbs:OrderItemList>
- <pbs:OrderVariableFields>
  <pbs:VariableField Sequence="5" FieldName="ACQUATEXEC" Value="JOE" />
</pbs:OrderVariableFields>
</pbs:Order>
</pbs:OrderList>

```

	CCNAME
>	RETAILERNAME
<	BROKERCODE
>>	BROKERCOMPANY
<<	ACQUATEXEC
	ACCOUNTNUM