



Universal Personalization Services: UPSGetTagOrder web method

A Basic Guide

Table of Contents

- Introduction 2
 - Goal 2
- Key Definitions 2
- Web Method..... 3
 - Method Syntax..... 3
 - Parameters..... 3
 - Return Value 3
 - Assumptions 5
 - CODE Example C# - UPSGetTagOrder 6
- Summary 7

UPS Console

Universal Personalization Services (UPS) supports on-site operations with a versatile solution called UPS Console. This solution provides the manufacturing site to integrate with HP custom integration services. This document illustrates one proposed method to save time and money on operator training at manufacturing sites. The reduced ramp-up eliminates the need for significant operator re-training that would normally ensue from a full-fledged integration of UPS with the factory shop-floor automation system.

Introduction

UPS offers extensive capabilities to support unit personalization in conjunction with support for Custom Integration Services (CIS) by HP in close cooperation with manufacturing sites across the world. A standard approach to supporting CIS services envisages that all factory sites process data in a service-oriented manner that relies on industry standards and best practices for an efficient manufacturing supply chain.

Goal

This document illustrates a web method that will allow site to store the tag information that will be printed later.

Objective

The use of this web method will permit the factory site to send all the Chassis/Box tag info order for the BarTender Fields to be used so they can be printed using other methods such as UPSClient, UPSGetTags or UPSGetATRP.

This web method must be executed separately for Chassis and Box ZWARs as each tag must require different information to be printed based on the Part Number AV specification.

Key Operation

The key operation is:

- Obtain a simple string that will then be provided to UPSUpdatePrintInfo web method to know the order of the data to be printed in the labels.

Important Assumption and Limitation

This document is not the comprehensive reference guide for any UPS activity. You should contact the UPS Development team for more detailed information on UPS. This document is limited to the illustration of the key operation described in the following sections.

Key Definitions

The proper consideration of the following terms will lead to a better understanding of the proposed web method:

- **ZWAR**
BOM Level 5 (Configurable Software), uses 6-3 format. It is the name of the component obtained from PRISM database where the web method will take the **ups.cmd** from.
- **HPPO**
Order reference number that was previously communicated to UPS Site server using a SendBOM method. This web method will be executed preferably once per order to avoid performance issues.
- **Ups.cmd**
Key file to be used, this file will contain the tag order information that UPSUpdatePrintInfo will need to place the sequence of the BarTender fields in the tag label.
- **Part Number AV**
UPS Part Number or SKU used by the HPPO which will contain the label format and information that will be printed (BarTender) as well as other info like asset number to be used, Print Routing, Order specific information and Unit specific information.

Web Method

Method Syntax

```
[WebMethod]
public TagOrderStruct UPSGetTagOrder(string ZWAR)
```

Parameters

The following table lists the parameters for the proposed **UPSGetTagOrder** method:

Parameter	Type	Description
ZWAR	String	6-3 Part Number. If empty or null, method will return code -1. i.e. "431679-00A, 439928-00A, etc..."

Return Value

Type: `TagOrderStruct`

Structure

The following figure illustrates the general response from the proposed **UPSGetTagOrder** method:

```
public struct TagOrderStruct
{
    public string version;
    public int retcode;
    public string message;
    public string datastring;
}
```

Field Values

The following table explains the data values in the response object:

Parameter	Type	Description
version	string	The web service version number.
retcode	int	The return code from the web service in response to the current parameters submitted by the client application. Typical return codes are: <ul style="list-style-type: none"> -1 - failure 0 - success 1 - Command failure 2 - Error trying to delete directory 3 - Error Loading Sub Folders 4 - Error reading file 5 - No UPS.cmd found 6 - No template found 7 - No chassis command found and Box command neither - 8 - Multiple command lines setting the TEMPLATE - 9 - Bad command line Calling UPS_PutTagData 10 - Multiple command lines Calling UPS_PutTagData - Box - 11 - Multiple command lines Calling UPS_PutTagData - Chassis -

		12 - Part number cannot be empty 13 - Error downloading component
message	string	Error or success message.
datastring	String	The return string containing the tag order sequence to be later used in UPSUpdatePrintInfo web method.

Risk

Wrong calls to the proposed **UPSGetTagOrder** method not passing the correct info specified in the Parameter table mentioned in previous section of this document may result in not receiving the expected results, for that reason, Site must configure the web method accordingly to avoid this issue.

Assumptions

This method has the following underlying assumptions:

- The Site uses one of the **SendBOM** and **UPSVerifyPOReady** web methods available before invoking the proposed **UPSGetTagOrder** web method.
- The Site allows sufficient time for UPS to service the range requests.
- It is recommended that Site must execute **UPSGetTagOrder** once per order to avoid performance issues and temporary store data at SFN, but it can be executed once per unit as well.
- Site will be responsible to manage exceptions that can be thrown by web service call.
- Examples provided in this document are exclusively for .NET and C# if site decides to use other programming language or Framework is OK as soon as the values returned are not changed.

The following code snippet illustrates the client side operations to:

- Submit the request to the UPS Site server
- Process the response data

The code snippet does not exhaustively illustrate all the operations needed in the client application for robust handling of shopfloor practices.

CODE Example C# - UPSGetTagOrder

```

public struct TagOrderStruct
{
    public string version;
    public int retcode;
    public string message;
    public string datastring;
}

namespace UPS_ATRPCConsumer
{
    class Program
    {
        static void Main(string[] args)
        {
            UPS_ATRPCConsumer localhost.ATMStruct retval = new
            UPS_ATRPCConsumer.localhost.ATMStruct();
            UPS_ATRPCConsumer.localhost.ATRP svc = new UPS_ATRPCConsumer.localhost.ATRP();

            /*** site must use its own URL and not necessarily the one illustrated below ***/
            svc.Url = "http://localhost/UPSATRP/UPS_ATRP.asmx";

            /*** test data; the values for your site may be different ***/
            string zwar = "431679-00A";
            int retcode = 1;
            string version = String.Empty;
            string message = String.Empty;
            string datastring = String.Empty;

            retval = svc.UPSTagOrder(wzar);
            if (retval.retcode == 0)
            {
                version = retval.version;
                message = retval.message;
                datastring = retval.datastring;
                /*** site should continue their normal workflow after success ***/
            }
            else
            {
                // error condition - read retval.message
            }
        }
    }
}

```

Enhancement

The feature to export the assigned range of asset numbers for external transmission is under consideration. Further scoping discussions with the manufacturing sites will provide the basis for a more efficient solution.

Summary

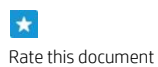
This document provides a brief description on the use of a proposed feature in the UPS solution for manufacturing sites – Retrieve Asset Number list. This feature will permit manufacturing sites to ramp up quickly on behalf of HP to deliver asset tag services to global customers. The UPS team cannot guarantee the delivery of any proposed feature without further consultation with the business teams responsible for delivery of the services. The manufacturing sites will have to integrate their current shop floor system to this new feature. There are many additional features in the UPS solution that can assist the manufacturer to improve its TAT and Quality. Analysis of these features and their corresponding integration into the shop floor computer application can occur after the initial adoption of UPS at the manufacturing site.

Resources, contacts, or additional links

You can find more information at
hp.com

Learn more at
hp.com

Sign up for updates
hp.com/go/getupdated



© Copyright 2013 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Trademark acknowledgments, if needed.

4AA4-123xxxENW, April 2013

