



Universal Personalization Services: UPSGetATM 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 4
- CODE Example C# - Simplified Version (If using UPSGetRange and Site is responsible to assign asset tag) 5
- CODE Example C# - Extended Version (If NOT using UPSGetRange) 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 to store the asset number and link it with the Serial Number of the Unit scanned in Shop floor Control System and provides how it should be executed.

Objective

The use of this web method will permit the factory site to store the asset tag number they want to assign to the Serial Number of the UUT being processed, so, it will be stored in the UPS database, also, the web method can be used, so UPS will get the next available asset tag number.

In case Site uses **simplified** version, it will perform audit tasks to check duplicates as well as have full tracking of the asset tags assigned in order to make troubleshooting easier.

If using **extended** version (UPS in charge to get the next available asset tag number and give it back to site).

Key Operation

The key operation is:

- Update UPS Site database to link the asset tag number and the Serial Number used, and, if requested by site, have the possibility to have UPS get the next available asset number for them instead of having to do that themselves.

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:

- **Asset Tag Number**
A Number (which most commonly consist in a PREFIX + Sequential Number + SUFFIX) that uniquely identifies an asset in UPS for a specific customer.
- **Serial Number**
A string that defines the bar code scanned value from the UUT being processed.
- **HPP0**
Order reference number that was previously communicated to UPS Site server using a SendBOM method.

Web Method

Method Syntax

```
[WebMethod]
public ATMStruct UPSGetATM(string SerialNumber, string PartNumber, string
AssetNumber, string HPPO, string AssetTagNumber)
```

Parameters

The following table lists the parameters for the proposed UPSGetATM method:

Parameter	Type	Description
SerialNumber	string	UUT Serial Number. If empty or null, method will return code -1. i.e. "SHG1234567, CZC4362L1Z, etc..."
PartNumber	String	UPS Part Number AV Setup. If empty or null, method will return code -1. i.e. "AY123AV"
AssetNumber	string	Name of the asset number as defined by CS Engineering. If empty or null, method will return code -1. i.e. "AY123AV" (It is usually named the same as the UPS Part Number AV)
HPPO	string	Order reference number that was previously communicated to UPS Site server using a SendBOM method. If empty or null, method will return code -1. i.e. "123456-001, 475869362, etc..."
AssetTagNumber	string	This is a key parameter , it is the Asset tag number assigned to Serial Number. If parameter is NULL , it means that UPS will be in charge of assigning this number from the available range . i.e. "PRE00001SUF" or NULL

Return Value

Type: ATMStruct

Structure

The following figure illustrates the general response from the proposed UPSGetATM method:

```
public struct ATMStruct
{
    public string version;
    public int retcode;
    public string message;
    public string assetTagNum;
}
```

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: 0 = success -1 = failure 1 = success – But, means the Serial Number was already processed.
message	string	Error or success message

assetTagNum	String	The return string containing the asset tag number that UPS assigned to Serial Number at the Site UPS DB.
--------------------	--------	--

Risk

Wrong calls to the proposed `UPSGetATM` method not passing the correct info specified in the Parameter table mentioned in previous section of this document may result on not getting the expected results, for that reason, Site must configure the web method accordingly in case they want to use.

Assumptions

This method has the following underlying assumptions:

- The Site uses one of the `SendBOM` web methods available in the UPS Primary Interface (PUPS) to notify UPS of the order previously before invoking the proposed `UPSGetATM` method.
- The Site allows sufficient time for UPS to service the range requests.
- The Site uses the same HPP0 value for the invocation of the chosen `SendBOM` method first and the proposed `UPSGetATM` method subsequently.
- 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.
- *These steps occur in case Site wants to use `UPSGetRange` web method.*
 - The Site has the responsibility to generate the data for the asset tag string for printing.
 - The Site must use the asset number mask with the range of numbers returned by the proposed `UPSGetRange` method to generate the customer asset number for the corresponding asset tag field. This step is in case Site wants to use `UPSGetRange` web method.
 - Site will use their own printing solution and this method will serve as a quality assurance and troubleshooting.
- *These steps occur in case Site **DO NOT** want to use `UPSGetRange` web method.*
 - UPS will have the responsibility to generate the new Asset Tag number and assign it to the Serial Number scanned.
 - This web method **DOES NOT** save tag data so it can be later used by `UPSClnt` or `UPSGetATRP`, a new web method called `UPSUpdatePrintInfo` will be in charge of storing the tag information so it can be printed.

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# - Simplified Version (If using UPSGetRange and Site is responsible to assign asset tag)

```

public struct ATMStruct
{
    public string version;
    public int retcode;
    public string message;
    public string assetTagNum;
}

namespace UPS_ATRPCConsumer
{
    class Program
    {
        static void Main(string[] args)
        {
            UPS_ATRPCConsumer localhost = new UPS_ATRPCConsumer();
            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 serialNo = "SHG1234567";
            string partNumber = "AY115AV";
            string assetNumber = "AY115AV";
            string hppo = "TestHPP0";
            string assetTagNo = "PRE00001SUF";

            int retcode = 1;
            string version = String.Empty;
            string message = String.Empty;
            string assetTagNumber = String.Empty;

            retval = svc.UPSGetATM(serialNo, partNumber, assetNumber, hppo, assetTagNo);
            if (retval.retcode == 0)
            {
                version = retval.version;
                message = retval.message;
                assetTagNumber = retval.assetTagNum;
                /*** site should continue their normal workflow after success ***/
            }
            else
            {
                // error condition - read retval.message
            }
        }
    }
}

```

CODE Example C# - Extended Version (If NOT using UPSGetRange)

```

public struct ATMStruct
{
    public string version;
    public int retcode;
    public string message;
    public string assetTagNum;
}

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 serialNo = "SHG1234567";
            string partNumber = "AY115AV";
            string assetNumber = "AY115AV";
            string hppo = "TestHPPPO";
            string assetTagNo = null;

            int retcode = 1;
            string version = String.Empty;
            string message = String.Empty;

            retval = svc.UPSGetATM(serialNo, partNumber, assetNumber, hppo, assetTagNo);
            // Data stored and ready to be printed in UPS Site DB
            if (retval.retcode == 0)
            {
                version = retval.version;
                message = retval.message;
                assetTagNumber = retval.assetTagNum;
                **** 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

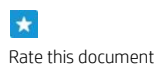
This document provides a brief description on the use of a proposed feature in the UPS solution for manufacturing sites – Retrieve Asset Number list. This feature will permit manufacturing sites to ramp up quickly on behalf of HP to deliver asset tag services to global customers. The UPS team cannot 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

