



Universal Personalization Services: UPSGetUSI 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#	5
Summary	6

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 Serial Number of the Unit scanned in Shop floor Control System in the Site UPS Database and assign it to the next available USI (Unit Specific Information), as well as generate the unattend.xml file the site will use for Unit personalization.

Objective

The usage of this web method will permit the factory site to perform Unit personalization tasks to the UUT being processed, and also, it will be stored in the Site UPS database for quality/metric/troubleshooting purposes.

Key Operation

The key operation is:

- Update UPS Site database to link the USI info and the Serial Number used, as well as generate the unattend.xml used by sites to make the Unit Personalization.

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:

- **Serial Number**

A string that defines the bar code scanned value from the UUT being processed.

- **HPPO**

Order reference number that was previously communicated to UPS Site server using a SendBOM method.

- **Part Number**

UPS Part Number AV used previously on a SendBOM method to let UPS Site server know there was a USI request from Site System.

Web Method

Method Syntax

```
[WebMethod]
public USIStruct UPSGetUSI(string SerialNumber, string PartNumber, string HPPO,
string FGPartNum)
```

Parameters

The following table lists the parameters for the proposed `UPSGetUSI` 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"
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..."
FGPartNum	String	Material Number SKU or Factory Localization number that will allow factories to split a same order number into many as needed, NULL must be passed if this parameter is not needed. i.e. "B3V51EC#ABA, A3N84EC#B1L, etc..."

Return Value

Type: `USIStruct`

Structure

The following figure illustrates the general response from the proposed `UPSGetUSI` method:

```
public struct USIStruct
{
    public string version;
    public int retcode;
    public string message;
    public string unattend;
}
```

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, Serial Number already assigned to USI, will retrieve the unattend.xml back again.
message	string	Error message (if applicable)
unattend	string	The return string which will contain (in XML format) the unattend.xml file to be used for Unit personalization.

Risk

Wrong calls to the proposed `UPSGetUSI` 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 - or not – the UPS printing solutions (UPS Client or `UPSGetATRP`) or their own print solution.

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 `UPSGetUSI` method.
- The Site allows sufficient time for UPS to service the USI range requests.
- The Site uses the same HPPD value for the invocation of the chosen `SendBOM` method first and the proposed `UPSGetUSI` method subsequently.
- UPS will have the responsibility to generate the new `unattend.xml` file and assign the USI returned to the Serial Number scanned.
- UPS will store the Serial Number and USI in the UPS DB.
- Site must create a web service consumer that will get the `unattend.xml` back from UPS, but, the rest of the OS injection process of this XML file will be Site responsibility.
- 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 `UPSGetUSI` 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#

```

public struct USIStruct
{
    public string version;
    public int retcode;
    public string message;
    public string unattend;
}

namespace UPS_ATRPCConsumer
{
    class Program
    {
        static void Main(string[] args)
        {
            UPS_ATRPCConsumer localhost.USIStruct retval = new
UPS_ATRPCConsumer.localhost.USIStruct();
            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 hppo = "TestHPP0";
            string partNumber = "AY115AV";
            string fgPartNumber = "A3N84EC#B1L";

            int retcode = 1;
            string version = String.Empty;
            string message = String.Empty;
            string unattendStr = String.Empty;
            XmlDocument doc = new XmlDocument();

            retval = atrp.UPSGetUSI(serialNo, hppo, partNumber, fgPartNumber);
            if (retval.retcode == 0)
            {
                version = retval.version;
                message = retval.message;
                unattendStr = retval.unattend;
                doc.LoadXml(unattendStr);

                doc.Save("C:\\unattend.xml");
                /*** site should continue their normal workflow after success ***/
            }
            else
            {
                // error condition - read retval.message
            }
        }
    }
}

```

Summary

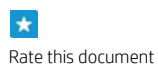
This document provides a brief description on the use of a proposed feature in the UPS solution for manufacturing sites – Retrieve Asset Number list. This feature will permit manufacturing sites to ramp up quickly on behalf of HP to deliver asset tag services to global customers. The UPS team cannot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

