Example of a Catalog Interface

The following excerpts are examples of the HTML and XML source code generated by a catalog engine after a user has finished selecting items for a shopping cart. You can extract the source code and display the data in a Web browser.

**HTML version**

In the HTML version, all the line items are placed as name/value pairs in an HTML form as shown below. This can be done as text or hidden fields. The parameter `HOOK_URL` has to be placed in the action parameter of that form, so that the HTML form can be sent to the SAP Business-to-Business Procurement system.

**HTML Form (interface.asp):**

```html
<HTML>
<HEAD>
</HEAD>
<H1>SAP Open Catalog Interface: HTML - Example</H1>
</HEAD>

<P>This is an example, what needs to be generated by the catalog engine, if the user is done with picking items in his shopping basket.</P>

<P>View the source of this file to see all the text inputs.</P>

<FORM action="<%=Request.QueryString("HOOK_URL")%>" method=post target=_top>
<!--For more information about the interface fields, refer to the catalog interface description-->
<!--Item number 1-->
<input type="hidden" name="NEW_ITEM-DESCRIPTION[1]" value = "Pen, Easytouch, Medium, Black">
<input type="hidden" name="NEW_ITEM-MATNR[1]" value = ">
<input type="hidden" name="NEW_ITEM-MATGROUP[1]" value = ">
<input type="hidden" name="NEW_ITEM-QUANTITY[1]" value = "1">
<input type="hidden" name="NEW_ITEM-UNIT[1]" value = "EA">
<input type="hidden" name="NEW_ITEM-PRICE[1]" value = "0.50">
<input type="hidden" name="NEW_ITEM-PRICEUNIT[1]" value = "5">
<input type="hidden" name="NEW_ITEM-CURRENCY[1]" value = "USD">
<input type="hidden" name="NEW_ITEM-LEADTIME[1]" value = "1">
<input type="hidden" name="NEW_ITEM-VENDOR[1]" value = "1768">
<input type="hidden" name="NEW_ITEM-VENDORMAT[1]" value = "648570">
<input type="hidden" name="NEW_ITEM-MANUFACTCODE[1]" value = "4711">
<input type="hidden" name="NEW_ITEM-MANUFACTMAT[1]" value = "4712">
<input type="hidden" name="NEW_ITEM-CONTRACT[1]" value = "111">
<input type="hidden" name="NEW_ITEM-CONTRACT_ITEM[1]" value = "1111">
<input type="hidden" name="NEW_ITEM-SERVICE[1]" value = "">
<input type="hidden" name="NEW_ITEM-EXT_QUOTE_ID[1]" value = "111">
<input type="hidden" name="NEW_ITEM-EXT_QUOTE_ITEM[1]" value = "11111">
<input type="hidden" name="NEW_ITEM-EXT_PRODUCT_ID[1]" value = "11111">
<input type="hidden" name="NEW_ITEM-LONGTEXT_1:132[1]" value = "longtext_1: The best pen in our program, lightweight and easy to handle">
<input type="hidden" name="NEW_ITEM-CUST_FIELD1[1]" value = "custf 1.1">
```
XML version

In the XML version, the data from a shopping cart can be generated into a separate file which is then linked to the HTML form. The XML data is expected to be in the value of an input field with the name `~xmlDocument`. It can be set there dynamically using the function `SAP_encode_b64(Str)`, as shown in the example below. In the input field `~xmlType`, you have to specify the type of schema used to generate the XML data. Possible values for this field are: ESAPO...
(Encoded SAP Object) and ERNO (Encoded RosettaNet Object). To view the SAP XML schema refer to the file pdi_oci.xsd in SAPNet. To access this files, enter the URL http://www.sap.com/, specifying the alias CSP/scenarios/ and choose the menu options SAP B2B Procurement ® B2B-OCI.

**HTML Form (interface.asp)**

```html
<HTML>
<script language="JavaScript1.2">
function SAP_encode_b64(Str) {
    var encStr = "";
    var base64 = [
        'A','B','C','D','E','F','G','H','I','J','K','L','M',
        'N','O','P','Q','R','S','T','U','V','W','X','Y','Z',
        'a','b','c','d','e','f','g','h','i','j','k','l','m',
        'n','o','p','q','r','s','t','u','v','w','x','y','z',
        '0','1','2','3','4','5','6','7','8','9','+','/ 
    ];
    for (var i = 0; i < Str.length; i += 3) {
        encStr += base64[(Str.charCodeAt(i) >>> 2)];
        if(!Str.charAt(i+1)) {encStr += '=='; break;}
        encStr += base64[((Str.charCodeAt(i) & 0x03) << 4) | Str.charCodeAt(i+1) >>> 4];
        if(!Str.charAt(i+2)) {encStr += '='; break;}
        encStr += base64[((Str.charCodeAt(i+1) & 0x0F) << 2) | Str.charCodeAt(i+2) >>> 6];
        encStr += base64[(Str.charCodeAt(i+2) & 0x3F)];
    }
    return encStr;
}
</script>

<HEAD>
<H1>SAP Open Catalog Interface: XML - Example</H1>
</HEAD>

<P>This is an example, what needs to be generated by the catalog engine, if the user is done with picking items in his shopping basket.</P>

<P>View the source of the file 'example1.xml' to see all the items.</P>

<!-- insert the order -->

<BODY bgcolor="#DED6C2">

<XML ID=xmlid src="example1.xml"></XML>

<!-- Transfer the order to the B2B Application -->

<form action="<%=Request.QueryString("HOOK_URL")%>" method="post" name="OrderForm" onsubmit="OrderForm['~xmlDocument'].value = SAP_encode_b64(xmlid.xml)"

<input type="hidden" name="~xmlDocument" value=""/>
<input type="hidden" name="~xml_type" value="ESAP0"/>
</form>
</BODY>

</HTML>
```
Example of a Catalog Interface (SAP Library - Open Catalog Interface)

http://help.sap.com/saphelp_crm20c/helpdata/en/30/67483936dd7607e1...
The award-winning Palm Computing organizers, designed as companion products to personal computers, enable mobile users to manage their schedules, contacts and other critical personal and business information on their desktops and remotely. Palm Computing organizers automatically synchronize their information with a personal computer locally or over a local or wide area network at the touch of a button. Their most distinguishing features include shirt-pocket size, instant response, an elegant graphical user interface and an innovative desktop docking cradle which facilitates two-way synchronization between the PC and organizer.
The award-winning Palm Computing organizers, designed as companion products to personal computers, enable mobile users to manage their schedules, contacts, and other critical personal and business information on their desktops and remotely. Palm Computing organizers automatically synchronize their information with a personal computer locally or over a local or wide area network at the touch of a button. Their most distinguishing features include shirt-pocket size, instant response, an elegant graphical user interface and an innovative desktop docking cradle which facilitates two-way synchronization between the PC and organizer.