



EXPO LOGIC
Exposition Solutions Made Simple

Confidential and Proprietary

badge[on]demand™

Personify Integration Setup

Last updated: September 14, 2016

Overview

Expo Logic's **badge[on]demand™** seamlessly integrates with the Personify AMS for badge and ticket printing. Enabling the integration requires the following steps in your Personify instance:

- Installing a database view
- Generating and publishing a web service for the view using the Personify Web Services Designer
- Updating a baseline Personify web service (OrderDetailBadges)
- Adding a taskbar link in Personify

We then need from you:

- The oData endpoint URI for the above UserDefinedExpologicBadgeViews resource.

e.g. *[your domain]/PersonifyDataServicesCustom/PersonifyDataR1.svc/
UserDefinedExpologicBadgeViews*

- The oData endpoint URI for retrieving meeting information from the baseline Personify WebProductViews resource.

e.g. *[your domain]/PersonifyDataServices/PersonifyData.svc/WebProductViews*

- The oData endpoint URI for updating badge print dates and counts via the updated OrderDetailBadges resource.

e.g. *[your domain]/PersonifyDataServicesCustom/PersonifyDataR1.svc/OrderDetailBadges*

- Credentials for access to the above oData resources.

Technical Requirements

The badge[on]demand integration with Personify requires:

- Personify360 version 7.4.0 SP1 or greater.
- PDS 1.4.0 or greater.

Special Note

When using Personify360 7.5+, any fields that will be used on badges produced by badge[on]demand should be included in a Personify badge configuration (even though Personify's badge configuration is not used by badge[on]demand). Otherwise those fields will not be available for editing in Personify360 during the event.

Installing the Database View

Below is the DDL for the view. Once installed, you may revise it to better suit your needs. However, any modified version of the view must meet the following requirements:

- All columns shown in the SELECT clause below must still be present. If a column is not relevant to your use of Personify, please alias an appropriate static value.
- Only one record should be returned for the primary registrant badge (BADGE_TYPE_CODE <> 'GUEST' and BADGE_CONFIGURATION_CODE <> 'SESSION_TICKET' or 'TICKET')
- There may be one or more records returned for guest badges (BADGE_TYPE_CODE = 'GUEST' and BADGE_CONFIGURATION_CODE <> 'SESSION_TICKET' or 'TICKET')
- There may be one or more records returned for tickets (BADGE_CONFIGURATION_CODE = 'SESSION_TICKET' or 'TICKET')

```
CREATE VIEW DBO.USR_EXPOLOGIC_BADGE_VW AS
```

```
SELECT
```

```
    OD.ORDER_NO,  
    OD.ORDER_LINE_NO,  
    OD.SHIP_MASTER_CUSTOMER_ID,  
    OD.SHIP_SUB_CUSTOMER_ID,  
    OD.PARENT_PRODUCT,  
    OD.PRODUCT_CODE,  
    OD.PRODUCT_ID,  
    P.LONG_NAME,  
    MP.START_DATE,  
    MP.END_DATE,  
    OD.ORDER_QTY,  
    ODB.BADGE_SEQ,  
    ODB.BADGE_TYPE_CODE,  
    ODB.NICKNAME,  
    ODB.FIRST_NAME,  
    ODB.LAST_NAME,  
    ODB.FULL_NAME,  
    ODB.PRIMARY_JOB_TITLE,  
    ODB.COMPANY_NAME,  
    ODB.CITY,  
    ODB.STATE,  
    ODB.POSTAL_CODE,  
    ODB.COUNTRY_CODE,  
    AC.COUNTRY_DESCR,
```

```

ODB.MTG_BADGE_CONFIG_MASTER_ID,
MBCM.BADGE_CONFIGURATION_CODE,
OD.RATE_STRUCTURE,
ODB.RATE_CODE,
RC.DESCR AS RATE_CODE_DESCR,
MFR.DESCRPTION AS LOCATION,
A.ADDRESS_1,
A.ADDRESS_2,
A.ADDRESS_3,
A.ADDRESS_4,
I.PRIMARY_EMAIL_ADDRESS,
I.PRIMARY_PHONE,
I.PRIMARY_FAX,
SUM(B.ACTUAL_BALANCE_AMOUNT) AS BALANCE,
'' as DISCLAIMER,
'' as ANNOTATION,
'' as RIBBONS,
'' as NO_PRINT_MSG,
'' as DESCRIPTION,
'' as custom01,
'' as custom02,
'' as custom03,
'' as custom04,
'' as custom05,
'' as custom06,
'' as custom07,
'' as custom08,
'' as custom09,
'' as custom10
FROM
DBO.ORDER_DETAIL OD (NOLOCK)
LEFT OUTER JOIN DBO.ORDER_DETAIL_BADGE ODB (NOLOCK) ON
    OD.ORDER_NO=ODB.ORDER_NO AND
    OD.ORDER_LINE_NO=ODB.ORDER_LINE_NO
JOIN DBO.CUSTOMER I (NOLOCK) ON
    I.MASTER_CUSTOMER_ID=OD.SHIP_MASTER_CUSTOMER_ID AND
    I.SUB_CUSTOMER_ID=OD.SHIP_SUB_CUSTOMER_ID
JOIN DBO.FAR_BALANCE_VW B (NOLOCK) ON
    OD.ORDER_NO=B.ORDER_NO AND
    OD.ORDER_LINE_NO=B.ORDER_LINE_NO
JOIN DBO.CUS_ADDRESS_VW A (NOLOCK) ON
    OD.SHIP_MASTER_CUSTOMER_ID=A.MASTER_CUSTOMER_ID AND
    OD.SHIP_SUB_CUSTOMER_ID=A.SUB_CUSTOMER_ID AND
    OD.SHIP_ADDRESS_ID=A.CUS_ADDRESS_ID AND
    OD.SHIP_ADDRESS_TYPE_CODE=A.ADDRESS_TYPE_CODE
JOIN DBO.APP_COUNTRY AC ON
    AC.COUNTRY_CODE=ODB.COUNTRY_CODE
JOIN DBO.PRODUCT_RATE_CODE RC ON
    OD.PRODUCT_ID=RC.PRODUCT_ID AND
    OD.RATE_STRUCTURE=RC.RATE_STRUCTURE AND
    OD.RATE_CODE=RC.RATE_CODE
JOIN DBO.MTG_BADGE_CONFIG_MASTER MBCM ON
    MBCM.MTG_BADGE_CONFIG_MASTER_ID=ODB.MTG_BADGE_CONFIG_MASTER_ID

```

```

JOIN DBO.PRODUCT P ON
    P.PRODUCT_ID=OD.PRODUCT_ID
JOIN DBO.MTG_PRODUCT MP ON
    MP.PRODUCT_ID=OD.PRODUCT_ID
LEFT OUTER JOIN MTG_ROOM_SCHEDULE_MASTER MRSM ON
    MRSM.PRODUCT_ID=P.PRODUCT_ID
LEFT OUTER JOIN MTG_FACILITY_ROOM MFR ON
    MFR.FACILITY_MAST_CUST=MRSM.FACILITY_MAST_CUST AND
    MFR.FACILITY_ROOM_ID=MRSM.FACILITY_ROOM_ID

WHERE
    OD.LINE_STATUS_CODE != 'C'
GROUP BY
    OD.ORDER_NO,
    OD.ORDER_LINE_NO,
    OD.SHIP_MASTER_CUSTOMER_ID,
    OD.SHIP_SUB_CUSTOMER_ID,
    OD.PARENT_PRODUCT,
    OD.PRODUCT_CODE,
    OD.PRODUCT_ID,
    P.LONG_NAME,
    MP.START_DATE,
    MP.END_DATE,
    OD.ORDER_QTY,
    ODB.BADGE_SEQ,
    ODB.BADGE_TYPE_CODE,
    ODB.NICKNAME,
    ODB.FIRST_NAME,
    ODB.LAST_NAME,
    ODB.FULL_NAME,
    ODB.PRIMARY_JOB_TITLE,
    ODB.COMPANY_NAME,
    ODB.CITY,
    ODB.STATE,
    ODB.POSTAL_CODE,
    ODB.COUNTRY_CODE,
    AC.COUNTRY_DESCR,
    ODB.MTG_BADGE_CONFIG_MASTER_ID,
    MBCM.BADGE_CONFIGURATION_CODE,
    MBCM.BADGE_DESCR,
    OD.RATE_STRUCTURE,
    ODB.RATE_CODE,
    RC.DESCR,
    MFR.DESCRPTION,
    A.ADDRESS_1,
    A.ADDRESS_2,
    A.ADDRESS_3,
    A.ADDRESS_4,
    I.PRIMARY_EMAIL_ADDRESS,
    I.PRIMARY_PHONE,
    I.PRIMARY_FAX

```

Generating and Publishing the Web Service

You may watch a video of how to use the Personify Web Services Designer (WSD) to generate and publish the web service for the above view at <http://www.personifycorp.com/presents/personify-nation/pn-episode-120-personify360-data-services>

Note that as long as the view output columns don't change, the view can be edited without re-publishing the web service.

Updating a Baseline Personify Web Service

The badge[on]demand integration uses the baseline Personify OrderDetailBadges entity to update badge print dates and counts in Personify. The process for exposing that entity as a web service varies by PDS version.

PDS 1.4.0

Create a read/write OrderDetailBadge entity from the existing OrderDetailBadge API property and then generate and publish it as a web service. This will automatically include the print count/date fields we need.

PDS 1.5.0

Two columns need to be added to the baseline OrderDetailBadges web service: PrintDate and PrintCount. Using the Personify WSD, drag-and-drop these two columns into the web service and then regenerate and publish the web service.

Installing the badge[on]demand Taskbar Link

Here is the code behind for the taskbar link; no screen DLL or other new components are required. Please note that we will separately provide your client_code and client_key values for entry in the indicated locations near the top of the code.

```
<CommandInfo("QuickPrintExpoLogicBadge", "Quick Print Expo Logic Badge")> _
    Public Function QuickPrintExpoLogicBadge(ByVal sender As Object, ByVal e As
CommandEventArgs) As Object

        Dim clientCode As String = "[client code provided by Expo Logic]"
        Dim clientKey As String = "[client key provided by Expo Logic]"

        Dim gridLineNumber As Integer = 0
        'Bades Change - Making it available for Master Product
        Dim oDetail As TIMSS.API.OrderInfo.IOrderDetail
        If (HasChangeBeenMade()) Then
            Dim oSaveResult As TIMSS.ICommandResult
            If ((dgOrderDetail.Grid.ActiveRow IsNot Nothing) AndAlso
(dgOrderDetail.Grid.ActiveRow.Band.Columns.Exists("OrderLineNumber"))) Then
                gridLineNumber =
dgOrderDetail.Grid.ActiveRow.Cells("OrderLineNumber").Value
            End If
        End If
```

```

        _boolSave = False
        oSaveResult = TIMSS.Win.CommandProcessor.Exec(TheForm,
"cmd://Form/ExecuteSave")
        If Not (_boolSave) Then
            Exit Function
        End If
    End If

    If Not dgOrderDetail.Grid.ActiveRow Is Nothing Then
        If
(dgOrderDetail.Grid.ActiveRow.Band.Columns.Exists("OrderLineNumber"))
Then
            If Not (Context Is Nothing) Then
                Dim oMaster As TIMSS.API.OrderInfo.IOrderMaster
                oMaster = Context
                If (gridLineNumber = 0) Then
                    gridLineNumber =
dgOrderDetail.Grid.ActiveRow.Cells("OrderLineNumber").Value
                End If
                'Badges Change - Making it available for Master Product
                oDetail = oMaster.Details.FindObject("OrderLineNumber",
gridLineNumber)
                If (oDetail.Subsystem = "MTG" AndAlso
oDetail.Product.MasterProductFlag _
                    AndAlso (oDetail.ProductTypeCode.Code = "M" OrElse
oDetail.ProductTypeCode.Code = "S")) _
                    OrElse _
                    (oDetail.Subsystem = "XBT" AndAlso
(oDetail.ProductTypeCode.Code = "BOOTH" OrElse oDetail.ProductTypeCode.Code = "SHARE"))
                _
                    OrElse _
                    (oDetail.Subsystem = "FAC") Then
                        TIMSS.Win.CommandProcessor.Exec(TheForm,
String.Concat("http://teampro.badgeondemand.com/print/index.cfm?clientCode=",
clientCode, "&client_key=", clientKey, "&evt_key=", oDetail.ProductId, "&reg_key=",
oMaster.OrderNumber))
                Else
                    MessageBox.Show("Badges not available for
this Product. ", _
                        "Badge Information", MessageBoxButtons.OK, _
                        MessageBoxIcon.Information,
MessageBoxDefaultButton.Button1)
                End If
            End If
        End If
    End Function

```