

Accelerator – Request Update Hook v81

User Guide

Contents

1.0	Introduction.....	3
2.0	Installation.....	3
2.1	Prerequisites.....	3
2.2	Adapter Installation.....	3
3.0	Configuration	4
3.1	Adapter configuration	4
3.2	Edit the xml configuration file	5
3.3	Create Request Update Hook Adapter	7
3.4	Create Field Value changed by user hook	8
3.5	Create Fulfilment Hook	9
3.6	Create Request Event Hook (Front Office Version 8.4+)	10

1.0 Introduction

The Front Office Request Update Hook Accelerator can be configured to populate request fields from any Microsoft SQL Server data source.

This version supports Front Office version 8.1+

2.0 Installation

2.1 Prerequisites

- Front Office v8.1+
- Microsoft .NET (Version 4.5)
- IIS 7+

2.2 Adapter Installation

- log in on the target installation server with administrator privileges
- Ensure that "IIS Metabase and IIS 6 configuration compatibility" is enabled (windows features under: IIS/Web Management Tools/IIS6 Management Compatibility)
- **Un-install any old versions**
- Extract the files from the zip file into a temporary folder
- Using Windows explorer open the temporary folder, then run setup.exe as an administrator
- Click the next button
- Set the virtual directory (default AcceleratorRequestUpdateHook81)
- Set the target application pool (must support ASP.NET v4).
- Click the next button, then again to start the installation
- Grant the NETWORK SERVICE account (or account that the selected application pool runs under) "Full Control" rights to: \inetpub\wwwroot\<selected virtual directory>\logs

3.0 Configuration

3.1 Adapter configuration

On the server where the adapter has been installed, edit the adapters web.config (for default installations this will be in inetpub\wwwroot\ AcceleratorRequestUpdateHook81) and update the following settings:

- DapiURL – Should be set to the url of your Front Office installations Public Web Service (this can be found on the Admin/Configuration Check window)
- DapiUser & DapiPassword – set these if your Front Office Api has been secured with basic authentication.

If required the appSettings section can be encrypted by following the instructions here:

[http://msdn.microsoft.com/en-us/library/vstudio/zhhddkxy\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/vstudio/zhhddkxy(v=vs.100).aspx)

3.2 Edit the xml configuration file

Using an XML editor open up the configuration file found in \inetpub\wwwroot\<selected virtual directory>\xml\ruhconfig.xml.

Repeat (and edit) the following block of xml for each request update\field change\fulfilment hook required:

```
<RUHConfig>
  <ClientID>GETUSERDATA</ClientID>
  <SQLConnectionString>Data Source=localhost;Initial
Catalog=FrontOffice81;Integrated Security=False;User
Id=webdirecta;Password=directaweb</SQLConnectionString>
  <Header>
    <SQLStatement>Select UserName as DISUSER from users where
UserID='{n:bXML/n:Request/n:RequestHeader/n:RequestedByUser/n:ID}'</SQLState
ment>
    <XPathMaps>
      <XPathMapping>

<XPathStatement>n:bXML/n:Request/n:RequestHeader/n:RequestedByUser/n:ID</XPa
thStatement>
      <RequestFieldCode>REQBYUSR</RequestFieldCode>
    </XPathMapping>
  </XPathMaps>
</Header>
<Line>
  <SQLStatement>Select PartDescription AS RF00000011 from
price_list_text where
partnumber='{1:RequestItem/1:PartNumber}'</SQLStatement>
  <XPathMaps>
    <XPathMapping>

<XPathStatement>n:bXML/n:Request/n:RequestHeader/n:RequestedByUser/n:ID</XPa
thStatement>
    <RequestFieldCode>REQBYUSR</RequestFieldCode>
  </XPathMapping>
</XPathMaps>
</Line>
</RUHConfig>
```

Element	Description
<ClientID>	If using for a request update hook, this must match the Client ID entered into the adapter setting (see 3.2.1). If using with a field change hook this must match the parameter entered against the field that is used to trigger the hook.
<SQLConnectionString>	The SQL connection string (.net format)
<Header>	Container for requester header configuration items
<SQLStatement>	The SQL statement to run (see below)
<XPathMapping>	Repeat this element for each xpath statement required
<XPathStatment>	XPath statement to run over the request xml (see below)
<RequestFieldCode>	Code of the request field to populate with the xpath query result
<Line>	Container for requester line configuration items
<SQLStatement>	The SQL statement to run (see below)
<XPathMapping>	Repeat this element for each xpath statement required, run for each line item on request
<XPathStatment>	XPath statement to run over the request xml (see below)
<RequestFieldCode>	Code of the request field to populate with the xpath query result

The SQL statements can contain xpath statements that run over the xml that is passed to the request update hook by Front Office (xsd schema is here: \inetpub\wwwroot\<selected virtual directory>\schemas\Request_v4.xsd), for example:

```
Select UserName as USRNAME, EMailAddress as EMAILADD, AccessProfileID as ACCESSPRO,
ApprovalLevel as APPLVL from users where
UserID=' [n:bxML/n:Request/n:RequestHeader/n:RequestedByUser/n:ID] '
```

Values after the SQL “AS” statements are the request field codes that will be updated with the SQL field value.

For line fields the SQL statement defined in the configuration is run for each line item in the request. Line item data can be accessed by using an l: prefix in the xpath, e.g:

Select PartDescription AS RF00000011 from price_list_text where partnumber='{1:RequestItem/1:PartNumber}'

Note that header data can also be referenced at line level using the n: prefix

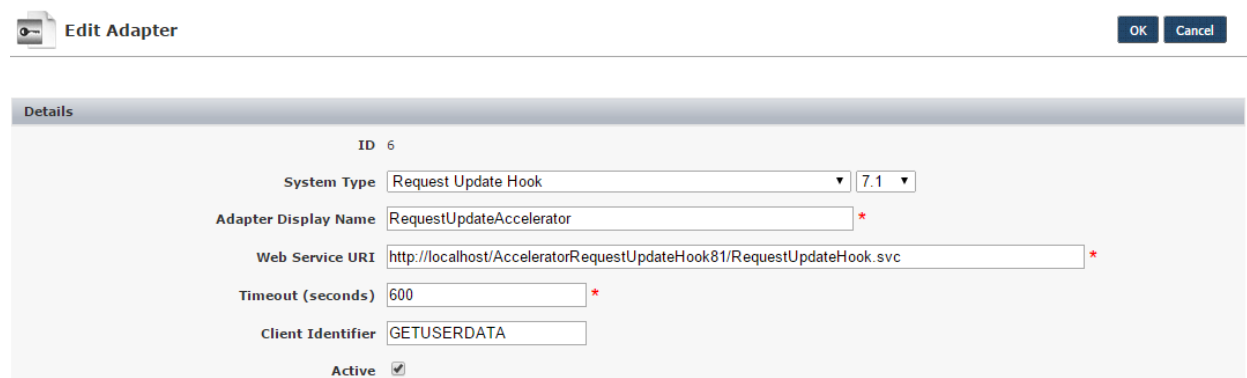
In this version single select external list boxes are now supported, use an xpath statement like this to retrieve the list box data, adding the name of the column to be used in the SQL statement at the end of the xpath with a pipe (|) delimiter:

```
{n:bxXML/n:Request/n:RequestHeader/n:UserFields/n:UserField[n:FieldCode='RF00000012']/n:SingleValueType/n:Value|UserName}
```

The adapter will detect if there is external list box xml in the value returned and if so will load the xml, and search for the column name defined at the end of the xpath statement. If no column is defined it will use the key column id.

3.3 Create Request Update Hook Adapter

Create a new adapter as below (adjusting server / virtual directory as per install):



Edit Adapter [OK] [Cancel]

Details

ID 6

System Type: Request Update Hook 7.1

Adapter Display Name: RequestUpdateAccelerator *

Web Service URI: http://localhost/AcceleratorRequestUpdateHook81/RequestUpdateHook.svc *


Timeout (seconds): 600 *

Client Identifier: GETUSERDATA

Active: ☒

Important: Set the Client Identifier to match the value set in the xml configuration file (ClientID)

3.4 Create Field Value changed by user hook

 Edit Adapter

Details

ID 5

System Type

Field Value changed by User hook8.0

Adapter Display Name

Field Changed*

Web Service URI

http://localhost/AcceleratorRequestUpdateHook81/FieldChangedUpdateHook.svc*

Timeout (seconds)

600*

Client Identifier

Active

☒

Important: For each field that is required to “fire” the hook set the Webservice parameters field to the code specified in the configuration xml (ClientID)

Edit Request Type - Activate account

Save OK Cancel

Details Form Rules Roles Approval Fulfilment SLA Request List Demand Variant

Switching tabs will save outstanding changes

Language English (United Kingdom)

Standard

- ☒ Check Box
- Single line text box
- Multi-line text box
- 123 Number
- @ Email
- Hyperlink
- File
- Date
- Time
- Radio Button
- Listbox

Section

Advanced

Check Box 1

External Listbox 1

Copy Delete

Visual Configuration Help

Field Code RF00000012 *

Field change by User calls Adapter ☒

Websocket Parameters GETUSERDATA

External Listbox Adapter Get Users * Get Configuration

Parameters USERS

Multi-select

Enabled Features Paging, Searching, User Filtering

Download Schema External Listbox_RF00000012.xsd

Download Value Schema External Picklist Value_RF00000012.xsd

Key Column UserId

3.5 Create Fulfilment Hook

Details

ID 8

System Type
Request Fulfilment Hook
8.0

Adapter Display Name
Accelerator Request Update Hook - Fulfilment

Web Service URI
http://localhost/AcceleratorRequestUpdateHook81/FulfilmentHook.svc

Timeout (seconds)
600

Client Identifier

Active
☒

Important: set the Parameters field to the code specified in the configuration xml (ClientID)

Adapter

Execute a system integration activity

Activity Label
Request update fulfilment hook

Adapter
Accelerator Request Update Hook - Fulfilment

Parameters
UPDATENAME

Retry on error
☐

Wait for response
☐

Cost

Duration
Days

OK
Cancel

3.6 Create Request Event Hook (Front Office Version 8.4+)

Details

ID 2

System Type Request Event Hook 8.4 ▼

Adapter Display Name Event Hook *

Web Service URI http://localhost/AcceleratorRequestUpdateHook81/RequestEventHook.svc *



Timeout (seconds) 600 *



Client Identifier



Active ☒



Important: set the Request Event Hook parameters field to the code specified in the configuration xml (ClientID)



Integration



On Request create hook  



On 'Requested For' change hook (deprecated) ?  

On Field Value changed by User hook  
Parameters

On Update button click hook  

Prior to Approval page hook (Deprecated) ?  
☐ Do not allow request to progress on error

Request Validation Hook  
Parameters

Request Event Hook Event Hook  
On Load ☒ Parameters GETUSERDATA