Broker Integration Interface

External Interface design document

Document is designed to depict a detailed description about the process flow, request-execution-response structure, Data flow, Error handling and schema design.

2013

Abhijeet Kumar

BMW Group Financial Limited

1/21/2013

# Contents:

## 1. Welcome to BMW-DFE External Interface

### What is BMW-DFE External Interface? ………………………………………………. 3

### How do I use BMW-DFE External Interface Web Service? …………………...3

### Do I need to use a specific language or tool? .............................................. .... 3

### Is it easy to implement? …………………………………………………………………..… 3

### What is Web service method implemented? ………………………………......…. 3

### How should these methods be used? ………………………………………….…..….. 3

## 2. Implementing BMW-DFE External Interface

### 2.1 Web Service security standards.......................................................................…....5

### 2.2 Implementation...................................................................................................…......5

## 3. BMW-DFE External Interface methods

### 3.1 CreateApplication () Method..................................................................….............6

## 4. Schema and Design

### 4.1 Request Schema..........................................................................................................7

### 4.2 Response Schema.......................................................................................................7

### 4.3 Schema Description...................................................................................................7

### 4.4 Error Handling...........................................................................................................7

## 5. Frequently asked questions

### 5.1 General.........................................................................................................................8

### 5.2 Performance……………………………………………………………………………………..8

## 6. Appendix 1............................................................................................9

**9.** **Activities after application creation via Interface...................10**

1. Welcome to BMW-DFE External Interface

### What is BMW-DFE External Interface?

BMW-DFE external interface is developed to serve the purpose of Dealers and Financiers who want to utilize the internal DFE web service, and submit application in DFE via this interface. If user has access to this web service, he can call it with his request XML. Service will process the request, validate it, and execute it. The final step would be to submit application in DFE database and forward it to AMS/Imaging for Application management. This will eliminate the two step process to capture application in a different system and then replicate the data in DP+ website I order to submit application to BMW. The service is generic enough to receive the request from any system, provided the request is in valid format and calling machine has authorization to access this web service.

### How do I use BMW-DFE External Interface Web Service?

As with any web service, you must integrate your existing application or environment with

the BMW-DFE external interface. The good news is that with basic web development skills, this

should be a straightforward task, as the interface has been developed specifically for ease of use.

### Do I need to use a specific language or tool?

No. The service is not language or tool specific. Any language or application that conforms

to standard web practices (as defined by the World Wide Web Consortium or W3C2) can

connect to and use BMW-DFE external interface.

### 1.4 Is it easy to implement?

Yes. The service has been designed with simplicity as a key feature. There is only one

functional method. This will receive request XML and will send the response back to calling service. You should have a good working knowledge of web services in your selected implementation language, and also have a sound understanding of how to provide authentication via a service.

### What is Web service method implemented?

1. Response XML CreateApplication (Request XML) This Web Method accepts the request XML from calling service. It validates the request XML, and pushes to the DFE Web service. The DFE Web Service in turn,

Consumes this XML and performs the submission task. How should these methods be used? Typically this method is exposed on internet. Vendor must call this service with a proper credential.

The request XML is received, consumed and a response is sent back.

Request XML

**BMW-DFE Interface**

**Vendor Service**

Response XML (success/failure)

Vendor Service can call BMW-DFE Interface by pushing the request XML. The Interface picks up the request and forwards it for execution. If the request fails due to service unavailability or being offline, it is queued. System polls for any queued service request and execute it, as and when service is up.

In case of service failure, a proper message, stating “Your request is queued in system” is returned to vendor. If the service fails due to validation error, an error message is returned in response XML stating the reason for failure. In case application is submitted successfully, a success message is returned back in response XML.

2. Implementing BMW-DFE external interface Security

**2.1. Web Service Security Standards**

The BMW-DFE external interface security is implemented using the web services security standard for

Authentication (refer to www.w3.org). It uses SOAP for the encoding of messages over this service.

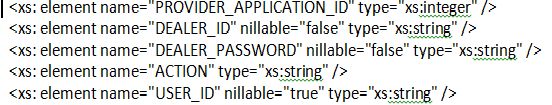
The hosted web service runs under a particular set of credentials. The calling system must impersonate their request with these credentials. Else 401 (access denied/unauthorized error will be raised.)

### Implementation

Authentication (User and System Access level)

Every request XML has a DEALER\_ID and DEALER\_PASSWORD tag.

The DEALER\_ID corresponds to the Id of Dealer who has access to DFE System, and DEALER\_PASSWORD corresponds to the password of this Dealer. These two are static entities and maintained in BMW Database. These credentials will be supplied to calling vendor. If user has access to DFE system the request will be forwarded. Else, an error message will be returned via the interface. Please make sure to send an active Dealer Id and correct password in this tag.



### Service URL:

Test URL: <https://proxy1uat.bmwfinance.com.au/BizTalk_External_Interface_Proxy/BizTalk_External_Interface_DFE_External_Interface_Orchestration_BMWDFE_AU.asmx>

Test WSDL: <https://proxy1uat.bmwfinance.com.au/BizTalk_External_Interface_Proxy/BizTalk_External_Interface_DFE_External_Interface_Orchestration_BMWDFE_AU.asmx?wsdl>

DFE URL to check applications:

<http://203.6.143.114/BMWFS.LS.DFE.UI.Web.AU/Login.aspx>

User Id: TestBrokerUser

Password: Testing#10

3. BMW-DFE External Interface Methods

### 3.1. CreateApplication () Method

The CreateApplication () method has following parameters:

1. Input: String.

This web method accepts XML string as input parameter. Please make sure that the input XML must be validated against the schema provided.

1. Output: String

The result of this function is a response XML string. The response can be of queuing, failure or success.

This is the only web method for this interface. It receives the request, processes it and sends back

the response to vendor service.

The response XML will contain Application Id if process is successful.

It will contain Error Code and Message if some validation has failed.

It will contain a queuing message, if request is queued due to service unavailability and network

Congestion.

4. Schema and design

### 4.1 Request Schema:



### 4.2 Response Schema



### 4.3 Schema Description



### 4.4 Error Handling:

Any error occurring in Service will be returned in response XML.

The format of error being returned will be in for of Id-Message pair. Every error handled in BMW-DFE external interface will be explicitly handled with an error Id and descriptive error message. This can be handled by vendor service to display in respective systems.

The response Schema contains a tag for ErrorDetails which is a multi occurrence node. The response can have multiple error messages based on validations occurring in interface.

ErrorId – Id of error message occurring in system (specific to interface).

Error Message – Descriptive error message corresponding to Error Id. Please refer to Appendix 1 for response XML.

Current set of response error codes are as below (please note that the list might increase till interface is finalized):

### Warnings:

DFE interface has a concept of warnings. Warnings are sent in response XML under WARNINGDETAILS tag. The application still saves in DFE but with a warning returned in response in addition to success message.



5. Frequently asked questions

### 5.1 General

### 5.1.1 How do I get BMW-DFE External Interface connection details?

The request data expects DEALER\_ID and DEALER\_PASSWORD of an Active Dealer to be supplied, who can access DFE System. If you do not have a DFE Active DEALER\_ID and Password, please contact BMW helpdesk and request for it.

### 5.1.2 Can our application connect directly to your web service?

If web service is called with proper credentials, the answer is most likely ‘Yes’.

The web service will receive and validate the request. If request is authorized, same will be forwarded for processing. Please make sure that the request XML is validated against the XSD.

### 5.2 Performance

### 5.2.1 What will be the average TAT for this web service?

The average turnaround time of this web service will depend on network congestion.

Since queuing is implemented, the TAT can vary between 15 sec to 40 sec.

### 5.2.2 Will the performance be impacted if we use Dot net 3.0 (and above)/JAVA/PHP

### for calling this web service?

The service is primarily developed on dot net. However it is platform independent. There will not be any major difference in performance if platform of calling system is varied.

6. Appendix 1

### Sample XML

### Request XML:



### Response XML

### Without Error



### With Error



Activities after application creation via Interface

Once application is creatd via interface,

1. System validates all finance and Information fields to be correct and consistent with application standards.
2. If any field is wrong, or invalidated, appropriate error message is returned.
3. Once application gets through, it is in Saved state in DFE database and application Id is retrned to Vendor service.
4. Then Vendor must Log in the system, fetch the application, and perform following changes :

a) Customer search for Existing customer to retrieve Costomer Id

b) Address Search, to validate if it’s correct address

c) Calculate Loyalty Bonus if applicable.

b) Submit the application in System

These activities will pust the application in Approval system for further processing.