Table of Contents

Integration With Other Systems

use existing database tables

This is explained in detail in the "Working with Data Stored in Existing Database Tables" section.

use data from several existing database tables

Sometimes you may want to connect to existing database tables but display data from several existing database table in a single query. With normal ***AwareIM*** objects you can do it using shortcuts. With existing database tables you can also define a "normal" ***AwareIM*** object that would have references to different objects stored in the existing database tables and then define shortcuts to different attributes in these objects.

A better approach, however, is to define a view in the existing database that would select data from different tables, and then create a special ***AwareIM*** object that would point to this view.

integrate a credit card payment system

Many credit card payment systems, such as PayPal, require that your web site communicate with them via a particular URL. Whenever your web site needs to perform credit card payment the users need to click on the link with this URL, which will bring them to the web site of the credit card payment system. As part of the URL that your web site supplies to the credit card payment system you must also specify the URL that the credit card payment web site will use to return to your web site upon successful or un-successful completion of the payment.

If you want to integrate the credit card payment into your ***AwareIM*** system you need to configure the following:

1. Most credit card systems require that you pass certain parameters to the system as part of their URL, for example, PayPal system requires the following parameters (among others) – `amount`, `currency_code`, `item_name` etc. First of all you need to define a business object representing parameters of the payment with the attribute names being exactly equal to the names of the parameters you want to pass to the credit card payment system. For example, we could define `PayPalPayment` business object with `amount`, `currency_code` and `item_name` attributes.
2. Define an intelligent business object representing the credit card payment system, for example you can define the business object with the name `PayPalSystem` – see the "Defining Intelligent Business Objects" section for details.
3. Define the URL channel as the default channel for this business object – see the "Setting Properties of URL Channel" section. Specify the base URL of the credit card payment system as the "URL of the Service Provider" channel property. Specify the names of the parameters that the credit card payment system uses to identify the URL to return to as channel properties. For example, PayPal system requires that the parameter names for the successful and unsuccessful completion of payment are `return` and `cancel_return` respectively. ***AwareIM*** will make sure that the values for these parameters are correctly set, so that the credit card system will

return to the ***AwareIM*** system on successful or un-successful completion of the payment.

4. Define the payment service of this business object, for example `ProcessPayment` – see the "Defining Services of Intelligent Business Object" section for details. Specify the business object you created in step 1 as the service input. Select `Standard Reply` as the service reply.
5. Define a process that will initialize the instance of the business object representing parameters of the payment (for example, PayPalPayment) request the service using the `REQUEST SERVICE` action, for example:

```
CREATE PayPalPayment WITH PayPalPayment.amount=100, ...
REQUEST SERVICE ProcessPayment of PayPalSystem USING PayPalPayment
```

6. Define a menu item that will start the above process – see the "Setting Menu Item Properties" section. Alternatively you can set up a form operation or a hyperlink on some presentation to start the process – see the "Adding/Editing Form Operations" and "Hyperlinks". When the process starts ***AwareIM*** will navigate to the web site of the payment system. After the payment has been completed the web site of the credit card payment system will return to the ***AwareIM*** system (***AwareIM*** will automatically display whether the payment request has been fulfilled).

communicate with another software system

This is explained in the "Communication with Other Systems" section.

communicate with a hardware device

This is explained in the "Communication with Other Systems" section.

expose a web-service

AwareIM will automatically expose the services you configure for your system as web services, so that other software systems or web sites can use them – see the "Adding/Editing Services" section.

expose a REST-ful service

AwareIM can expose a service you configure either as a SOAP-based web service or as a REST-ful service or both. For REST-ful services exposed by AwareIM applications that need to call the service need to call a particular URL using the HTTP protocol. This URL should include the AwareIM server name and port where Tomcat listens to requests and also the name of the service to be called and parameters. The default URL where AwareIM will listen to REST-ful requests has the following format:

http://ServerName:ServerPort/AwareIM/REST/BusinessSpace/ServiceName?parameters

Parameters are specified as paramName1=paramValue1&paramName2=paramValue2&...

Where paramName1, paramName2 etc are names of the attributes of a business object that is specified as input to the process implementing the service.

You can also specify your own custom URL in which case you also need to supply a file called

rest.props

All the details are explained in the “Adding/Editing Services” section.

consume a REST-ful service

To do this you need to define an intelligent business object by defining the REST communication channel with the object. This is described in the “Defining Intelligent Business Objects” section.

use a web-service

If you want to use an existing web service exposed by some web site then you need to do the following:

1. Define an intelligent business object representing the web site exposing the web service with System intelligence type – see the “Defining Intelligent Business Objects” section.
2. Define the SOAP channel as the default channel for this object. Specify the URL where the WDSL file describing the web service is exposed as the property of the channel – see the “Setting Properties of SOAP Channel” section (alternatively ask the web service provider for the WSDL file, copy this file to a local file system and specify the URL to this, file, for example: `file://c:/temp/lookup.wsidl`).
3. Discover services of the business object – see the “Discovering Services” section. The web service you want to use should appear in the list of services exposed by the object.
4. Define a rule that will request the service of the business object using the `REQUEST SERVICE` action.

register AwareIM application in Facebook

In order to allow users to login via Facebook you need to register your AwareIM application with Facebook. In order to do this:

- Go to the URL https://developers.facebook.com/apps
- Create a new application
- When specifying site URL for the application enter the following URL:

http://YourServerName:8080/AwareIM

You can use localhost instead of the server name when testing your application.

When specifying “redirect URI” of the application enter the following URI:

http://YourServerName:8080/AwareIM/req.awfb2

- Copy application id and secret that Facebook should assign to your application – you will need these when setting up login options (see 0910_login_fb_tw_ggl sections)

When users login to your ***AwareIM*** application they should use the following URL:

http://YourServerName:8080/AwareIM/logon.awfb?domain=YourBusinessSpaceName&testingMode=fa

lse

register AwareIM application in Twitter

In order to allow users to login via Twitter you need to register your AwareIM application with Twitter. In order to do this:

- Go to the URL https://dev.twitter.com/apps
- Create a new application
- When specifying callback URL for the application enter the following URL:

http://YourServerName:8080/AwareIM

You can use localhost instead of the server name when testing your application.

When specifying “redirect URI” of the application enter the following URI:

http://YourServerName:8080/AwareIM/req.awtw2

- Copy consumer key and secret that Twitter should assign to your application – you will need these when setting up login options (see 0910_login_fb_tw_ggl section)

When users login to your AwareIM application they should use the following URL:

http://YourServerName:8080/AwareIM/logon.awtw?domain=YourBusinessSpaceName&testingMode=false

register AwareIM application in Google

In order to allow users to login via Google you need to register your ***AwareIM*** application with Google. In order to do this:

- Go to the URL https://code.google.com/apis/console
- Create a new application
- When specifying redirect URI for the application enter the following URI:

http://yourservername:8080/awareim/req.awggl2

You can use localhost instead of the server name when testing your application.

When specifying “redirect URI” of the application enter the following URI:

http://YourServerName:8080/AwareIM/req.awggl2

- Copy client id and secret that Google should assign to your application – you will need these when setting up login options (see 0910_login_fb_tw_ggl section)
- If using synchronization with the Google Calendar make sure that access to the calendar API is allowed!

When users login to your ***AwareIM*** application they should use the following URL:

http://YourServerName:8080/AwareIM/logon.awggl?domain=YourBusinessSpaceName&testingMode=false