

Table of Contents

Adding custom channels 2

[Programmers Reference](#), [Adding Custom Channels](#)

Adding custom channels

Communication between external entities, such as other software systems and hardware devices (which are modeled in **Aware IM** as intelligent business objects) and **Aware IM** is performed through *channels*. A channel represents a media through which messages are sent between external systems and hardware devices (collectively called here as “external parties”) and **Aware IM**. A channel is responsible for converting messages into the format that can be understood by the **Aware IM** software on one end and the external party on the other end. There can be one-way or two-way channels. In a one-way channel messages are only sent in one direction – from **Aware IM** to the external party or from the external party to **Aware IM**. In a two-way channel messages are sent in both directions.

From **Aware IM** point of view there are only two types of messages – *service requests* and *notifications*. Service request is a request for execution of a particular service. The external party being an intelligent business object may expose a number of services that can be requested by **Aware IM**. Similarly the current business space version in **Aware IM** may expose a number of services that can be requested by the external party. The service requestor may expect a particular reply to the service as declared in the definition of the service.

A notification, on the other hand, represents just a piece of information that is sent to the external party or to **Aware IM** (the business space version in **Aware IM** may contain rules which define how different notifications are handled). It is up to a receiver of the notification to decide how it will react to this information or whether it will react at all. Generally no reply to the notification is expected from the receiver.

A channel therefore must translate the service request or notification message that is sent by **Aware IM** to the external party into whatever format the external party understands. Similarly a channel has to translate a message sent by the external party into the service request or notification message that can be understood by **Aware IM**.

Note that if an external party is sending service requests to **Aware IM** it must first log into **Aware IM** as a particular user (an intelligent business object representing the external party must be a member of the SystemUsers business object group), otherwise the service request will be sent on behalf of a “guest” user. Sending notifications to **Aware IM**, on the other hand, does not require logging in.

There are a number of predefined channels that **Aware IM** comes with, such as the E-mail channel, the SOAP channel and others (see the “Defining Intelligent Business Objects” section in the “Aware IM User Guide”). If your application connects to external software systems or hardware using protocols that are not supported by **Aware IM** out-of-the-box, you are likely to need to write your own custom channel that will connect your **Aware IM** system with the required software system or a hardware device.

The following section describes what is involved in adding a new custom channel to **Aware IM**.

To add a new custom channel to **Aware IM** follow the steps below:

1. Write the code for the component that will handle communication between **Aware IM** and your software system or a hardware device (see section 4.1 for details)

2. Write the code for the component that describes the capabilities of the new channel - this component must implement the `IChannelType` interface (see section 4.1.2 for details).
3. If your channel has properties that need to be controlled via the user interface in the Configuration Tool, write the code for the component that implements this user interface (see section 4.3 for details).
4. Compile the source code of your components and package them in a jar file(s). Make sure that the jar file is referenced in the CLASSPATH (see section 2)
5. Add the definition of the new channel to the `BASServer.props` file located in the BIN directory of your **Aware IM** installation. You have to add a line that provides the fully qualified name of the component that you wrote in step 2. The name of the property must start with the `ChannelType` prefix and the value is the fully qualified name of the component. For example:

```
ChannelTypeMail=com.bas.basserver.channels.EmailChannelType
```

If everything has been done correctly the name of the new channel must appear in the list of channels available for intelligent business objects in the Intelligence tab of the Business Object Property Editor (see “Defining Intelligent Business Objects” section in the “Aware IM User Guide”)

- [Implementing Channel’s Source and Sink](#)
- [Implementing Channel Type component](#)
- [Implementing Channel Settings Editor](#)

From:
<http://www.awareim.com/dokuwiki/> - **Documentation**

Permanent link:
<http://www.awareim.com/dokuwiki/docs/3500/0400?rev=1680676706>

Last update: **2023/04/05 06:38**

