



**Masoud Aghadavoodi Jolfaei**

April 14, 2014 6 minute read

## ABAP Channels Part 3: Collaboration Scenario Using ABAP Messaging and ABAP Push Channels

[Follow](#)

[RSS feed](#)

2 Likes 5,266 Views 16 Comments

### Introduction

For introduction to ABAP Channels please refer to [Introduction to ABAP Channels](#). For tangible examples in the system see also the blog [ABAP Channels Examples](#). See also [FAQ – ABAP Channels](#).

This blog focuses on the collaboration scenario, i.e. using **ABAP Push Channel (APC)** and **ABAP Messaging Channel (AMC)** to push messages from ABAP sessions to WebSocket clients by binding publish/subscribe channels to a WebSocket connection.

For live demonstration and step-by-step implementation look at [Collaboration Video](#).

### Collaboration of ABAP Messaging and Push Channels

The collaboration scenario (see figure 2.0) provides the possibility on the basis of publish/subscribe pattern of ABAP Messaging Channels to push messages from an ABAP session or from a WebSocket client (using ABAP Push Channel) to from publisher to subscribers of channels.

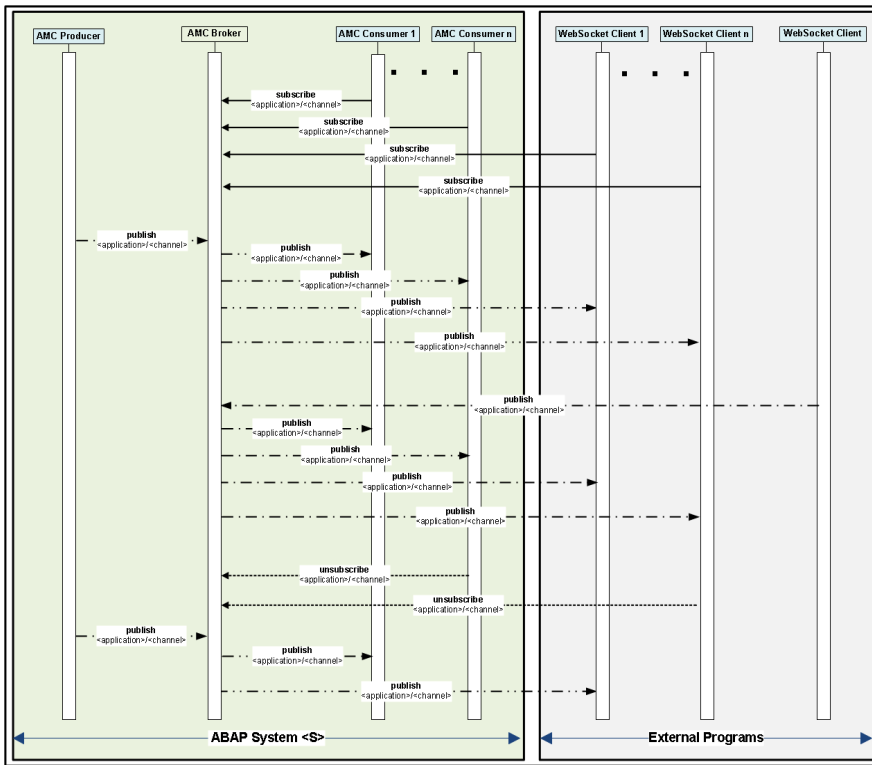


Figure 2.0: Simple interaction model of collaboration scenario.

One of the goals for the collaboration scenario is to overcome the manual refresh for an object list in the user-interfaces by introducing an “auto-update” or alternatively by using a notification pattern. This goal can easily be reached by notifying the user-agent having a WebSocket connection to the ABAP system about the changes on objects displayed in the user-agent.

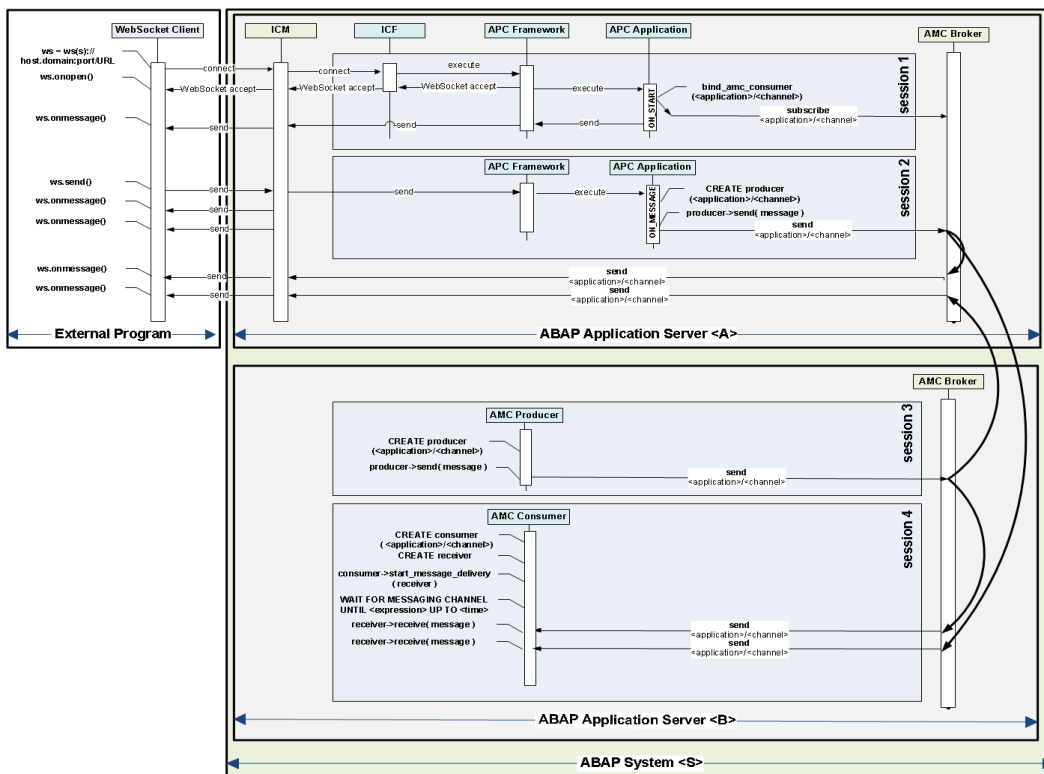


Figure 2.1 illustrates the technical interaction model for pushing messages between a WebSocket

In order to pass an event from an ABAP session to a WebSocket client and vice-versa the publish/subscribe infrastructure of the ABAP Messaging Channels is used. As in the figure 2.1 depicted the APC frame with the binding manager interface offers the possibility to bind an AMC channel to the underlying WebSocket connection to act as message consumer (receiver) for the corresponding channel. After a successful binding to an AMC channel, here `<application>/<channel>`, any messages published to the channel will be forwarded by the AMC broker to the WebSocket client using corresponding WebSocket connection.

## Creating an ABAP Push and Messaging Channel Application for Collaboration Scenario

This section outlines the additional steps required to extend the existing APC application (see [ABAP Channels Part 1: WebSocket Communication Using ABAP Push Channels \[1\]](#)) and AMC application (see [ABAP Channels Part 2: Publish/Subscribe Messaging Using ABAP Messaging Channels \[2\]](#)) to create the collaboration scenario.

First, the APC application will be extended:

- In the existing APC application `YCL_APC_WS_EXT_YAPC_WS_TEST` the coding of method `ON_START` (Figure 2.12 in [1]) will be replaced with coding to bind the WebSocket connection to the AMC application `YAMC_TEST` and channel `"/ping"`.

**METHOD `if_apc_ws_extension~on_start`.**

TRY.

*\* bind the WebSocket connection to the AMC channel*

`DATA(lo_binding) = i_context->get_binding_manager( ).`

```

lo_binding->bind_amc_message_consumer( i_application_id = 'YAMC_TEST'
i_channel_id    = '/ping' ).
CATCH cx_apc_error INTO DATA(lx_apc_error).
DATA(lv_message) = lx_apc_error->get_text( ).
MESSAGE lx_apc_error->get_text( ) TYPE 'E'.
ENDTRY.

```

**ENDMETHOD.**

- Same for method **ON\_MESSAGE** (figure 2.15 in [1]). Here the received messages from WebSocket client are forwarded to the AMC application **YAMC\_TEST** and channel “/ping”.

**METHOD if\_apc\_ws\_extension~on\_message.**

```

DATA: lo_producer TYPE REF TO if_amc_message_producer_text.
TRY.
* retrieve the text message
DATA(lv_text) = i_message->get_text( ).
lo_producer ?= cl_amc_channel_manager=>create_message_producer(

                                i_application_id = 'YAMC_TEST'

                                i_channel_id    = '/ping' ).
lo_producer->send( i_message = lv_text ).
CATCH cx_amc_error INTO DATA(lx_amc_error).
MESSAGE lx_amc_error->get_text( ) TYPE 'E'.
CATCH cx_apc_error INTO DATA(lx_apc_error).
MESSAGE lx_apc_error->get_text( ) TYPE 'E'.
ENDTRY.

```

**ENDMETHOD.**

- In the next step the access rights will be maintained. For the AMC application **YAMC\_TEST** and channel “ping” the activities “Receive via APC WebSocket” and Send” are referencing the implementation class **YCL\_APC\_WS\_EXT\_YAPC\_WS\_TEST** of APC application **YAPC\_WS\_TEST**. This is done in the AMC design time, transaction **SAMC** (figure 3.20 in [2]).

Open the definition of application **YAMC\_TEST** and click **Insert line** (see figure 3.1).

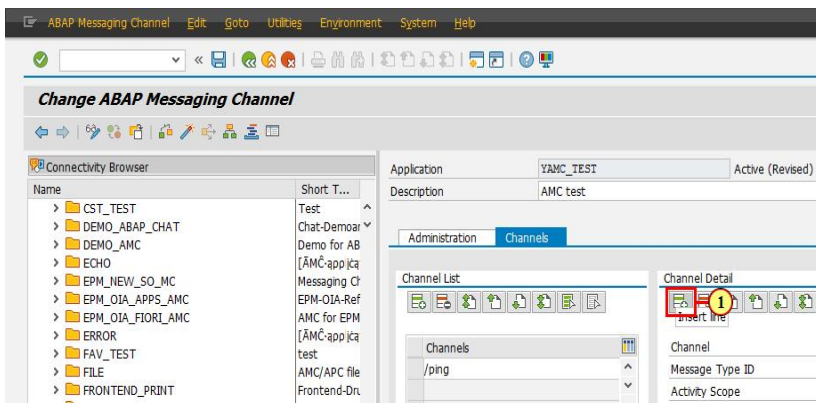


Figure 3.1

Open the value help in the **Authorized Program** field (see figure 3.2).

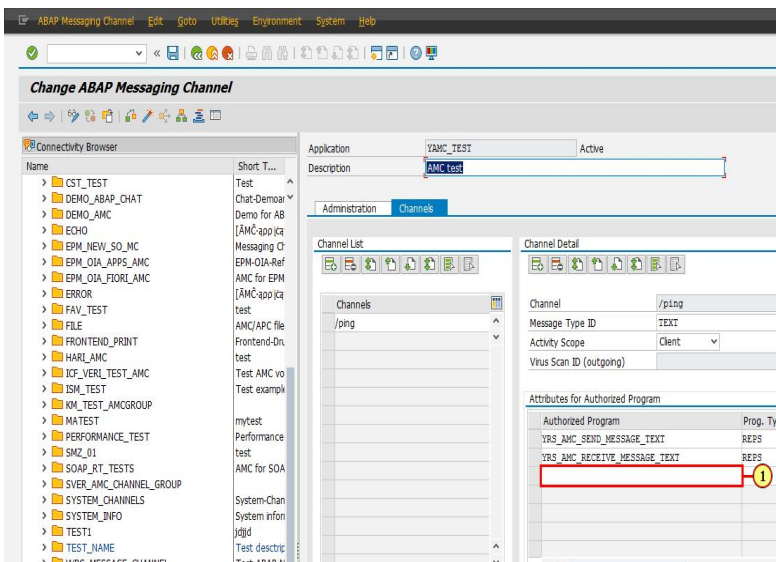


Figure 3.2

Insert the class **YCL\_APC\_WS\_EXT\_YAPC\_WS\_TEST** and click **Accept** (see figure 3.3).

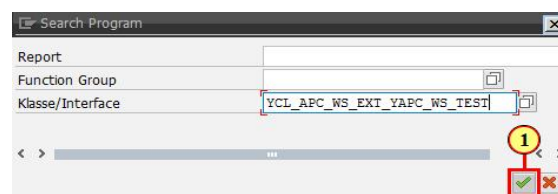


Figure 3.3

Select the entry **C Receive via APC WebSocket** (see figure 3.4).

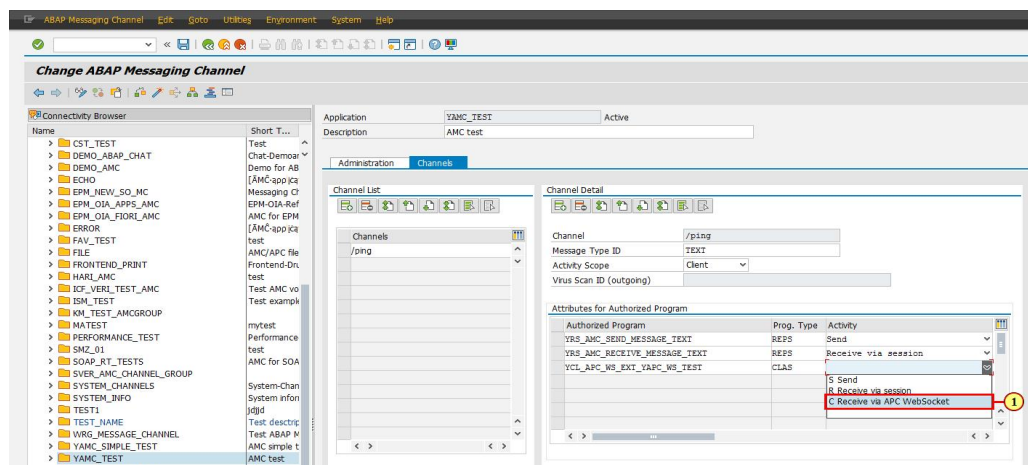


Figure 3.4

Insert additional line as above and select the entry **S Send** (see figure 3.5).

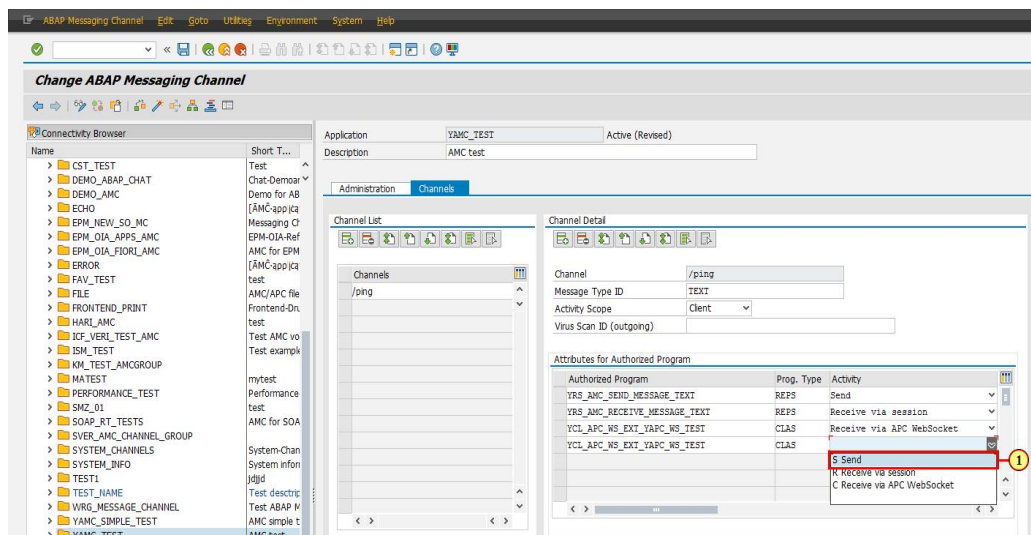


Figure 3.5

Optionally a **Virus Scan ID (outgoing)** can be specified (see figure 3.6).

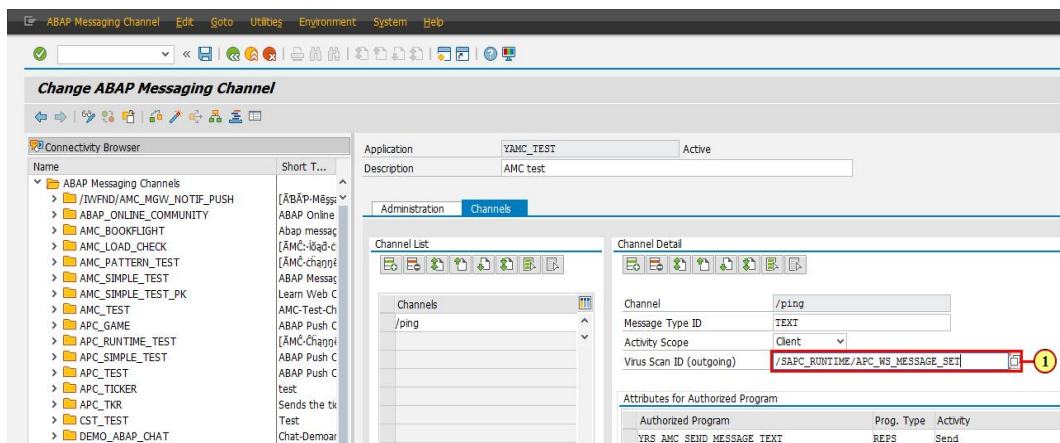


Figure 3.6

Click **Save** (see figure 3.7).

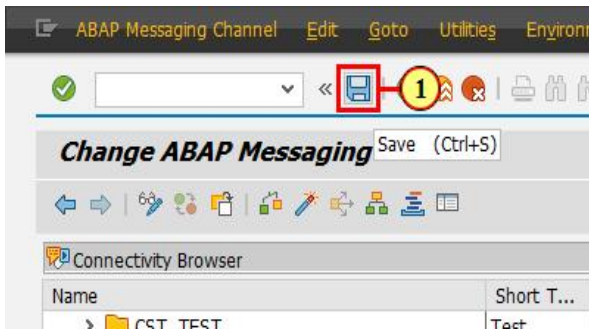


Figure 3.7

And **Activate**. (see figure 3.8).

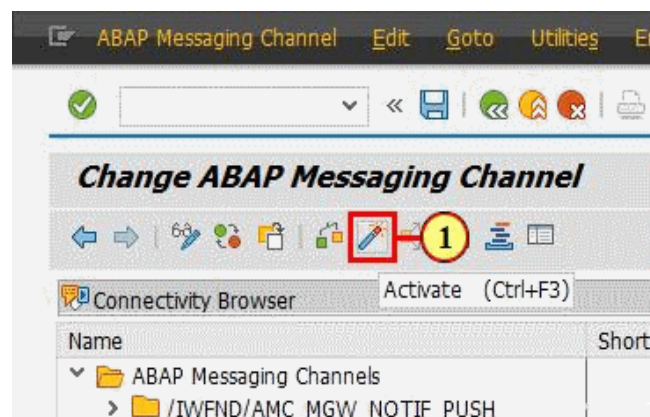


Figure 3.8

Click **Continue** .(see figure 3.9)..

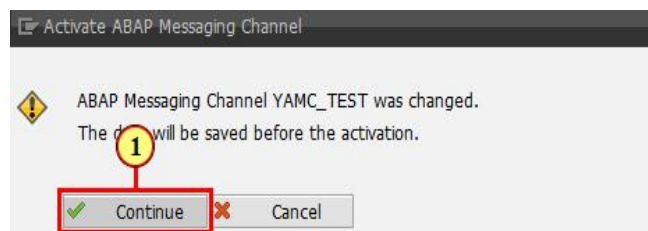


Figure 3.9

To test the application just press the **Test** icon (see figure 3.10).

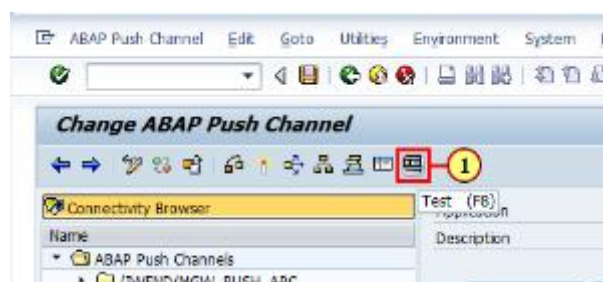


Figure 3.10

This action launches the Web Dynpro application **WDR\_TEST\_APC** in a browser (see figure 3.11) which supports the WebSocket protocol (RFC 6455) (e.g. Chrome version  $\geq 16$ , Firefox version  $\geq 11$ , Internet Explorer version  $\geq 10$  and Safari on iOS version  $\geq 6$ ). If it does not launch, double check the Web Dynpro service path `/sap/bc/webdynpro/sap/wdr_test_apc` in the transaction **SICF** is active.

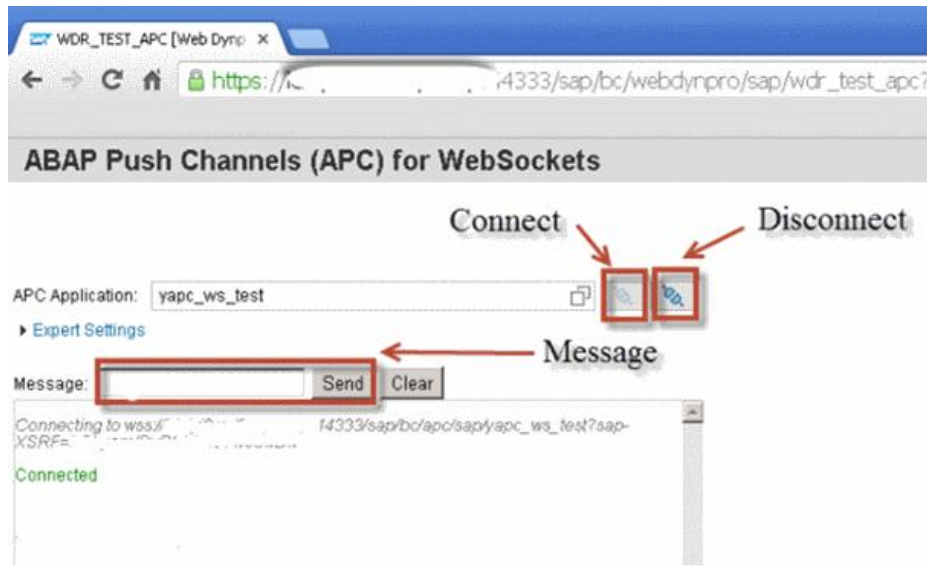


Figure 3.11

Just starts a second time the Web Dynpro test application for APC application **YAMC\_TEST** in browser (see figure 3.12). After a successful connection setup any messages send in a browser will be received also by the second browser.



Figure 3.12

Furthermore any report which apply any actions on the AMC application **YAMC\_TEST** and channel `/ping` will be involved in the collaboration interaction. In other words the execution of the AMC send report **YRS\_AMC\_SEND\_MESSAGE** (refer [2] figure 3.15) will lead to receiving the send message by the browser(see figure 3.13)



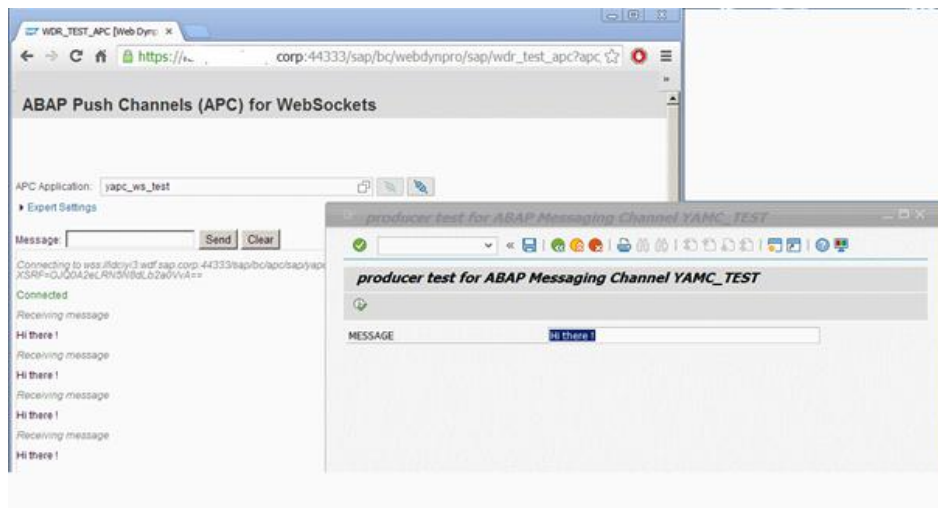


Figure 3.13

And the execution of the AMC receive report YRS\_AMC\_RECEIVE\_MESSAGE (refer blog [2]figure 3.25) will lead to receiving the send message by the browser(see figure 3.14).

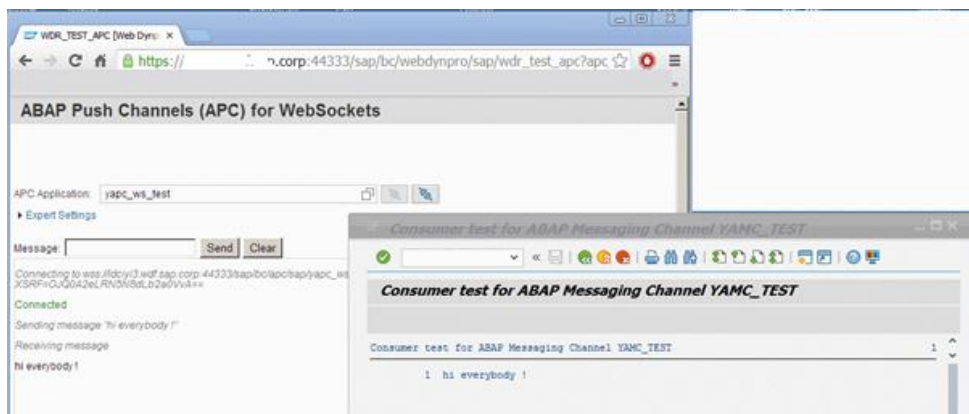



Figure 3.14

## Collaboration Scenario Security

The security of collaboration scenario is handled by the underlying APC and ac frameworks.

## Collaboration Scenario Supportability

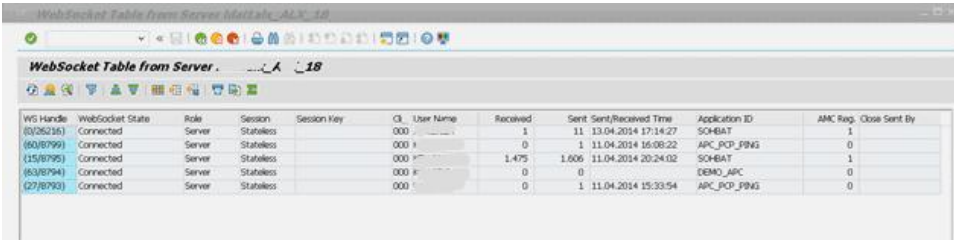
Whereas for AMC channels with the transaction **SMAMC** (refer blog [2] figure 4.0) can the subscribed channels be monitored also here a navigation to the bound WebSocket connection in **SMWS** (see [1] ) transaction. The transaction SMWS provides also the total number of bound AMC channels per WebSocket connection (see figure 4.0).



Channel	Type	Recipient ID	Session Key	Client	User Name	Received Messages
echo/sapgui/ping	SAP GUI	T57_U100613_M0_11	T57_U100613_M0_11	000		1
sohbat/command/skder	WebSocket	(0/26216)	T62_U11605_M0_10	000		
sohbat/command/skder	WebSocket	(15/8795)	T32_U06306_M0_10	000		1595

Figure 4.0

The transaction SMAMC (see figure 4.1) shows the list of the AMC channels which are registered in sessions (Type “Session”) or in case of “Collaboration scenario” the WebSocket connections (Type is “WebSocket”) in the system as AMC consumers.



WS Handle	WebSocket State	Role	Session	Session Key	Cl	User Name	Received	Sent	Sent/Received Time	Application ID	AMC Req.	Close Sent By
(0/26216)	Connected	Server	Stateless		000		1	11	13.04.2014 17:14:27	SO4BAT	1	
(60/8799)	Connected	Server	Stateless		000		0	1	11.04.2014 16:08:22	APC_RPC_PING	0	
(115/8795)	Connected	Server	Stateless		000		1,475	1,606	11.04.2014 20:24:02	SO4BAT	1	
(60/8794)	Connected	Server	Stateless		000		0	0		SO4BAT	0	
(27/8793)	Connected	Server	Stateless		000		0	1	11.04.2014 15:33:54	APC_RPC_PING	0	

Figure 4.1

## Conclusion and outlook

This article showed the collaboration scenario based on AMC and APC for exchange of messages between different WebSocket clients. Currently the message exchange based on best/effort strategy and no guaranteed delivery is available. In a future article best practices will be outlined.

---

## Assigned tags

---

ABAP Connectivity | channel | communication | event | messaging |

[View more...](#)

## Related Blogs

---

[Real-time notifications and workflow using ABAP Push Channels \(websockets\) Part 1: Creating the APC and AMC](#)

By **Brad Pokroy** , Oct 16, 2014

[Introduction to ABAP Channels](#)

By **Olga Dolinskaja** , Nov 27, 2014

[ABAP Channels Part 1: WebSocket Communication Using ABAP Push Channels](#)

By **Masoud Aghadavoodi Jolfaei** , Nov 18, 2013

## Related Questions

---

[ABAP Push Channel is not supported in the selected project](#)

By **Former Member** , Nov 03, 2016

[Websockets oon HCP & Cloud Connector](#)

By **Carlos Zaburlin** , Aug 08, 2018

[How can we connect to Eclipse Neon ABAP perspective with SAP SSO \( Single Sign On \)](#)

By **Former Member** , Nov 24, 2017

## 16 Comments

You must be [Logged on](#) to comment or reply to a post.

---



Former Member

[August 27, 2014 at 11:00 am](#)

Thanks for all the 3 articles 😊 Very useful!

Like (0)



Former Member

January 23, 2015 at 10:42 am

Hi Masoud,

Really Great explanation. could you please explain how to get notifications in my own bsp page rather than a web dynnpro application.

Thanks,

Santosh.

Like (0)



Masoud Aghadavoodi Jolfaei | Post author

February 2, 2015 at 7:06 am

Hi Santosh,

unfortunately I did not have time to contact the BSP team regarding your question. I see that you have already released the SCN blog [ABAP Push Channel with BSP application](#) explaining how to support notifications in BSP applications using APC & AMC. Thanks for the blog.

Masoud

Like (0)



Former Member

May 21, 2015 at 9:35 am

Hi Friends,

Is it Possible to achieve one to One Chat using this websocket Technique in WebDynPro ?

if so, pls Suggest me an Idea.

Thanks,

Siva

Like (0)



Masoud Aghadavoodi Jolfaei | Post author

May 26, 2015 at 6:46 am

Hi Siva,

with the help of the AMC channel extension ID the group of the recipients of a message can be defined fine grained. Just have look to my response to Ekansh on Dec. 23 2014 in my blog

[“ABAP Channels Part2. Publish/Subscribe Messaging Using ABAP Messaging Channels”](#).

Best regards,

Masoud

Like (0)



Chris Xu

May 26, 2015 at 1:38 pm

Hi Masoud,

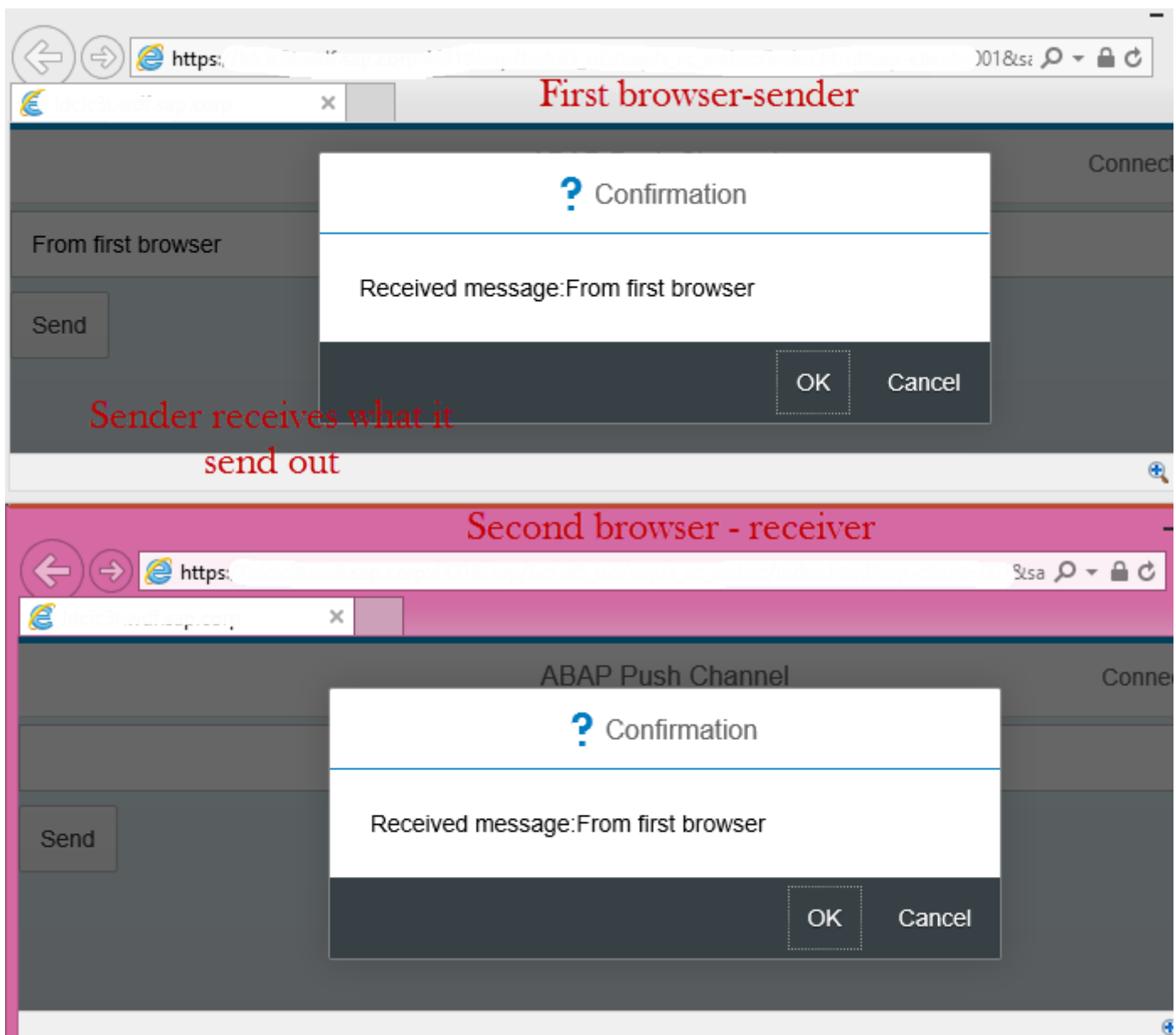
Thanks for your great post.

I followed this blog and created a demo like yours. And I have a question now.

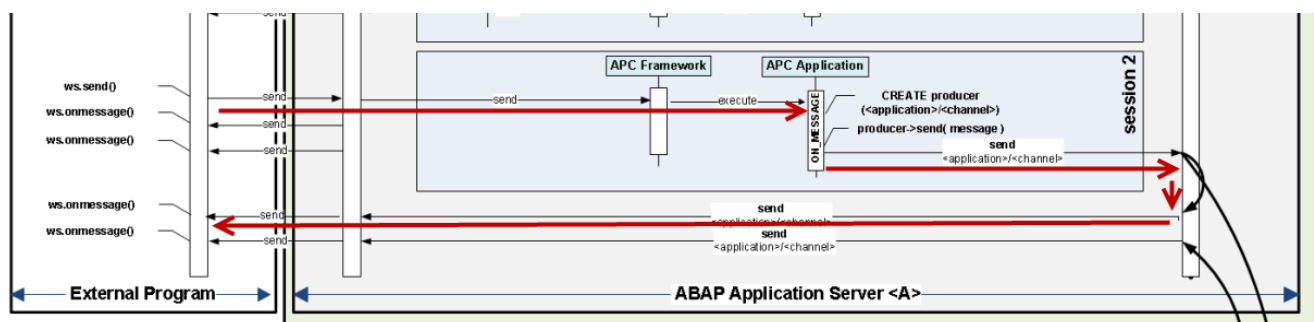
You mentioned in the post that:

Just starts a second time the Web Dynpro test application for APC application YAMC TEST in browser (see figure 3.12) .After a successful connection setup any messages send in a browser will be received also by the second browser.

I found that the sender browser also received the message what it sent out. Is this behavior correct? Like below. Top browser acts the sender, bottom acts the receiver.



From figure 2.1. I think the behavior is correct.



So my question is: will the sender always receive what it send out?

Thanks,

Chris Xu

Like (0)



Masoud Aghadavoodi Jolfaei | Post author

May 26, 2015 at 2:43 pm

Hi Chris,

yes, the sender always receives what it sent out. In this case you can add a kind of “sender id” which is generated as UUID per Web Dynpro instance and inserted to the PCP message as a PCP field. This allows the recipients to identify their own messages and if required to filter/skip those message (with JavaScript means).

Best regards,

Masoud

Like (0)



Chris Xu

May 27, 2015 at 1:19 am

Hi Masoud,

Thanks for your quick response and good solution.

So my understanding of the work mechanism is correct. Thank you again.

Chris.

Like (0)



Kamal el Marzgioui

November 17, 2015 at 12:58 am

Dr. Masoud,

Thank you for taking the time to write these 3 very elaborated and useful articles!

I have witnessed the APC/AMC Collaboration scenario working at the Barcelona TECHED (hands-on workshop).

It is definitely an interesting feature in ABAP and can be put to use right away.

Regards,

Kamal.

Like (0)



Kamal el Marzgioui

November 17, 2015 at 12:59 am

Dr. Masoud,

Thank you for taking the time to write these 3 very elaborated and useful articles!

I have witnessed the APC/AMC Collaboration scenario working at the Barcelona TECHED Hands-On Workshop.

It is definitely an interesting feature in ABAP and can be put to use right away.

Regards,

Kamal.

Like (0)



Dren Selimi

September 30, 2016 at 8:24 am

Hello Dr.Masoud,

I am having a problem Sending data from SAP to Web, and from Web to Web,

Could you help me out, I have followed all the steps as described by you.

Regards,

Dren Selimi

Like (0)



Masoud Aghadavoodi Jolfaei | Post author

September 30, 2016 at 11:46 am

Hi Dren,

to localize the reason of the issue I propose to follow the instructions in the note [2353453](#) and also double check the correctness of the application based on the [Collaboration Video](#).

Cheers,

Masoud

Like (0)



Dren Selimi

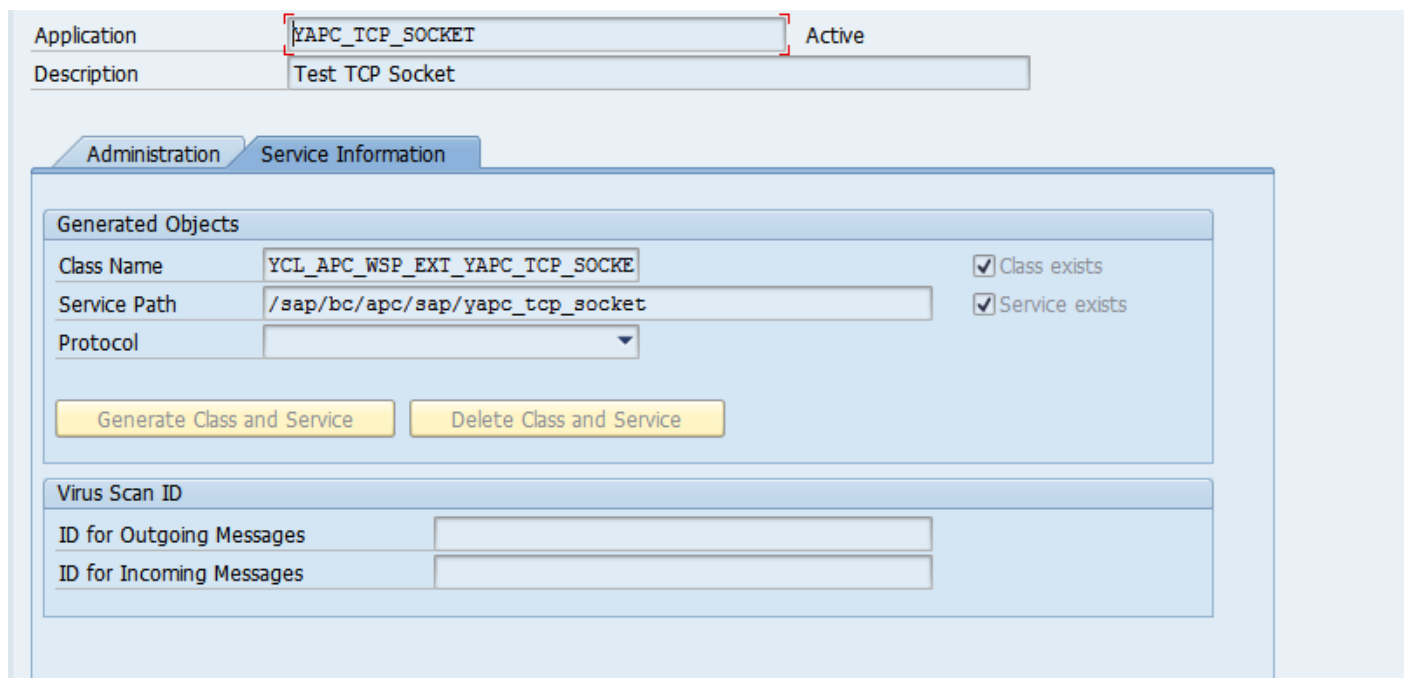
October 5, 2016 at 2:27 pm

Hi Dr.Masoud,



I am facing another problem, I am now trying to create a TCP Socket, but I do not get all the options that I found in the official documentation for creating a TCP Socket Server in SAP, I will attach two images and you can see the differences between the documentation and what I get in my system, in the part for setting the protocol.

My screen:



The screenshot shows the SAP configuration interface for a TCP Socket. At the top, the 'Application' field is set to 'YAPC\_TCP\_SOCKET' and is marked as 'Active'. Below it, the 'Description' is 'Test TCP Socket'. The interface has two tabs: 'Administration' and 'Service Information', with 'Service Information' currently selected. Under 'Service Information', there is a section for 'Generated Objects' containing fields for 'Class Name' (YCL\_APC\_WSP\_EXT\_YAPC\_TCP\_SOCKET), 'Service Path' (/sap/bc/apc/sap/yapc\_tcp\_socket), and 'Protocol' (a dropdown menu). To the right of these fields are checkboxes for 'Class exists' and 'Service exists', both of which are checked. Below this section are two buttons: 'Generate Class and Service' and 'Delete Class and Service'. At the bottom, there is a 'Virus Scan ID' section with fields for 'ID for Outgoing Messages' and 'ID for Incoming Messages'.

the doc screen:  [Creating a TCP Socket Application](#)

The link for the documentation

[Implementing a TCP Socket Server – ABAP Channels – SAP Library](#)

Do you have any idea for this problem ?

Thanks, Dren

Like (0)



**Masoud Aghadavoodi Jolfaei** | Post author

October 5, 2016 at 3:45 pm

Hi Dren,

it seems that your development system is not on ABAP release  $\geq 750$ . The TCP Socket protocol as part of ABAP Channels infrastructure is available with ABAP release  $\geq 750$ .

Cheers,

Masoud

Like (0)



Former Member

February 18, 2017 at 12:09 am

Hi Masoud,

It is a very nice blog and very useful feature to implement the User-interface automation for some of the requirements. There are two questions

1. There is a APC for CRM\_IU\_PFC is not working for testing. Is there anyway to know if all are released for customers to use or internally released.
2. Are there any example for SAP CRM WEBUI interfacing or any special tags introduced.

Like (0)



Henning Duerholt

March 9, 2017 at 3:30 pm

Hello Madhusudhanarao,

regarding your question 2.: an example for using the ABAP channels in SAP CRM Interaction Center is described in this [blog](#):

regarding your first question: could you explain in more detail waht you mean by CRM\_IU\_PFC ?

thanks a lot and best regards,  
Henning Dürholt

Like (0)

## Share & Follow

Privacy

Terms of Use

Legal Disclosure

Copyright

Trademark

Cookie Preferences

Sitemap

Newsletter