



Application Notes for configuring Outlook Calendar Routing Snap-in V1.1 from Pridis B.V. to interoperate with Avaya Breeze™ R3.4 and Avaya Aura® Communication Manager R7.1 – Issue 1.0

Abstract

These Application Notes describe the configuration steps for provisioning Outlook Calendar Routing Snap-in from Pridis B.V. to successfully interoperate with Avaya Breeze™ and Avaya Aura® Communication Manager. The snap-in created by Pridis B.V. is installed on Avaya Breeze™ and performs a database lookup based on the number being called and can check the Outlook calendar to see if the user is in a meeting or has an appointment and will re-route the call where necessary.

Readers should pay attention to Section 2, in particular the scope of testing as outlined in Section 2.1 as well as any observations noted in Section 2.2, to ensure that their own use cases are adequately covered by this scope and results.

Information in these Application Notes has been obtained through DevConnect Compliance Testing and additional technical discussions. Testing was conducted via the DevConnect Program at the Avaya Solution and Interoperability Test Lab.

1. Introduction

These Application Notes describe the configuration steps for provisioning Outlook Calendar Routing Snap-in from Pridis B.V. to successfully interoperate with Avaya Breeze™ and Avaya Aura® Communication Manager. The snap-in, created by Pridis, is installed on Avaya Breeze™ and performs a database lookup based on the number being called and can check Microsoft Exchange calendar to see if the user is in a meeting or has an appointment and will re-route the call where necessary.

With the Pridis Outlook Calendar Routing Snap-in (snap-in) for Avaya Breeze® Platform, businesses can easily change the default routing for incoming calls based on appointments in the Microsoft Exchange calendar. This means that routing can be changed simply by updating a calendar appointment.

Upon an inbound call, Avaya Breeze® Platform triggers the Pridis snap-in to check the current Exchange calendar information for the number called and redirects the call to the telephone number indicated in the appointment. For example, a calendar appointment can be used to route incoming calls to a specific service engineer on duty. If there is no calendar appointment, the call can be redirected to a default telephone number.

Note: The Outlook Calendar Routing Snap-in uses implicit application sequence. Breeze does not support explicit SIP sequencing, it's not possible to sequence Breeze between Communication Manager and the SIP station. Therefore, the snap-in will be applied only to incoming calls over a SIP trunk.

2. General Test Approach and Test Results

Testing included validation of correct operation of the snap-in by placing inbound calls over a SIP trunk to the Implicit Users defined on Breeze. Calls are routed to Breeze and if the number dialed is listed as an Implicit User the call will invoke the snap-in and a database lookup will be performed. These users, being present on the snap-in database, are associated with an email address.

The attributes of the snap-in allows the lookup on Exchange server of each user's calendar to check if a meeting or an appointment is scheduled. The snap-in checks the subject of this meeting/appointment and it routes the call based on the information provided in the subject. When setting up a meeting/appointment the user will place a number in the subject, for example {7000}, this will allow the call to be re-routed to 7000.

All calls are incoming over a SIP trunk to Session Manager. Session Manager is then responsible for routing the call to Breeze, where the snap-in is activated and the lookup on Exchange server is carried out. The call is then re-routed as per the information provided in the subject of the meeting or appointment.

The snap-in database:

- Pridis provided a build-in Cluster Database within the same snap-in (as described in Chapter 3 of the Avaya Breeze Snap-in Development Guide). This database was installed with the snap-in and will be added to the Service Databases tab on the System Manager.
- This database comes with a table that is already configured to work with the Outlook Calendar Routing Snap-in and has the default settings already placed in the Attributes page on Breeze, (see **Section 5.3**).
- The database table needs to be updated for the customer specific configuration. This is explained in **Section 8.1**.

DevConnect Compliance Testing is conducted jointly by Avaya and DevConnect members. The jointly-defined test plan focuses on exercising APIs and/or standards-based interfaces pertinent to the interoperability of the tested products and their functionalities. DevConnect Compliance Testing is not intended to substitute full product performance or feature testing performed by DevConnect members, nor is it to be construed as an endorsement by Avaya of the suitability or completeness of a DevConnect member's solution.

Avaya recommends our customers implement Avaya solutions using appropriate security and encryption capabilities enabled by our products. The testing referenced in this DevConnect Application Note included the enablement of supported encryption capabilities in the Avaya products. Readers should consult the appropriate Avaya product documentation for further information regarding security and encryption capabilities supported by those Avaya products.

Support for these security and encryption capabilities in any non-Avaya solution component is the responsibility of each individual vendor. Readers should consult the appropriate vendor-supplied product documentation for more information regarding those products.

2.1 Interoperability Compliance Testing

Three Communication Manager phones were used during compliance testing, a 96x1 H.323 deskphone, a 96x1 SIP deskphone and an Avaya Vantage running Avaya Equinox software. These extensions are matched with three Microsoft Exchange email addresses. These three extensions are also setup as Implicit Users on System Manager to route the calls to Breeze and invoke the snap-in. Once these extensions are called the snap-in is called upon and the database lookup begins with the corresponding email address used to check the calendar for meetings/appointments.

The interoperability compliance test included both feature functionality and serviceability testing. The feature functionality testing focused on verifying the calls are routed properly according to the Exchange/Outlook calendar of that particular user/extension. Calls are made to the extensions over a SIP trunk when the users are free, i.e., no meetings or appointments are scheduled and when a meeting or an appointment is scheduled so the call is re-routed. The serviceability testing focused on verifying the ability of the snap-in to recover from adverse conditions, such as disconnecting the Ethernet cable from the Breeze and the Session Manager servers.

2.2 Test Results

All test cases passed successfully. The following observations were noted.

1. Breeze does not support explicit SIP sequencing, it's not possible to sequence Breeze between Communication Manager H.323 and SIP stations. Internal calls are not supported and ISDN calls are not supported using this type of setup/snap-in. There is a possibility to route incoming ISDN trunk calls out to Session Manager and back to Communication Manager over a SIP trunk, so the snap-in is activated, but that setup was not tested and is outside the scope of these Application Notes. All calls made were incoming over a simulated SIP Trunk.
2. When a call is re-routed to another Communication Manager extension there is no information given to the routed station that the call was being re-routed. It appears as a new call.

2.3 Support

Technical support can be obtained for Pridis as follows:

- Email: support@pridis.com
- Website: www.pridis.com
- Phone: +31 (0) 33 2475715

3. Reference Configuration

The configuration in **Figure 1** was used to compliance test Outlook Calendar Routing Snap-in with Avaya Breeze™.

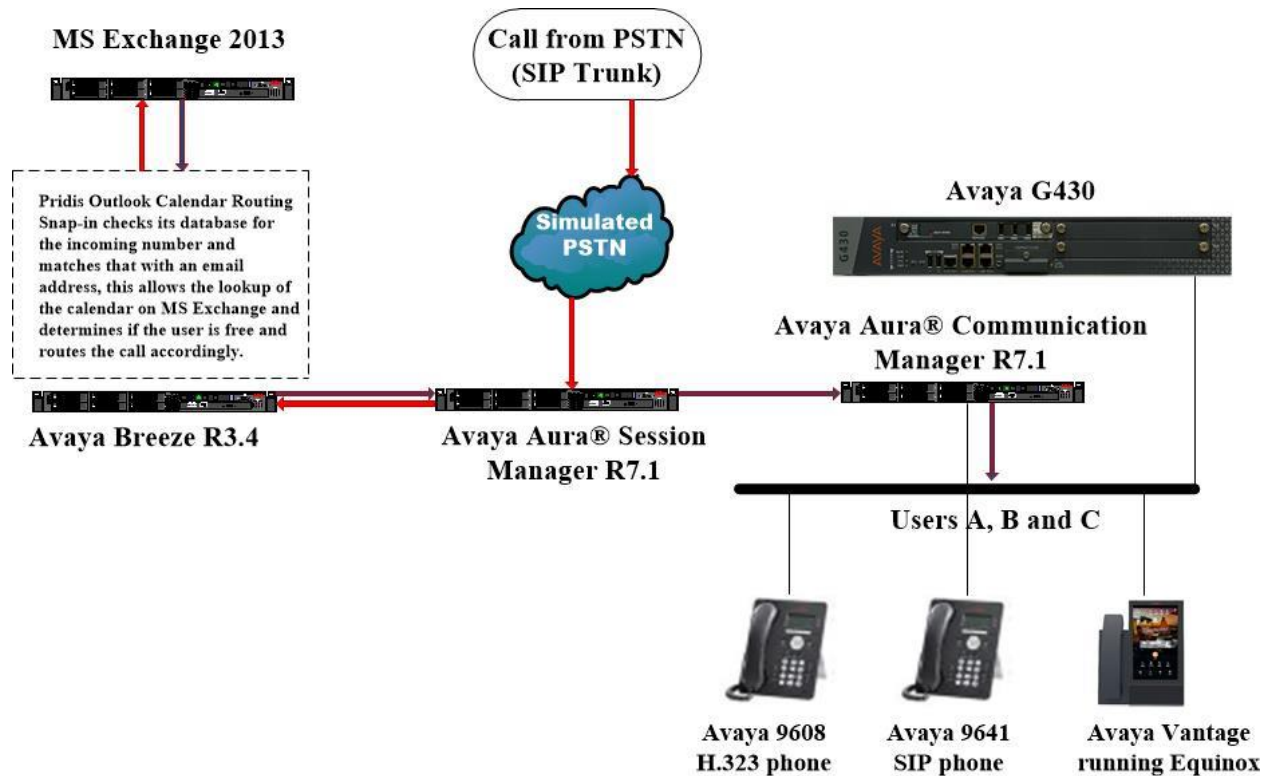


Figure 1: Connection of Pridis Outlook Calendar Routing Snap-in with Avaya Breeze™

4. Equipment and Software Validated

All the hardware and associated software used in the compliance testing is listed below.

Avaya Equipment	Software / Firmware Version
Avaya Aura® System Manager running on a virtual server	System Manager 7.1.3.0 Build No. - 7.1.0.0.1125193 Software Update Revision No: 7.1.3.0.37763 Feature Pack 3
Avaya Aura® Session Manager running on a virtual server	Session Manager R7.1 Build No. – 7.1.3.0.713014
Avaya Aura® Communication Manager running on a virtual server	R7.1 R017x.01.0.532.0 01.0.532.0-24515
Avaya Media Gateway G430	37.42.0/1
Avaya 96x1 H.323 Deskphone	96x1 H323 Release 6.6604
Avaya 96x1 SIP Deskphone	96x1 SIP Release 7.1.2.0.14
Avaya Vantage running Equinox	3.3.1
Avaya Breeze	3.4 Build 3.4.0.0.340003
Microsoft Exchange 2013 running on Windows 2012 R2 Standard Server	Exchange 2013
Pridis Outlook Calendar Routing Snap-in	1.1

5. Configure Avaya Breeze™

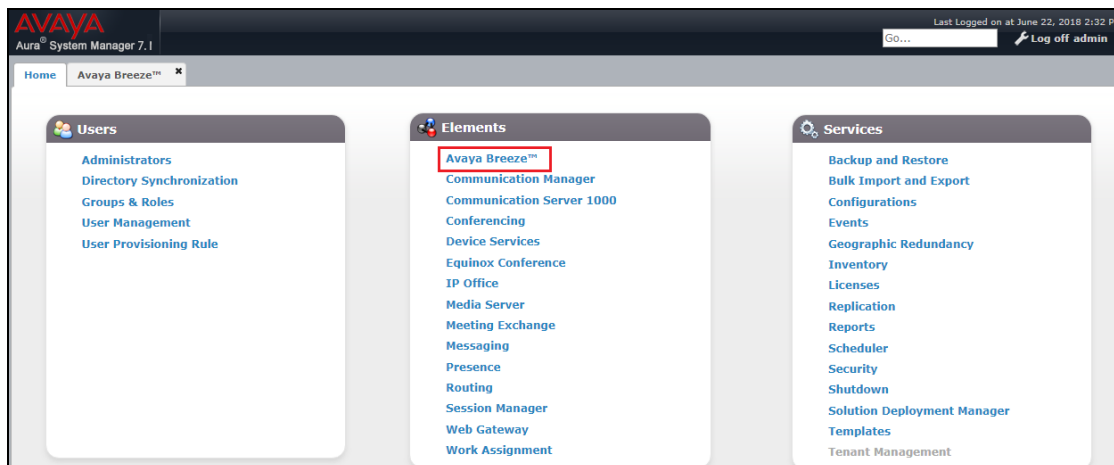
This section provides the procedures for configuring the snap-in on Breeze. Breeze and its snap-ins are configured via System Manager. The configuration on Breeze includes the following:

- Install the Outlook Calendar Routing Snap-in
- Create a Service Profile
- Configure Attributes
- Add Implicit User Profiles

Note: These Application Notes do not illustrate the deployment and setup of the Breeze Server or the Breeze Cluster. It is assumed that this is already in place and the snap-in will be installed on an existing Breeze Cluster.

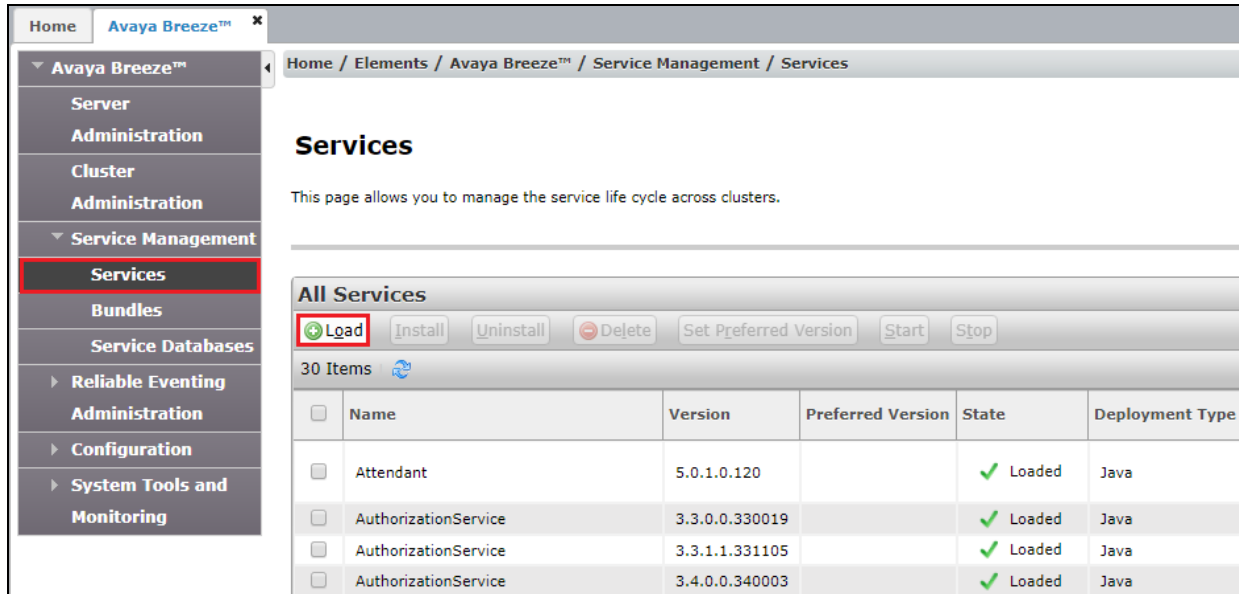
Access System Manager using a web browser by entering **http://<FQDN>/SMGR**, where **<FQDN>** is the fully qualified domain name of System Manager or **http://<IP Address>/SMGR**. Log in using appropriate credentials.

Select **Avaya Breeze™** from the **Elements** menu as shown below.

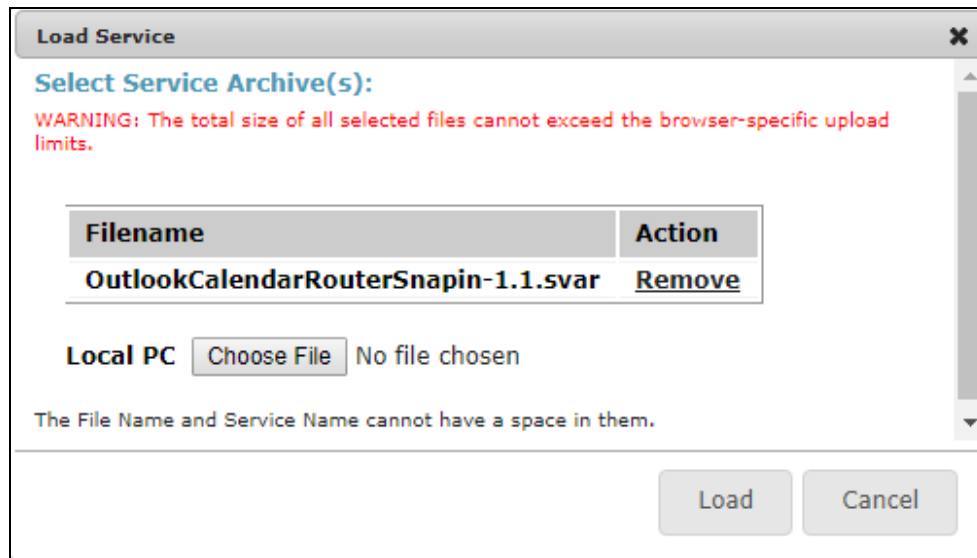


5.1 Install Outlook Calendar Routing Snap-in

Navigate to **Service Management** → **Services** in the left window. From the main window, click on **Load**.



Browse to the folder where the snap-in is located and click on **Load**.



During the loading of the snap-in, a license agreement page is shown where the user can accept the terms and conditions of the snap-in. Once the snap-in is loaded, the snap-in can then be installed onto a Breeze cluster. After the snap-in is loaded successfully, it will appear in the list of services as loaded. This snap-in must now be installed. The snap-in is selected, as shown below, and **Install** is clicked to begin the installation.

All Services								
<input type="button" value="Load"/> <input type="button" value="Install"/> <input type="button" value="Uninstall"/> <input type="button" value="Delete"/> <input type="button" value="Set Preferred Version"/> <input type="button" value="Start"/> <input type="button" value="Stop"/>								
30 Items								Filter: Enable
<input type="checkbox"/>	Name	Version	Preferred Version	State	Deployment Type	License Mode	Avaya Signed	Log Size(MB)
<input type="checkbox"/>	Attendant	5.0.1.0.120		✓ Loaded	Java	2018/05/16 16:01:45 IST	✓	100
<input type="checkbox"/>	AuthorizationService	3.3.0.0.330019		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	AuthorizationService	3.3.1.1.331105		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	AuthorizationService	3.4.0.0.340003		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	AuthorizationService	3.4.0.1.340120		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	CallEventControl	3.3.0.0.330019		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	CallEventControl	3.3.1.1.331105		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	CallEventControl	3.4.0.0.340003		✓ Installed	Java	Not Applicable	✓	100
<input type="checkbox"/>	CallEventControl	3.4.0.1.340120		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	EmailConnector	3.3.0.0.330019		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	EmailConnector	3.3.1.1.331105		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	EmailConnector	3.4.0.0.340003		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	EmailConnector	3.4.0.1.340120		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	EventingConnector	3.3.0.0.330019		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	EventingConnector	3.3.1.1.331105		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	EventingConnector	3.4.0.0.340003		✓ Installed	Java	Not Applicable	✓	100
<input type="checkbox"/>	EventingConnector	3.4.0.1.340120		✓ Loaded	Java	Not Applicable	✓	100
<input type="checkbox"/>	HelloWorld	3.3.0.0.0		✓ Loaded	Java	Not Applicable	Not Signed	100
<input type="checkbox"/>	HelloWorld	3.3.0.0.330019		✓ Loaded	Java	Not Applicable	Not Signed	100
<input type="checkbox"/>	HelloWorld	3.3.1.1.331105		✓ Loaded	Java	Not Applicable	Not Signed	100
<input type="checkbox"/>	HelloWorld	3.4.0.0.340003		✓ Loaded	Java	Not Applicable	Not Signed	100
<input type="checkbox"/>	HelloWorld	3.4.0.1.340120		✓ Loaded	Java	Not Applicable	Not Signed	100
<input checked="" type="checkbox"/>	OutlookCalendarRouterSnapin	1.1		✓ Loaded	Java	Not Applicable	Not Signed	100
<input type="checkbox"/>	PresenceServices	7.1.2.0.224		✓ Installed	Java	Not Applicable	✓	100
<input type="checkbox"/>	ScopiaConnector	3.3.0.0.330019		✓ Loaded	Java	Not Applicable	✓	100
Select : All, None								
								Page 1 of 2

Select the cluster that the snap-in will be installed on. Click on **Commit**.

Confirm Install service: OutlookCalendarRouterSnapin-1.1

1 Item

Filter: Enable

<input checked="" type="checkbox"/>	Cluster Name
<input type="checkbox"/>	For Presence
<input checked="" type="checkbox"/>	Pridis Calendar

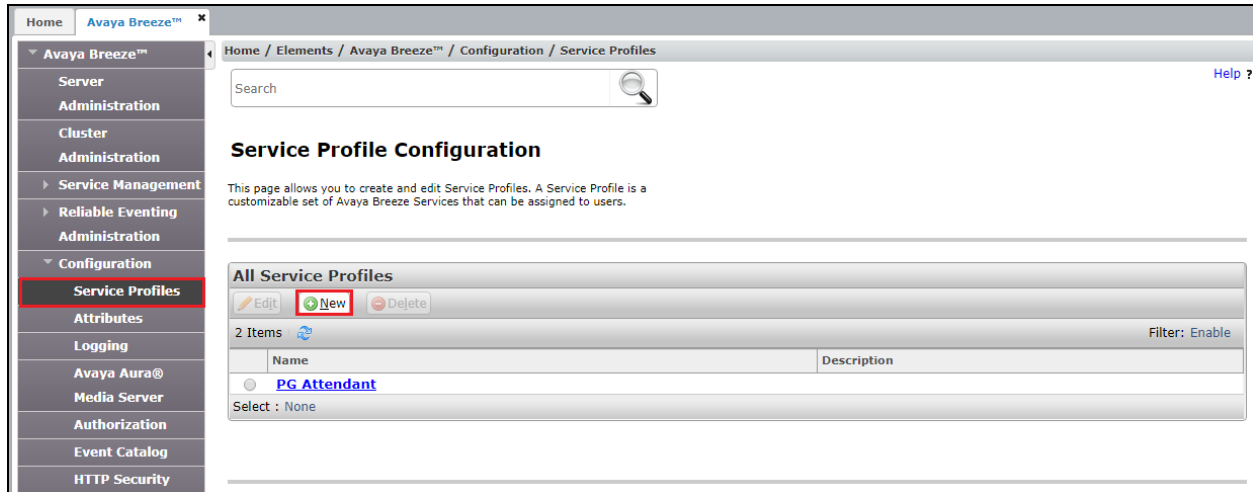
Select : All, None

Commit

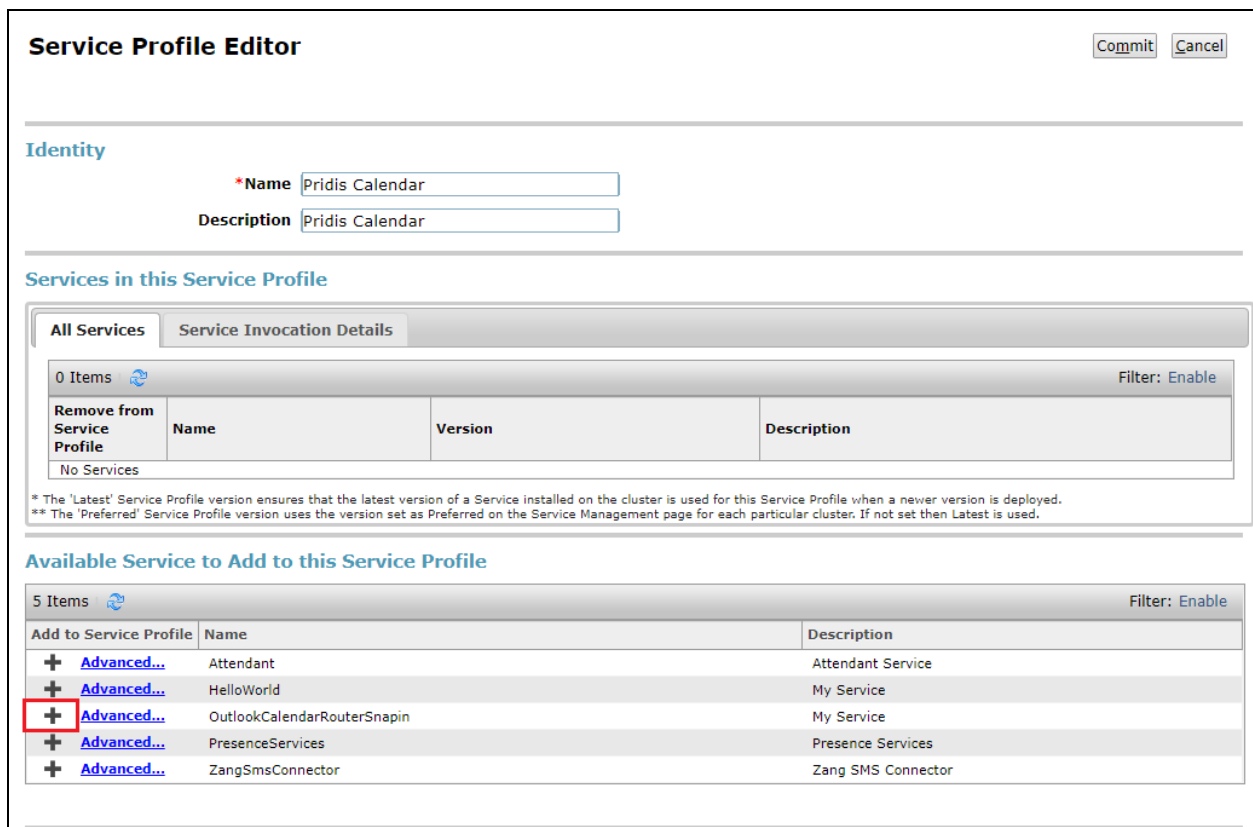
Cancel

5.2 Create a Service Profile

Navigate to **Configuration** → **Service Profiles** from the left window. In the main window, select **New** to add a new Service Profiles for the snap-in.



Enter a suitable **Name** for the profile and add the new snap-in. Click on the + icon beside the new snap-in to add this to the profile.



With the new Service added to the profile, click on **Commit**.

Service Profile Editor

CommitCancel

Identity

*Name

Pridis Calendar

Description

Pridis Calendar

Services in this Service Profile

All ServicesService Invocation Details

1 ItemFilter: Enable

Remove from Service Profile	Name	Version	Description
X	OutlookCalendarRouterSnapin	Latest*	My Service

* The 'Latest' Service Profile version ensures that the latest version of a Service installed on the cluster is used for this Service Profile when a newer version is deployed.

** The 'Preferred' Service Profile version uses the version set as Preferred on the Service Management page for each particular cluster. If not set then Latest is used.

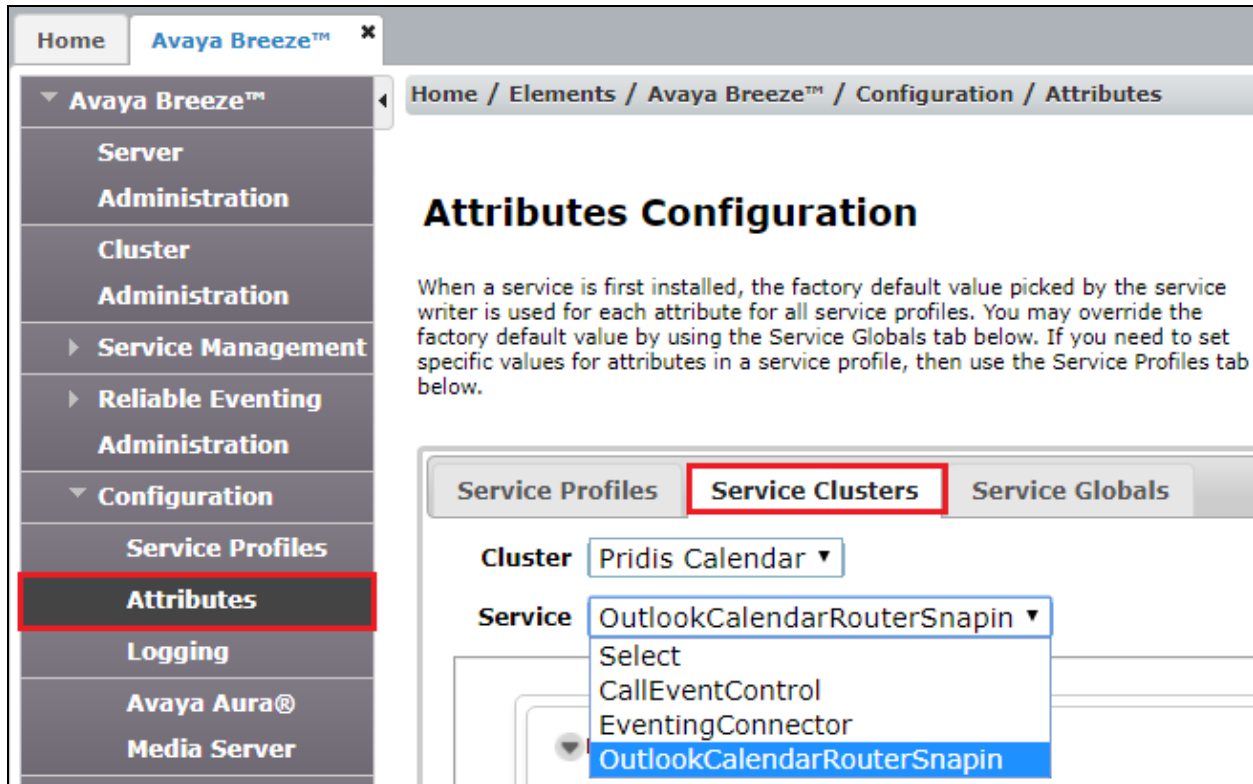
Available Service to Add to this Service Profile

5 ItemsFilter: Enable

Add to Service Profile	Name	Description
+ Advanced...	Attendant	Attendant Service
+ Advanced...	HelloWorld	My Service
+ Advanced...	OutlookCalendarRouterSnapin	My Service
+ Advanced...	PresenceServices	Presence Services
+ Advanced...	ZangSmsConnector	Zang SMS Connector

5.3 Configure Attributes

Navigate to **Configuration** → **Attributes** in the left window. From the main window, select the **Service Clusters** tab, highlighted below. Select the Cluster that the snap-in is installed on (**Pridis Calendar**) and select the snap-in.



Many of the Attributes are already filled in and may be correct but there are some that need to be amended as shown below.

- **License key** – This will be provided by Pridis.
- **Master email** – This is the email account created in **Section 7.1** to allow the snap-in access to the Exchange server and check the user calendars.
- **Master Password** – This is the password associated with the master email created.
- **Outlook Exchange URL** – This is the URL used to gain access to the Exchange server.

Service Profiles | Service Clusters | Service Globals

Cluster: Pridis Calendar

Service: OutlookCalendarRouterSnapin

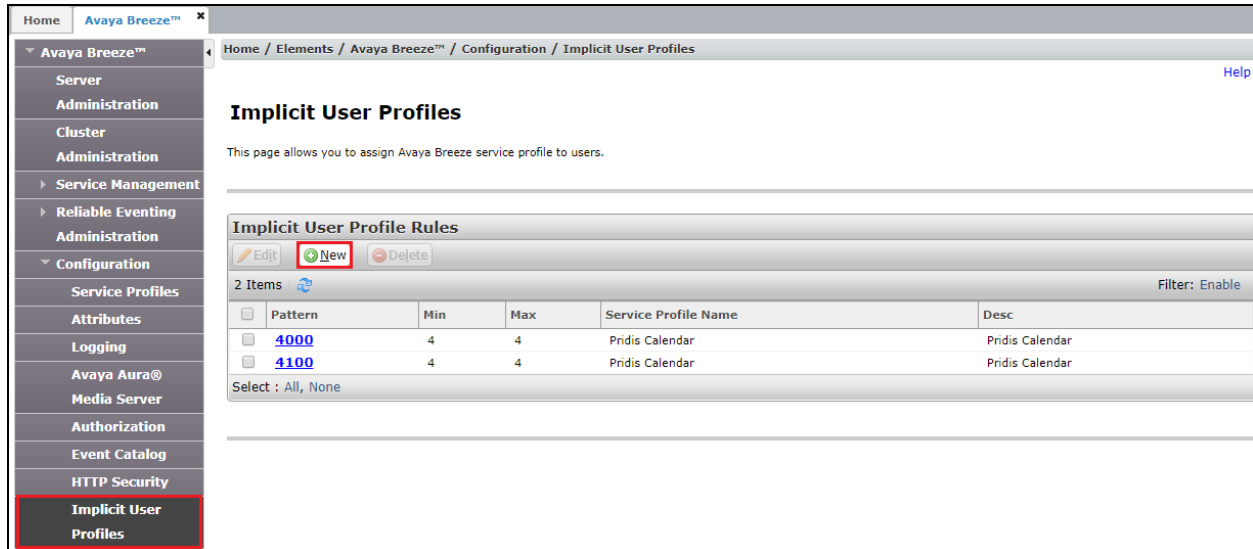
DEFAULT_GROUP

12 Items

Name	Override Default	Effective Value	Description
Announcement Message	<input type="checkbox"/>		A default announcement message that will be played if the call has to be redirected
Database Password	<input type="checkbox"/>	*****	The password for the database account
Database table name	<input type="checkbox"/>	Emails	The name of the database table.
Database URL	<input type="checkbox"/>	jdbc:postgresql://edphadbmaster:5433/outlookcalendaroutersnapin_callroutingdb	The URL to the JDBC database server. jdbc:dbtype://ip:port/dbname
Database Username	<input type="checkbox"/>	outlookcalendaroutersnapin	The username for the database account
Default truststore location	<input type="checkbox"/>	/usr/java/default/jre/lib/security/cacerts	Location of the cacerts file within the servers. Empty for default location
Default truststore password	<input type="checkbox"/>	*****	Password of the cacerts file within the servers. Empty for default password
License key	<input checked="" type="checkbox"/>	*****	License key that holds the license info for the snap-in.
Master email	<input checked="" type="checkbox"/>	pridismaster@devconnect.local	The email address for the master outlook account
Master password	<input checked="" type="checkbox"/>	*****	The password for the master outlook account
Outlook Exchange URL	<input checked="" type="checkbox"/>	https://exchange2013/	The URL to the Outlook webmail server. (don't include the "/owa")
Supplier Id	<input type="checkbox"/>	*****	Avaya provided supplier id

5.4 Add Implicit User Profiles

Navigate to **Configuration → Implicit User Profiles** in the left window. From the main window click on **New**.



Home / Elements / Avaya Breeze™ / Configuration / Implicit User Profiles

Implicit User Profiles

This page allows you to assign Avaya Breeze service profile to users.

Implicit User Profile Rules

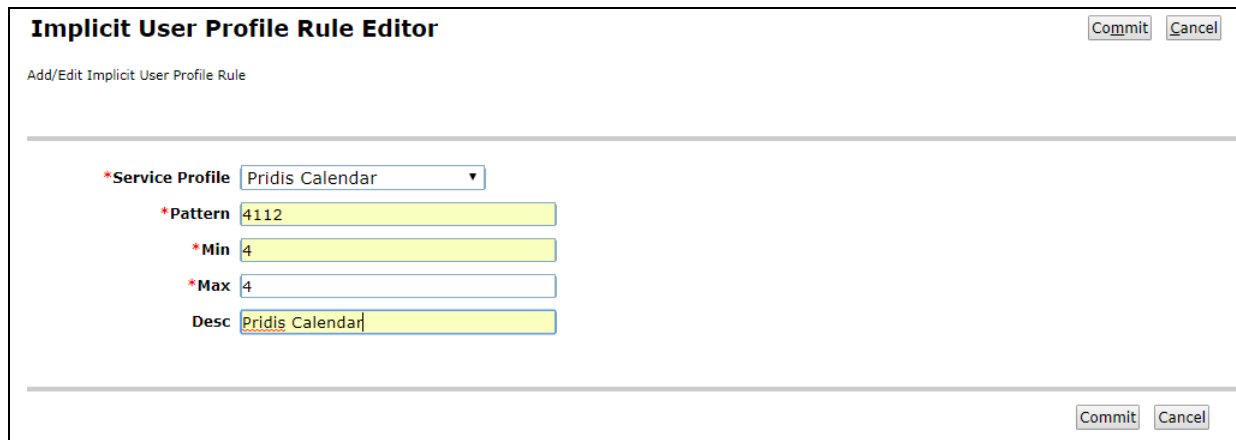
Edit New Delete

2 Items Filter: Enable

<input type="checkbox"/>	Pattern	Min	Max	Service Profile Name	Desc
<input type="checkbox"/>	4000	4	4	Pridis Calendar	Pridis Calendar
<input type="checkbox"/>	4100	4	4	Pridis Calendar	Pridis Calendar

Select : All, None

Select the **Service Profile** created in **Section 5.2**. Enter the appropriate **Pattern** and once completed click on **Commit**.



Implicit User Profile Rule Editor

Add/Edit Implicit User Profile Rule

*Service Profile: Pridis Calendar

*Pattern: 4112

*Min: 4

*Max: 4

Desc: Pridis Calendar

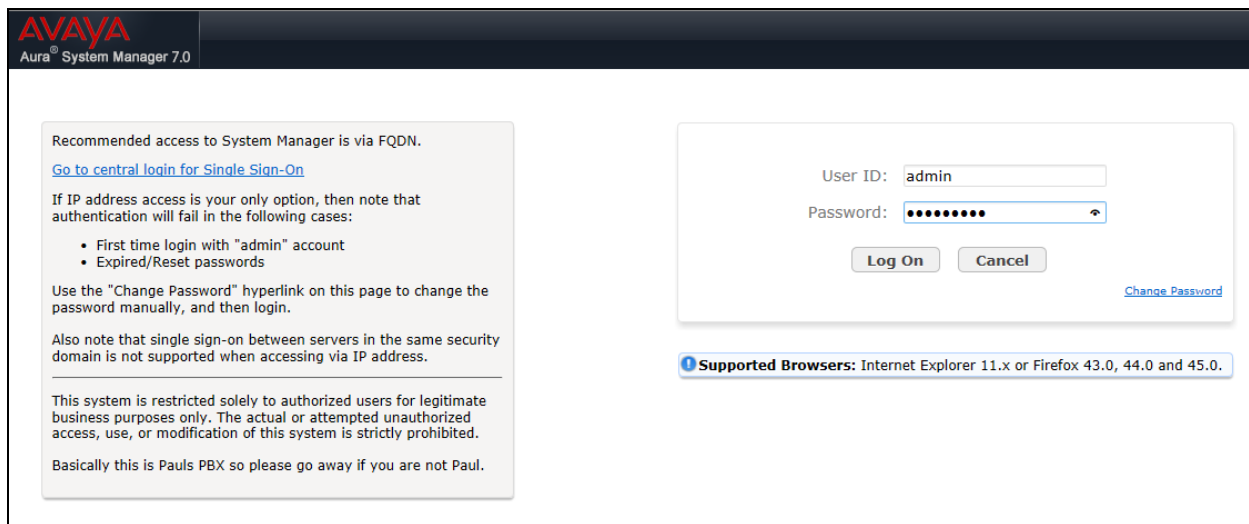
Commit Cancel

6. Configuring Avaya Aura® Session Manager

This section provides the procedures for configuring Session Manager. Like with Breeze in **Section 5**, Session Manager is configured via System Manager. The procedures include the following:

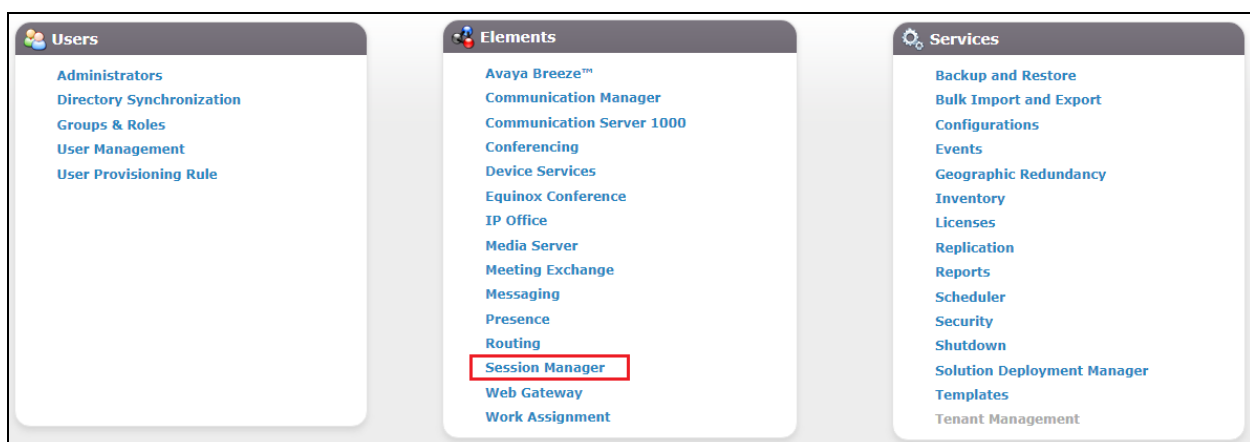
- Configure Application
- Configure Application Sequence
- Configure Implicit Users

Access System Manager using a web browser by entering **http://<FQDN>/SMGR**. Log in using appropriate credentials.



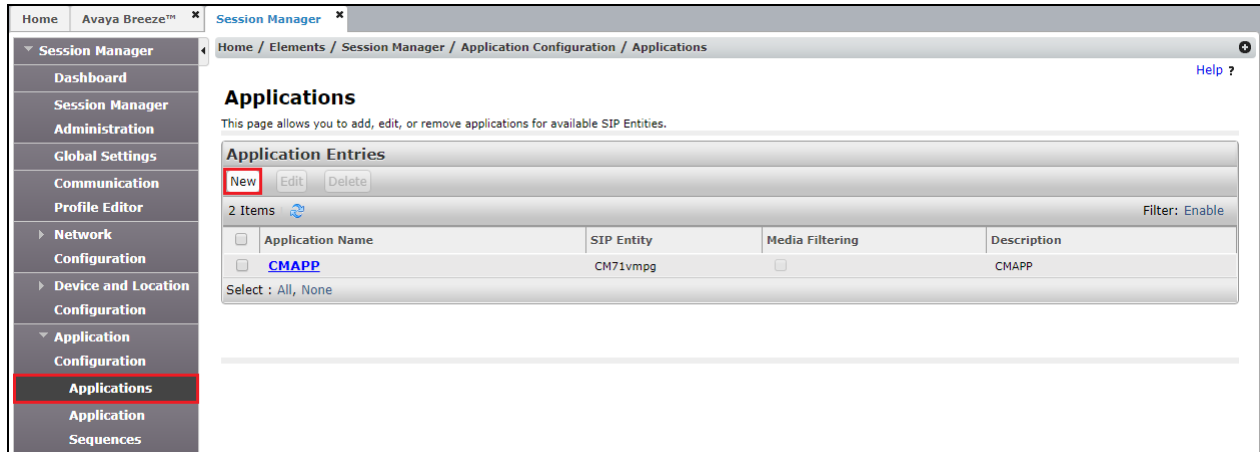
The screenshot shows the Avaya Aura System Manager 7.0 login interface. On the left, there is a text box with instructions: "Recommended access to System Manager is via FQDN. Go to central login for Single Sign-On. If IP address access is your only option, then note that authentication will fail in the following cases: First time login with 'admin' account, Expired/Reset passwords. Use the 'Change Password' hyperlink on this page to change the password manually, and then login. Also note that single sign-on between servers in the same security domain is not supported when accessing via IP address. This system is restricted solely to authorized users for legitimate business purposes only. The actual or attempted unauthorized access, use, or modification of this system is strictly prohibited. Basically this is Pauls PBX so please go away if you are not Paul." On the right, there is a login form with fields for "User ID" (containing "admin") and "Password" (masked with dots). Below the fields are "Log On" and "Cancel" buttons, and a "Change Password" link. At the bottom, a banner states "Supported Browsers: Internet Explorer 11.x or Firefox 43.0, 44.0 and 45.0."

From the main menu click on **Session Manager** highlighted below.



6.1 Add a New Application

From the left window, navigate to **Application Configuration** → **Applications**, from the main window click in **New**.



Enter a suitable **Name** and add the appropriate **SIP Entity**, this will be the SIP Entity of the Breeze server associated with the Pridis Calendar Cluster. Click on **Commit** to save it.

Application Editor Commit Cancel

Application

*Name

*SIP Entity

Description

Application Attributes (optional)

Name	Value
Application Handle	<input type="text"/>
URI Parameters	<input type="text"/>

Application Media Attributes

Enable Media Filtering ☐

Audio	Video	Text	Match Type	If SDP Missing
<input type="text" value="YES"/>	<input type="text" value="YES"/>	<input type="text" value="YES"/>	<input type="text" value="NOT_EXACT"/>	<input type="text" value="ALLOW"/>

*Required Commit Cancel

6.2 Add a new Application Sequence

Navigate to **Application Configuration** → **Application Sequences**. Click on **New** as shown.

The screenshot shows the 'Application Sequences' page in the Session Manager. The left sidebar has a menu with 'Application Sequences' highlighted. The main content area has a breadcrumb trail: 'Home / Elements / Session Manager / Application Configuration / Application Sequences'. Below the title 'Application Sequences', there is a sub-header 'Application Sequences' and a description: 'This page allows you to add, edit, or remove sequences of applications.' There are three buttons: 'New' (highlighted with a red box), 'Edit', and 'Delete'. Below these buttons, it says '2 Items' with a refresh icon. A table lists the items:

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	AppSEQ_PG	AppSEQ_PG

Below the table, it says 'Select : All, None'.

Enter a suitable **Name**. Add the Application created above by clicking on the + icon at the bottom of the screen. Also add the existing Application for Communication Manager Sequencing. The two Applications should be associated with this Application Sequence with the Pridis Application at the top, that is important to note.

The screenshot shows the 'Application Sequence Editor' window. At the top, there are 'Commit' and 'Cancel' buttons. The main section is titled 'Application Sequence' and contains two input fields: '*Name' (with value 'AppSeq_Pridis') and 'Description' (with value 'BreezePridis_AppSEQ'). Below this is a section titled 'Applications in this Sequence' with buttons 'Move First', 'Move Last', and 'Remove'. It shows '2 Items' in a table:

<input type="checkbox"/>	Sequence Order (first to last)	Name	SIP Entity	Mandatory	Description
<input type="checkbox"/>	1	BreezePridis_App	Breeze1SIP	<input checked="" type="checkbox"/>	BreezePridis_App
<input type="checkbox"/>	2	CMAPP	CM71vmppg	<input checked="" type="checkbox"/>	CMAPP

Below the table, it says 'Select : All, None'. At the bottom, there is a section titled 'Available Applications' with a 'Filter: Enable' button. It shows '2 Items' in a table:

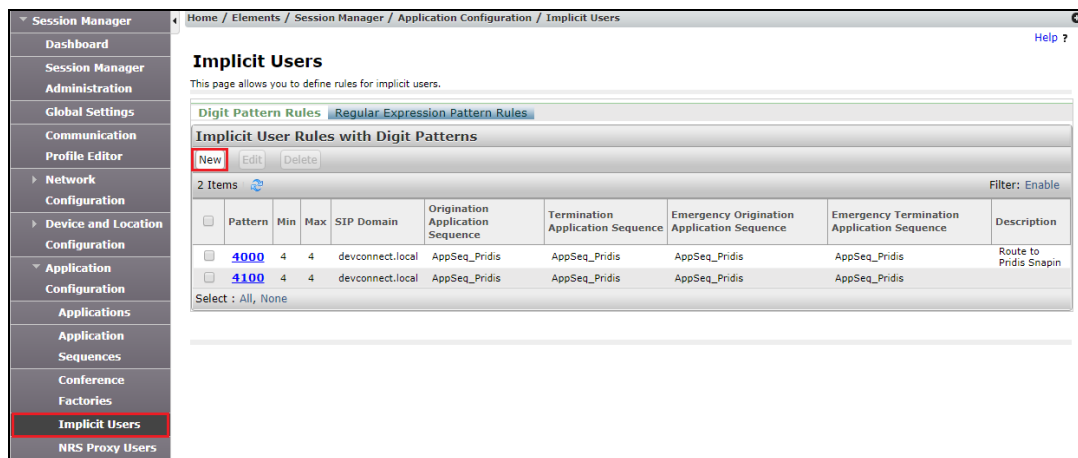
<input type="checkbox"/>	Name	SIP Entity	Description
<input checked="" type="checkbox"/>	BreezePridis_App	Breeze1SIP	BreezePridis_App
<input checked="" type="checkbox"/>	CMAPP	CM71vmppg	CMAPP

At the bottom, there is a '*Required' label and 'Commit' and 'Cancel' buttons.

6.3 Adding Implicit Users

Calls are routed to Breeze using Implicit Users. When calls come into Session Manager over a SIP trunk and the extension is listed in Implicit Users the call will follow the Application Sequence that is associated with the user. In this case the Application Sequence created in **Section 6.2** is associated with the Breeze server and so calls are routed to that Breeze server where the snap-in is activated.

Navigate to **Application Configuration → Implicit Users** and click on **New** from the main window.



The appropriate Pattern is entered with the Application Sequence in **Section 6.2** chosen from the drop-down menus as shown below. Click on **Commit** to save this.

Implicit User Rule Editor

Implicit User Rule

*Pattern: 4112

*Min: 4

*Max: 4

Description: Route to Pridis Snapin

SIP Domain: devconnect.local

Origination Application Sequence: AppSeq_Pridis

Termination Application Sequence: AppSeq_Pridis

Emergency Origination Application Sequence: AppSeq_Pridis

Emergency Termination Application Sequence: Select Termination Application Sequence...
AppSEQ_PG
AppSeq_Pridis

*Required

Commit Cancel

7. Configure Microsoft Exchange/Outlook

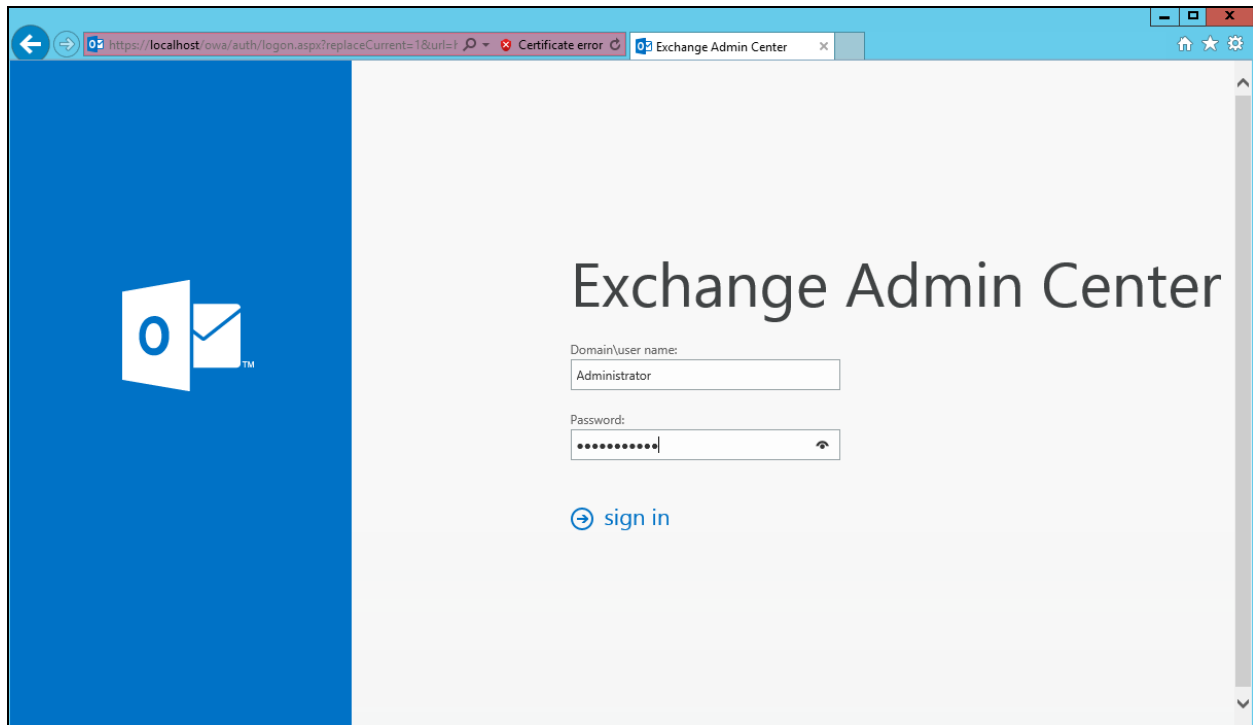
It is assumed that Microsoft Exchange is already installed and configured for a number of users. This section goes through the setup of the 'master email account' for the snap-in. The snap-in uses this account to observe the calendar information on those users who have their calendar shared with the Pridis master email account. Some configurations for Exchange were performed using Microsoft Outlook.

7.1 Configure the Pridis Master Email account

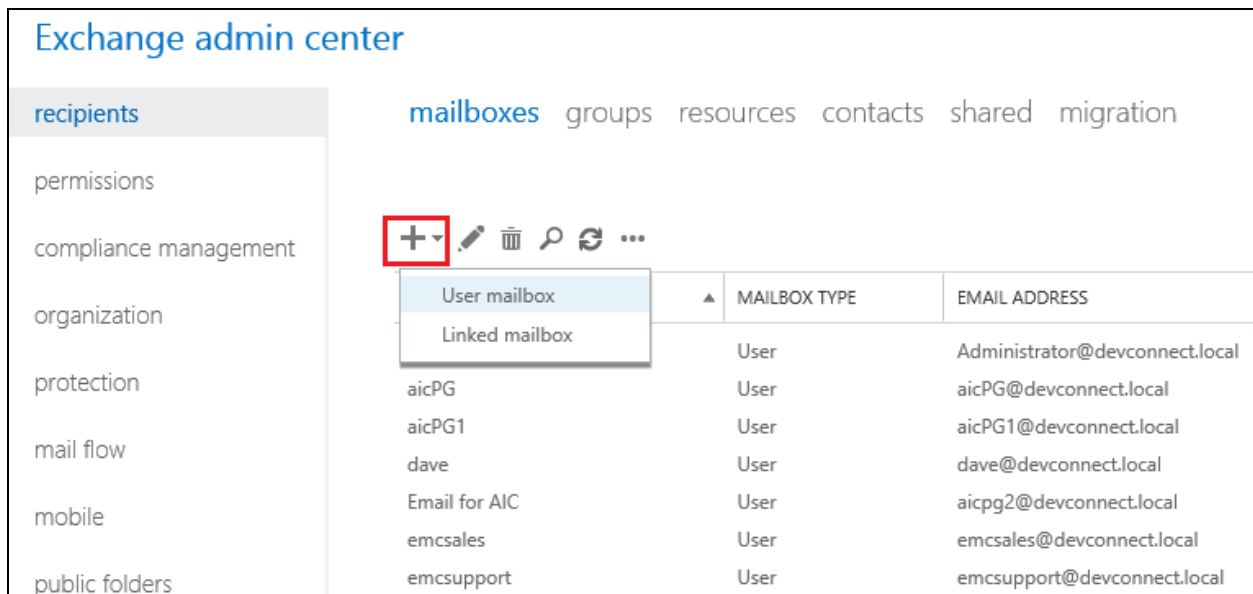
This user should be added to the domain before adding the email account. A user called 'pridismaster' was added to the DevConnect lab domain prior to adding the email account. To add a new email, open the **Exchange Administrative Center** on the Exchange server.



Enter the appropriate credentials and click on **sign in**.



From the main page, click on the + icon to add a new **User mailbox**.



To assign the pridismaster user an email account, either click on **Existing user**, where the user can be searched and added or click on **New user** as shown below, adding the appropriate details. Click on **Save** to complete.

new user mailbox

☐ Existing user

browse...

☒ New user

First name:

Initials:

Last name:

*Display name:

*Name:

Organizational unit:
 browse...

*User logon name:
 @

*New password:

*Confirm password:

☐ Require password change on next logon

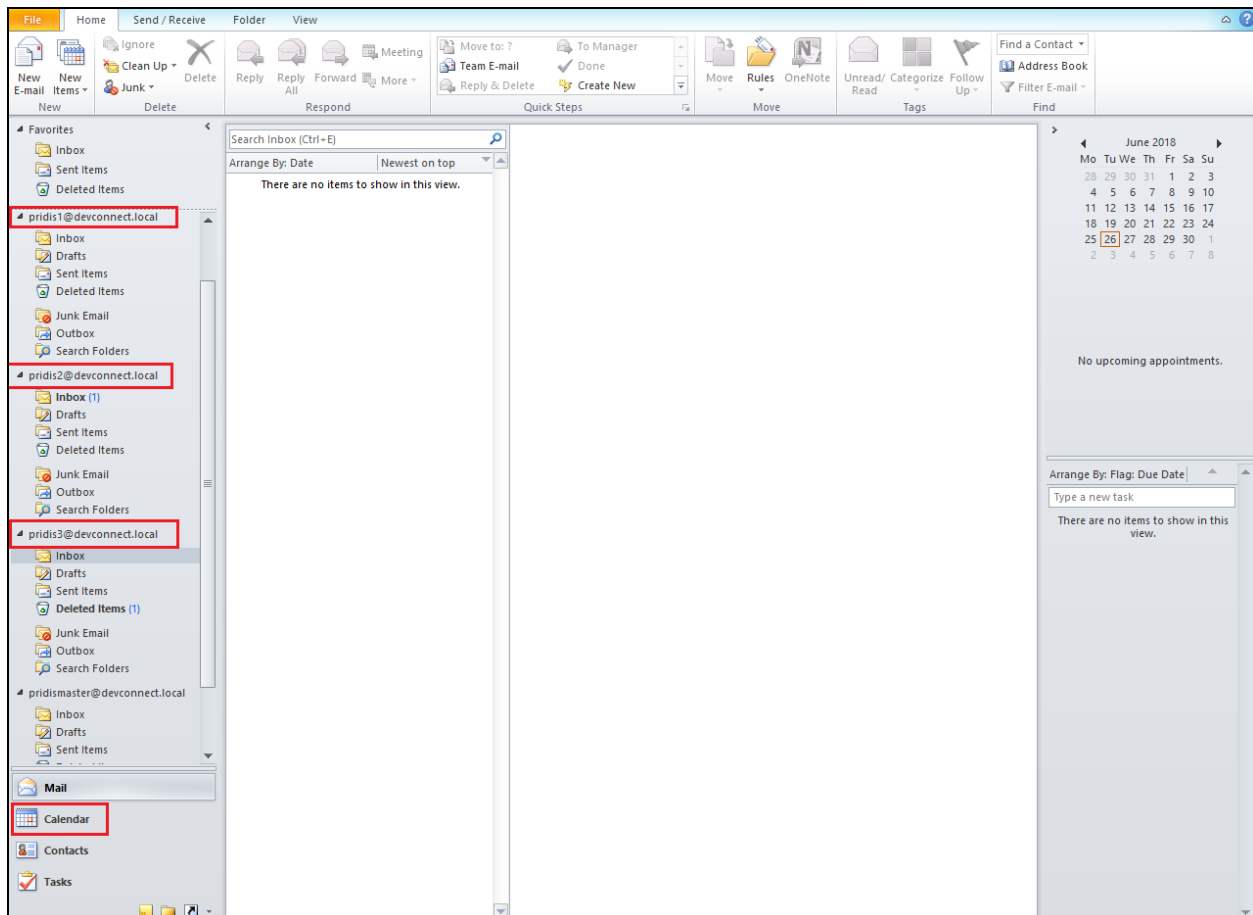
[More options...](#)

7.2 Configure Exchange calendar for sharing

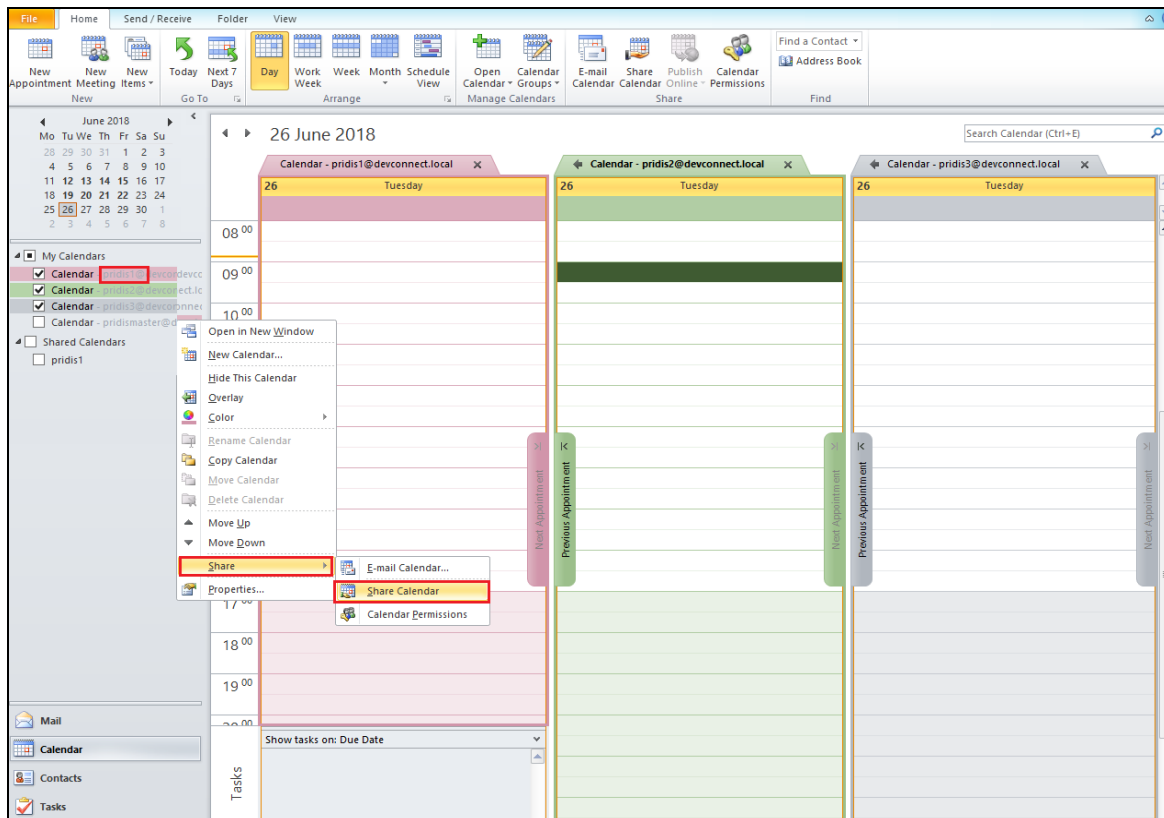
This section shows how to share the calendar of existing Exchange users with the newly added pridismaster email account.

Note: For compliance testing four email accounts were added to Outlook running on the Exchange server, pridismaster, pridis1, pridis2 and pridis3. These being the master email account and the three user accounts associated with the three extensions on Communication Manager.

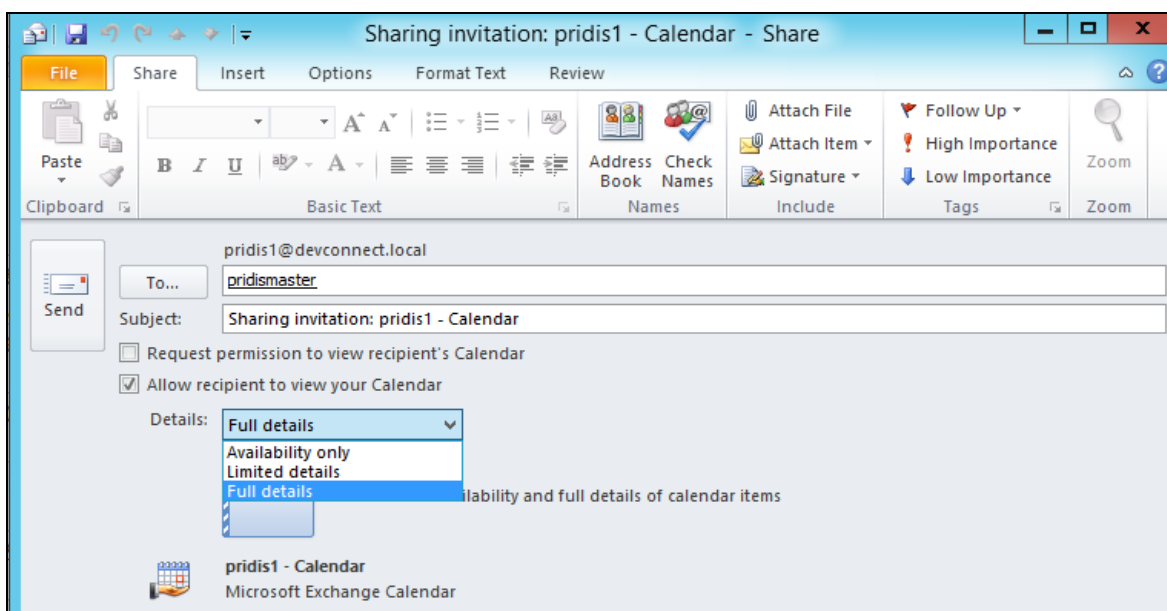
Open Outlook and click on **Calendar** at the bottom left of the screen. Note the three email accounts shown that were configured for compliance testing.



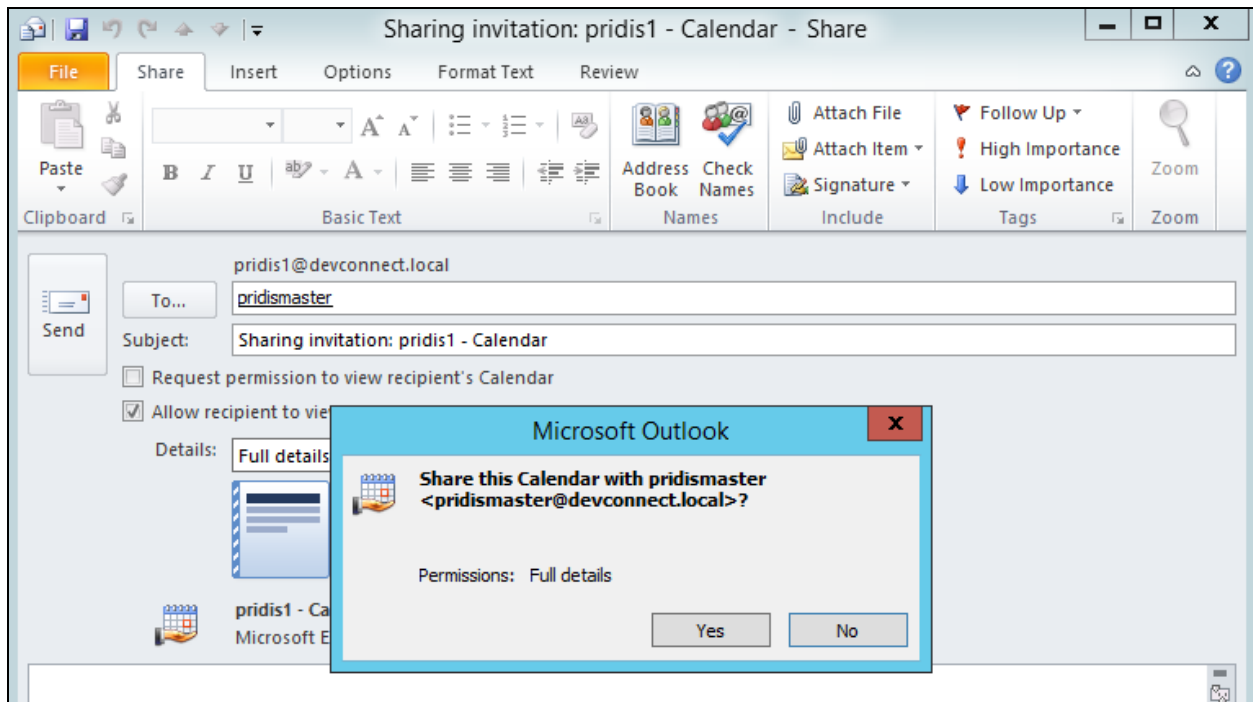
Right click on the account that is to be shared. Select **Share** and **Share Calendar** as shown below.



Enter **pridismaster** (the Pridis master email account) into the **To** field and ensure that **Full details** is chosen for the **Details** drop-down box. **Allow recipient to view your Calendar** is ticked.



Click on **Send** and an email will be sent to pridismaster where the sharing is accepted. Press **Yes** to confirm this as shown below.



8. Configure the Outlook Calendar Routing Snap-in

While there is no direct configuration required on the snap-in, there are changes that need to be made to the database associated with the snap-in. The database must be populated with the extension number and email addresses for the site.

Pridis will need to obtain a public key to generate a license, although this procedure will be carried out by a Pridis engineer it is outlined in **Section 8.2**.

8.1 Configure the Outlook Calendar Routing Snap-in Database

Open the Breeze node with PuTTY (or a similar product using SSH). Log in with the appropriate credentials and execute the command **psql -h edphadbmaster -p 5433 -U outlookcalendaroutersnapin -d outlookcalendaroutersnapin_callroutingdb -W** as shown below.

```
cust@BREEZEvmgpg:~$
login as: cust
This system is restricted solely to authorized users for legitimate business purposes only. The actual or attempted unauthorized access, use, or modification of this system is strictly prohibited.

Unauthorized users are subject to company disciplinary procedures and or criminal and civil penalties under state, federal, or other applicable domestic and foreign laws.

The use of this system may be monitored and recorded for administrative and security reasons. Anyone accessing this system expressly consents to such monitoring and recording, and is advised that if it reveals possible evidence of criminal activity, the evidence of such activity may be provided to law enforcement officials.

All users must comply with all corporate instructions regarding the protection of information assets.
Using keyboard-interactive authentication.
Password:
Last login: Fri Jun 22 13:33:20 IST 2018 from 10.10.40.222 on pts/2
Last login: Tue Jun 26 10:51:40 2018 from 10.10.40.222
[cust@BREEZEvmgpg ~]$ psql -h edphadbmaster -p 5433 -U outlookcalendaroutersnapin -d outlookcalendaroutersnapin_callroutingdb -W
```

Type the password **outlookcalendaroutersnapin01**. It is highly advised to change this password for security. To change the password, use the command **\password** after having logged in. When changing this password change the database password attribute for the snap-in accordingly in **Section 5.3**.

```
cust@BREEZEvmgpg:~$
login as: cust
This system is restricted solely to authorized users for legitimate business purposes only. The actual or attempted unauthorized access, use, or modification of this system is strictly prohibited.

Unauthorized users are subject to company disciplinary procedures and or criminal and civil penalties under state, federal, or other applicable domestic and foreign laws.

The use of this system may be monitored and recorded for administrative and security reasons. Anyone accessing this system expressly consents to such monitoring and recording, and is advised that if it reveals possible evidence of criminal activity, the evidence of such activity may be provided to law enforcement officials.

All users must comply with all corporate instructions regarding the protection of information assets.
Using keyboard-interactive authentication.
Password:
Last login: Fri Jun 22 13:33:20 IST 2018 from 10.10.40.222 on pts/2
Last login: Tue Jun 26 10:51:40 2018 from 10.10.40.222
[cust@BREEZEvmgpg ~]$ psql -h edphadbmaster -p 5433 -U outlookcalendaroutersnapin -d outlookcalendaroutersnapin_callroutingdb -W
Password for user outlookcalendaroutersnapin:
```

Once logged in commands such as **\db** or **\list** can be executed to show the connection is good.

```
Last login: Tue Jun 26 10:51:40 2018 from 10.10.40.222
[cust@BREEZEvmpeg ~]$ psql -h edphadbmaster -p 5433 -U outlookcalendaroutersnapin -d outlookcalendaroutersnapin_callroutingdb -W
Password for user outlookcalendaroutersnapin:
psql (9.6.1)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256, compression: off)
Type "help" for help.

outlookcalendaroutersnapin_callroutingdb-> \db
      List of tablespaces
   Name   | Owner | Location
-----+-----+-----
 pg_default | hadb  |
 pg_global | hadb  |
(2 rows)

outlookcalendaroutersnapin_callroutingdb-> \list
      List of databases
   Name   | Owner          | Encoding | Collate | Ctype  | Access privileges
-----+-----+-----+-----+-----+-----
 attendant_attendantdb | attendant      | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 hadb      | hadb           | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 outlookcalendaroutersnapin_callroutingdb | outlookcalendaroutersnapin | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | hadb           | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | hadb           | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/hadb          +
           |                |          |             |             | hadb=CTc/hadb   +
 template1 | hadb           | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/hadb          +
           |                |          |             |             | hadb=CTc/hadb   +
(6 rows)
```

The following command is run to add the three extensions and associated emails.

INSERT INTO Emails

VALUES

('4000', 'pridis1@devconnect.local'),
('4100', 'pridis2@devconnect.local'),
('4112', 'pridis3@devconnect.local');

The command below adds three rows with 4000 being associated with pridis1@devconnect.local, 4100 with pridis2@devconnect.local and 4112 with pridis3@devconnect.local.

```
outlookcalendaroutersnapin_callroutingdb-> INSERT INTO Emails
outlookcalendaroutersnapin_callroutingdb-> VALUES
outlookcalendaroutersnapin_callroutingdb-> ('4000', 'pridis1@devconnect.local'),
outlookcalendaroutersnapin_callroutingdb-> ( '4100', 'pridis2@devconnect.local'),
outlookcalendaroutersnapin_callroutingdb-> ( '4112', 'pridis3@devconnect.local');
```

Enter the command **SELECT * FROM Emails;**. This should return three rows with the correct information as shown below.

```
outlookcalendaroutersnapin_callroutingdb=> SELECT * FROM Emails;
 called_handle | email_address
-----+-----
 4000          | pridis1@devconnect.local
 4100          | pridis2@devconnect.local
 4112          | pridis3@devconnect.local
(3 rows)

outlookcalendaroutersnapin_callroutingdb=>
```

8.2 Obtain the Public Key for license

Log into the Breeze node using PuTTY or something similar. Once logged in navigate to `/var/log/Avaya/services/OutlookCalendarRouterSnapin/`

```
cust@BREEZEvmpeg:~  
login as: cust  
This system is restricted solely to authorized users for legitimate business purposes only. The actual or attempted unauthorized access, use, or modification of this system is strictly prohibited.  
  
Unauthorized users are subject to company disciplinary procedures and or criminal and civil penalties under state, federal, or other applicable domestic and foreign laws.  
  
The use of this system may be monitored and recorded for administrative and security reasons. Anyone accessing this system expressly consents to such monitoring and recording, and is advised that if it reveals possible evidence of criminal activity, the evidence of such activity may be provided to law enforcement officials.  
All users must comply with all corporate instructions regarding the protection of information assets.  
Using keyboard-interactive authentication.  
Password:  
Last login: Tue Jun 26 10:51:40 IST 2018 from 10.10.40.222 on pts/2  
Last login: Tue Jun 26 11:00:34 2018 from 10.10.40.222  
[cust@BREEZEvmpeg ~]$ cd /var/log/Avaya/services/OutlookCalendarRouterSnapin/
```

Open the **OutlookCalendarRouterSnapin.log** file as shown below.

```
cust@BREEZEvmpeg:/var/log/Avaya/services/OutlookCalendarRouterSnapin  
login as: cust  
This system is restricted solely to authorized users for legitimate business purposes only. The actual or attempted unauthorized access, use, or modification of this system is strictly prohibited.  
  
Unauthorized users are subject to company disciplinary procedures and or criminal and civil penalties under state, federal, or other applicable domestic and foreign laws.  
  
The use of this system may be monitored and recorded for administrative and security reasons. Anyone accessing this system expressly consents to such monitoring and recording, and is advised that if it reveals possible evidence of criminal activity, the evidence of such activity may be provided to law enforcement officials.  
All users must comply with all corporate instructions regarding the protection of information assets.  
Using keyboard-interactive authentication.  
Password:  
Last login: Tue Jun 26 10:51:40 IST 2018 from 10.10.40.222 on pts/2  
Last login: Tue Jun 26 11:00:34 2018 from 10.10.40.222  
[cust@BREEZEvmpeg ~]$ cd /var/log/Avaya/services/OutlookCalendarRouterSnapin/  
[cust@BREEZEvmpeg OutlookCalendarRouterSnapin]$ cat OutlookCalendarRouterSnapin.log
```

The log file will display a line containing the text **Your public key**, this will be followed by the key which can be given to Pridis in order to generate a license for the snap-in.

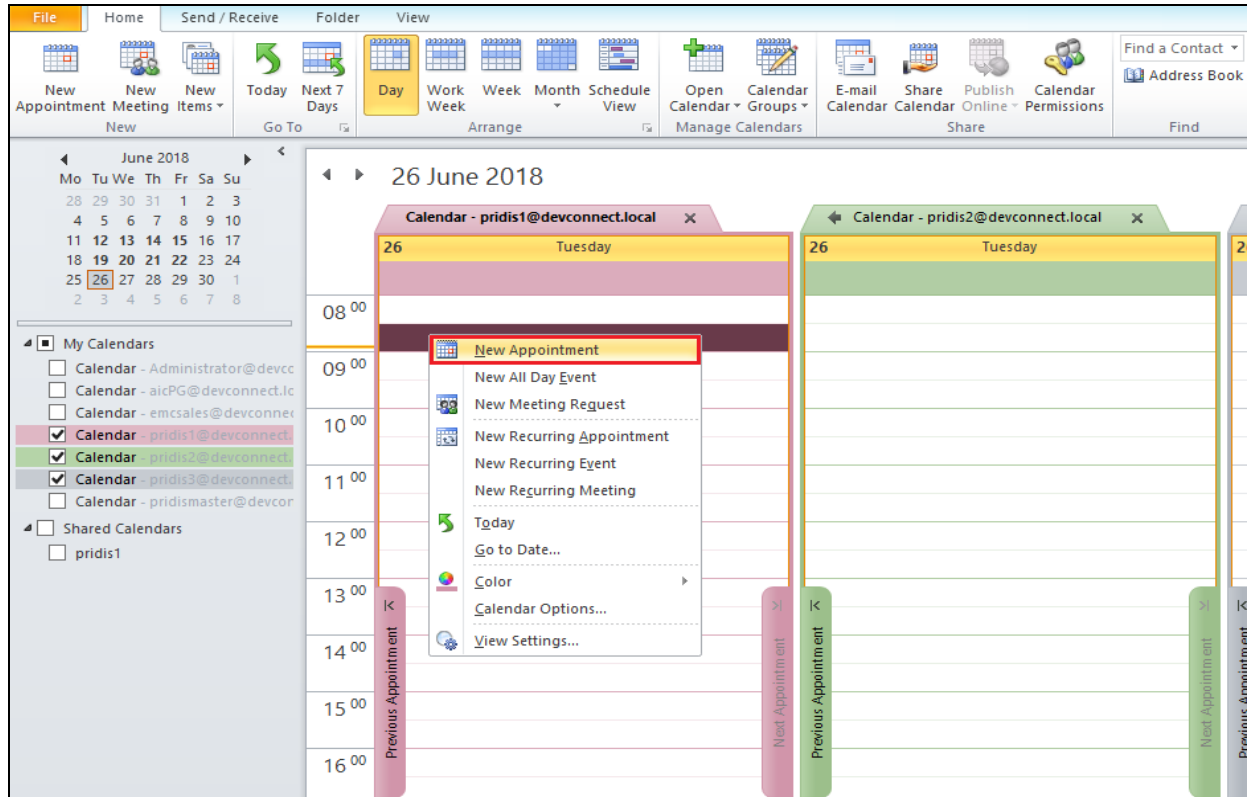
```
cust@BREEZEvmgpg:/var/log/Avaya/services/OutlookCalendarRouterSnapin
e]
2018-06-26 08:57:31,717 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Connected party 4001: com.avaya.collabora
tion.call.ParticipantImpl@791ee78f[participationId=local.1529997635320_748;10.10.40.136-515625135-10.10.40.136, mediaConnectionId=local.1529997635320_748_748
_1, routeId=<null>, handleFarEnd=4001, domainFarEnd=devconnect.local, handleNearEnd=4001, domainNearEnd=devconnect.local, protocolType=SIP, display name=4001
, H323User, participantState=CONNECTED, presented handle=<null>, presented domain=<null>, presented display name=<null>, isAgent=false, hasIndicatedVideoSupp
ort=false]
2018-06-26 08:57:34,903 [SipContainerPool : 2] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Participant dropped cause: PARTICIPANT_DR
OPPED
2018-06-26 08:57:34,903 [SipContainerPool : 2] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Participant dropped participant: com.avaya
a.collaboration.call.ParticipantImpl@791ee78f[participationId=local.1529997635320_748;10.10.40.136-515625135-10.10.40.136, mediaConnectionId=local.1529997635
320_748_748_1, routeId=<null>, handleFarEnd=4001, domainFarEnd=devconnect.local, handleNearEnd=4001, domainNearEnd=devconnect.local, protocolType=SIP, displa
y name=4001, H323User, participantState=IDLE, presented handle=<null>, presented domain=<null>, presented display name=<null>, isAgent=false, hasIndicatedVid
eoSupport=false]
2018-06-26 08:57:34,903 [SipContainerPool : 2] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Participant dropped cause: AFTER_ANSWER
2018-06-26 08:57:34,904 [SipContainerPool : 2] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Participant dropped participant: com.avaya
a.collaboration.call.ParticipantImpl@8772ac01[participationId=local.1529997635320_748;10.10.40.136-788176222-10.10.40.136, mediaConnectionId=local.1529997635
320_748_747_1, routeId=<null>, handleFarEnd=3000, domainFarEnd=devconnect.local, handleNearEnd=3000, domainNearEnd=devconnect.local, protocolType=SIP, displa
y name=<null>, participantState=CONNECTED, presented handle=<null>, presented domain=<null>, presented display name=<null>, isAgent=false, hasIndicatedVideoS
upport=false]
2018-06-26 08:57:34,905 [SipContainerPool : 2] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Call terminated: AFTER_ANSWER
2018-06-26 08:57:38,058 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Service attributes:
2018-06-26 08:57:38,058 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Exchange URL: https://exchange2013/
2018-06-26 08:57:38,058 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Master email address: pridismaster@devcon
nect.local
2018-06-26 08:57:38,059 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Database URL: jdbc:postgresql://edphadba
ster:5433/outlookcalendarrouteroutersnapin_callroutingdb
2018-06-26 08:57:38,059 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Table name: Emails
2018-06-26 08:57:38,059 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Database username: outlookcalendarrouteru
tersnapin
2018-06-26 08:57:38,059 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Announcement message:
2018-06-26 08:57:38,061 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Your public key : vZ05uvRBSYeZ4ciNoddeLBP
XN94vSMRMLIsu7htJd8=
2018-06-26 08:57:38,468 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Showing appointments: count: 1
2018-06-26 08:57:38,525 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - redirecting call to: "4101"
2018-06-26 08:57:40,177 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Call Answered.
2018-06-26 08:57:40,178 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Connected party 3000: com.avaya.collabora
tion.call.ParticipantImpl@a62ccbc7[participationId=local.1529997635320_759;10.10.40.136-15185935-10.10.40.136, mediaConnectionId=local.1529997635320_759_758
_1, routeId=<null>, handleFarEnd=3000, domainFarEnd=devconnect.local, handleNearEnd=3000, domainNearEnd=devconnect.local, protocolType=SIP, display name=<null>
, participantState=CONNECTED, presented handle=<null>, presented domain=<null>, presented display name=<null>, isAgent=false, hasIndicatedVideoSupport=false]
2018-06-26 08:57:40,178 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Connected party 4101: com.avaya.collabora
tion.call.ParticipantImpl@a60d1be8[participationId=local.1529997635320_759;10.10.40.136-363554934-10.10.40.136, mediaConnectionId=local.1529997635320_759_759
_1, routeId=<null>, handleFarEnd=4101, domainFarEnd=devconnect.local, handleNearEnd=4101, domainNearEnd=devconnect.local, protocolType=SIP, display name=4101
, SIPUser, participantState=CONNECTED, presented handle=<null>, presented domain=<null>, presented display name=<null>, isAgent=false, hasIndicatedVideoSuppo
rt=false]
2018-06-26 08:57:42,598 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Participant dropped cause: PARTICIPANT_DR
OPPED
2018-06-26 08:57:42,598 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Participant dropped participant: com.avaya
a.collaboration.call.ParticipantImpl@a60d1be8[participationId=local.1529997635320_759;10.10.40.136-363554934-10.10.40.136, mediaConnectionId=local.1529997635
320_759_759_1, routeId=<null>, handleFarEnd=4101, domainFarEnd=devconnect.local, handleNearEnd=4101, domainNearEnd=devconnect.local, protocolType=SIP, displa
y name=4101, SIPUser, participantState=IDLE, presented handle=<null>, presented domain=<null>, presented display name=<null>, isAgent=false, hasIndicatedVide
oSupport=false]
2018-06-26 08:57:42,598 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Participant dropped cause: AFTER_ANSWER
2018-06-26 08:57:42,598 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Participant dropped participant: com.avaya
a.collaboration.call.ParticipantImpl@a62ccbc7[participationId=local.1529997635320_759;10.10.40.136-15185935-10.10.40.136, mediaConnectionId=local.15299976353
20_759_758_1, routeId=<null>, handleFarEnd=3000, domainFarEnd=devconnect.local, handleNearEnd=3000, domainNearEnd=devconnect.local, protocolType=SIP, display
name=<null>, participantState=CONNECTED, presented handle=<null>, presented domain=<null>, presented display name=<null>, isAgent=false, hasIndicatedVideoSu
pport=false]
2018-06-26 08:57:42,599 [SipContainerPool : 1] OutlookCalendarRouterSnapin INFO - OutlookCalendarRouterSnapin-1.1 - Call terminated: AFTER_ANSWER
[cust@BREEZEvmgpg OutlookCalendarRouterSnapin]$
```

9. Verification Steps

This section provides steps that may be performed to verify that the solution is configured correctly.

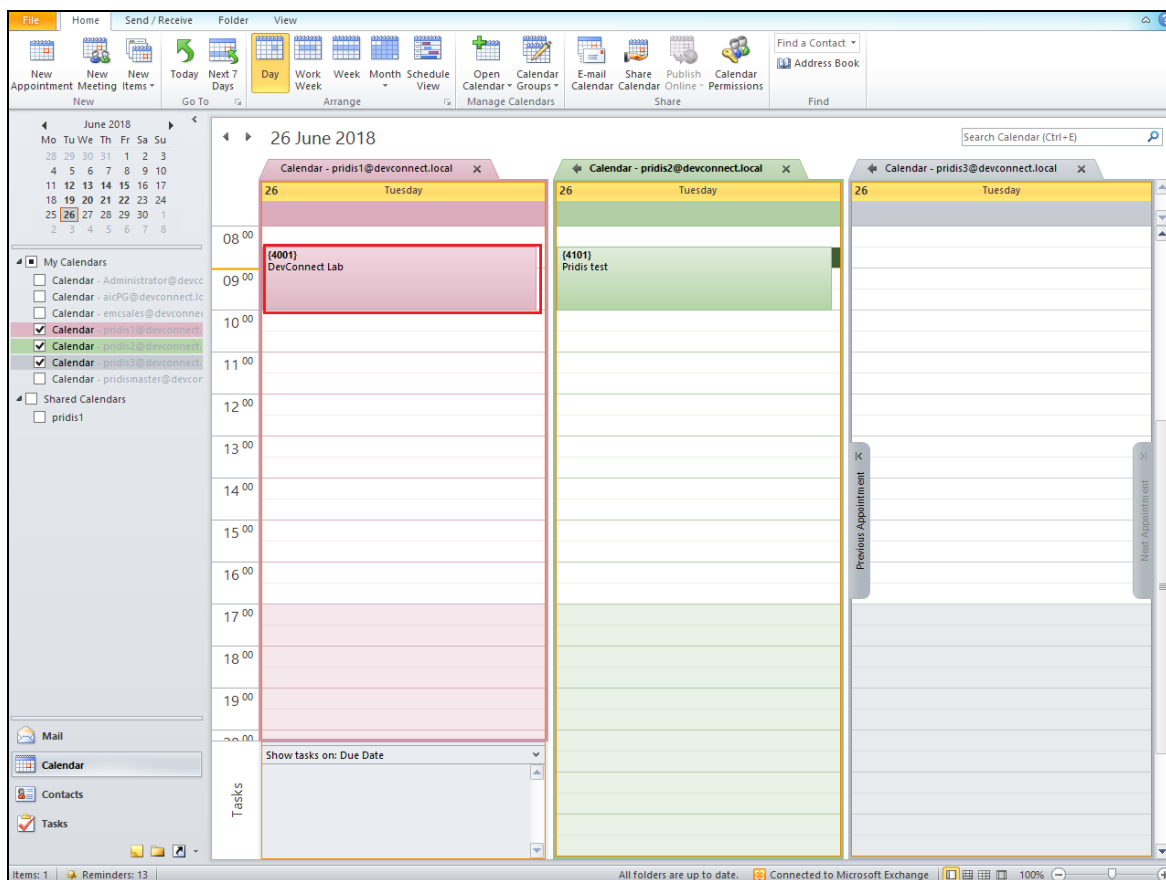
9.1 Create an Appointment on the Outlook calendar

Pick a user and right click in the calendar. The example below shows a **New Appointment** being created for **pridis1**.



The **Subject** is where the re-routing of calls is taken from, enter {xxxx} into the subject header where xxxx is the extension where the call is to be re-routed. In this case **4001** was chosen. The extension number is placed between the two { } brackets. Save the new Appointment by clicking on **Save & Close** at the top left of the screen.

The new Appointment for pridis1 is clearly shown, highlighted below.



9.2 Run TraceCE on Avaya Breeze™

Open an SSH session to the Breeze node as per **Sections 8.1 & 8.2**, using PuTTY or some equivalent program. Once logged in type traceCE and the trace will open allowing the capture of various components, ensure that snap-in is selected. Once the trace was running the following calls were made.

9.2.1 Call to Extension 4100 with no Meeting/Appointment scheduled

A call was made from a simulated PSTN over a SIP trunk to extension 4100. At this time 4100 did not have any Appointment or Meeting scheduled in Outlook. Looking at the trace, the **INVITE** can be observed from Session Manager to Breeze and the Implicit User **4100** is found. The information following that shows the snap-in being activated and the database being looked up with the email calendar being checked. The line **Showing appointments: count: 0** shows there is no appointment and the following line states **No redirection, calling: "4100"**. The call should then proceed to extension 4100 and get answered.

SM71vmpg		SM100
11:06:45.497	INVITE	(2) T:4100 F:3000 U:4100 P:imsterm
11:06:45.501	No explicit template found	Phase:imsterm Method:INVITE
11:06:45.502	No explicit template found	User: 4100@devconnect.local
11:06:45.502	No explicit template found	User: 4100
11:06:45.504	Trying	(2) 100 Trying
11:06:45.504	No explicit template found	User: 4100@devconnect.local
11:06:45.504	No explicit template found	User: 4100
11:06:45.505	Implicit user found	pfa,CallEventControl-3.4.0.0.340003,OutlookCalendarRouterSnapin-1.1,pfa,
11:06:45.528	Service list	CallEventControl 3.4.0.0.340003
11:06:45.571	OutlookCalendarRouterSnapin	OutlookCalendarRouterSnapin 1.1
11:06:45.575	Service attributes:	OutlookCalendarRouterSnapin: Service attributes:
11:06:45.575	Exchange URL: https://exchange2	OutlookCalendarRouterSnapin: Exchange URL: https://exchange2013/
11:06:45.575	Master email address: pridismas	OutlookCalendarRouterSnapin: Master email address: pridismaster@devconnect.local
11:06:45.575	Database URL: jdbc:postgresql://	OutlookCalendarRouterSnapin: Database URL: jdbc:postgresql://edphadmater:5433/outlookcalendarrouter snapsi
11:06:45.575	Table name: Emails	OutlookCalendarRouterSnapin: Table name: Emails
11:06:45.576	Database username: outlookcalen	OutlookCalendarRouterSnapin: Database username: outlookcalendarrouter snapsin
11:06:45.576	Announcement message:	OutlookCalendarRouterSnapin: Announcement message:
11:06:45.578	Your public key : v205uvRBSYe24	OutlookCalendarRouterSnapin: Your public key : v205uvRBSYe24ciNoddeLBPJX9iv9MPMLIsIu7btJd8=
11:06:45.578	Fetch database url and credenti	OutlookCalendarRouterSnapin: Fetch database url and credentials.
11:06:45.579	Try to connect to the database.	OutlookCalendarRouterSnapin: Try to connect to the database.
11:06:45.587	Connected to the database.	OutlookCalendarRouterSnapin: Connected to the database.
11:06:45.587	SELECT COUNT(called_handle) FRO	OutlookCalendarRouterSnapin: SELECT COUNT(called_handle) FROM Emails called_handle
11:06:45.589	DB users: 3	OutlookCalendarRouterSnapin: DB users: 3
11:06:45.589	Allowed: 80	OutlookCalendarRouterSnapin: Allowed: 80
11:06:45.589	Entered callIntercepted.	OutlookCalendarRouterSnapin: Entered callIntercepted.
11:06:45.590	Fetching truststore properties.	OutlookCalendarRouterSnapin: Fetching truststore properties.
11:06:45.590	Setting new truststore: /usr/ja	OutlookCalendarRouterSnapin: Setting new truststore: /usr/java/default/jre/lib/security/cacerts
11:06:45.590	Setting new truststore password	OutlookCalendarRouterSnapin: Setting new truststore password changeit
11:06:45.590	Fetch email from database.	OutlookCalendarRouterSnapin: Fetch email from database.
11:06:45.590	Fetch database url and credenti	OutlookCalendarRouterSnapin: Fetch database url and credentials.
11:06:45.591	Try to connect to the database.	OutlookCalendarRouterSnapin: Try to connect to the database.
11:06:45.598	Connected to the database.	OutlookCalendarRouterSnapin: Connected to the database.
11:06:45.598	SELECT email address FROM Email	OutlookCalendarRouterSnapin: SELECT email address FROM Emails WHERE called_handle = CAST(4100 AS VARCHAR)
11:06:45.602	Email: pridis2@devconnect.local	OutlookCalendarRouterSnapin: Email: pridis2@devconnect.local
11:06:45.602	Start fetching the appointments	OutlookCalendarRouterSnapin: Start fetching the appointments.
11:06:45.642	Creating service	OutlookCalendarRouterSnapin: Creating service
11:06:45.642	Fetching useripassword attribut	OutlookCalendarRouterSnapin: Fetching useripassword attributes
11:06:45.642	Fetching Server URL	OutlookCalendarRouterSnapin: Fetching Server URL
11:06:45.643	Request appointments	OutlookCalendarRouterSnapin: Request appointments
11:06:45.909	Showing appointments: count: 0	OutlookCalendarRouterSnapin: Showing appointments: count: 0
11:06:45.910	No redirection, calling: "4100"	OutlookCalendarRouterSnapin: No redirection, calling: "4100"
11:06:45.926	INVITE	(3) T:4100 F:3000 U:4100 P:termdone
11:06:45.930	Trying	(3) 100 Trying
11:06:46.019	Ringling	(3) 180 Ringling
11:06:46.059	Ringling	(2) 180 Ringling
11:06:46.159	Ringling	(3) 180 Ringling
11:06:46.190	Ringling	(2) 180 Ringling

9.2.2 A call to Extension 4100 with an Appointment scheduled

With an Appointment scheduled (as per **Section 9.1**) make a call to that user. The example below a call was made to 4100 which had an Appointment scheduled with the subject as {4101} thus re-routing the incoming caller to 4101. Looking at the trace below similar information to the call in **Section 9.2.1** can be observed, however, on this occasion an appointment is found and the line redirecting call to “4101” can be observed.

SM71vmpg			SM100
11:07:54.335	← INVITE →		(1) T:4100 F:3000 U:4100 P:imsterm
11:07:54.339	Getting Service List		Phase:imsterm Method:INVITE
11:07:54.340	No explicit template found		User: 4100@devconnect.local
11:07:54.340	Implicit User found		User: 4100
11:07:54.342	No explicit template found		User: 4100@devconnect.local
11:07:54.342	Implicit User found		User: 4100
11:07:54.343	Service List		pfa,CallEventControl-3.4.0.0.340003,OutlookCalendarRouterSnapin-1.1,pfa,
11:07:54.349	← Trying →		(1) 100 Trying
11:07:54.362	OutlookCalendarRouterSnapin writeCall		CallEventControl 3.4.0.0.340003
11:07:54.444	OutlookCalendarRouterSnapin writeCall		OutlookCalendarRouterSnapin 1.1
11:07:54.447	Service attributes:		OutlookCalendarRouterSnapin: Service attributes:
11:07:54.448	Exchange URL: https://exchange2		OutlookCalendarRouterSnapin: Exchange URL: https://exchange2013/
11:07:54.448	Master email address: pridismas		OutlookCalendarRouterSnapin: Master email address: pridismaster@devconnect.local
11:07:54.448	Database URL: jdbc:postgresql://edphadbmaster:5433/outlookcalendarrouter		OutlookCalendarRouterSnapin: Database URL: jdbc:postgresql://edphadbmaster:5433/outlookcalendarrouter
11:07:54.448	Table name: Emails		OutlookCalendarRouterSnapin: Table name: Emails
11:07:54.448	Database username: outlookcalen		OutlookCalendarRouterSnapin: Database username: outlookcalendarrouter
11:07:54.449	Announcement message:		OutlookCalendarRouterSnapin: Announcement message:
11:07:54.451	Your public key : v205uvRBSYeZ4		OutlookCalendarRouterSnapin: Your public key : v205uvRBSYeZ4ciNoddeLBFX9iv9MFMPLi1u7btJd8=
11:07:54.451	Fetch database url and credenti		OutlookCalendarRouterSnapin: Fetch database url and credentials.
11:07:54.452	Try to connect to the database.		OutlookCalendarRouterSnapin: Try to connect to the database.
11:07:54.621	Connected to the database.		OutlookCalendarRouterSnapin: Connected to the database.
11:07:54.621	SELECT COUNT(called_handle) FRO		OutlookCalendarRouterSnapin: SELECT COUNT(called_handle) FROM Emails called_handle
11:07:54.623	DB users: 3		OutlookCalendarRouterSnapin: DB users: 3
11:07:54.623	Allowed: 80		OutlookCalendarRouterSnapin: Allowed: 80
11:07:54.623	Entered callIntercepted.		OutlookCalendarRouterSnapin: Entered callIntercepted.
11:07:54.624	Fetching truststore properties.		OutlookCalendarRouterSnapin: Fetching truststore properties.
11:07:54.624	Setting new truststore: /usr/ja		OutlookCalendarRouterSnapin: Setting new truststore: /usr/java/default/jre/lib/security/cacerts
11:07:54.624	Setting new truststore password		OutlookCalendarRouterSnapin: Setting new truststore password changeit
11:07:54.624	Fetch email from database.		OutlookCalendarRouterSnapin: Fetch email from database.
11:07:54.624	Fetch database url and credenti		OutlookCalendarRouterSnapin: Fetch database url and credentials.
11:07:54.625	Try to connect to the database.		OutlookCalendarRouterSnapin: Try to connect to the database.
11:07:54.633	Connected to the database.		OutlookCalendarRouterSnapin: Connected to the database.
11:07:54.633	SELECT email address FROM Email		OutlookCalendarRouterSnapin: SELECT email address FROM Emails WHERE called_handle = CAST(4100 AS VARCHAR)
11:07:54.636	Email: pridis2@devconnect.local		OutlookCalendarRouterSnapin: Email: pridis2@devconnect.local
11:07:54.636	Start fetching the appointments.		OutlookCalendarRouterSnapin: Start fetching the appointments.
11:07:54.636	Creating service		OutlookCalendarRouterSnapin: Creating service
11:07:54.711	Fetching userpassword attribut		OutlookCalendarRouterSnapin: Fetching userpassword attributes
11:07:54.712	Fetching Server URL		OutlookCalendarRouterSnapin: Fetching Server URL
11:07:54.712	Request appointments		OutlookCalendarRouterSnapin: Request appointments
11:07:55.054	Showing appointments: count: 1		OutlookCalendarRouterSnapin: Showing appointments: count: 1
11:07:55.234	SUBJECT: {4101}		OutlookCalendarRouterSnapin: SUBJECT: {4101}
11:07:55.234	LOCATION: Pridis test		OutlookCalendarRouterSnapin: LOCATION: Pridis test
11:07:55.235	SUBJECT: {4101}		OutlookCalendarRouterSnapin: SUBJECT: {4101}
11:07:55.235	LOCATION: Pridis test		OutlookCalendarRouterSnapin: LOCATION: Pridis test
11:07:55.235	ENTERED GetWhitelist		OutlookCalendarRouterSnapin: ENTERED GetWhitelist
11:07:55.235	EXIT GetWhitelist		OutlookCalendarRouterSnapin: EXIT GetWhitelist
11:07:55.235	Found phonenumber (4101)		OutlookCalendarRouterSnapin: Found phonenumber (4101)
11:07:55.235	redirecting call to: "4101"		OutlookCalendarRouterSnapin: redirecting call to: "4101"
11:07:55.284	← INVITE →		(2) T:4100 F:3000 U:4101 P:termdone
11:07:55.288	← Trying →		(2) 100 Trying
11:07:55.513	← Ringing →		(2) 180 Ringing
11:07:55.558	← Ringing →		(1) 180 Ringing

10. Conclusion

These Application Notes describe the configuration steps required for Outlook Calendar Routing Snap-in from Pridis B.V. to successfully interoperate with Avaya Breeze™ and Avaya Aura® Communication Manager. All functionality and serviceability test cases were completed successfully with any observations noted in **Section 2.2**.

11. Additional References

This section references the Avaya product documentation that are relevant to these Application Notes.

Product documentation for Avaya products may be found at <http://support.avaya.com>.

[1] *Administering Avaya Aura® Communication Manager* – Release 7.1

[2] *Administering Avaya Aura® Session Manager* – Release 7.1

[3] *Deploying Avaya Breeze™* Release 3.4

Documentation on the Outlook Calendar Routing Snap-in can be obtained from Pridis by contacting their support team.

- Email: support@pridis.com
- Website: www.pridis.com
- Phone: +31 (0) 33 2475715

©2018 Avaya Inc. All Rights Reserved.

Avaya and the Avaya Logo are trademarks of Avaya Inc. All trademarks identified by ® and ™ are registered trademarks or trademarks, respectively, of Avaya Inc. All other trademarks are the property of their respective owners. The information provided in these Application Notes is subject to change without notice. The configurations, technical data, and recommendations provided in these Application Notes are believed to be accurate and dependable, but are presented without express or implied warranty. Users are responsible for their application of any products specified in these Application Notes.

Please e-mail any questions or comments pertaining to these Application Notes along with the full title name and filename, located in the lower right corner, directly to the Avaya DevConnect Program at devconnect@avaya.com.