



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Performance DNA Knowledge Base Articles

Making Exchange-based Rooms visible to Performance DNA

12/14/2025

Contents

- 1 Making Exchange-based Rooms visible to Performance DNA
 - 1.1 Symptoms
 - 1.2 Resolution
 - 1.3 Setting up a PowerShell session with access to the Exchange Server
 - 1.4 Creating and updating Room Lists
 - 1.5 Permissions required for the user accessing Exchange Server
 - 1.6 Disconnecting PowerShell from the Exchange

Making Exchange-based Rooms visible to Performance DNA

Software product name	Performance DNA
SLS Version Number	9.0
Subject	Making Exchange-based rooms visible to Performance DNA

Symptoms

You have configured Performance DNA for use with Exchange, but no **Rooms** are appearing.

Resolution

Start a Windows PowerShell session and use the following commands to create and maintain a **Room List** within the Exchange.

Setting up a PowerShell session with access to the Exchange Server

Use the following command to setup a PowerShell session with access to the exchange server.

```
$UserCredential = Get-Credential
$Session = New-PSSession -ConfigurationName Microsoft.Exchange -ConnectionUri
https://outlook.office365.com/powershell-liveid/ -Credential $UserCredential -Authentication
Basic -AllowRedirection
Import-PSSession $Session
```

Creating and updating Room Lists

You can create the **Room Lists** by using the **Exchange Online PowerShell cmdlet** `New-DistributionGroup` and the `RoomList` parameter as shown below. That is, you must change the `ExampleRoomList` and `testroom` options to match the **Room List** and **Room Names** for your environment.

```
New-DistributionGroup -RoomList -Name 'ExampleRoomList' -Members
('testroom1','testroom2','testroom3')
```

Displaying all the Rooms in the List

Use the following command to display all the **Rooms** in the **List**:

```
Get-DistributionGroupMember 'ExampleRoomList'
```

Adding a Room to the Room List

Use the following command to add a **Room** to the **Room List**:

```
Add-DistributionGroupMember -Identity 'ExampleRoomList' -Member 'testroom4'
```

Removing a Room from the Room List

Use the following command to remove a **Room** from the **Room List**:

```
Remove-DistributionGroupMember -Identity 'ExampleRoomList' -Member 'testroom3'
```

Removing the Room List

Use the following command to remove a **Room List**:

```
Remove-DistributionGroup 'ExampleRoomList'
```

Permissions required for the user accessing Exchange Server

Owner Permission

The user that you are using to access Exchange Server from within Performance DNA must have **Owner** permissions on the **Calendar** folder of each **Room** that you want to use. If the user doesn't have the required permissions, the connectivity test will pass but Performance DNA will not create entries in the **Room Calendars**.

In such cases, your log file shows error as shown in the following example:

```
04/10/2018 13:32:06 - ERROR - An exception of type
Microsoft.Exchange.WebServices.Data.ServiceResponseException was caught
04/10/2018 13:32:06 - ERROR - The specified folder could not be found in the store.
04/10/2018 13:32:06 - WARN - at
Microsoft.Exchange.WebServices.Data.ServiceResponse.InternalThrowIfNecessary()
    at Microsoft.Exchange.WebServices.Data.MultiResponseServiceRequest`1.Execute()
...
```

Checking Permissions

You can check the current permissions for a **Room Calendar** by using the Get-MailboxFolderPermission cmdlet, by passing the email address of the **Room**, followed immediately by a colon, backslash, and the word Calendar. For example, consider that you want the user with email id **pdnroomuser@yourexchange.com** to have access to your **Rooms**. Using the **Room's**

email id **roomname@yourexchange.com**, you can check the permissions by executing the following command:

```
Get-MailboxFolderPermission roomname@yourexchange.com:\Calendar
```

The command result gives you the details of permissions that a user/group have on a **Room's Calendar**:

FolderName	User	AccessRights	SharingPermissionFlags
Calendar	Default	{AvailabilityOnly}	
Calendar	Anonymous	{None}	

Granting Owner Permissions to the Room User

In the example above, the room user doesn't have any permission. To add a permission, use the `Add-MailboxFolderPermission` cmdlet with other details as shown below:

```
Add-MailboxFolderPermission roomname@yourexchange.com:\Calendar -User  
pdnroomuser@yourexchange.com -AccessRights Owner
```

This would give you an output similar to the following:

FolderName	User	AccessRights	SharingPermissionFlags
Calendar	PDNA Room User	{Owner}	

Updating Room User Permissions to the Owner

If the user already has permissions to the **Room Calendar**, but NOT the Owner permissions, use the `Set-MailboxFolderPermission` cmdlet to update the permission.

For example, the user has following permissions for the **Room**.

FolderName	User	AccessRights	SharingPermissionFlags
Calendar	Default	{AvailabilityOnly}	
Calendar	Anonymous	{None}	
Calendar	PDNA Room User	{Reviewer}	

You can update the permissions by using the following command:

```
Set-MailboxFolderPermission roomname@yourexchange.com:\Calendar -User  
pdnroomuser@yourexchange.com -AccessRights Owner
```

Disconnecting PowerShell from the Exchange

After completing your tasks in the PowerShell session, disconnect it from the Exchange session by

running the following command:

```
Get-PSSession | Where-Object {$_.ComputerName -eq "outlook.office365.com"} | Remove-PSSession
```

Important

Genesys recommends the above scripts. However, you can change them or write an alternative script that comply with your organization's local security policy.