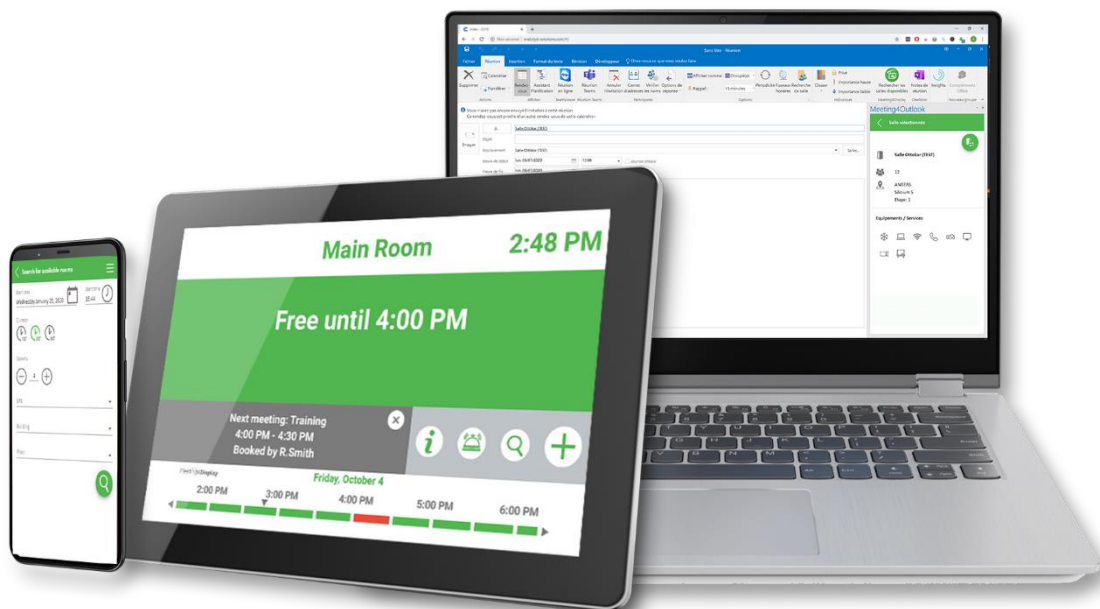


Workspace management solution



CONFIGURATION GUIDE

V3.2

1. Introduction	3
2. Exchange configuration.....	4
3. Meeting4Display Back Office	10
4. Meeting4Mobile	10
5. Outlook Add-in.....	10

1. Introduction

The Meeting4Display solution uses Exchange Web Services (EWS).

This service must be enabled and accessible.

The Client Access Server (CAS) role must be enabled to allow a third-party application to connect to it.

2. Exchange configuration

The mailbox is configured via PowerShell using the “**Exchange Management Shell**” command line interface available in Microsoft’s Exchange management tools.

PowerShell commands must be executed using an Exchange administrator account. This can be done from any workstation with access to the Exchange server.

The Meeting4Display suite uses a service account to perform actions on Exchange.

Available rooms are managed via the room list functionality in the application (Exchange RoomLists).

*Note: Items noted in red italics in the PowerShell commands to be executed are examples only and should be replaced with your own values.
All these commands are available on the Microsoft website.*

Obtaining the link to EWS defined on the Exchange Server:

This link will be used in the Meeting4Display back office to handle web service calls.

View
Get-WebServicesVirtualDirectory

Creating a service account:

Create
New-Mailbox -Name "Account_Name" -UserPrincipalName <i>account_name@mydomain</i> -Password (ConvertTo-SecureString -String <i>Password</i> -AsPlainText -Force)

Verify
Get-Mailbox -Identity "Account_Name" Format-List Name,DisplayName,Alias,PrimarySmtpAddress,Database

Creating a room mailbox:

Create
<pre>New-Mailbox -Name "Room_name" -PrimarySmtpAddress room_name@mydomain -Room -EnableRoomMailboxAccount \$true -RoomMailboxPassword (ConvertTo-SecureString -String Password -AsPlainText -Force)</pre>
Verify
<pre>Get-Mailbox -Identity "Room_name" Format-List Name,DisplayName,Alias,PrimarySmtpAddress,Database</pre>

Creating a room list:

Create
<pre>New-DistributionGroup -Name "Room_list_name" -OrganizationalUnit "Domain_Name.local/Users" -RoomList</pre>
Verify
<pre>Get-DistributionGroup -Identity "Room_list_name" Format-List</pre>

Adding a room to the room list:

Add
<pre>Add-DistributionGroupMember -Identity "Room_list_name" -Member "room_name@mydomain"</pre>
Verify
<pre>Get-DistributionGroupMember -Identity "Room_list_name"</pre>

Assigning delegation permissions to the service account:

Room delegation permissions can be assigned to the service account for a single room or all the rooms in a room list (RoomList).

The "full access" permission setting is assigned to the service account.

Add room

```
Add-MailboxPermission  
-Identity "Room_name"  
-User "Account_Name"  
-AccessRights FullAccess
```

Verify

```
Get-MailboxPermission  
-Identity "Room_name" | Format-List
```

Add list

```
Get-DistributionGroupMember  
-Identity "Room_list_name" |  
ForEach-Object  
{  
    Add-MailboxPermission  
    $_.Identity  
    -User "Account_Name"  
    -AccessRights FullAccess  
}
```

Verify

```
Get-DistributionGroupMember  
-Identity "Room_list_name" |  
ForEach-Object  
{  
    Get-MailboxPermission  
    $_.Identity | Format-List  
}
```

Configuring the service account for impersonation:

Impersonation can be performed:

- By using a role group which the service account is a member of
- By associating the role directly to the service account

Using a role group

By creating a role group allowing impersonation and containing the service account as a member, impersonation can be configured for the service account in a single command (New-RoleGroup).

Note: To add impersonation permissions to an existing group, use the command Set-RoleGroup. Then, to add the service account to the group, use the command Update-RoleGroupMember.

Add

```
New-RoleGroup  
-Name "Group_Name"  
-Roles ApplicationImpersonation  
-Members "Account_Address"
```

Verify

```
Get-RoleGroup  
-Identity "Group_name"  
  
Get-RoleGroupMember  
-Identity "Group_name"
```

Adding the role directly to the service account (without adding a role group)

Add

```
New-ManagementRoleAssignment  
-Name "Role_Name"  
-Role ApplicationImpersonation  
-User "Account_Address"
```

Verify

```
Get-ManagementRoleAssignment  
"Role_Name"
```

Configuring the room options:

Common options:

Room option settings can be assigned to the service account for a single room or all the rooms in a room list (RoomList).

The options required for Meeting4Display to function properly are as follows:

- **DeleteComments** allows you to indicate that the body text of incoming meeting request messages should be saved.
- **RemovePrivateProperty** specifies that you should not clear the private flag for incoming meetings sent by the host in the original requests.
- **DeleteSubject** indicates that the subject of incoming meeting requests should be saved.
- **AddOrganizerToSubject** specifies that the name of the meeting organizer is not used as the subject of the meeting request.
- **AutomateProcessing** enables the processing of calendar items in the mailbox. This means that the Calendar Wizard updates the calendar and the Resource Reservation Wizard accepts the meeting according to the policies.

Add room

```
Set-CalendarProcessing
-Identity "Room_name"
-DeleteComments $false
-RemovePrivateProperty $false
-DeleteSubject $false
-AddOrganizerToSubject $false
-AutomateProcessing AutoAccept
```

Verify

```
Get-CalendarProcessing
-Identity "Room_name" | Format-List
```

Add list

```
Get-DistributionGroupMember
-Identity "Room_list_name" |
ForEach-Object
{
    Set-CalendarProcessing
    -Identity $_.Identity
    -DeleteComments $false
    -RemovePrivateProperty $false
    -DeleteSubject $false
    -AddOrganizerToSubject $false
    -AutomateProcessing AutoAccept
}
```

Verify

```
Get-DistributionGroupMember
-Identity "Room_list_name" |
ForEach-Object
{
    Get-CalendarProcessing
    $_.Identity | Format-List
}
```


User-dependent options (to be configured for Meeting4Mobile) :

Meeting4Mobile requires additional room rights to be configured for users.

Add room
<pre>Add-MailboxFolderPermission -Identity Room_address:\calendar -User "By default" -AccessRights noneditingauthor</pre>

Verify
<pre>Get-MailboxFolderPermission -Identity Room_address:\calendar</pre>

Add list
<pre>Get-DistributionGroupMember -Identity "Room_list_name" ForEach-Object { Add-MailboxFolderPermission -Identity "\$(\$_.Identity):\calendar" -User "By default" -AccessRights noneditingauthor }</pre> <p>Note</p> <p>If the Add-MailboxFolderPermission command does not work because permissions are already configured, use the Set-MailboxFolderPermission command to manage the permissions:</p> <pre>Get-DistributionGroupMember -Identity "Room_list_name" ForEach-Object { Set-MailboxFolderPermission -Identity "\$(\$_.Identity):\calendar" -User "By default" -AccessRights noneditingauthor</pre>

Verify
<pre>Get-DistributionGroupMember -Identity "Room_list_name" ForEach-Object { Get-MailboxFolderPermission "\$(\$_.Identity):\calendar" Format-List }</pre>

3. Meeting4Display Back Office

The application can be accessed here:
`http(s)://{hostnameorIPAddress}/Meeting4Display`

The back office uses the basic authentication of the service account previously created.

The settings to use the service account for the Meeting4Display suite are configured from the "Calendar Configuration" tile under the "Settings" menu or the "Settings" tile on the "Home" page.

The elements to be defined are as follows:

Calendar system	Exchange
Server address	Address of the Exchange EWS service e.g.: <code>https://exchange-server-hostname/ews/exchange.asmx</code>
Account name	Service account name: Example: <i>Meeting4Display</i>
Password	Password for the service account
Domain	Domain used (optional, depending on Exchange configuration)

The "Test" button is used to check that the Meeting4Display application communicates correctly with Exchange Web Services (EWS).

4. Meeting4Mobile

The application can be accessed here:
`https://{hostnameorIPAddress}/Meeting4DisplayMobile`

The Meeting4Mobile application uses basic authentication. It allows you to connect with a user account (login/password) defined in Exchange.

5. Outlook Add-in

The application can be accessed here:
`http(s)://{hostnameorIPAddress}/Meeting4DisplayOutlook/`

The Outlook Add-in application uses basic authentication. It allows you to log on with the service account configured in the back office.

In order to use it, you must enter the URL and password required to log on to the company.