OpenCode4Connections

Guest Model Customizer extension

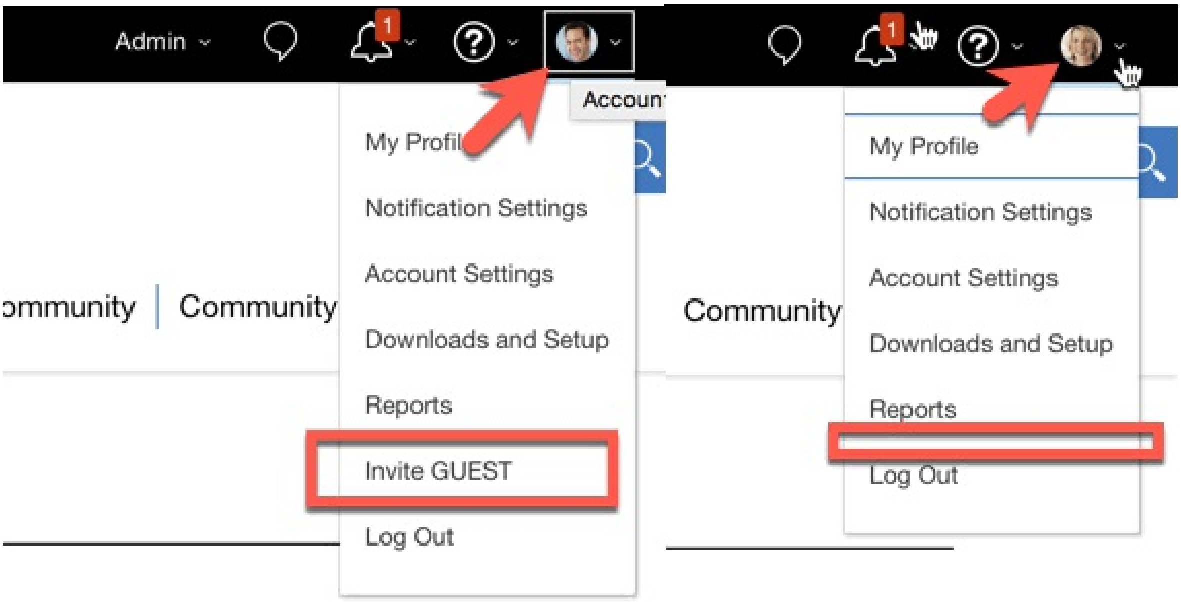
This set of scripts addresses 3 areas:

1. Custom Guest Model
2. Privacy
3. Sametime Cloud AutoLogin

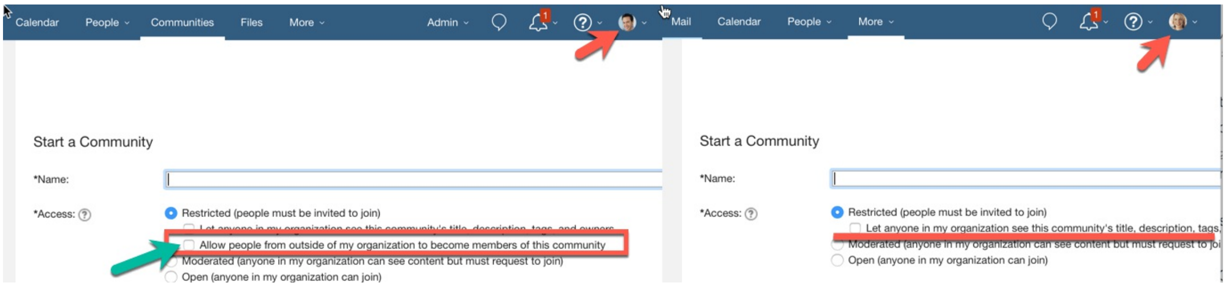
Custom Guest Model

The goal of this area is to provide a mean by which "external guest" sharing and invitations is hidden to a selected category of users. The following features are available:

- The "Invite Guest" menu item is hidden to the users not having the rights to invite external people



- Users not having the rights to invite external people cannot create a Restricted Community open to External users



Start a Community

*Name:

*Access:
● Restricted (people must be invited to join)
□ Let anyone in my organization see this community's title, description, tags, and owners
□ Allow people from outside of my organization to become members of this community
○ Moderated (anyone in my organization can see content but must request to join)
○ Open (anyone in my organization can join)

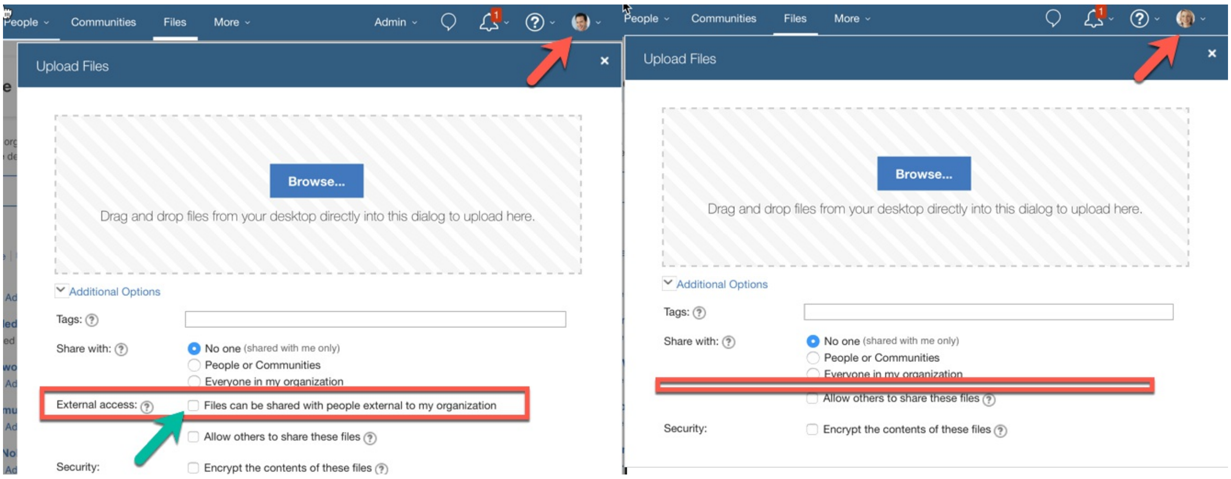
Start a Community

*Name:

*Access:
● Restricted (people must be invited to join)
□ Let anyone in my organization see this community's title, description, tags,
○ Moderated (anyone in my organization can see content but must request to joi
○ Open (anyone in my organization can join)

1. **Note:** for the users having the rights, the default is also changed as shown by the green arrow

- Users not having the rights to invite external people cannot upload a file which becomes open to external people nor create a folder which is accessible to external people



Upload Files

Drag and drop files from your desktop directly into this dialog to upload here.

Additional Options

Tags:

Share with: ● No one (shared with me only)
○ People or Communities
○ Everyone in my organization

External access: □ Files can be shared with people external to my organization

□ Allow others to share these files

Security: □ Encrypt the contents of these files

Upload Files

Drag and drop files from your desktop directly into this dialog to upload here.

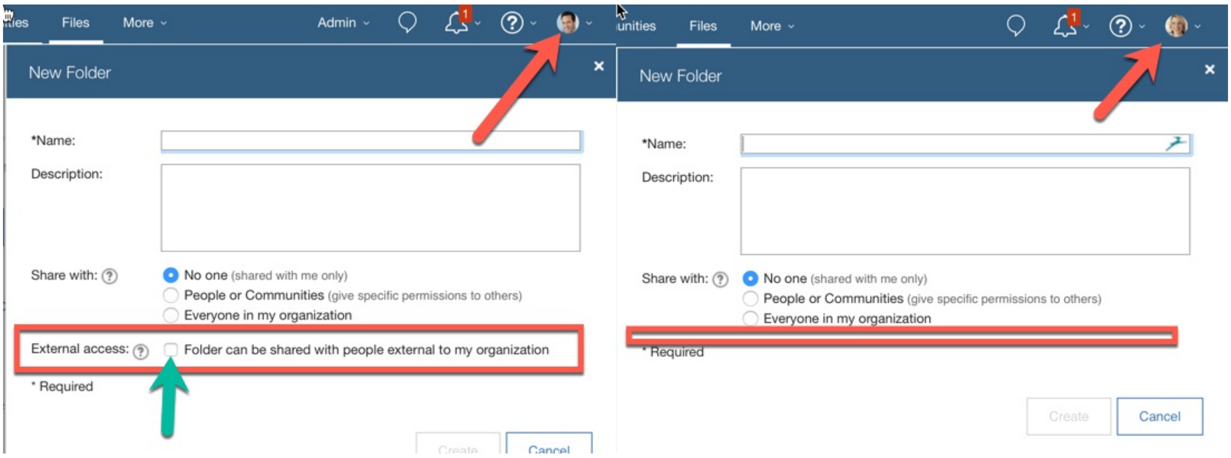
Additional Options

Tags:

Share with: ● No one (shared with me only)
○ People or Communities
○ Everyone in my organization

□ Allow others to share these files

Security: □ Encrypt the contents of these files



New Folder

*Name:

Description:

Share with: ● No one (shared with me only) ○ People or Communities (give specific permissions to others) ○ Everyone in my organization

External access: □ Folder can be shared with people external to my organization

* Required

New Folder

*Name:

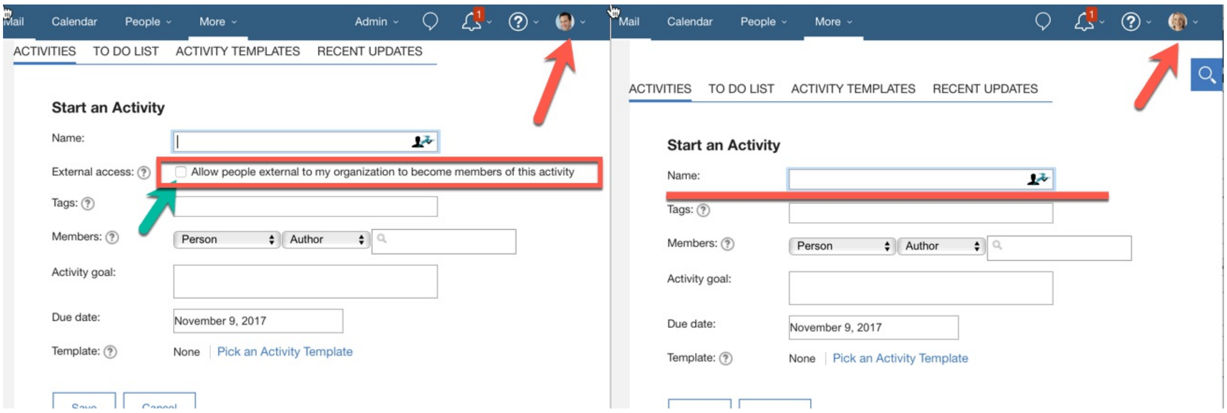
Description:

Share with: ● No one (shared with me only) ○ People or Communities (give specific permissions to others) ○ Everyone in my organization

* Required

1. **Note:** for the users having the rights, the default is also changed as shown by the green arrow
2. **Note:** The same applies when the user creates a "New Document", "New Spreadsheet" and "New Presentation"

- Users not having the rights to invite external people cannot create an Activity which becomes open to external people



Start an Activity

Name:

External access: □ Allow people external to my organization to become members of this activity

Tags:

Members: Person Author

Activity goal:

Due date: November 9, 2017

Template: None | Pick an Activity Template

Start an Activity

Name:

Tags:

Members: Person Author

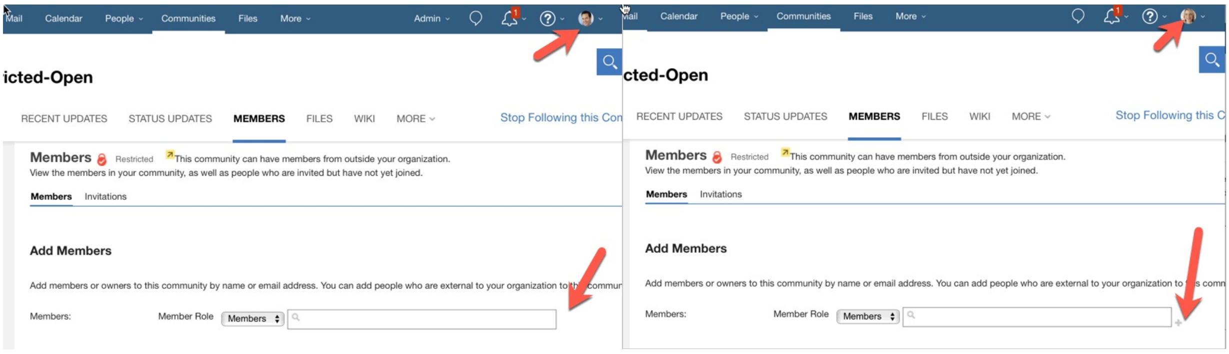
Activity goal:

Due date: November 9, 2017

Template: None | Pick an Activity Template

1. **Note:** for the users having the rights, the default is also changed as shown by the green arrow

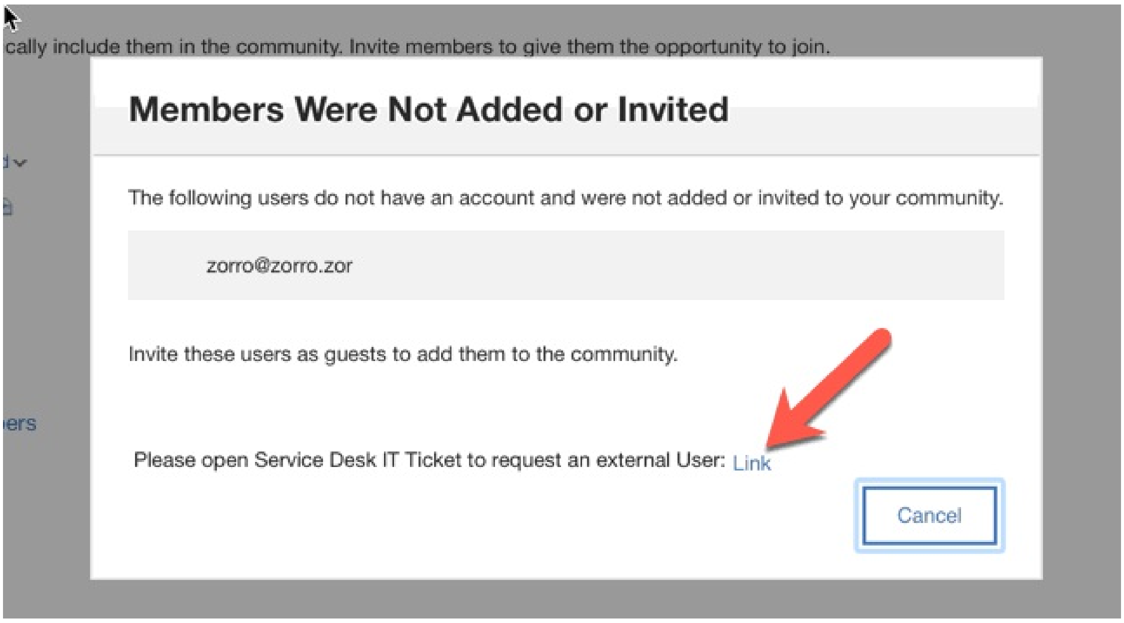
- Users not having the rights to invite external people cannot add or invite an external person to a Restrcited Community for which they are owners



This community can have members from outside your organization.
View the members in your community, as well as people who are invited but have not yet joined.

Add members or owners to this community by name or email address. You can add people who are external to your organization to th commun

1. **Note** : as shown in the above picture, **the rights** to invite external people into an existing Restricted Community and the rights to create a Restricted Community open to external people **can be different** (this means that someone may not be able to create a Restricted Community open to external people but if she is the owner of an existing Restricted Community she may be able to invite external people into it)
2. **Note** : if the user created a CONTACT for a person that is not yet a registered user or guest in Connections Cloud, referencing that Contact in the Add/Invite Member section actually displays the contact in the dropdown list. By hitting the "Save" button, the user could actually add the Contact to the Community even if the "+" sign is disabled. In order to address this situation, the script intercepts the resulting invitation form and proposes to follow a LINK which points to a custom page where the request for inviting the person should be dealt with by authorized people. See the follwoing picture:



cally include them in the community. Invite members to give them the opportunity to join.

Members Were Not Added or Invited

The following users do not have an account and were not added or invited to your community.

zorro@zorro.zor

Invite these users as guests to add them to the community.

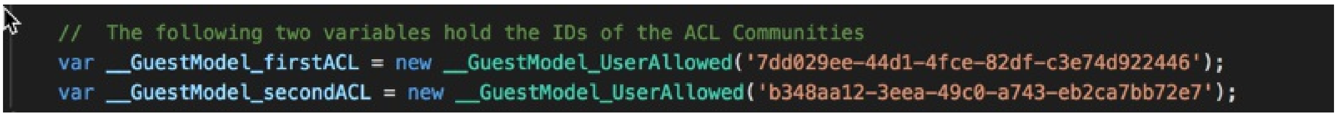
Please open Service Desk IT Ticket to request an external User: Link

Installation

You can install these extensions directly by adding the GuestModel_common.json JSON file into the Organization Extensions of your IBM Connections Cloud organization.

Customization

You may want to customize the "Access Control Lists" which define the rights previously described. In order to do this, you need to modify the following lines at the bottom of the GuestModel_Common.js file:

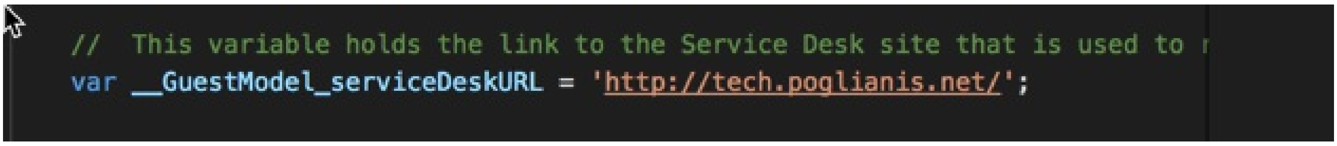


```
//  The following two variables hold the IDs of the ACL Communities
var __GuestModel_firstACL = new __GuestModel_UserAllowed('7dd029ee-44d1-4fce-82df-c3e74d922446');
var __GuestModel_secondACL = new __GuestModel_UserAllowed('b348aa12-3eea-49c0-a743-eb2ca7bb72e7');
```

- Simply change the UUID of the two communities referenced by the global variables shown in the picture above to point to two Communities you have previously created in your organization.
- you need to create two communities (I suggest you create **RESTRICTED Communities** as the script is able to properly manage the 403 errors from APIs) and add, as members for those Communities, the people you want to have the rights
    1. __GuestModel_firstACL represents the ACL allowing people to invite external users and to share content with external users

2. __GuestModel_secondACL represents the ACL allowing people to add/invite members to an already created Restricted Community.

You may also want to customize the Link to the "service desk" page which manages invitations for external people who are referenced in the personal Contacts. In that case you need to modify the following value in the GuestModel_Common.js file:



```
//  This variable holds the link to the Service Desk site that is used to r
var __GuestModel_serviceDeskURL = 'http://tech.poglianis.net/';
```

Technical impact

The scripts creates a Cookie with this name:

```
Customizer-<userId>-<communityId>
```

The "communityId" refers the Community Uuid of the relevant ACL Community. The Cookie can hold a value of **true** or **false** .

The Cookie is used to store the authorization status for the two ACLs previously described. Since getting the authorization status requires a REST call to the IBM Connections APIs (to validate the membership of the current user against the two Communities modelling the ACLs), saving the status in a Cookie avoids unnecessary calls and speeds up the execution of the scripts.

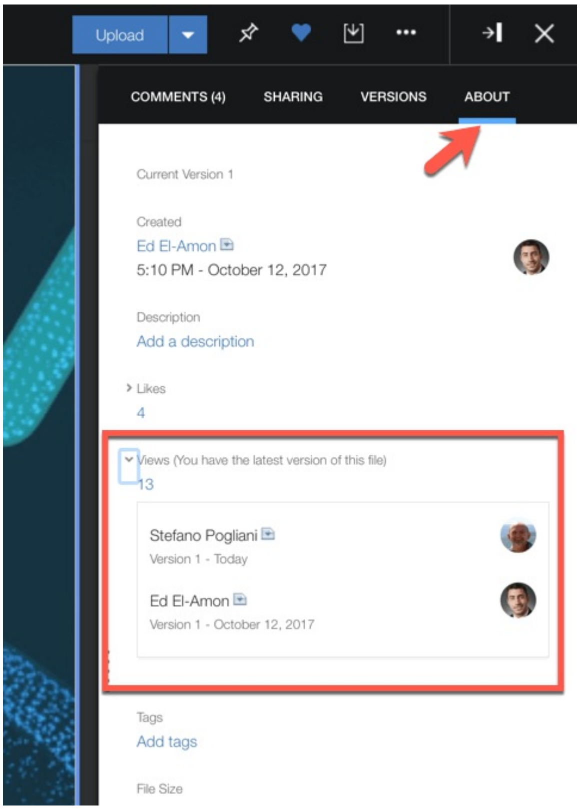
The **side effect** is that if the ACL memembership is changed when the user has a browser open, the user will need to close the browser in order to have the new authorization status reflected in the behavior of the scripts.

Privacy

The goal of this area is to provide a mean by which an organization could selectively hide the details of whom viewed/downloaded any given file. The idea is that sometimes, to comply with local policies, organizations need to protect the details of certain actions performed by the user and which did not require an explicit **consent**. Viewing or Downloading a file is one of these activities.

The script automatically removes from the "View" widget in the "About" tab of the Files Preview page the details of people whose identity needs to be protected.

In the following picture, both Franck Adams and Heather Reeds viewed the document but their details are not shown:



Current Version 1

Created
Ed El-Amon
5:10 PM - October 12, 2017

Description
Add a description

Likes
4

Views (You have the latest version of this file)
13

Stefano Pogliani
Version 1 - Today

Ed El-Amon
Version 1 - October 12, 2017

Tags
Add tags

File Size

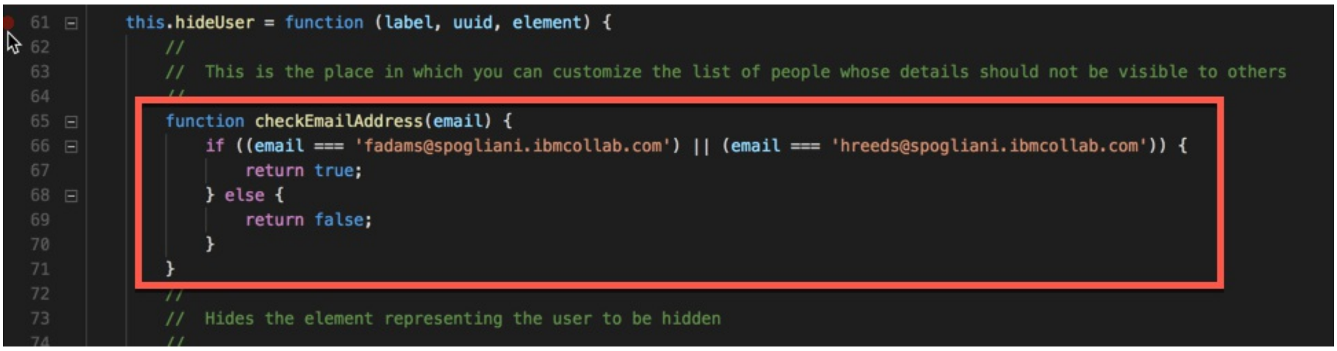
Installation

You can install these extensions directly by adding the HideUserActivity.json JSON file into the Organization Extensions of your

IBM Connections Cloud organization.

Customization

You may want to customize the criteria used by the script to select the people to hide, you need to modify the following function in the HideUserActivity.js script:



```
this.hideUser = function (label, uuid, element) {
    //
    //  This is the place in which you can customize the list of people whose details should not be visible to others
    //
    function checkEmailAddress(email) {
        if ((email === 'fadams@spogliani.ibmcollab.com') || (email === 'hreeds@spogliani.ibmcollab.com')) {
            return true;
        } else {
            return false;
        }
    }
    //
    //  Hides the element representing the user to be hidden
    //
```

In the example, we want to hide the View/Download actions performed by *fadams@spogliani.ibmcollab.com* and *hreeds@spogliani.ibmcollab.com*. You may want to use a more sophisticated algorithm, involving the use of regular expressions for instance.

Technical impact

The scripts does not have any technical impact on the behavior of your browser session.

Sametime Cloud AutoLogin

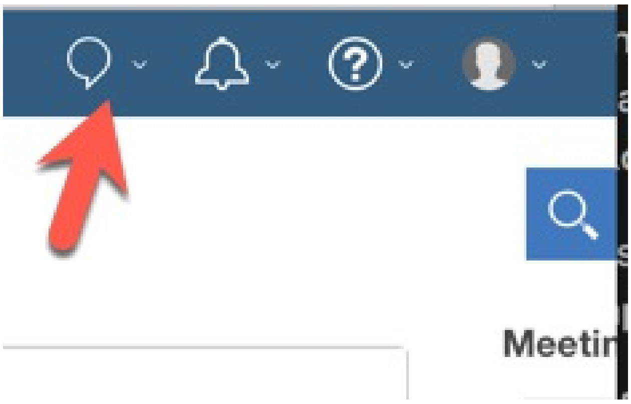
The goal of this area is to provide a mean by which an organization replace the default behavior of the IBM Connections Cloud embedded Sametime Autologin. By default, today, when a user logs into IBM Connections Cloud, and automatic Login is performed on the emebedded Sametime.

This may not be adapted to organizations where many users still use the

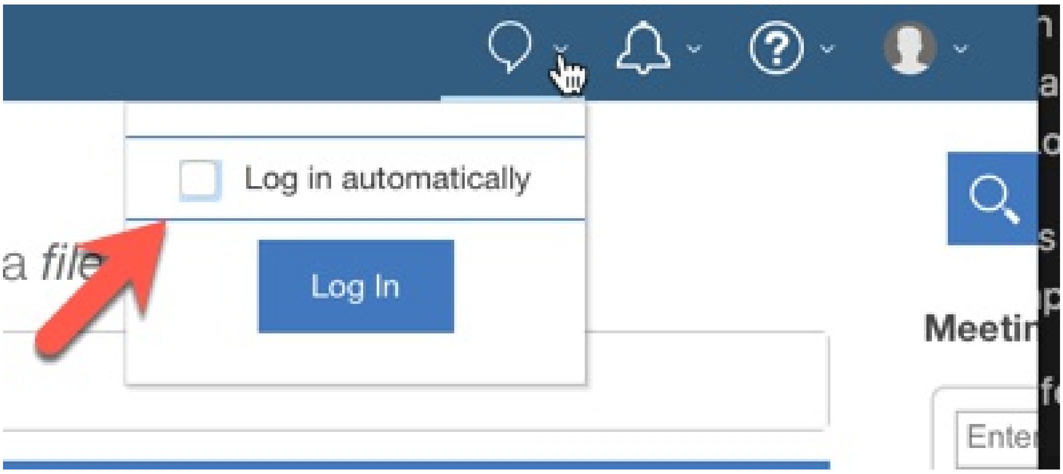
IBM Notes client to access their email and have the IBM Sametime application embedded in the IBM Notes client.

The way in which the script works is the following:

```
When the user logs the first time, the AutoLogin
feature is removed
```



```
The user could force AutoLogin by setting the
relevant checkbox:
```



```
At this point, AutoLogin will be re-enabled.

The user can prevent AutoLogin again by unchecking
```

```
the relevant checkbox
```

Installation

You can install these extensions directly by adding the SametimeNoAuto.json JSON file into the Organization Extensions of your IBM Connections Cloud organization.

Customization

No customization options for this script

Technical impact

The scripts creates a Cookie with this name:

```
CustomizerSametimeCookie
```

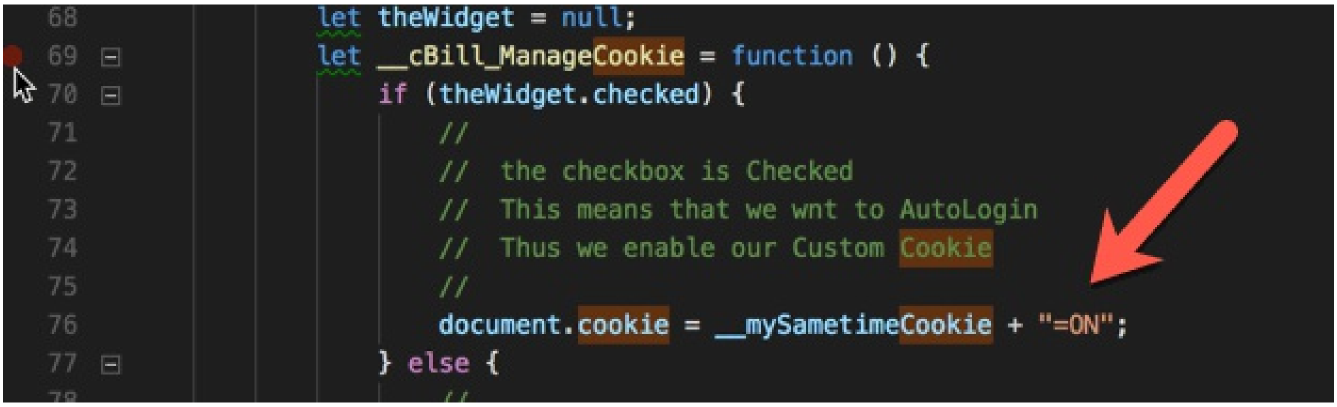
When this Cookie is present, the AutoLogin behavior is determined by the user having checked/unchecked the relevant UI option.

The scripts also manages a standard IBM Sametime Cookie

```
stproxy.dock.notremembered
```

This cookie is set to "no-auto-connect" when the new default behavior of preventing Auto-Login is in place.

I do not manage "persistent cookies". You may need to modify the source code to manage them:



```
            let theWidget = null;
            let __cBill_ManageCookie = function () {
                if (theWidget.checked) {
                    //
                    //  the checkbox is Checked
                    //  This means that we wnt to AutoLogin
                    //  Thus we enable our Custom Cookie
                    //
                    document.cookie = __mySametimeCookie + "=ON";
                } else {
```