MEP Builder

Admin Guide

Version 3.3

2024

Document Change Record

Version Number	Date	Description
1.0	31 October 2021	Original
1.0	22 November 2021	Updated with server installation procedures
1.0	19 November 2021	Final Edit
2.0	27 July 2022	Removed ability for Sysadmin to remove users
2.0	27 July 2022	Added the ability for Sysadmin to create initial SuperAdmin users only
2.0	27 July 2022	Added Admin Module, SuperAdmin and Admin roles and associated permissions.
3.0	14 September 2023	Added steps for Importing a New Release
3.1	28 September 2023	Added steps for implementing a Plug-In
3.1	07 November 2023	Updated config.json modification instructions to include new configuration items plus definitions and guidance for each.
3.2	21 February 2024	Updated steps for Importing a New Release
3.2	1 March 2024	Added Transferring packages to another user section
3.3	6 May 2024	Add Revoke as a user status option, Add Point of Contact Update in Transfer Package section, Add Audit Logs section

Table of Contents

1. INSTALLATION INSTRUCTIONS

A. Server Deployment Procedures:

Prerequisites: Server Instances will require HTTPS

1. As a server admin, follow the Build Installer instructions located in the User Manual Document to install the project on a server.

2. The Build Installer instructions will add the source code for the application in the Build_Installer folder. Navigate to this folder.

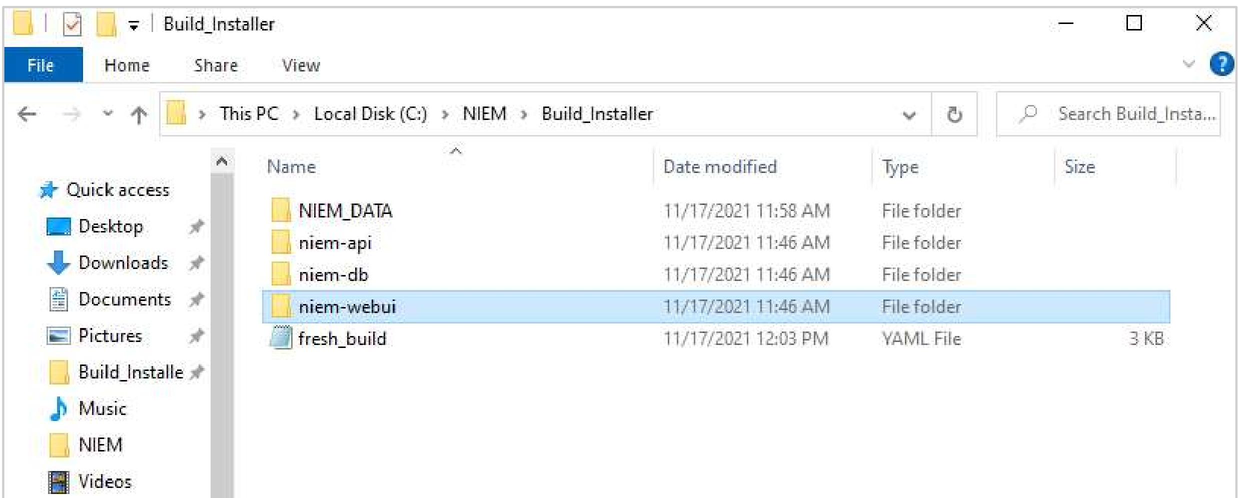


Figure 1.1

3. From this folder location, navigate to:

   \Build_Installer\niem-webui\src\config

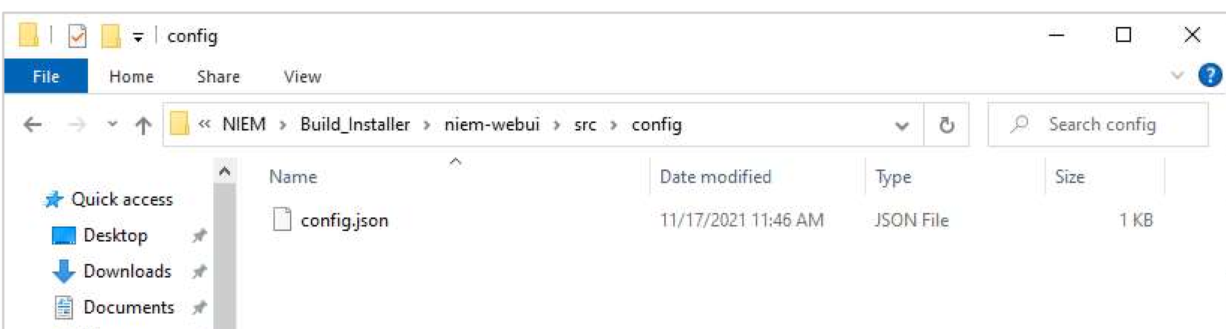


Figure 1.2

4. Open the config.json file using any text editor application

config.json - Notepad

File Edit Format View Help

```
{
        "isStandAloneSys": false,
        "translateTypes": ["CMF", "JSON LD", "XML", "JSON Schema", "OWL"],
        "niemReferenceBaseURL": "https://reference.niem.gov/niem/",
        "niemContactEmail": "info@niemopen.org"
}
```

Figure 1.3

5. See that the value for ‘isStandAloneSys’ is ‘true’…
6. Change the ‘true’ value to ‘false’, without quotation marks

config.json - Notepad

File Edit Format View Help

```
{
        "isStandAloneSys": false,
        "translateTypes": ["CMF", "JSON LD", "XML", "JSON Schema", "OWL"],
        "niemReferenceBaseURL": "https://reference.niem.gov/niem/",
        "niemContactEmail": "info@niemopen.org"
}
```

Figure 1.4

7. Make any additional updates to the config document as appropriate. See Table 1 below for guidance.

Table 1: Definitions and guidance for config.json keys and values.

Key	Value Definition	Notes
isStandAloneSys	Flag to determine if the tool is being loaded as a local/stand-alone instance or as a server instance. Only accepts values of true and false.	- true: Indicates a local/stand-alone instance that will not require user login. - false: Indicates a server instance that will require user accounts and sign-in credentials.
translateTypes	A list of formats users may translate their MEP Builder packages to.	- This configuration list only affects the front-end dropdown list presented to the user. Actual format translation functionality will need further backend development to be functional. - Any formats added to this list that do not have back-end functionality will be shown as a disabled option to the user. - Follow the current syntax. Each translation format is added between the square brackets as a comma separated list and surrounded by double quotation marks. They are in user-friendly formats with proper casing and spacing as appropriate because these will be the labels presented to the user.
niemReferenceBaseURL	Base URL for any NIEM Reference documentation	- Current value is the current base URL for NIEM Reference documentation. The MEP Builder Tool will continue to utilize the established relative URLs in the event of any changes to this base URL.
niemContactEmail	General email contact for MEP Builder Management Office	- This email is presented to MEP Builder users throughout the application to contact as support.

Figure 1.5

8. Save and close the config document
9. The application needs to be restarted to apply the new change. Navigate back to inside the Build_Installer directory.

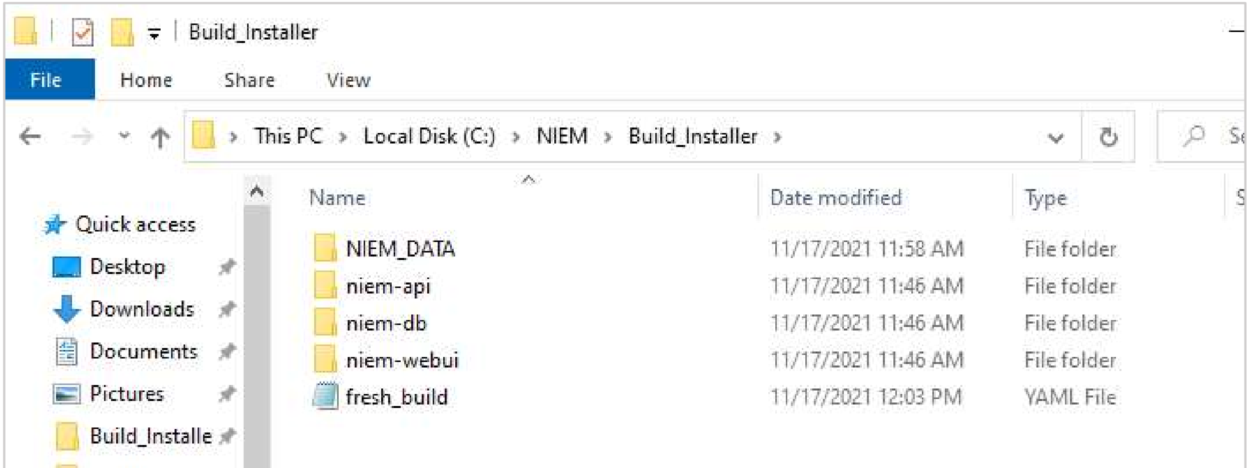


Figure 1.6

10. Type ‘powershell’ in the directory path field and either type ‘Enter’ on your keyboard or click the arrow to go to powershell.

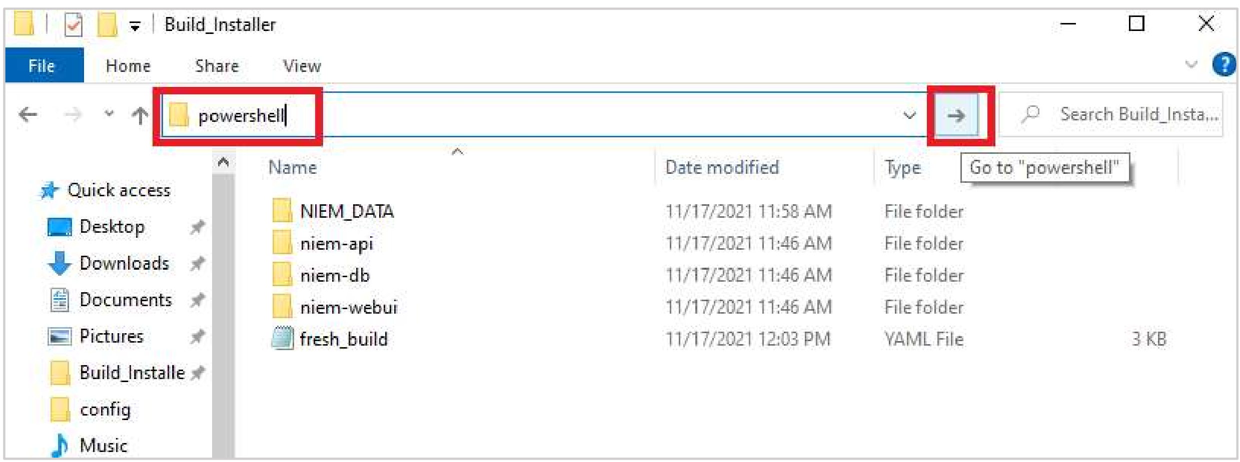


Figure 1.7

11. Type or copy/paste the following command:
    a. docker-compose -f fresh_build.yaml down

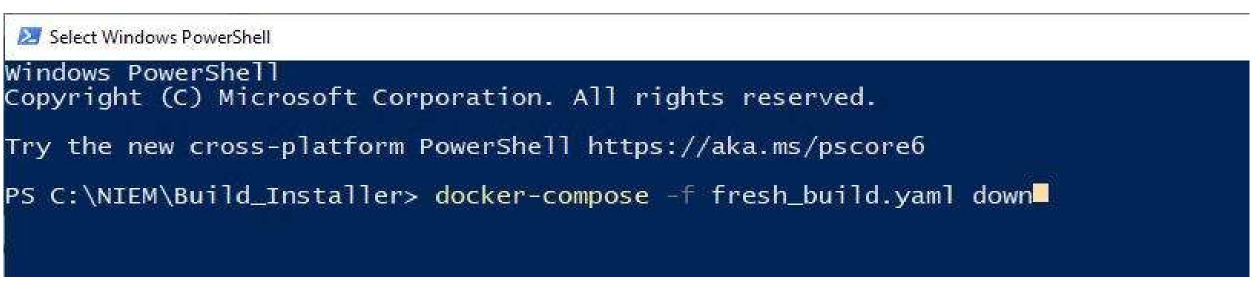


Figure 1.8

12. Click the Enter key on the keyboard to run the above command

13. It will take a few moments for the command to stop and remove all of the containers. It will be finished when you see six 'done' statuses and the directory path as shown in the image below

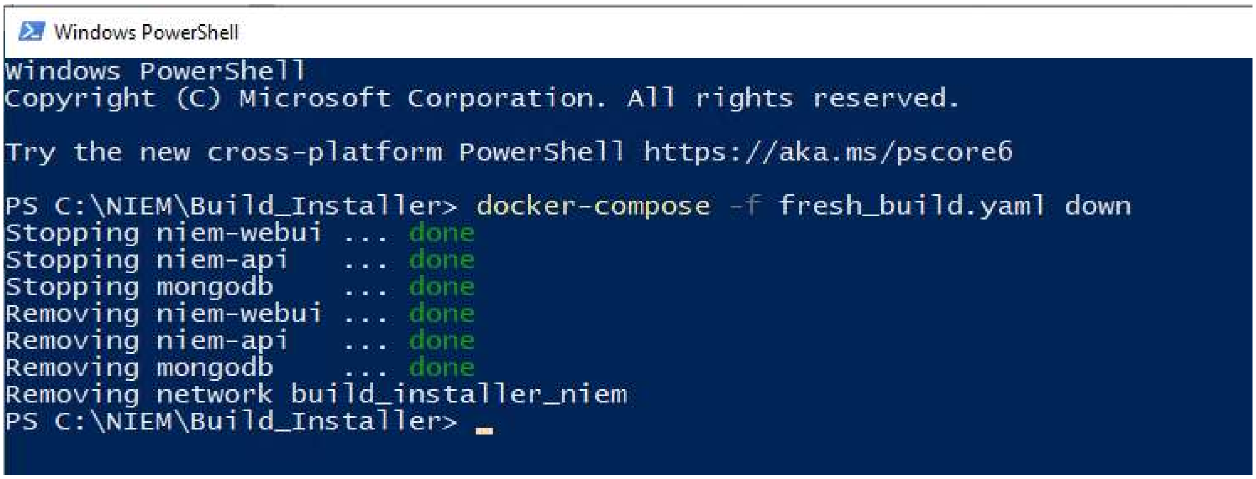


Figure 1.9

14. To rebuild, run the following command in powershell:
    a. docker-compose -f fresh_build.yaml up -d --build
    b. Note there are two '-' marks before 'build'

Figure 1.10

15. The rebuilding process will be complete when you see three new 'done' statuses and the directory path as shown at the bottom of the image below

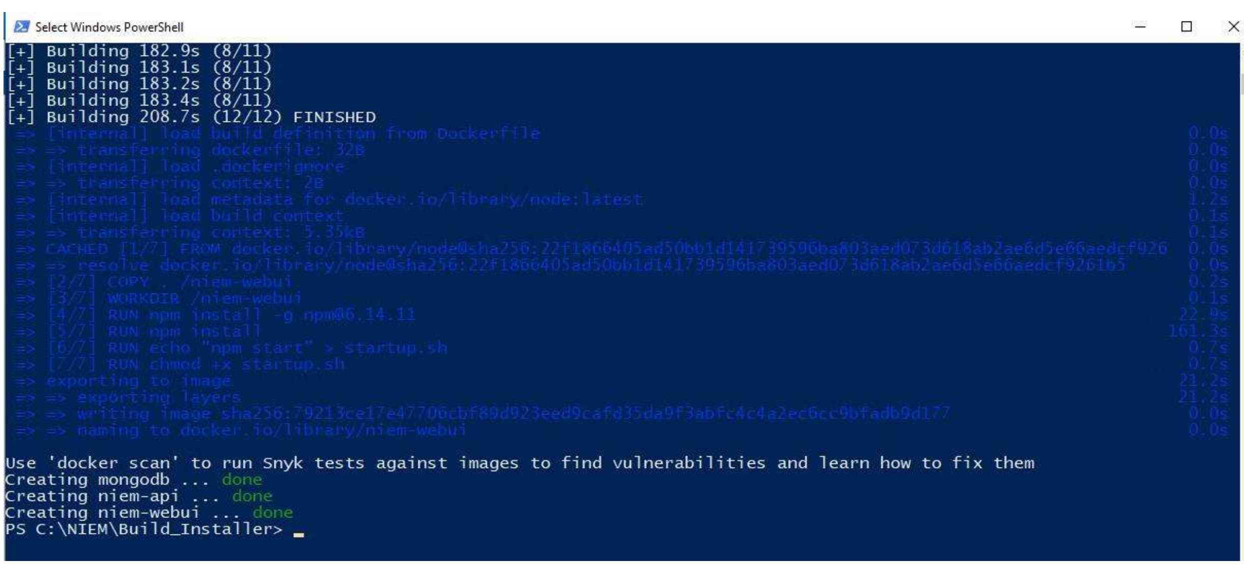


Figure 1.11

16. In the browser, refresh the page or navigate to localhost:3000. The application will now present the server instance that will require the user to sign in for access

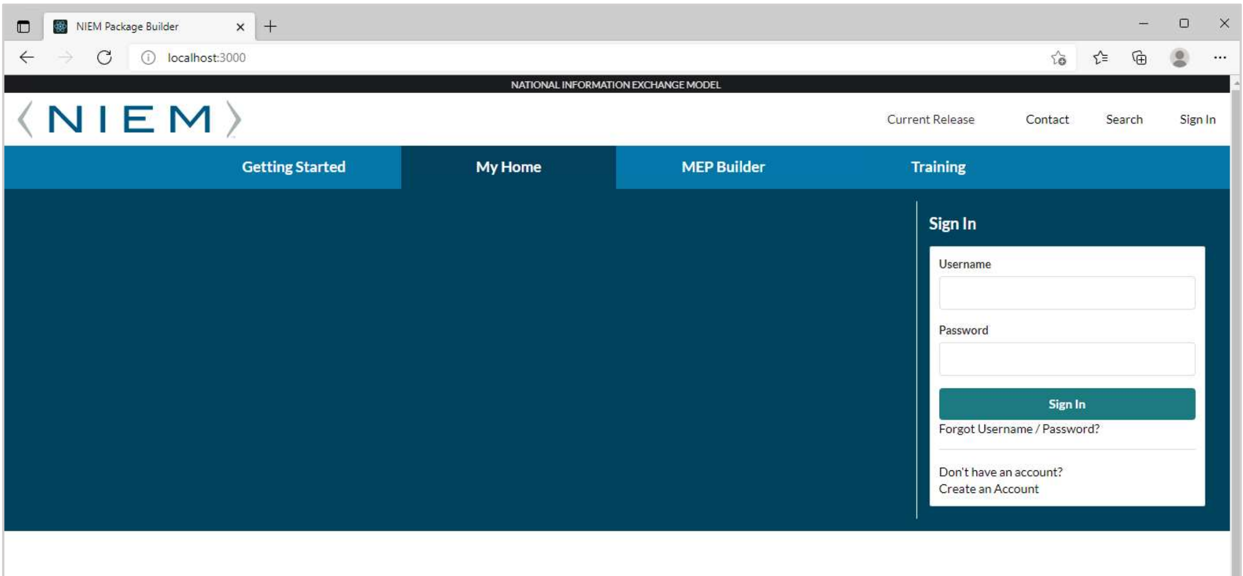


Figure 1.12

2. ADD INITIAL SUPER-ADMIN USERS (SYSADMIN)

A. Log in using supplied Sysadmin credentials.
B. This role will set up the application on the environment and set up the initial SuperAdmin accounts. The only accounts the Sysadmin can create are "SuperAdmin" accounts.
C. Local policy will dictate how required information will be provided to SysAdmin for account creation and other permissions. After login and required user information is provided, add user accounts using the following procedures:
    a. Upon login a pop-up modal will be available to Add Super Admin user accounts with the following required fields. (Note: See Figure 2.1)
        - First Name*
        - Last Name*
        - Email*
        - Phone*
        - Password*
        - Password Confirm* (i.e. the standard enter the same password again)

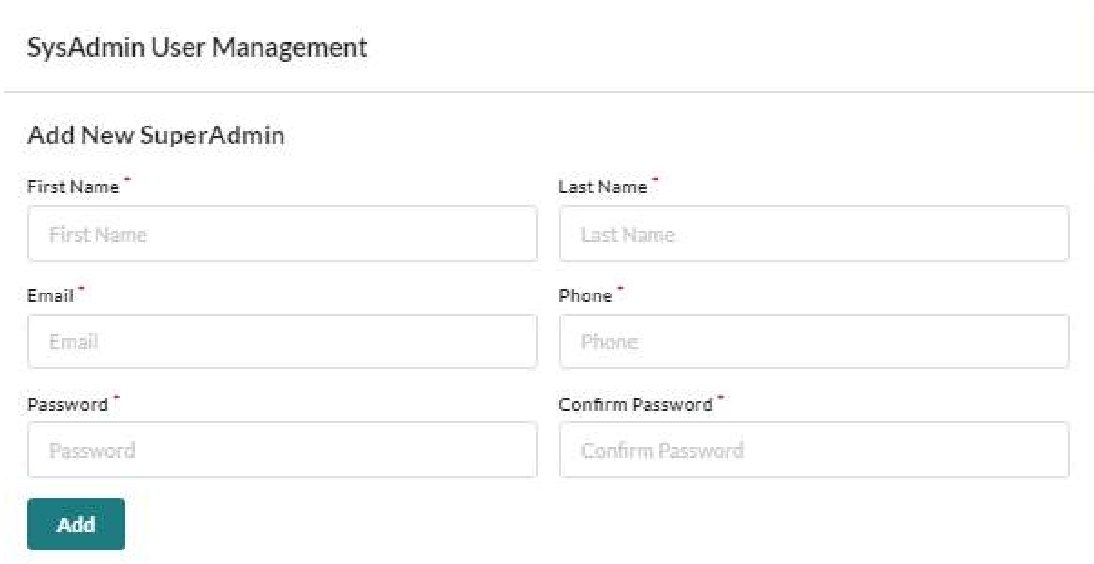


Figure 2.1

D. After filling in the required fields, click "**Add**" and a confirmation will render if the add was successful.

3. SUPER-ADMIN/ADMIN ROLES

A. There are three roles that are given permission to use the application. General User, Admin and SuperAdmin. Admin and SuperAdmin have special access to the Admin Module in addition to General User permissions. These roles and associated permissions are listed below:
   a. General User (User):
      - Account Registration
      - Ability to update their own account profile (e.g. email address)
      - Ability to create, read, update, and delete their own data
      - Ability to upload and delete their own artifacts (e.g. mapping spreadsheet, supporting documentation)
      - Ability to submit Contact form to request NIEM Management Office (NMO) assistance
      - Ability to import and export valid NIEM Schemas
      - Ability to browse and search NIEM components
      - Ability to add custom NIEM components (e.g. augmentations)
      - Additional features described within this document
   b. Admin:
      - *All permissions of a General User*
      - Ability to create, read, and update General User & Administrator Account profiles
      - Ability to review Contact form submissions

c. SuperAdmin:

- All permissions of an Administrator
- Ability to review Audit logs of all General User & Administrator CRUD activity
- Ability to review Audit logs of any developed API services
- Ability to create, read, update, and delete all User Account profiles

B. Admin Module

a. Default landing page once you click the “Administrator Tab” see Figure 3.1

Figure 3.1

b. Pending User Tab

i. Admin will approve or disapprove users from this tab using the green checkmark or the red X as depicted in figure 3.2. Note the number on the Pending Users tab will show how many users are pending Admin action. The numbers will decrease with each successful action completed.

Figure 3.2

1. Approve a user account request.
    a. Note: All users approved from this interface are give the role of "General User" by default. If you desire to assign another role to the approved user, you can do that on the "All Users" tab interface.
    b. To approve the user account, click the green check mark and affirm that you want to approve the user by clicking "Yes" as illustrated in figures 3.3 and 3.4

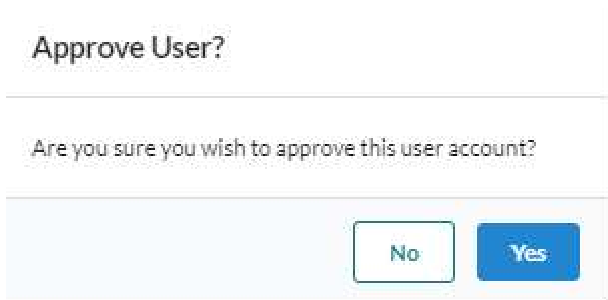


Figure 3.3

Figure 3.4

2. Deny a user account request
    a. Note: Any user denied an account will still be added to the "All Users" table where they can be approved if other information is provided that may change the course of action or deleted depending on the circumstances.
    b. To deny a user click the read "X"

c. Select a reason from the default reasons. Obviously, if the reason isn't provided you can select the last selection "Other" and provide further details in the "details" field. Note: both the reason and details are mandatory fields. See figures 3.5 through 3.8.

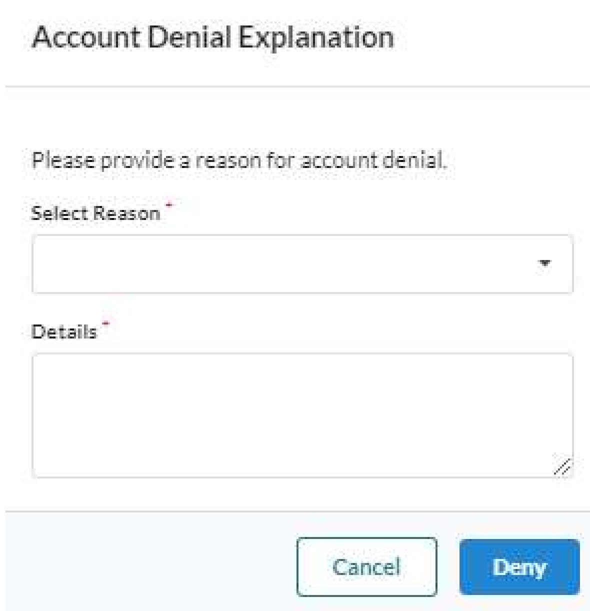


Figure 3.5

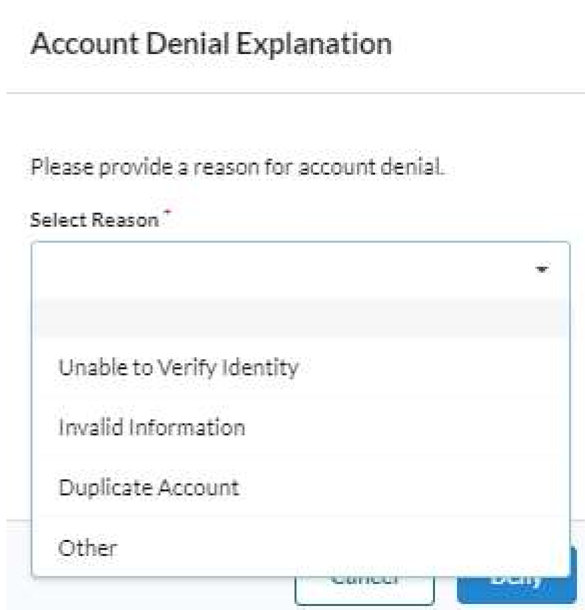


Figure 3.6

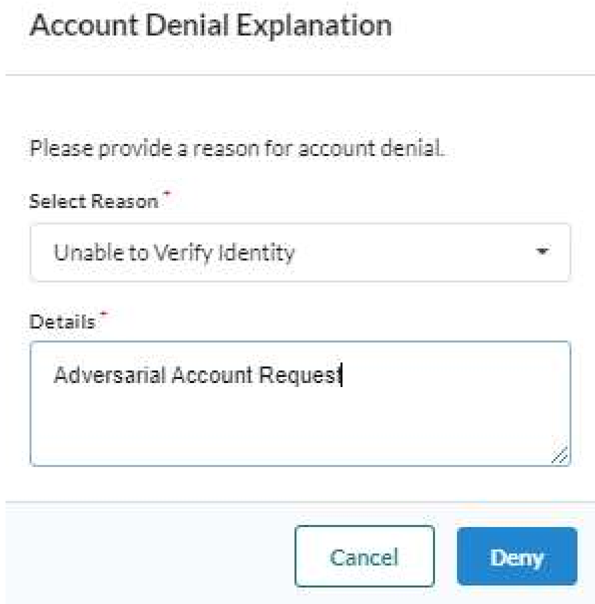


Figure 3.7

Figure 3.8

c. All Users Tab. The All Users tab (Figure 3.9) is where most of the administration of user accounts will occur. From this page admin can perform the following actions:
   i. Reset Password
   ii. Edit user account profile information
   iii. Delete a user account
   iv. Change Status of a user:
      1. Lock account
      2. Unlock account
      3. Revoke account
      4. Grant access for a denied user
   v. Add a user
   vi. Search using the integrated search filter
   vii. Transfer Packages

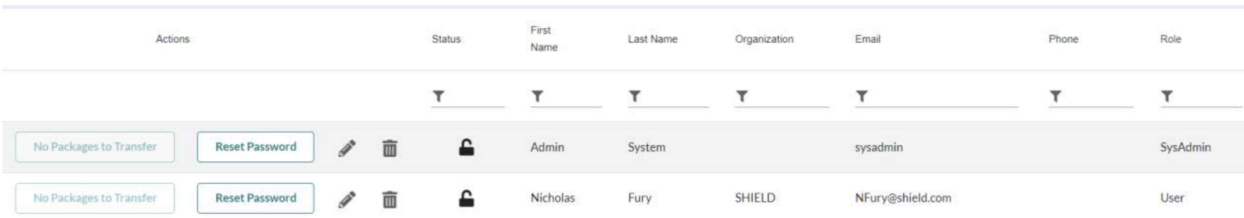


Figure 3.9

d. Reset Password. To reset a password, click the “Reset Password button.
    i. Enter a new password and confirm the new password. Figure 3.10. This password will act as a temporary password.
    ii. A pop-up notification will advise Admin to contact the user and inform the user that the temporary password will expire in 48 hours and only a login within that timeframe will prevent a lockout. Figure 3.11. Figures 3.12 and 3.13 depict what the user will see.

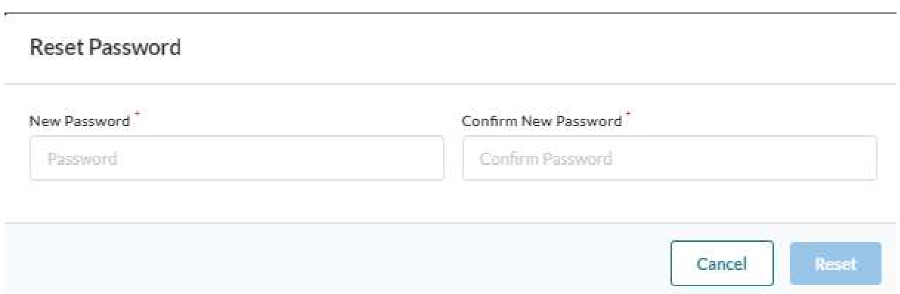


Figure 3.10

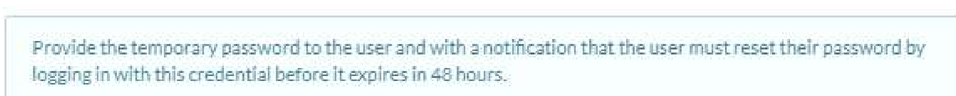


Figure 3.11

Figure 3.12

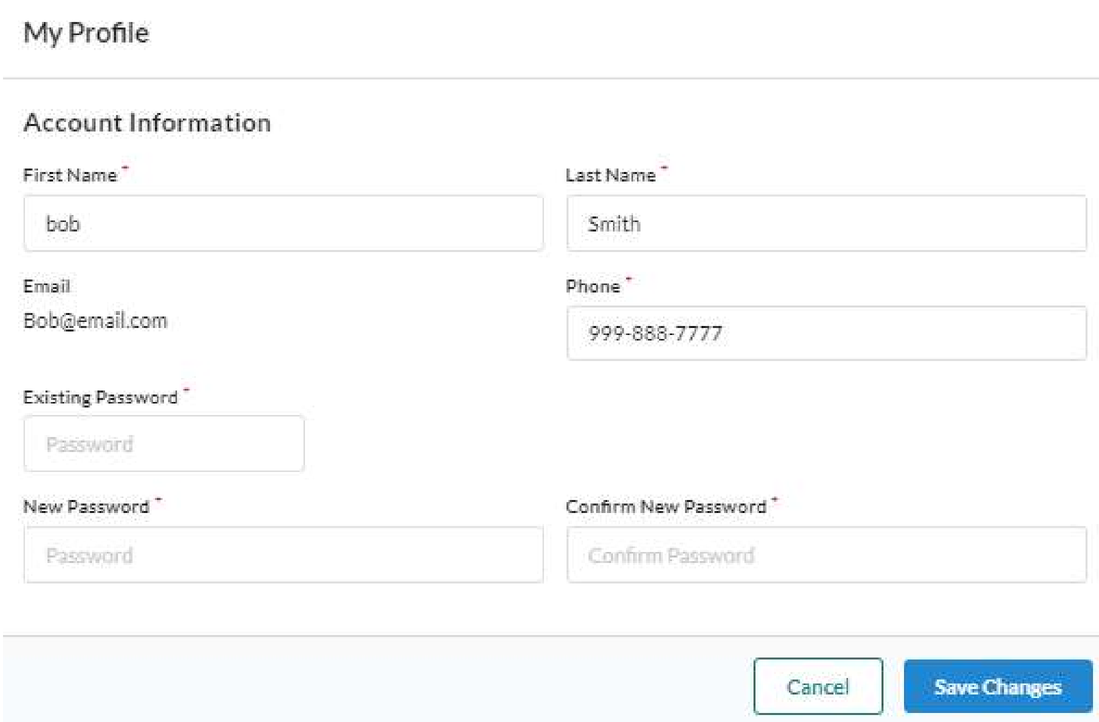


Figure 3.13

e. Edit an Account Profile. Users account information can be modified by clicking the pencil icon. 

   i. After clicking the pencil, you can update the following profile information:
      1. First Name
      2. Last Name
      3. Organization
      4. Email
      5. Phone
      6. Role (Note: Admin role permission rules apply.
   ii. Click the red X to cancel or the green checkmark to confirm.

f. Status. A user's status can be updated by clicking on the status icons. These icons are defined below:
   i. Unlocked - This is the normal state showing that a user is active and authorized to use the application
   ii. Locked - A locked icon can mean a few things
      1. The user has locked themselves out of the application.
      2. Admin has locked the account by clicking the unlocked icon and elected to lock the account.
   iii. Revoked – A revoked account can only be a user status if an Admin revokes the account. Similar to a locked account, the user cannot access the tool.
   iv. Denied - A user may have been denied access from the pending users page by an admin. However, here you have an opportunity change that status by clicking the denied icon.

g. Changing Status of an account from the All-User's Table.
   i. Unlock to locked or revoked.
      1. Click the Edit Icon
      2. Click the drop-down to see the status options.

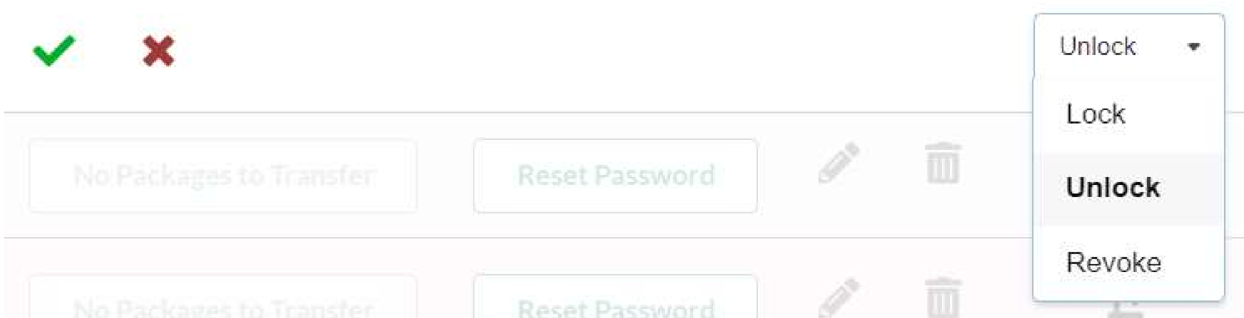


Figure 3.14

      3. Select Lock or Revoke.
      4. Enter the Reason for the status change and select 'Yes, Lock' (or 'Yes, Revoke')

   ii. Locked/Revoke to Unlocked
      1. Click the edit icon
      2. Select Unlock

3. Add an unlock reason in the Description field. Figure 3.15

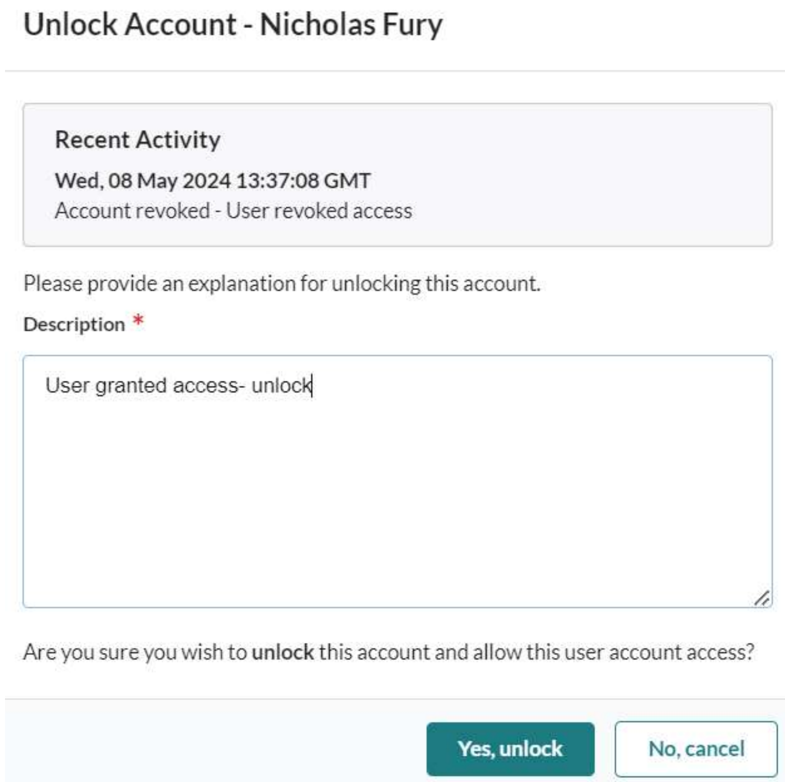


Figure 3.15

4. Click ‘Yes, Unlock’ to unlock the account.

iii. Give Access to a Denied User.

1. Click the edit icon 
2. Select Unlock
3. Add an unlock reason in the Description field. Figure 3.16

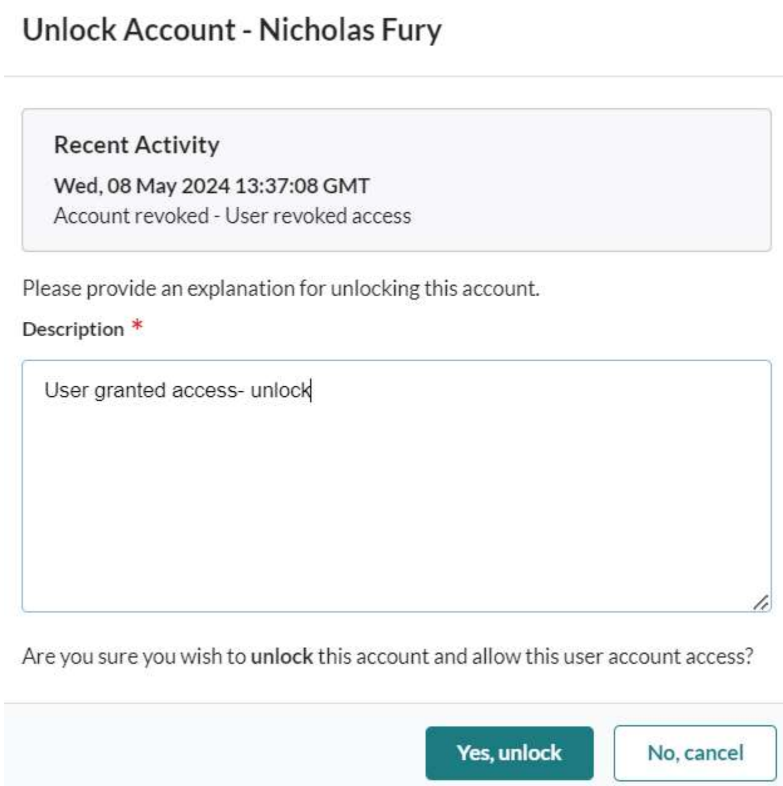


Figure 3.16

4. Click ‘Yes, Unlock’ to unlock the account.

h. Adding a user. A new user can be added by clicking the plus sign at the top of the page. See figure 3.17

Figure 3.14

i. After clicking the checkmark, enter the users profile information. On the open form as shown on figure 3.18. All of the following is mandatory:
   1. First Name

2. Last Name
3. Organization
4. Email
5. Phone
6. Role. Note only User, Admin or SuperAdmin the only available options.

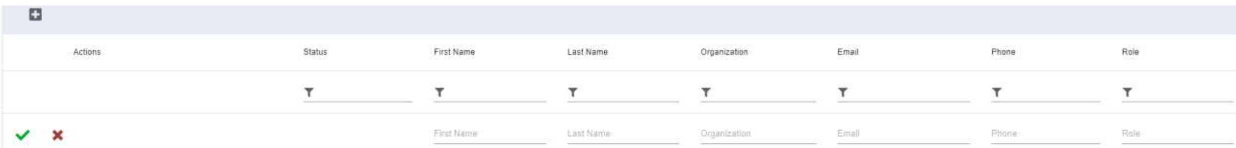
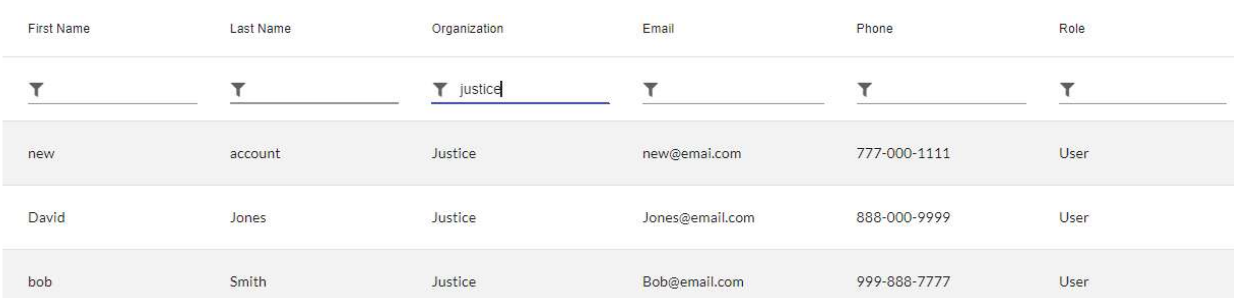


Figure 3.15

ii. Click the red X to cancel or the green checkmark to confirm.
iii. Adding a password to the newly created account using this feature. Now, go and click the reset password button for the new account and continue the same workflow described for resetting a password to include the notification to the user and the same 48-hour limitations apply.

i. Filter. A filter is provided to assist the admin with finding users. Standard filter functionality features apply. Note. The filter does not filter on status as of release 2.0. Figure 3.19 depicts a typical filtering action.



First Name	Last Name	Organization	Email	Phone	Role
		justice			
new	account	Justice	new@emai.com	777-000-1111	User
David	Jones	Justice	Jones@email.com	888-000-9999	User
bob	Smith	Justice	Bob@email.com	999-888-7777	User

Figure 3.16

4. IMPORTING NEW NIEMOPEN RELEASE

A. Importing a new NIEMOpen Release is done via the Administrator tab in the MEP Builder Tool. There are options to: follow a link to receive more information about NIEMOpen releases, Check for Updates, and Update the MEP Builder Tool to the newest version of NIEMOpen (if available). This process is only available for users with Admin or higher permissions.

B. It is recommended that the process to Import a new NIEMOpen Release should occur during non-business hours. Users should not be actively using the application to avoid unexpected behavior while the import is in progress.

C. Steps for Updating NIEMOpen Release:

   a. Navigate to Release Tab in the Administrator section

Figure 4.1

   b. Select Check for Update button, if there are no updates then a message will appear (Figure 4.2). if there are updates available, then you will be prompted to update the release.

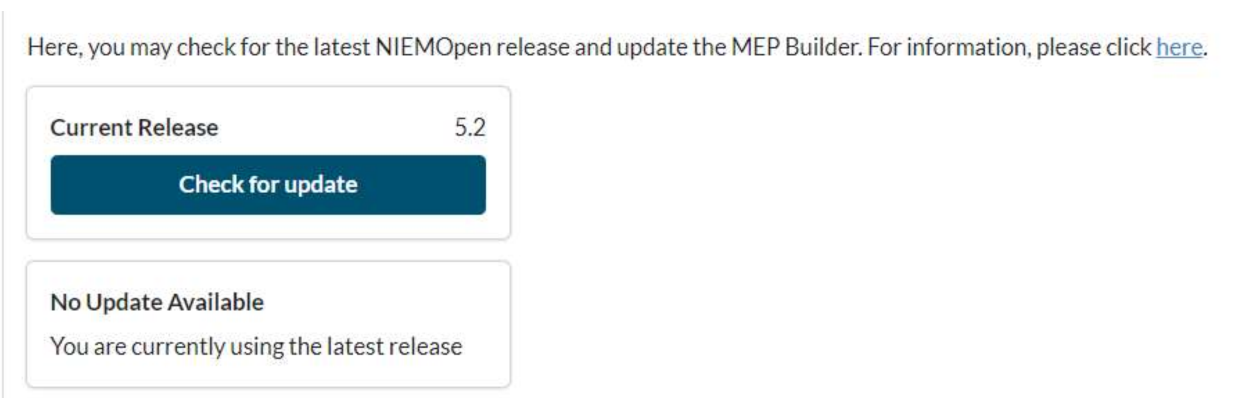


Figure 4.2

5. TRANSFERRING PACKAGES TO ANOTHER USER

A. An Administrator can Transfer Packages from one user to another. To start this process, navigate to the All Users tab of the admin section.

Figure 5.1

a. Find the user that you would like to transfer package from and click the **Transfer Packages** button.

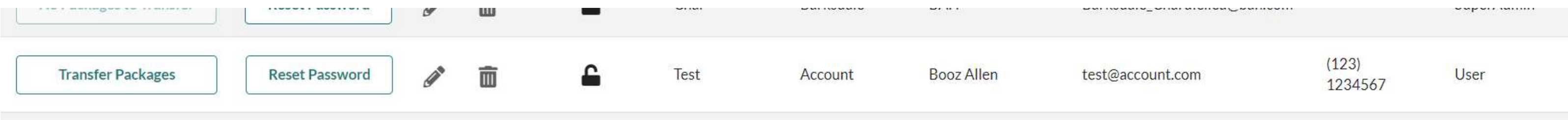


Figure 5.2

b. The Transfer modal will now open and give instructions on how to proceed.

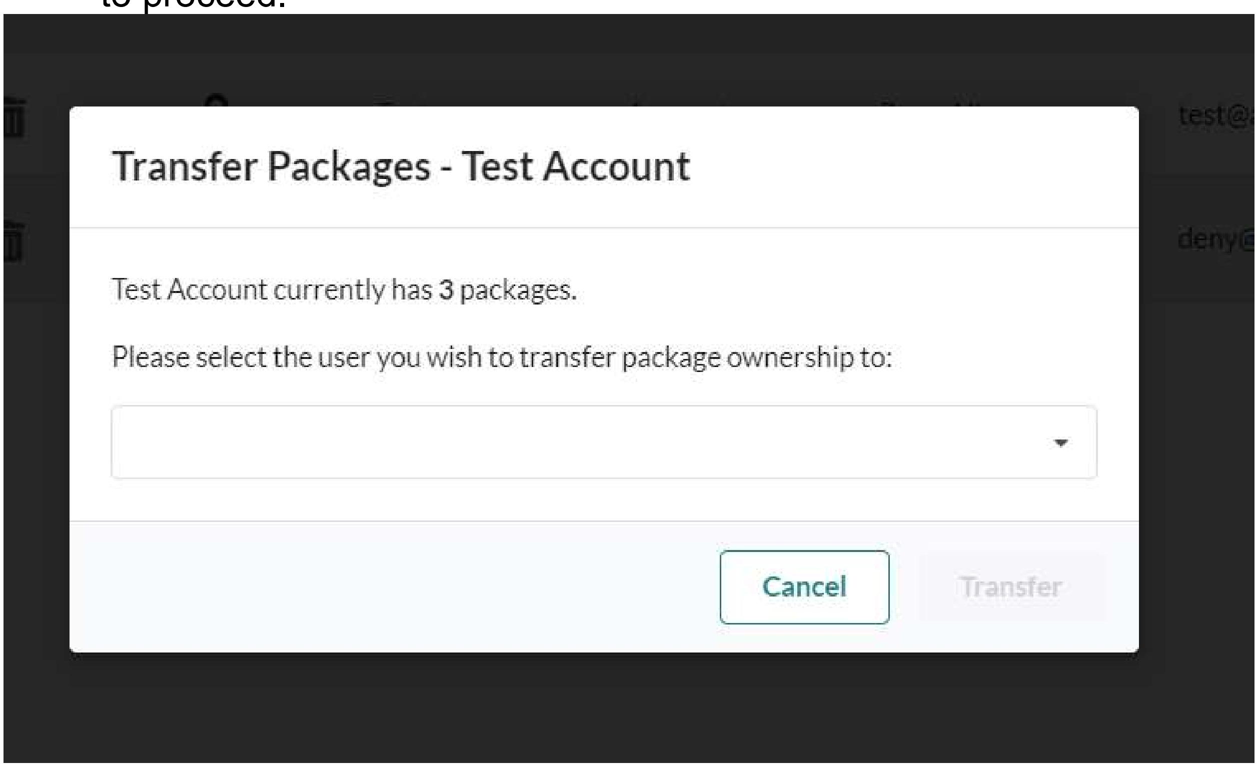


Figure 5.3

c. Select the User you want to transfer packages to using the drop-down.

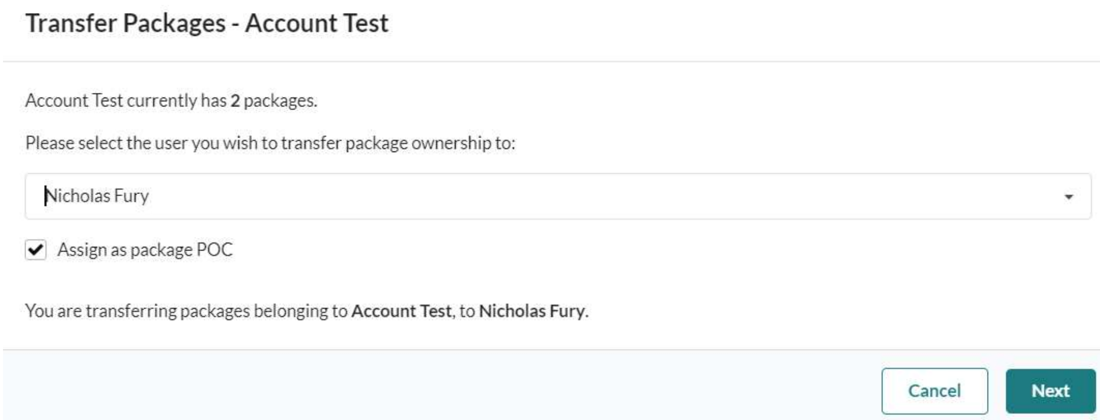


Figure 5.4

d. An option to auto-update the transferred packages POC to the new owner is available. The checkbox is auto-selected, so if the transferred packages do not need to have the POC updated, uncheck the box.

e. Click Next.

f. Packages can now be selected for transfer. Available actions include:

   i. Manually Selecting/Unselecting packages to transfer
   ii. Individually updating POC Name and POC Email
   iii. Select All/Unselect All for Transfer

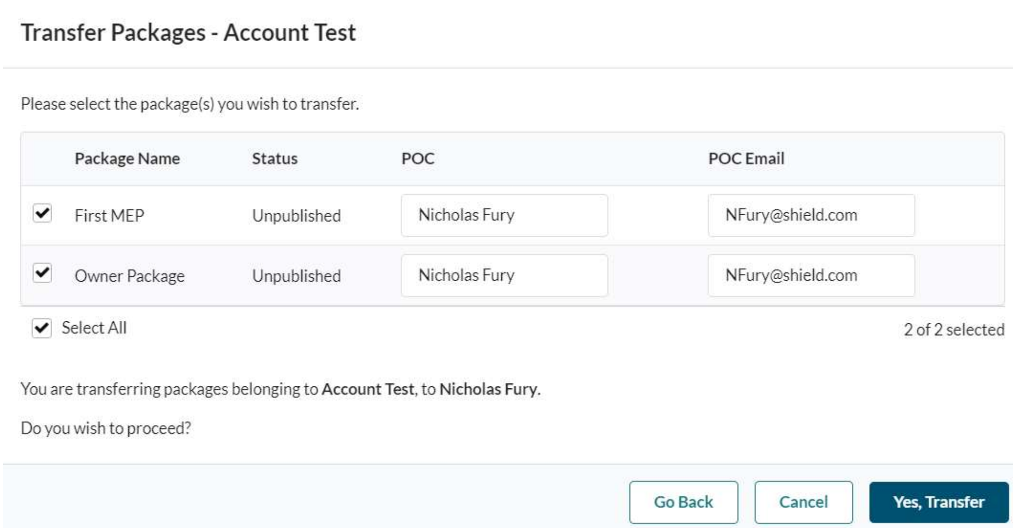


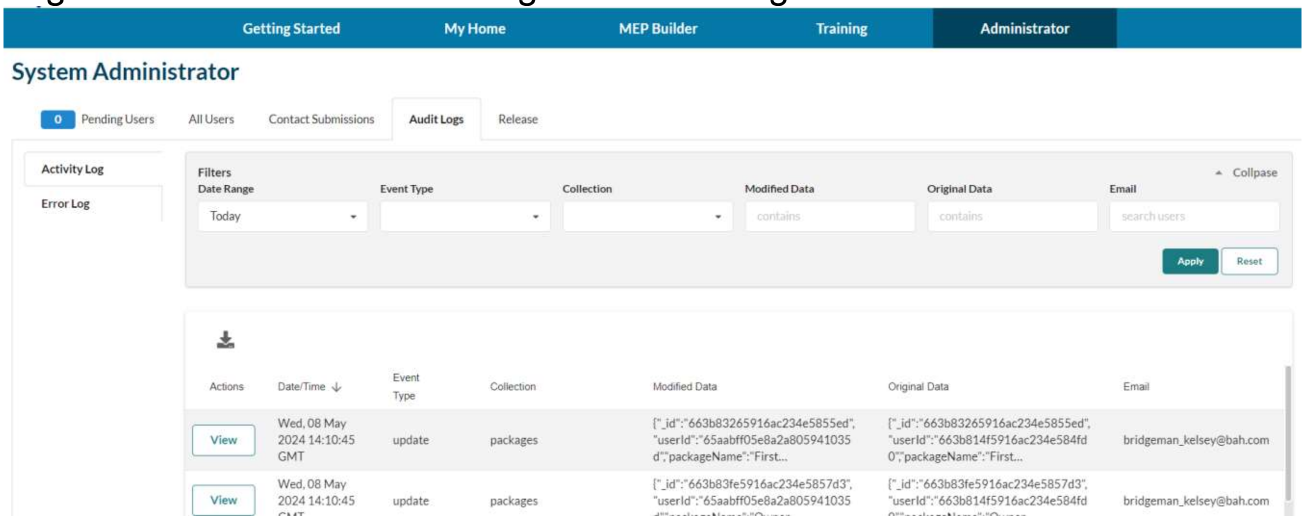
Figure 5.5

g. Once transfer selections have been made, click ‘Yes, Transfer’. Transfer success message will display, and packages will be transferred to the appropriate account.

Figure 5.6

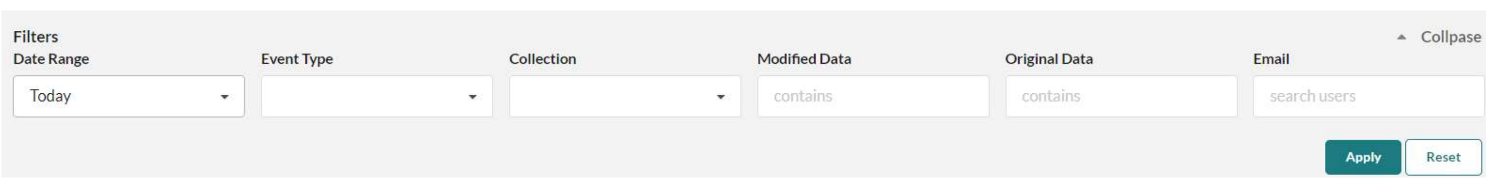
6. AUDIT LOGS

A. The Audit Logs can be found on the Audit Logs tab of the Administrator module. Here, an Admin can look into Activity Logs as well as Error Logs to assist in troubleshooting or user management.

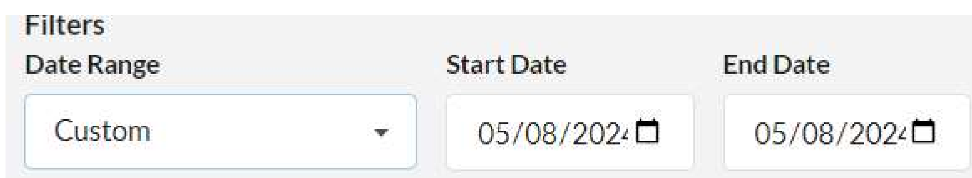


B. The Activity Logs tab includes entries of all user actions involving creating, updating, or deleting.

a. The top section of the screen includes filtering options. The options avaible are described below.



i. Date Range
1. Today (Default)
2. This Week
3. Last Week
4. Last Month
5. Custom



ii. Event Type
1. Create
2. Update

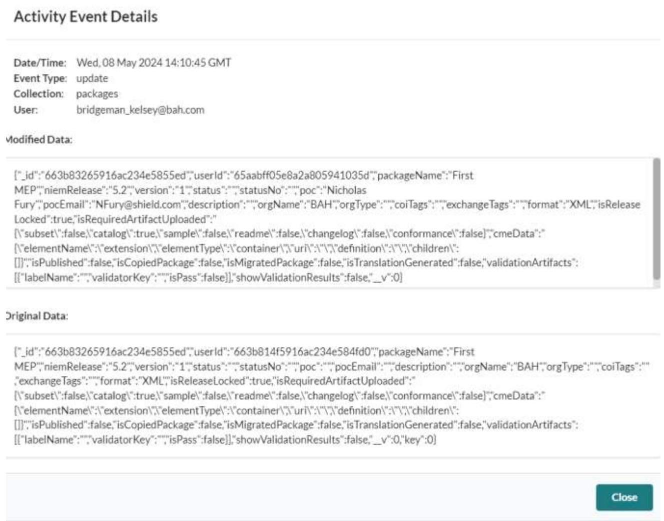
        3. Delete
    iii. Collection
        1. Artifacttrees
        2. Auditlogs
        3. Errorlogs
        4. Fileblobs
        5. Mappingdocs
        6. Packages
        7. Users
    iv. Modified Data - Freeform text field utilizing the ‘contains’ search condition.
    v. Original Data - Freeform text field utilizing the ‘contains’ search condition.
    vi. Email - Freeform text field to search using a user’s email.
    vii. Apply Button - Click to apply the filter conditions selected.
    viii. Reset Button – Click to clear the filter conditions selected.

b. The Download Icon allows for an Activity Log download to user’s computer. If there are filter conditions applied to the Log results, then the download will only include those in the search results.

c. The Activity Log entries display in a table format.

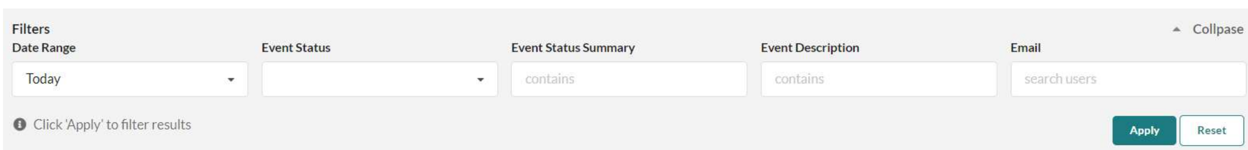
Actions	Date/Time ↓	Event Type	Collection	Modified Data	Original Data	Email
View	Wed, 08 May 2024 14:10:45 GMT	update	packages	{"_id":"663b83265916ac234e5855ed", "userId":"65aabff05e8a2a805941035d","packageName":"First...	{"_id":"663b83265916ac234e5855ed", "userId":"663b814f5916ac234e584fd0","packageName":"First...	bridgeman_kelsey@bah.com
View	Wed, 08 May 2024 14:10:45 GMT	update	packages	{"_id":"663b83fe5916ac234e5857d3", "userId":"65aabff05e8a2a805941035d","packageName":"Owner...	{"_id":"663b83fe5916ac234e5857d3", "userId":"663b814f5916ac234e584fd0","packageName":"Owner...	bridgeman_kelsey@bah.com

d. The View option on an entry gives more detailed information on the entry.

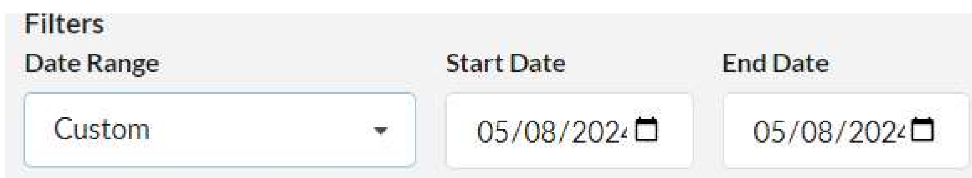


C. The Error Logs tab includes entries of all user actions involving different errors encountered.

a. The top section of the screen includes filtering options. The options avaible are described below.



i. Date Range
   1. Today (Default)
   2. This Week
   3. Last Week
   4. Last Month
   5. Custom



ii. Event Status – Various types of errors are able to be filtered here.

iii. Event Status Summary - Freeform text field utilizing the ‘contains’ search condition.
iv. Event Description - Freeform text field utilizing the ‘contains’ search condition.
v. Email - Freeform text field to search using a user’s email.
vi. Apply Button - Click to apply the filter conditions selected.
vii. Reset Button – Click to clear the filter conditions selected.

b. The Download Icon allows for an Error Log download to user’s computer. If there are filter conditions applied to the Log results, then the download will only include those in the search results.

c. The Error Log entries display in a table format.

Actions	Date/Time ↓	Event Status	Event Status Summary	Event Description	Email
View	Wed, 08 May 2024 14:07:15 GMT	400	Bad Request	{"message":"Request failed with status code 400","name":"AxiosError","stack":"AxiosError: Request failed with status code 400\n at settle (/app/node_modules/axios/lib/core/settle.js:19:12)\n at...	joynes_kazare@bah.com
View	Wed, 08 May 2024 14:07:15 GMT	500	Internal Server Error	{"status":500,"message":"Error when saving cmf file."}	joynes_kazare@bah.com

d. The View option on an entry gives more detailed information on the entry.

Error Event Details

Date/Time: Wed, 08 May 2024 14:07:15 GMT
Event Status: 400
Event Status Summary: Bad Request
User: joynes_kazare@bah.com

Event Description:

{"message":"Request failed with status code 400","name":"AxiosError","stack":"AxiosError: Request failed with status code 400\n at settle (/app/node_modules/axios/lib/core/settle.js:19:12)\n at Unzip.handleStreamEnd (/app/node_modules/axios/lib/adapters/http.js:585:11)\n at Unzip.emit (node:events:530:35)\n at endReadableNT (node:internal/streams/readable:1696:12)\n at processTicksAndRejections (node:internal/process/task_queues:82:21)","config":{"transitional":{"silentJSONParsing":true,"forcedJSONParsing":true,"clarifyTimeoutError":false},"adapter":["xhr","http"],"transformRequest":[null],"transformResponse":[null],"timeout":0,"xsrfCookieName":"XSRF-TOKEN","xsrfHeaderName":"X-XSRF-TOKEN","maxContentLength":-1,"maxBodyLength":-1,"env":{},"headers":{"Accept":"application/json, text/plain, */*","Content-Type":"multipart/form-data; boundary=------------------------566801582479620317470438","User-Agent":"axios/1.6.2","Content-Length":"151573","Accept-Encoding":"gzip, compress, deflate, br"},"method":"post","url":"https://tools.niem.gov/api/v2/transforms/models?from=cmf&to=cmf","data":{"_overheadLength":159,"_valueLength":0,"_valuesToMeasure":[{"fd":null,"path":"temp.cmf.xml","flags":"r","mode":438,"end":null,"bytesRead":151358,"_events":{},"_readableState":{"highWaterMark":65536,"buffer":[],"bufferIndex":0,"length":0,"pipes":[],"awaitDrainWriters":null},"_eventsCount":2}],"writable":false,"readable":true,"dataSize":0,"maxDataSize":2097152,"pauseStreams":true,"_rele
[],"_currentStream":null,"_insideLoop":false,"_pendingNext":false,"_boundary":"------------------------
-566801582479620317470438","_events":{},"_eventsCount":3}},"code":"ERR_BAD_REQUEST","status":400}

Close

7. ADDING A PLUG-IN

A. Adding new functionality, or a plugin, to the tool can be done with a NestJs Dynamic Module. A Dynamic Module is a module that is created at runtime.

e. To make a Dynamic Module, simply configure a new NestJs Module with the desired functionality.
f. Inside the module file, be sure to import DynamicModule from @nestjs/common, and declare a new method called ‘register()’ which returns a DynamicModule with the desired module, services, controllers, and exports.
g. Finally, import the plugin module into the app.module.ts file. Inside the imports array, add: ‘moduleName’.register(args).

B. Screenshots with an example are attached below, along with external documentation.

```
import { Module, DynamicModule } from '@nestjs/common';
import { MyPluginController } from './my-plugin.controller';
import { MyPluginService } from './my-plugin.service';

@Module({})
export class MyPluginModule {
  static register(/** arguments */): DynamicModule {
    return {
      module: MyPluginModule,
      providers: [MyPluginService],
      controllers: [MyPluginController],
      exports: [MyPluginService],
    }
  };
}
```

Figure 7.1

```
import { Controller } from '@nestjs/common';
import { MyPluginService } from './my-plugin.service';

@Controller('my-plugin')
export class MyPluginController {
  // myPlugin provider is injected from the constructor.
  constructor(private myPluginService: MyPluginService) {}
}
```

Figure 7.2

```
import { Injectable } from '@nestjs/common';

@Injectable()
export class MyPluginService {
    private readonly collection: string[] = [];

    getData(): string[] {
        return this.collection;
    }

    postData(data: string): void {
        this.collection.push(data);
    }
}
```

Figure 7.3

```
import { MyPluginModule } from './plugins/my-plugin/my-plugin.module';

@Module({
  imports: [
    UserModule,
    MongoRepoModule,
    FilesModule,
    AuthModule,
    AuditLogModule,
    ReleasesModule,
    ErrorLogModule,
    ConfigModule.forRoot({
      isGlobal: true,
      load: [config],
    }),
    // MongoDB Connection
    MongooseModule.forRootAsync({
      imports: [ConfigModule],
      useClass: DatabaseConfig,
    }),
    SsgtModule,
    MyPluginModule.register(/** arguments */),
  ],
  controllers: [],
  providers: [],
})
export class AppModule implements NestModule {
  configure(consumer: MiddlewareConsumer) {
    // throw new Error('Method not implemented.');
  }
}
```

Figure 7.4

C. External Resources:

- How to Build A Plugin System with Node.js
- Plugin Architecture for NestJs
- Walkthrough of Building a Plugin System with Node.js from scratch
- Documentation about dynamic modules

NOTES