Miltech Mission Hub

Mission Package

To distribute multiple missions together you can create a mission package.

Creating the mission package

Work folder (easy)

You need to create a folder for your mission package in \work\MissionPacks\ . Inside that folder, put all your mission .json files, thumbnails, briefing images, and other assets (Sounds, Scripts, Images, Weather).

In your mission package folder, create a new file called MissionPackDefinition.json and fill in the fields:

```
{
"GUID": "YOUR-GENERATED-GUID",
"Title": "Your Missionpack Title",
"Description": "Your Missionpack Description",
"Author": "YourName",
"Thumbnail": "YourMissionPack-thumbnail.jpg",
"Files": [
{
"FileName": "1MissionName.json",
"Thumbnail": "1MissionName_thumbnail.jpg",
"BriefingImage": "1MissionName_briefing.jpg"
},
{
"FileName": "2MissionName.json",
"Thumbnail": "2MissionName_thumbnail.jpg",
"BriefingImage": "2MissionName_briefing.jpg"
}
]
}
```

Caution
Generate a random GUID for your mission pack. You can use https://guidgenerator.com/

Community folder (harder)

You can distribute your mission package as package that can be placed in the community folder. The user then has to manually add the pack to the mission hub. The community folder method is also more complex. Therefore, it's recommended to use the work folder to distribute mission packages.

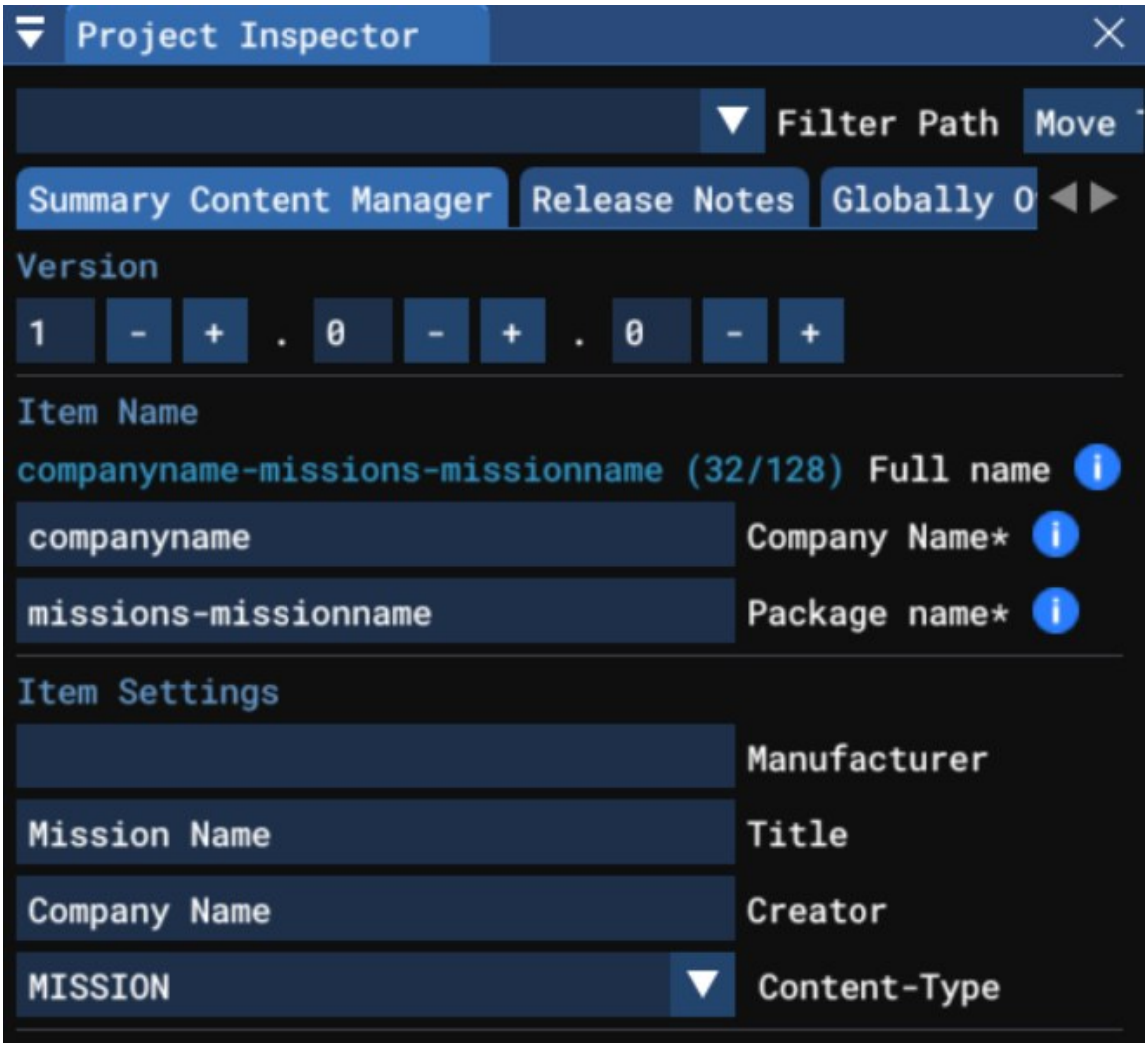
Template project

This zip contains an example Devmode project:

↓ Download attachment

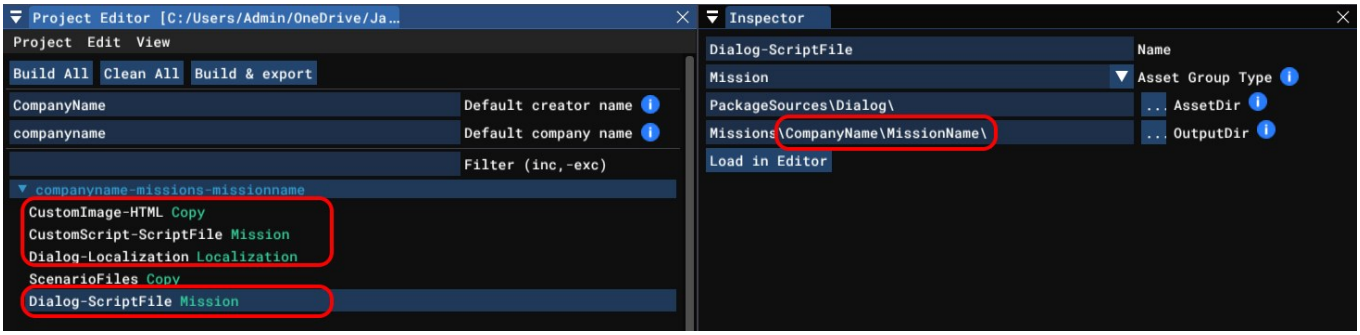
Download and extract the .zip file. Open the project in MSFS devmode.

Once opened, you need to change the *Package name* and other details in the project inspector.



And you need to change the *OutputDirs* for the all asset groups **except for the ScenarioFiles asset group**.

Replace CompanyName with your name and MissionName with the name of the mission.

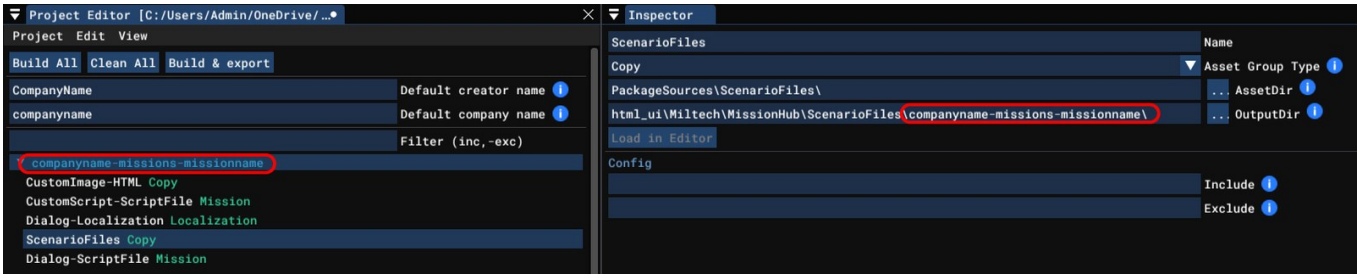


Caution
Be sure to change the package and folder names, to avoid conflicts with other missions.

Defining your mission pack details

Preparing ScenarioFiles asset group

In the MSFS project editor, edit the **OutputDir** of the *ScenarioFiles* asset group. The last part of the OutputDir should exactly match your package name:



Adding missions

Paste your mission files, thumbnail images, and briefing images in the PackageSources\ScenarioFiles\ folder (See: Sharing and Installing Missions).

In that folder, open the MissionPackDefinition.json file:

```
{
    "GUID": "6CFA8116-A9A6-41DF-A3A6-8E253A21E6E8",
    "Title": "Example Mission Pack",
    "Description": "This is an example Miltech Mission Hub mission pack. This pack includes a variety of missions, and custom content like dialog and images.",
    "Author": "YourName",
    "Thumbnail": "ExampleMissionPack-thumbnail.jpg",
    "Files": [
        {
            "FileName": "1MissionName.json",
            "Thumbnail": "1MissionName_thumbnail.jpg",
            "BriefingImage": "1MissionName_briefing.jpg"
        },
        {
            "FileName": "2MissionName.json",
            "Thumbnail": "2MissionName_thumbnail.jpg",
            "BriefingImage": "2MissionName_briefing.jpg"
        }
    ]
}
```

Edit all the fields with your own information. You can add more files if needed.

You can add multiple briefing images if you want, like this:

```
"BriefingImage": ["1MissionName_briefing1.jpg", "1MissionName_briefing2.jpg", "1MissionName_briefing3.jpg"]
```

> Caution
> Generate a random GUID for your mission pack. You can use https://guidgenerator.com/

Adding custom content

Custom Dialog

Follow the steps in Custom Dialog.

Custom Image

Add the custom image(s) to the PackageSources\CustomImage\ folder. In the Custom Image *Image Path*, you can specify the image like: CompanyName/MissionName/ImageName.jpg

Custom Script

Add the custom script(s) to the PackageSources\CustomScript\ folder. In the Custom Script *Script Path*, you can specify the script like: Missions\CompanyName\MissionName\CustomScriptName