

eftsure

DEPLOYMENT GUIDE

**eftsure inside Microsoft
Dynamics 365 for
Finance and Operations
Enterprise Edition**

Change History

Version	Date	Change Summary	Author
1.0	28/02/2020	Initial draft	Greg
1.9	5/10/2023	Updates for troubleshooting and issues	Greg
1.10	20/10/2023	Updates for Upgrades	Greg
1.11	14/11/2023	Updates for DevBox copies via GetLatest	Greg
1.12	28/01/2024	Updates for Azure hosted build pipelines	Greg

Table of contents

Updates from Version 925 and BELOW	3
Introduction.....	3
PACKAGES.....	4
Deployment from LCS	5
Deployment FROM COMMAND LINE	6
Clean Packages	6
Deployable Package.....	7
Source control	7
UPGRADES	9
All in one package	10
Upload package to LCS.....	10
Deployments to cloud hosted environments (DEV,Test, Live)	12
Apply a package to a production environment by using LCS	13
Deployment to Developer environments	14
Post deployment activities.....	14
Post DB Refresh activities.....	16
Uninstall	17
Troubleshooting	18



UPDATES FROM VERSION 925 AND BELOW

Since v 925, a new license framework is being used, you should delete all PESLicense files from source control, these will no longer be used.

If such files are still deployed to Tier2+ environments, the following error is likely to occur and rollback entire deployment.

```
please investigate before synchronizing again: 'ClrErrorException:ApplicationException
  at Microsoft.Dynamics.Ax.Xpp.ReflectionCallHelper.MakeStaticCall(Type type, String MethodName,
Object[] parameters) in D:\dbs\sh\123t\0505_210821\cmd\k\Source\Kernel\Xppil\XppSupport
\ReflectionCallHelper.cs:line 1634
  at Microsoft.Dynamics.AX.Deployment.DeploymentHelper.AppDBOperations.ImportLicenseFile(String
fileName)
  at
Microsoft.Dynamics.AX.Framework.Database.Tools.LegacyCodepath.<>c__DisplayClass36_0.<NOTE_LeavingS
ynchronizer_CallStackAboveThisLineIsCustomCode>b__0()
  at Microsoft.Dynamics.AX.Framework.Database.Tools.LegacyCodepath.ExecuteWithinAOS(SyncOptions
syncOptions, String sqlConnectionString, IMetadataProvider metadataProvider, Func`1 func, Action`1
errorHandler)
Inner exception:
ApplicationException:Error: method: SysLicenseCodeReadFile::serverISVProcessingInternal, error
message: License code PESLicense [9999999995] is missing in Metadata.
  at Dynamics.AX.Application.SysLicenseCodeReadFile.`run()
  at Dynamics.AX.Application.SysLicenseCodeReadFile.run()
  at Dynamics.AX.Application.SysOperationSandbox.`startOperation(SysOperationIMarshalledRun
operationInstance)
```

INTRODUCTION

This user guide will walk you through all the processes relevant to deployment of eftsure's extension: eftsure inside Microsoft Dynamics for Finance and Operations.

Three distinct methods are used for deployment depending on the target environment.

The **first deployment** is for your initial **Development** environment. You will receive / download a copy of our latest release (deployable package) that your deployment team will first deploy into your Development environment and **source control**.

For **all other Development** environments, your developers will deploy our latest release to their development machine by calling a **Get Latest in Visual Studio**.

For all **Tier2+**, the Deployable Package will need to be built including all other packages (All In One Package) and deployed **using LCS features**.

Note: We deploy binaries using Software Deployable Packages, the source code is not deployed. The Software deployable packages files are shipped by eftsure. We encourage you to be on the latest version that you will find on LCS.

For Live environments, the same Deployable Package approved in Test will be deployed using LCS features, after testing and approval in Test.

It's important to understand each deployment technique prior to performing a deployment of our latest release.

For more information, please refer to the following:

<https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/deployment/cloud-deployment-overview>

For on premise deployments, please refer to the following:

<https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/deployment/on-premises-deployment-landing-page>

Updates are applied using deployable packages. Applying updates causes system downtime. All relevant services will be stopped, and you won't be able to use your environments while the package is being applied. You should plan accordingly.

On Premise

This user guide describes the release process for all environments deployed through Lifecycle Services, on-premise release documentation is not available at this date.

This user guide describes the LCS deployment features only, the Deployable Packages are standard, and therefore can also be deployed via command line to on premise environments or cloud environments:

<https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/deployment/install-deployable-package>

Upgrades

Upgrades are slightly different from first time installs, please see Upgrades for more details.

Troubleshoot package deployment failures

If package deployment fails, see the [Troubleshoot package application issues topic](#).

PACKAGES

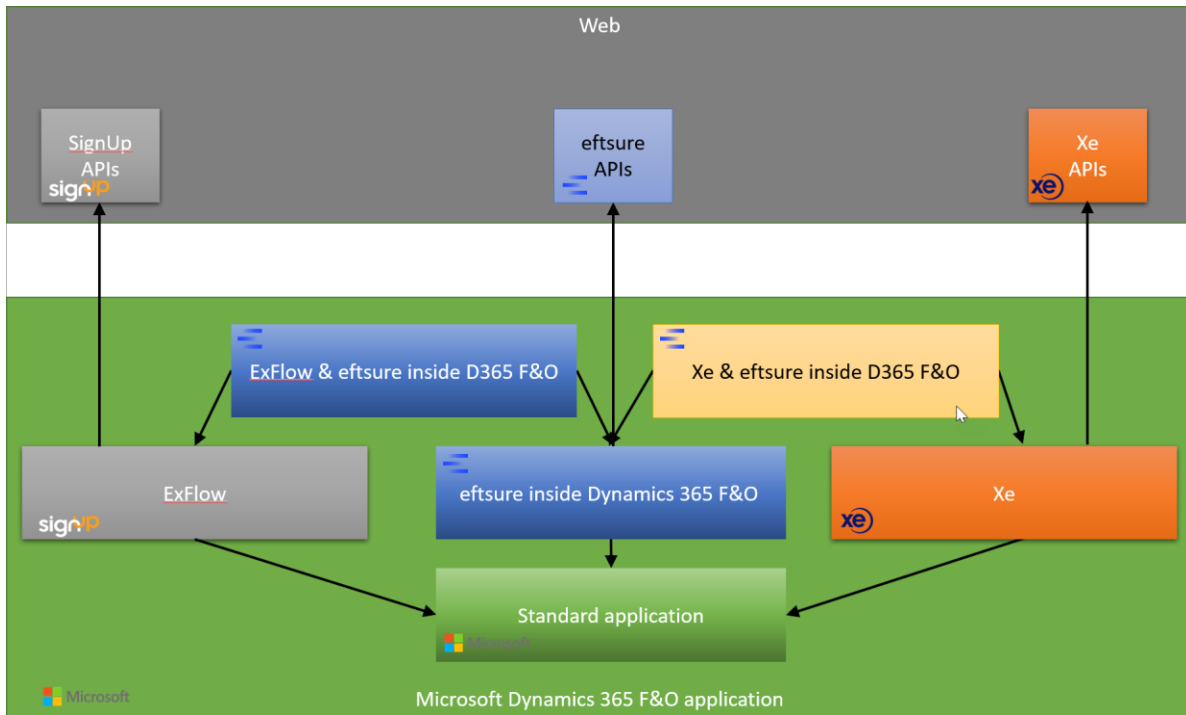
Eftsure ships the following 3 packages:

- Eftsure inside D365 (core features)

- ExFlow and eftsure inside D365 (to be installed for ExFlow customers)

- Xe and eftsure inside D365 (to be installed for Xe customers)

The packages architecture looks like the following:



The main package can be deployed on it's own,

The two partner packages require their matching partner packages and will fail if not available.

Please see our partners to get their packages.

Important:

In all scenarios, make sure to clean the packages folder for each package. For more information,

Please refer to Clean Packages.

DEPLOYMENT FROM LCS

Receive an LCS invitation for the project from eftsure,

Go to LCS, then Solution Management.

If the eftsure solution is not yet available, select Import and import the solution.

Go to LCS > **Project** > Asset Library > Software deployable package > Find eftsure package(s).

Deployable package to your environment.

DEPLOYMENT FROM COMMAND LINE

Alternatively, the deployment can be achieved via command line.

Find the latest eftsurre package and download it to c:\Temp

Extract eftsurre package to C:\Temp (No spaces in the folder name)

Run the following script in PowerShell as Admin

```
dir -Path c:\temp -Recurse | Unblock-File
```

Close PowerShell,

Note: Due to previous issues, it is **recommended to clean eftsurre package(s), see clean Packages**

Deploy the latest eftsurre package:

```
AxUpdateInstaller QuickInstallAll
```

Download file, copy to c:\Temp

CLEAN PACKAGES

We recommend cleaning packages before any attempt to deploy any new version of any eftsurre package. To clean packages, Kill the following processes:

- D365 batch (batch)
- IIS Service (w3wp)
- IISExpress (iisexpress)
- xppcAgent (xppcagent)

Then delete the following eftsurre packages, if they exist:

Eftsurre, ExFlowAndeftsureInsideD365, XeAndeftsureInsideD365

Make sure the folder no longer exists on disk,

Disable Gated Checkins if necessary

Checkin deleted folder, make sure folder has been deleted from DevOps.

Enable Gated Checkins if necessary

DEPLOYABLE PACKAGE

If you have a deployable package, install deployable packages from the command line or from LCS:

<https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/deployment/install-deployable-package>

<https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/deployment/apply-deployable-package-system>

SOURCE CONTROL

After successful deployment, you can now **check-in** our new release into your source control for further references.

Go to **Visual Studio** and open the **Source Control Explorer** window,

For initial deployments, **add the contents** of the eftsure folder into your source control, please make sure **no files are excluded** while adding our release folder to source control!

Add to Source Control

? X

Use the Include/Exclude button at the bottom right to include and exclude items. All items in the "items to add" tab will be added to source control.

Items to add (1) Excluded items (97)

Name	Destination folder	Type	Size
Dynamics.AX.SomelSV.0.netm...	\$/Fleet Management/Trunk/Main/Metadat...		6.50
Dynamics.AX.SomelSV.0.pdb	\$/Fleet Management/Trunk/Main/Metadat...		13.5
Dynamics.AX.SomelSV.ChainO...	\$/Fleet Management/Trunk/Main/Metadat...		18 b
Dynamics.AX.SomelSV.ChainO...	\$/Fleet Management/Trunk/Main/Metadat...		3.00
Dynamics.AX.SomelSV.ChainO...	\$/Fleet Management/Trunk/Main/Metadat...		9.50
Dynamics.AX.SomelSV.ClassExt...	\$/Fleet Management/Trunk/Main/Metadat...		18 b
Dynamics.AX.SomelSV.DeleteA...	\$/Fleet Management/Trunk/Main/Metadat...		125
Dynamics.AX.SomelSV.dll	\$/Fleet Management/Trunk/Main/Metadat...		5.00
Dynamics.AX.SomelSV.EdtExt...	\$/Fleet Management/Trunk/Main/Metadat...		3 by
Dynamics.AX.SomelSV.EventH...	\$/Fleet Management/Trunk/Main/Metadat...		3.00
Dynamics.AX.SomelSV.EventH...	\$/Fleet Management/Trunk/Main/Metadat...		9.50
Dynamics.AX.SomelSV.Extensio...	\$/Fleet Management/Trunk/Main/Metadat...		20 b
Dynamics.AX.SomelSV.ODataE...	\$/Fleet Management/Trunk/Main/Metadat...		3 b

97 items selected

Include item(s)

Destination source control folder:
\$/Fleet Management/Trunk/Main/Metadata

< Previous Finish Cancel

Include all items in source control, until there are no more excluded items:

Add to Source Control

Use the Include/Exclude button at the bottom right will be added to source control.

Items to add (98) Excluded items (0)

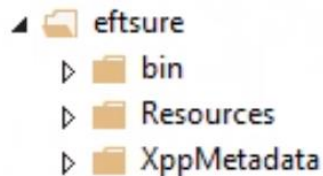
Name	Destination
------	-------------

Important: When you check-in the upgraded version to TFS, there should be exactly the same files in repository as if you would install and check-in the new version on a clean environment. TFS does not automatically detect and include all upgrade changes for check-in. **It requires some manual steps that can be error prone.**

Go to **Team Explorer – Pending changes** and check-in your code, following your Team rules, making sure to only check-in code from this release.

Note: The files resulting from compilation might appear in your **Pending Changes**, and are not required to be version control, just make sure all files from the original zip file are checked-in.

Your source control folder should look like:



UPGRADES

During upgrade installation, the model folders and descriptor files will be automatically updated. New files will be added and old ones removed, except **DLL files** in the **bin** folder of the **any** package, where the old files may remain.

When you check-in the upgraded version to TFS, there should be exactly the same files in repository as if you would install and check-in the new version on a clean environment. TFS does not automatically detect and include all upgrade changes for check-in. It requires some manual steps that can be error prone.

If you would like to check-in all the changes of the upgraded models, make sure all these changes are listed in **Team Explorer\Included Changes**, before checking-in them to **TFS**. Updates are included automatically, while deleted files have to be manually promoted from **Team Explorer\Excluded Changes > Detected**. New files have to be added manually as well (**Add Items to Folder**). Obsolete **DLLs** in **bin** folder have to be manually deleted.

Therefore, we recommend the following, for a more reliable UPGRADE PROCEDURE:

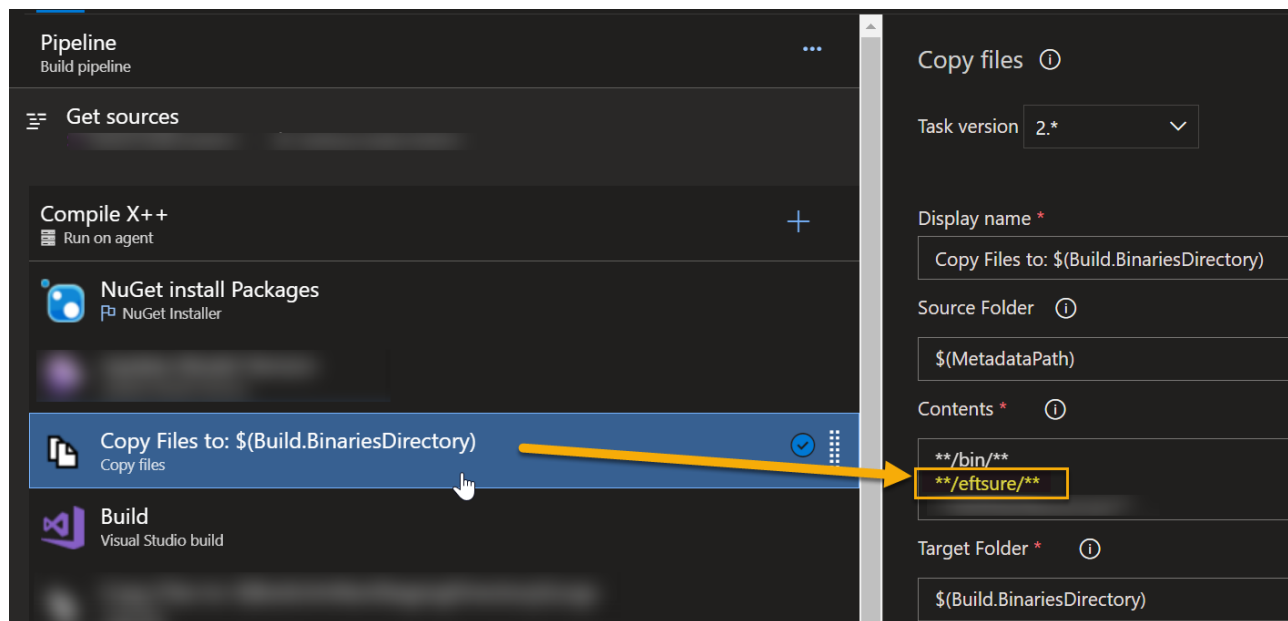
1. Stop **IIS** (or **IIS Express**) and the **Batch** service.
2. Delete all **eftsurre** models (i.e. packages) from **TFS** and check-in the changes.
3. Delete all the **eftsurre** package folders from the hard drive:.
4. Redeploy eftsurre packages from LCS or command line
5. Add all the artifacts from **eftsurre** models to **TFS** . Make sure that the correct DLLs from the **\bin** folder are added to **TFS**.

6. Check-in the changes and verify on the **Azure DevOps** portal that you really have the right code in place in the repository.
7. Build the **new eftsure** models.
8. Synchronize the database.
9. Start **IIS** (or **IIS Express**) and the **Batch** service.

ALL IN ONE PACKAGE

Your package can be created using DevOps pipelines or Visual Studio.

For Cloud hosted build pipelines, we recommend the following step to be added into the pipeline:



UPLOAD PACKAGE TO LCS

This deployable package should now be **uploaded** to LCS for testing in non-developer environments.

Head to <http://lcs.dynamics.com/>, select your **project(s)**, go to the **Asset library**, Click the **Add** button, enter a Deployable Package **name and description**, **upload** your newly created Deployable Package, and **Confirm**.

Asset library

Select asset type

Software deployable package files

Upload Software deployable package file

Name

Description

efsure - version 2.0.32.47

Add a file

Confirm Cancel

Asset library

Select asset type

Software deployable package files

Additional details

ExFlowAndEftsureInsideD365Finance v.2.14.48.829

Show component details

Description

ExFlow and eftsure inside Dynamics 365 F&O version 2.14.48.829 - Build 2022.10.6.1 on Platform Update 14.48

Asset ID

a5ff9076-4eac-4808-b81e-8354e6c21cdd

Compatibility with 8.1 and later releases

Yes

Compatibility with sealed releases

Yes

Package ID

d1966224-5d59-43c1-8917-0ecda5e9764

Package type

Application deployable package

Platform build

7.0.6253.76

Platform version

Update48

Publisher

Non-Microsoft

Validation status

Valid

Created by

Della

Created Date

10/6/2022 9:41 PM

Modified by

Your new deployable package is now ready to be deployed into your sandbox environments.

DEPLOYMENTS TO CLOUD HOSTED ENVIRONMENTS (DEV,TEST, LIVE)

Before you begin, verify that the deployable package has been uploaded to the Asset library in LCS.

Open the **Environment details view** for the environment where you want to apply the update.

Click **Maintain > Apply updates** to apply an update.

Select the **package to apply**. Use the filter at the top to find your package.

The screenshot shows the Lifecycle Services interface. On the left, the 'Manage environment' view is visible, showing details for an environment named 'AOS' in the 'Australia Southeast' region. The 'ENVIRONMENT DETAILS' section includes fields for Environment Id, Deployed on, Deployed by, Environment administrator, Latest activity ID, and Primary region (Active). The 'ENVIRONMENT VERSION INFORMATION' section shows Application release (Microsoft Dynamics 365), Platform release (Update32 (7.0.5)), and a link to 'View detailed version information'. The 'DEPLOYMENT STATUS' section shows a table with VM Name, Description, Status, and Region. The 'DATABASE ACCOUNTS' section shows a table with SQL Server, Database Name, User name, and Password. On the right, a modal titled 'Select the software deployable package to apply' is open. It contains a filter input field and a table of available packages. The table has columns: Name, Description, Package type, Asset version, Platform version, Modified date, Modified by, and Publisher. The first row shows 'eftsure - version 2.0...' with Package type 'Application depl...', Asset version '1', Platform version 'Update30', Modified date '28/02/2020 1:59 AM', and Publisher 'Non-Microsoft'. At the bottom of the modal are 'Apply' and 'Cancel' buttons.

Name	Description	Package type	Asset version	Platform version	Modified date	Modified by	Publisher
eftsure - version 2.0...		Application depl...	1	Update30	28/02/2020 1:59 AM		Non-Microsoft

Click **Apply**. Notice that the status in the upper-right corner of the **Environment details view** changes from Queued to **In Progress**, and an Environment updates section now shows the progress of the package. You can refresh the page to check the status.

Continue to **refresh** the page to see the status updates for the package application request. When the package has been applied, the environment status changes to **Deployed**, and the servicing status changes to **Completed**.

APPLY A PACKAGE TO A PRODUCTION ENVIRONMENT BY USING LCS

In a production environment, customers can **schedule a downtime** for when they want the update to be applied.

Important: A prerequisite for applying a package to a production environment is that the package must be successfully applied to at least one sandbox environment in the same project.

After the update is successfully applied in a sandbox environment, go to the project's asset library. On the Asset library page, select the **Software deployable package** tab, select the **package that you want to move to production**, and click **Release candidate**. This indicates that this package is **ready for production deployment**.

Open the **Environment details view** for the **production** environment where you want to apply the package.

Select **Maintain > Apply updates** to apply the package.

Select the **package to apply in your production environment**, and then click **Schedule** to submit a request to apply it.

Note: The list of packages includes only the packages that have been successfully signed off in the sandbox environment, and that have been marked as release candidates.

Specify the date and time to **schedule the package application**. Click **Submit**, and then click **OK** to confirm.

Note that your environments will be unavailable to perform business while the package is being applied.

At the scheduled downtime, package deployment will start.

After the environment is serviced, you can monitor the status. The Servicing status field indicates the status of package application. Additionally, a progress indicator shows the number of steps that have been run, out of the total number of steps that are available.

After the deployment is successfully completed, the Servicing status field is set to **Completed**.

If package application isn't successfully completed, Microsoft will investigate the issue. The Servicing status field will indicate that package application has failed. The environment will be **rolled back to a good state**.

DEPLOYMENT TO DEVELOPER ENVIRONMENTS

For all **developer machines** that require to run our extension,

Clean the packages folder for all eftsure extensions (see Clean Packages)

Go to **Visual Studio**, Connect to your **Team Project**, locate the appropriate **branch** where the eftsure extension has been added, perform a **Get Latest** on the eftsure folder and/or entire branch.

This process will **download** the copy of our extension **onto your local machine**,

Refresh Models in Visual Studio > Extensions > Dynamics 365 > Model Management > Refresh Models.

Then, **compile** the **new or updated models** and dependencies if any custom extension has been created on top of our extension.

Then **synchronize** your database.

Important: Resource files are not automatically deployed as part of the Get Latest process, see [this article](#) for more information. In order to force the deployment of resources, we recommend each developer to run [this powershell script](#) to redeploy resources manually after each GetLatest.

Then your developer machine is fully updated, ready to be configured and start testing / debugging / troubleshooting your environment.

POST DEPLOYMENT ACTIVITIES

After the deployment of any new version of eftsure inside Dynamics 365, you will also need to perform the following actions:

Data entity refresh

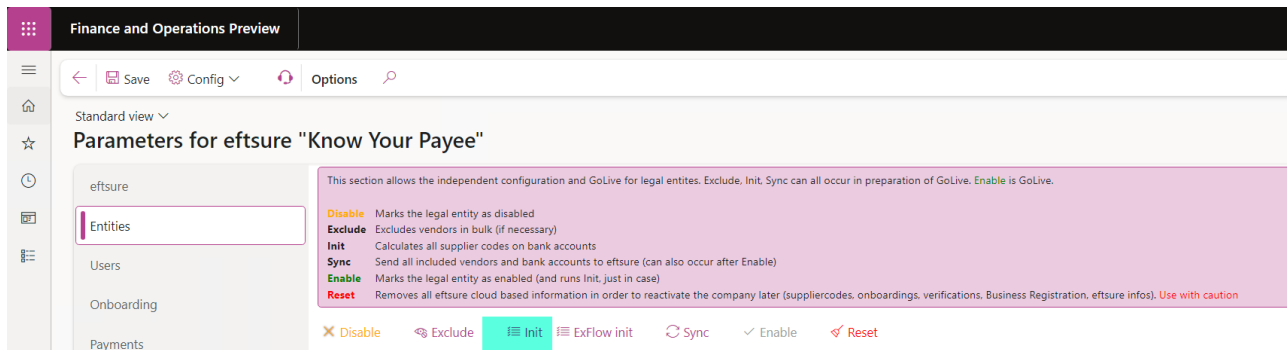
New or updated entities are likely to be published in between new versions. In order to make sure that new entities are deployed, please run the standard entity refresh script available in **Data Management > Framework Parameters > Entity Settings > Refresh Entity List**.

Cross company data sharing

If the configuration is using cross company data sharing, check for new PES fields on VendTable, VendBankAccount. Deprecated fields to be deleted, otherwise will result in SQL error – See CCDS White Paper.

Re-Init legal entity

Go to eftsure parameters and run init on all enabled legal entities:



POST DB REFRESH ACTIVITIES

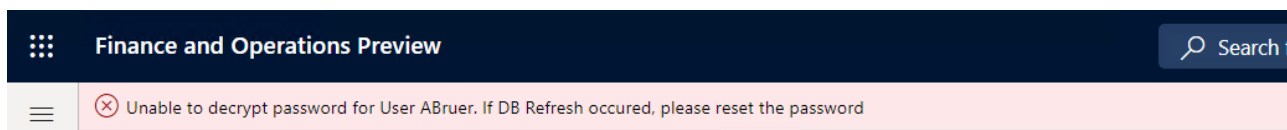
We recommend adding the following script to your SQL DB Refresh scripts, in order to avoid Password encryption errors for passwords coming from another environment.

These SQL Scripts will clear all eftsure credentials that are no longer valid on the new target environment.

```
UPDATE PESUSERINFO SET PASSWORDENCRYPTED = NULL
```

```
UPDATE PESPARAMETERS SET PASSWORDENCRYPTED = NULL
```

If such scripts aren't performed as part of DB Refresh, the following errors are likely to occur throughout the system:



Data elements that aren't exported

When you export a database backup from an environment, some elements of the database aren't exported in the backup file. Here are some examples:

- Email addresses in the LogisticsElectronicAddress table.
- Batch job history in the BatchJobHistory, BatchHistory, and BatchConstraintHistory tables.
- SMTP Relay server in the SysEmailParameters table.
- Print Management settings in the PrintMgmtSettings and PrintMgmtDocInstance tables.
- Environment-specific records in the SysServerConfig, SysServerSessions, SysCorpNetPrinters, SysClientSessions, BatchServerConfig, and BatchServerGroup tables.
- Document attachments in the DocuValue table. These attachments include any Microsoft Office templates that were overwritten in the source environment.
- All users except the admin will be set to **Disabled** status.
- All batch jobs are set to **Withhold** status.
- All users will have their partition value reset to the "initial" partition record ID.
- All Microsoft-encrypted fields will be cleared, because they can't be decrypted on a different database server. An example is the **Password** field in the SysEmailSMTPPassword table.

See <https://learn.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/database/export-database>

UNINSTALL

If you no longer require our ISV product or haven't renewed the license, you must remove the package.

Follow the steps in:

<https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/deployment/uninstall-deployable-package>

Customers using Cross Company Data Sharing will need to start by removing their CCDS configuration (see Cross Company Data Sharing White Paper)

TROUBLESHOOTING

Using a second dev box:

In case of deployment issues, the developer can try to use a second dev environment to validate that no files are missing from source control.

On the second dev environment:

- Clean package
- Perform get latest
- Redeploy package
- Go to Source Control Explorer
- Select "Add items to folder" on the EFTSure folder
- Make sure that there are no files listed (in either include or exclude)
- Check-in / Redeploy

Other ideas that might have helped on troublesome deployments:

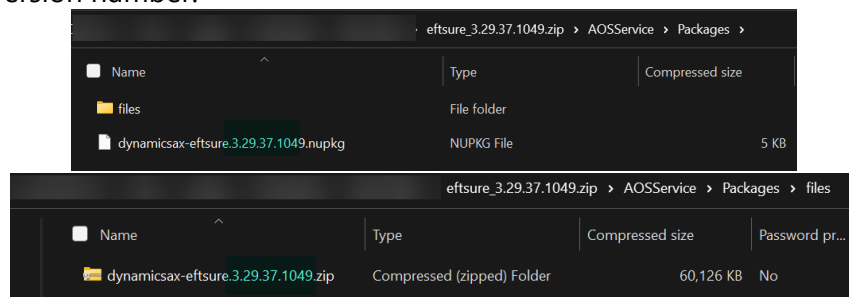
Deploying ISV extensions on an empty DB, then restoring DB over.

Turning off all flighting options (Feature Management)

Reprovisioning environments

Deployable packages:

The deployable packages shipped at eftsire are zip files containing a single model, all clearly labelled with a version number.



This software package is usually bundled by the customer or their partner into an 'All in One' package to be deployed with additional code to Tier2+ environments.

This 'all in one' package should eftsire files with the same version number that you are expecting and the eftsire zip file should contain the same number of files as the SDP shipped by eftsire.

If that's not the case or the eftsire package is versioned as 1.0.0.0, an issue is likely present and files might be missing, this software deployable package will be unlikely to deploy successfully. Please revisit your deployment and release pipeline, checking for version number and file numbers after each and every step. To this day, the likely causes for such errors are the following:



Issues with services not stopping
Files being locked by the system during deployment
Missed files during checkin/merge

Eftsure does not ship any package with v1.0.0.0, clearly indicating an error. Please involve support if you are struggling.