



Event Handling Framework

Release 27

Last Modified: 05/11/2026

Copyright Information

Copyright © 2026 Aras Corporation. All Rights Reserved.

Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Phone: 978-691-8900

Notice of Rights

Copyright © 2026 by Aras Corporation and/or its affiliates. All rights reserved.

This document is protected by U.S. and international copyright laws and conventions. No copyright may be obscured or removed from this document. This document may not be modified or altered, or reproduced or transmitted in any form, without the explicit permission of the copyright holder.

Aras Innovator, Aras, and the Aras Corp "A" logo are registered trademarks of Aras Corporation in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

Notice of Liability

This document is provided for informational purposes only, and the contents hereof are subject to change without notice. The information contained in this document is distributed on an "as is" basis, without warranty of any kind, express or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose or a warranty of non-infringement. Aras shall have no liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this document or by the software or hardware products described herein.

Table of contents

Release Notes	4
Platform Support Matrix	5
Key Features and Known Issues	6
Known Issues in Aras Event Handling Framework 27	8
Administration Guide	9
Introduction	10
Scope	11
Target Audience	12
Introduction to Event Handling Framework (EHF)	13
Event Handling Framework Identities	14
Subscriptions	15
Create Subscription	16
Event Types	18
Custom Event Type	23
Event Handlers	32
Create Event Handlers	33
Create Custom Event Handler Method	35
Add an Existing Event Types and Event Handlers to Subscription	38
Installation Guide	40
Introduction	41
Scope	42
Target Audience	43
Prerequisites	44
Checking for Eligibility	45
Installing Event Handling Framework	46
Notification and Backup	47



Installing the Event Handling Framework Platform Component	50
Automated Installation Option	51
Manual Installation Option	56
Confirming the Installation	59
Post Installation	60



Release Notes



Platform Support Matrix

For the software required for the Aras Event Handling Framework 27:

Aras Innovator Release Client Operating System Browser

27	Windows: 8.1, 10, 11	<ul style="list-style-type: none">• Chrome• Edge• Firefox: 91 ESR and 102 ESR
----	----------------------	---



Key Features and Known Issues

Key Features in Aras Event Handling Framework 27

The following key features are available in this initial release of the Aras Event Handling Framework application:

Publish Events from Aras Innovator Feature

This feature provides high-level functionality of the Aras Event Handling Framework. It enables administrators to configure subscriptions that describe types of events and how those events should be handled. The platform component's publishing queue uses these subscriptions to asynchronously publish event data to external systems and services.

Publishing Queue Feature

This feature enables administrators and developers to add event data to the EHF Publishing Queue via AML and custom method code. The method is used for creating Custom Event Types.

Publish Items OnAfterAdd Event Data Feature

This feature provides the OnAfterAdd Item Event Type, which enables administrators to publish an event after an instance of an item is added. This Item Event Type corresponds to the OnAfterAdd ItemType Server Event.

Publish Items OnAfterUpdate Event Data Feature

This feature provides the OnAfterUpdate Item Event Type, which enables administrators to publish an event after an instance of an item is modified. This Item Event Type corresponds to the OnAfterUpdate ItemType Server Event.

Publish Items OnAfterCopy Event Data Feature

This feature provides the OnAfterCopy Item Event Type, which enables administrators to publish an event after the "Save As" action is executed on an instance of an item. This Item Event Type corresponds to the OnAfterCopy ItemType Server Event.



Publish Items OnAfterDelete Event Data Feature

This feature provides the OnAfterDelete Item Event Type, which enables administrators to publish an event after an instance of an item is deleted. This Item Event Type corresponds to the OnAfterDelete ItemType Server Event.

Publish Items OnAfterVersion Event Data Feature

This feature provides the OnAfterVersion Item Event Type, which enables administrators to publish an event after an instance of an item is versioned. This Item Event Type corresponds to the OnAfterVersion ItemType Server

Event.

Publish Items OnAfterPromote Event Data Feature

This feature provides the OnAfterPromote Item Event Type, which enables administrators to publish an event after an instance of an item is promoted to a new state. This Item Event Type corresponds to the OnAfterPromote ItemType Server Event.

Create Event Publishing Method Feature

This feature provides the ability for an administrator to create an Event Handler, which uses custom method code to publish event data from the queue.

Publish Custom Events Feature

This feature enables administrators to define custom event types that can be used by EHF subscriptions. Publishing custom events requires custom method code to add the event data to the queue.



Known Issues in Aras Event Handling Framework 27

There are no Known Issues in the Aras Event Handling Framework 27 release.



Administration Guide



Introduction

[Purpose](#)

The Administration Guide describes the features of the Aras Event Handling Framework (EHF) 27 platform component.



Scope

This document provides instructions to create and manage the **Subscriptions, Event Types, and Event Handlers** of EHF.



Target Audience

This document is intended for Aras Event Handling Framework Administrators responsible for creating and managing all features within EHF.



Introduction to Event Handling Framework (EHF)

[Event Handling Framework Overview](#)

EHF provides the capability to publish Aras Innovator event data to external systems and services.

The following is an overview of the three main EHF features:

Feature	Description
Event Types	Listeners for common ItemType events are included and can be configured with custom event types to support various business cases.
Event Handlers	Publishing methods that send event data to external systems and services.
Subscriptions	Configures Event Type(s) that are published by the Event Handler(s) .



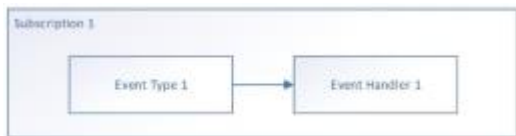
Event Handling Framework Identities

1. **ehf_EventAdministrators** - manages Event Types, Event Handlers and Subscriptions.
2. **ehf_EventPublishers** - controls execution of Event publishing methods.
3. **ehf_EventHandlers** - controls execution of Event Handlers methods.

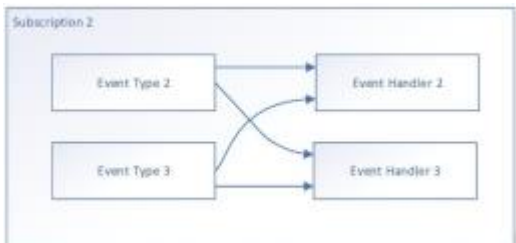


Subscriptions

The Subscription defines which **Event Handler** will process all events produced by appropriate **Event Type**. It is possible to have multiple **Event Handlers** and/or multiple **Event Types** on a **Subscription**.



In the above diagram, **Subscription 1** contains **Event Type 1** and **Event Handler 1**. All events produced by **Event Type 1** will be processed by **Event Handler 1**.

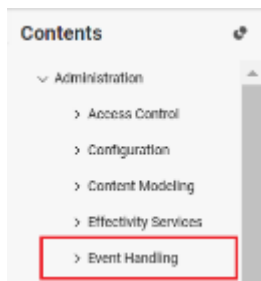


In the above diagram, **Subscription 2** contains 2 **Event Types** and 2 **Event Handlers**. Each event from **Event Type 2** or **Event Type 3** will be processed by both **Event Handler 2** and **Event Handler 3**. The same **Event Type** and **Event Handler** can be a part of multiple **Subscriptions**.

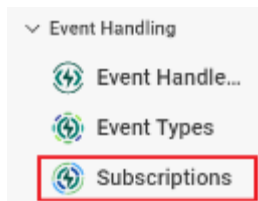
Create Subscription

The following steps outline the process of creating the **Subscription** to link an **Event Type** to an **Event Handler**:

1. From the **Table of Contents**, expand **Administration** and expand **Event Handling**.

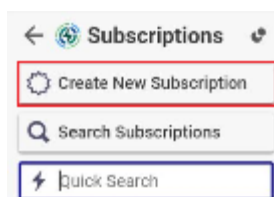


2. Select **Subscriptions**.




The **Subscription Table of Contents** appears.

3. Click **Create New Subscription**.



The Subscription form appears.

 **Subscription 3**

Name

Description

4. Enter the **Name** and **Description** (Optional) of the **Subscription**.
5. Click **Done** to save.
6. Next step is to create new **Event Types** and link them to the **Event Handlers**.

Event Types

Create Item Event Types

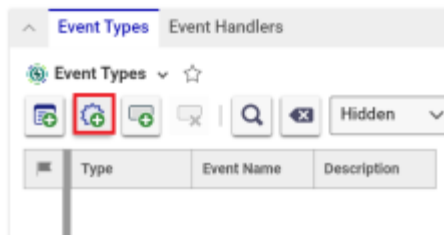
Item Event Type allows to subscribe to specific system events like “OnAfterAdd”, “OnAfterUpdate”, “OnAfterDelete”, “OnAfterCopy”, “OnAfterPromote” and “OnAfterVersion”. Every time an appropriate system event occurs, an EHF Item Event is generated and added to the publishing events queue.

Important

Since the publishing queue is in the server memory, the publishing queue will be lost if the Aras Innovator server is restarted.

The following steps outline the process of creating an **Item Event Type**:

1. On the **Subscription** form, click **Edit**.
2. From the **Subscription** form, scroll down to the **Event Types** tab and click **New Event Type**.



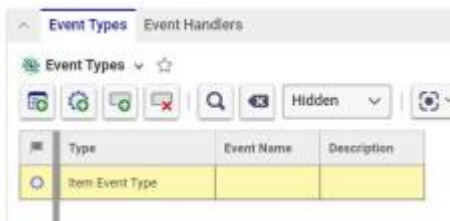
The **Select Item Type** dialog box appears.



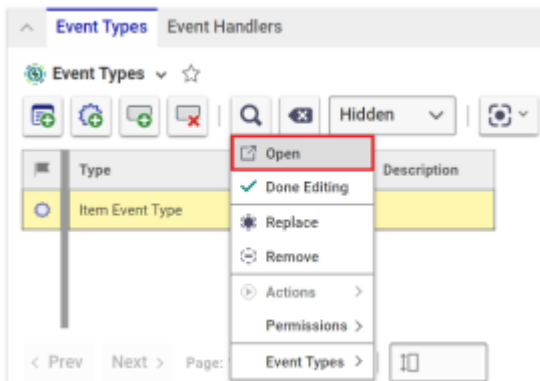
3. Select **Item Event Type** and click **OK**.



A new row appears on **EventTypes** tab.



4. Right-click on the newly added row and click **Open**.

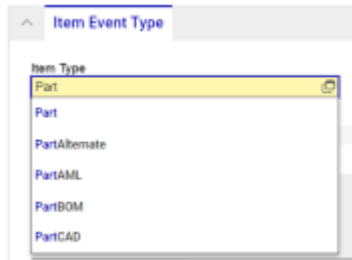


The **Item Event Type** form appears.

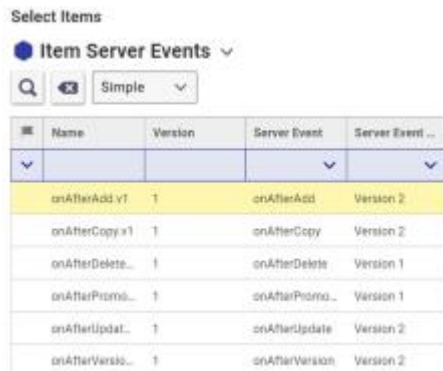


5. In the **ItemType** field, enter the **ItemType**.





6. From the **Item Server Event**, select the appropriate system event that needs to be subscribed.



As a part of **Event Handling Framework**, Aras provides the following **Item Server Events**:

- **onAfterAdd.v1**: The event occurs after an Item is added to the database.
- **onAfterCopy.v1**: The event occurs after an Item is copied.
- **onAfterDelete.v1**: The event occurs after an Item is deleted.
- **onAfterPromote.v1**: The event occurs after an Item is promoted to the next state.
- **onAfterUpdate.v1**: The event occurs after an Item is updated in the database.
- **onAfterVersion.v1**: The event occurs after an Item is versioned.

Event Name is automatically generated by the server.

7. Add the event description in the **Description** field.

8. Click **Done** to save.

The **Handled By** tab displays all **Event Handlers** that process this **Item Event Type** grouped by **Subscription**. The **Event Handlers** will be visible only when the **Event Type** is linked to **Event Handler** using **Subscription**. Make sure to save the **Subscription** item, otherwise the **Subscription** will not be visible.



9. Click **Refresh** to view the **Subscription** and linked **Event Handler**.



Creating **Item Event Type** will inform Aras Innovator when a specific event occurs and add the event data to the publishing queue to publish.

The following example displays the **Item Event Type** data that will be added to the publishing queue:

```
<Item>
```

```
<data>
```

```
<Item type="Part" id="04850C519DFD45F0A994961CA0F843CD" />
```

```
</data>
```

```
<subject>04850C519DFD45F0A994961CA0F843CD</subject>
```

```
<type>com.aras.innovator.items.Part.onAfterAdd.v1</type>
```

```
<id>C4187AA25AFC4C5E97607EF082ED297F</id>
```

```
<time>2023-03-02T12:26:31.8746280Z</time>
```

```
<source>14_0_15_37629@http://localhost/InnovatorServer_14_0_15_37629</source>
```

```
<datacontenttype>application/aml+xml</datacontenttype>
```

```
</Item>
```

where:

- **data** – Contains information about the item that triggered the system event.
- **subject** – Event subject that contains id of the item that triggered the system event.

- **type** – The type property contains the name of the Item Event Type.
- **id** – Unique event id.
- **time** – UTC time when event was generated.
- **source** – Contains information about event source in the form of <database name>@<innovator_server_url>.
- **datacontenttype** – Content type of data value.



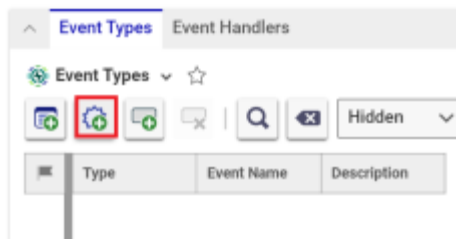
Custom Event Type

A **Custom Event Type** allows administrators to use method code to add events to the publishing queue. This capability enables publishing for events that fall outside the scope of the **Event Handling Framework's** standard **Item Event Types**.

Create Custom Event Type

The following steps outline the process of creating a **Custom Event Type**:

1. On the **Subscription** form, click **Edit**.
2. From the **Subscription** form, scroll down to the **Event Types** tab and click **New Event Type**.



The **Select Item Type** dialog box appears.



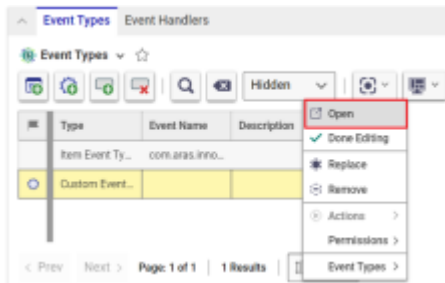
3. Select **Custom Event Type** and click **OK**.



A new row appears on **EventTypes** tab.



4. Right-click on the newly added row and click **Open**.



The **Custom Event Type** form appears.



5. Add the Event Name and Description.

The **Event Name** is required to map the event in the publishing queue to the type of event.

6. Click **Done** to save.

The **Handled By** tab shows all the **Event Handlers** that process this **Custom Event Type** grouped by **Subscriptions**. Make sure to save the **Subscription** otherwise the **Subscription** will not be visible.

The Administrators need to develop the method to generate the **Server Events** that triggers to add events in the publishing queue.

Important

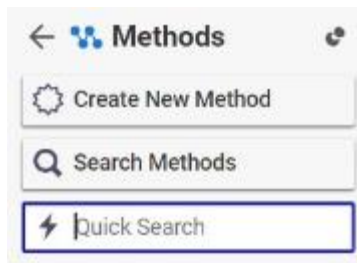


The Execution allowed to field should be set to **ehf_EventPublishers** when creating a **Server Events** method.

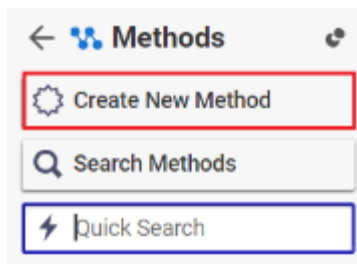
Create Custom Event Method

The following steps outline the process of creating the **Custom Event Methods**:

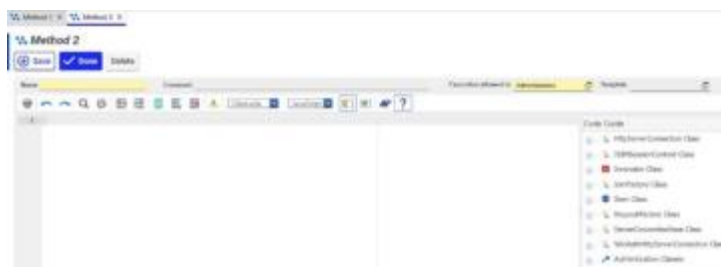
1. From the **Table of Contents**, expand **Administration** and select **Methods**. The **Methods Table of content** appears.



2. Click **Create New Method**.



The Method form appears.



3. Add the Method **Name** and the following properties:

- **Method Location:** Server-Side



- **Method Type:** C#
- **Execution Allowed To:** ehf_EventPublishers

4. Write the method code that will add a custom event to the publishing queue.

The following sample code demonstrates how to use the **ehf_PublishEvent** action to add a **Custom Event** to the queue. Replace the highlighted text with the data that will be processed by an **Event Handler Method** (see section “Create Event Handler Method”):

```
Item publishEvent = this.newItem("Method", "ehf_PublishEvent");  
  
Item eventItem = this.newItem();  
  
eventItem.setProperty("type", "Custom Event Type Name");  
  
eventItem.setProperty("subject", "context item id or subject string");  
  
eventItem.setProperty("data", "context item data");  
  
publishEvent.setPropertyItem("event", eventItem);  
  
Item result = publishEvent.apply();  
  
return this;
```

5. Click **Done** to save.

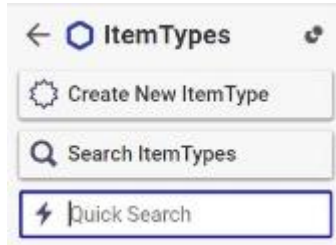
Configure Custom Event Method Execution

To publish a **Custom Event**, the **Custom Event Method** created in section “Create Custom Event Method” must be executed as an ItemType Server Event, an Action, or some other process.

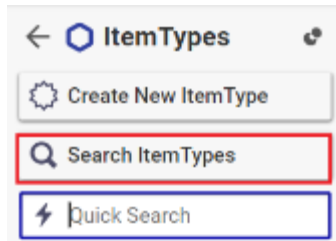
The following steps outline the process of adding the **Custom Event Method** as a **Server Event** on an **ItemType**:

1. From the **Table of Contents**, expand **Administration** and select **ItemTypes**. The **ItemTypes Table of Contents** appears.





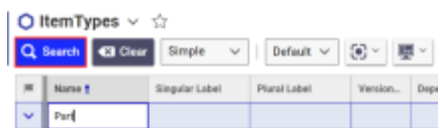
2. Click **Search ItemTypes**.



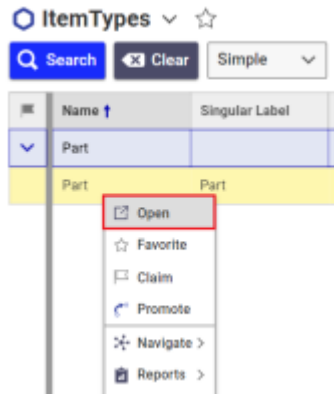
The **ItemTypes** grid appears.



3. In the Name field, enter the required ItemType whose data should be published and click **Search**.

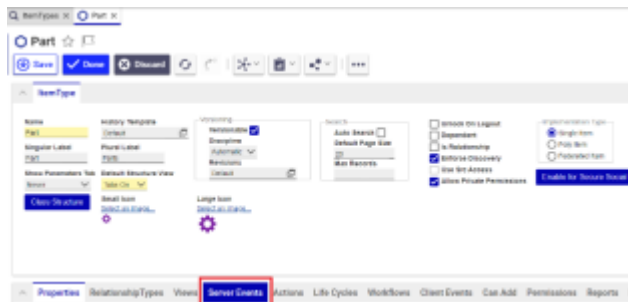


4. Right click the ItemType in the grid and click **Open**.

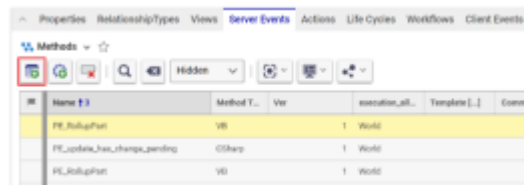


5. Click **Edit**.

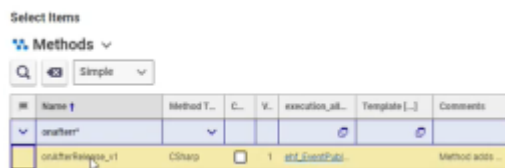
6. Click the **Server Events** tab.



7. Click **Add Methods**.

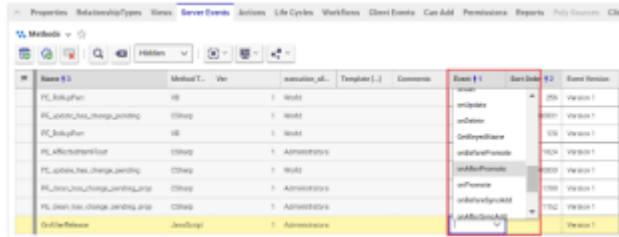


8. In the **Select Items** dialog box, search for the method created in the **Create Custom Events Method** section and click **OK**.



9. In the newly created **Server Event** row, choose the **Event** that should trigger the execution of the **Method**.





10. Click **Done** to save.

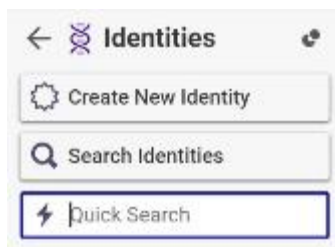
For more information on configuring Server Methods, please see the **Developing Solutions Training Guide** or the **Aras Innovator Programmer's Guide**.

Configure Event Publisher Identity

The **ehf_EventPublishers Identity** is a group Identity used to grant permissions for adding event to the publishing queue. Any users who may cause an event to be added to the publishing queue should be added to this Identity.

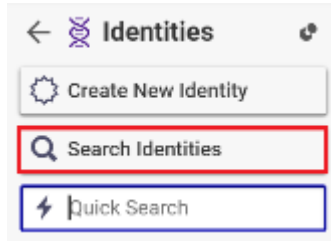
The following steps outline the process of adding one or more Users (or group Identities) to the **ehf_EventPublishers Identity**:

1. From the **Table of Contents**, expand **Administration** and select **Identities**. The Identities Table of Contents appear:

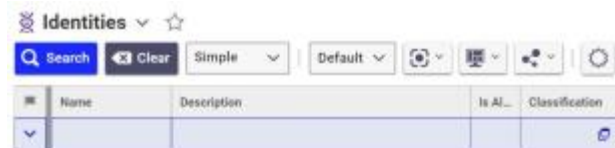


2. Click **Search Identities**.



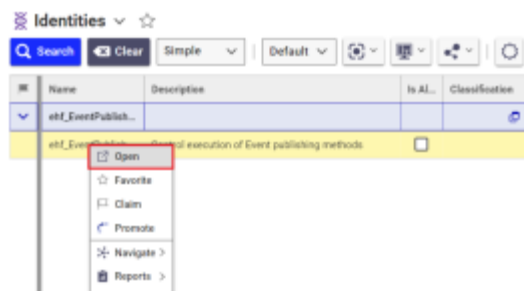


The Identities Search grid appears.



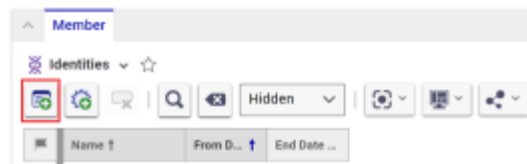
3. In the Name field, enter **ehf_EventPublishers** and click **Search**.

4. Right-click on the **ehf_EventPublishers** and click **Open**.



5. Click **Edit**.

6. In the Member tab, click **Add Identities**.



7. In the Select Items dialog box, select one or more Identities to include in the event publishing group and click **OK**.

Select items

Identities

Search [X] [Back] [Simple]

Name	Description	In Use	Classification
atp			
etf_eventAdmin...	Manage event types, event handlers and subscrip...	<input type="checkbox"/>	
etf_eventHandler...	Control execution of event handlers methods	<input type="checkbox"/>	
etf_eventPublish...	Control execution of event publishing methods	<input type="checkbox"/>	

8. Click **Done** to save.



Event Handlers

[Ad Hoc Event Handler](#)

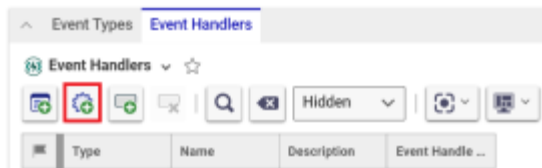
The **Event Handlers** allow the Administrator to configure how the events will be processed. The **Ad Hoc Event Handler** allows to specify arbitrary C-Sharp (C#) method with custom logic that will be executed each time when the new events are processed.



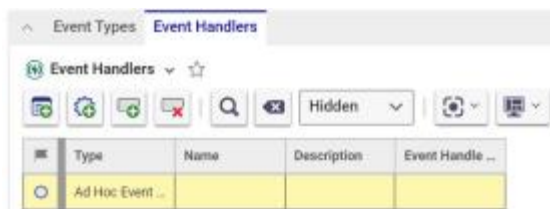
Create Event Handlers

The following steps outline the process of creating the **Event Handler**:

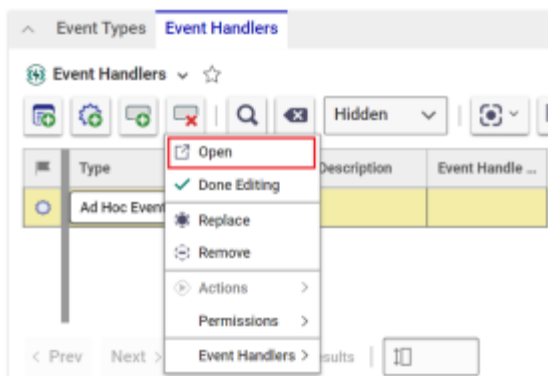
1. On the **Subscription** form, click **Edit**.
2. From the **Subscription** form, scroll down and switch to the **Event Handlers** tab and click **New Event Handler**.



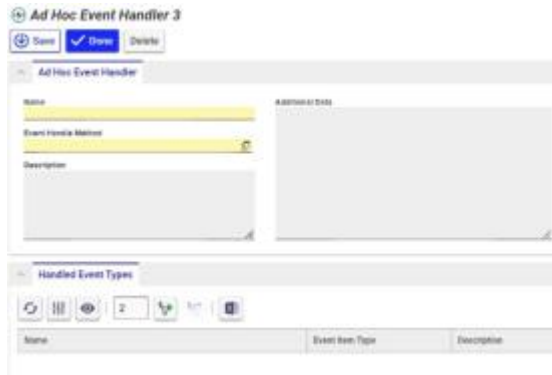
A new row appears on **Event Handlers** tab.



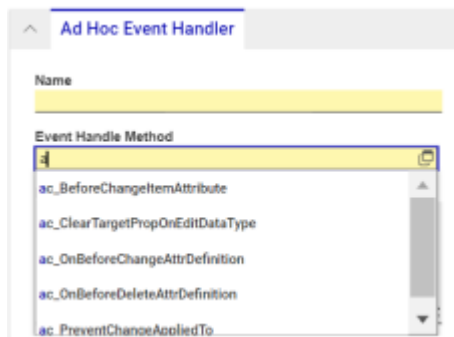
3. Right-click on the newly added row and click **Open**.



The **Ad Hoc Event Handler** form appears.



4. In the **Name** field, enter the name of the **Event Handler**.
5. In the **Event Handle Method** field, select the C-Sharp (C#) method with custom logic that will be processing events. The Administrators need to add their own custom methods. For more information on how to create a custom method, see **Create Custom Event Handler Method**.



6. Add the description of the **Event Handler** in the **Description** field.
7. This step is Optional. In the **Additional Data** field, Administrators can pass any parameters or any additional data that is required for the custom method.
8. Click **Done** to save.

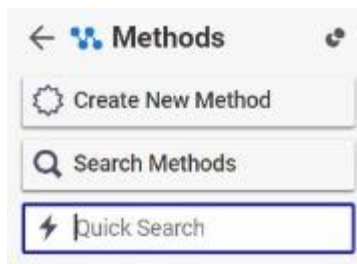
The **Handled Event Types** tab displays all **Event Types** that are processed by this **Ad Hoc Event Handler** grouped by **Subscriptions**. Make sure to save the **Subscription**, otherwise Subscription will not be visible.

Create Custom Event Handler Method

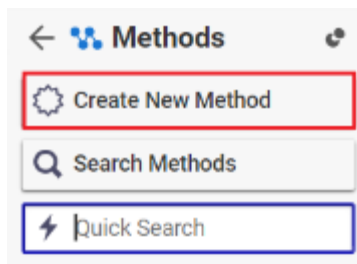
The Administrators need to create **Event Handler** Methods that will process all events produced by appropriate **Event Type**.

The following steps outline the process of creating custom event handler method:

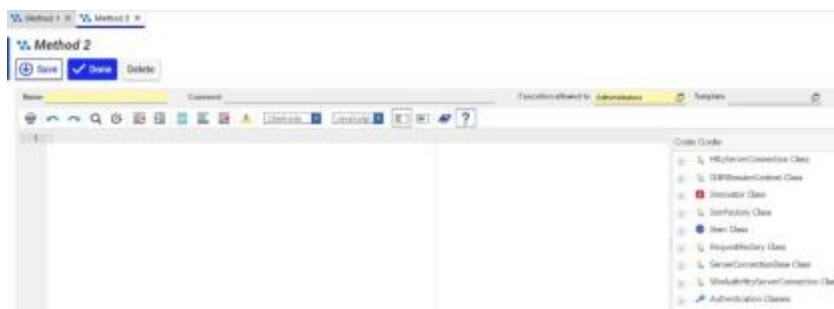
1. From the Table of Contents, expand Administration and select Methods. The Methods Table of content appears.



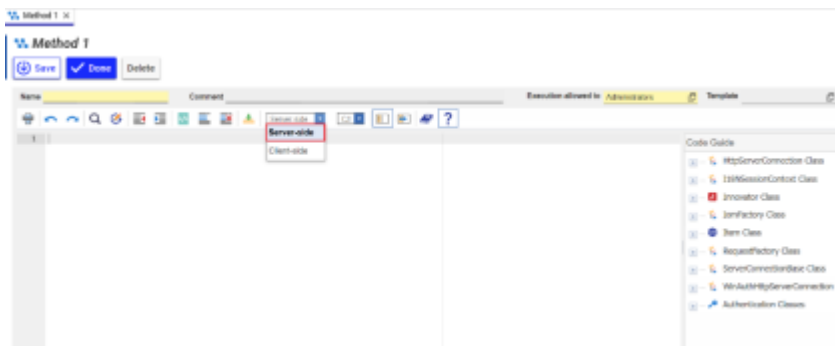
2. Click **Create New Method**.



The Method form appears.



3. In the **Name** field, enter the name of the method.
4. The **Execution allowed to** field should be set to **ehf_EventHandlers**.

5. Select **Server-Side** method.

6. Add the custom method code for publishing event data from the queue.

Here is a code sample that demonstrates how to retrieve the event data from the queue for publishing.

```
Item relationships = this.getRelationships();
int eventItemsCount = relationships.getItemCount();
/* for each event in this batch, get the event details and perform
custom publishing logic */
for (int i = 0; i < eventItemsCount; i++) {
Item eventItem = relationships.getItemByIndex(i);
var data = eventItem.node.SelectNodes("./data");
string dataXml = string.Empty;
if (data.Count > 0) {
dataXml = data[0].InnerXml;
}
string subject = eventItem.getProperty("subject", string.Empty);
```



```
string type = eventItem.getProperty("type", string.Empty);  
string eventId = eventItem.getProperty("id", string.Empty);  
string time = eventItem.getProperty("time", string.Empty);  
string source = eventItem.getProperty("source", string.Empty);  
  
// add your publishing logic here  
  
}  
  
return this;
```

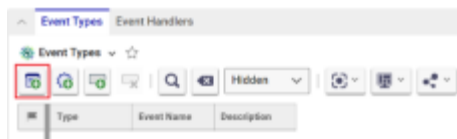
7. Click **Done** to save the changes.



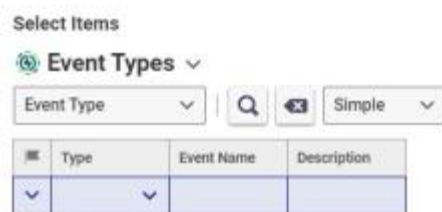
Add an Existing Event Types and Event Handlers to Subscription

The following steps outline the process of adding an existing **Event Types** and **Event Handlers**:

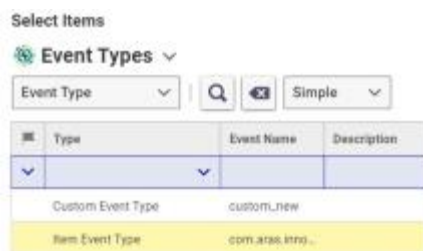
1. On the Subscription form, click **Edit**.
2. From the Subscription form, scroll down to the **Event Types** tab and click the **Add Event Types**.



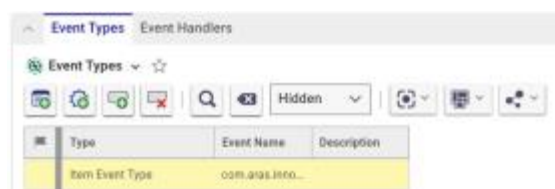
The Search Item dialog box appears.



3. Enter the search criteria and select the **Event Types** from the search results.

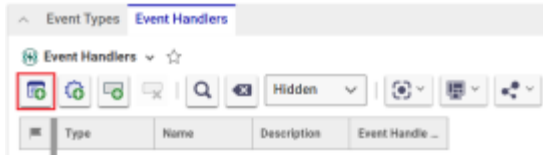


4. Click **OK**. The selected **Event Type** appear on the **Event Types** tab on **Subscription** form.

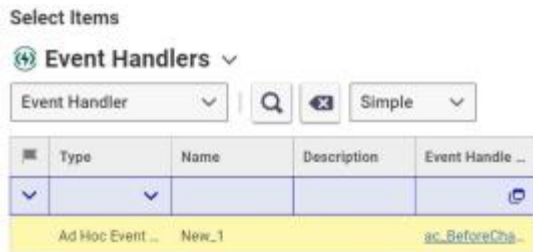


5. Click **Done** to save.

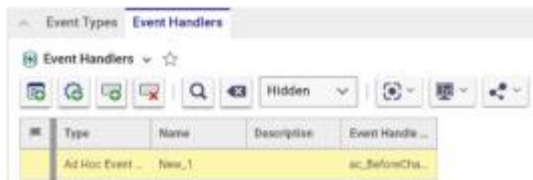
6. To add **Event Handlers**, switch to the **Event Handlers** tab and click **Add Event Handlers**.



7. Enter search criteria and select the **Event Handler** from the search results.



8. Click **OK**. The selected **Event Handlers** appear on the **Event Handlers** tab.



9. Click **Done** to save.



Installation Guide



Introduction

[Purpose](#)

The Installation Guide describes how to install Aras Event Handling Framework 27.



Scope

The Installation Guide provides detailed information for installing Aras Event Handling Framework for Aras Innovator. The installation involves updates to both the code tree and database.



Target Audience

This Installation Guide is intended for administrators responsible for installing and maintaining Aras Innovator and Aras Event Handling Framework.



Prerequisites

To install Event Handling Framework 27, users need to confirm that the installation of Aras Innovator meets the minimum requirements to run the Aras Event Handling Framework.



Checking for Eligibility

The following steps outline the process to verify the code tree and database are eligible for installing the Aras Event Handling Framework 27:

1. Log into **Aras Innovator** as an administrator.
2. Click the **User** icon in the top-right corner and select **About**.
3. Verify Release 27 appears.
4. From the **TOC**, select **Administration, Variables**.

VersionMajor	VersionMinor	VersionServiceUpdate	VersionLabel
14	0	15	Release 27

If it does not meet the service pack requirements listed in the previous steps, contact Aras to discuss options.



Installing Event Handling Framework

The Aras Event Handling Framework must modify the Aras Innovator database with the metadata required to run the component. Before applying these changes, verify the following requirements:

Requirements:

- Aras Innovator Release 27
- Aras Update Tool 1.20+
- or
- Aras Innovator Package Import Export Utilities



Notification and Backup

1. Notify users that the system will be down at a scheduled time, and they should log out of the system prior to the start of the process.

Important

It is best to give at least 24-hour notice, as well as a reminder 15 minutes prior to the upgrade.

2. Backup the code tree.
3. The “Code Tree” refers to files and folders installed to the disk when Aras Innovator was first installed.

The default path for the Code Tree installation would be something like:

`C:\Program Files (x86)\Aras\Innovator`

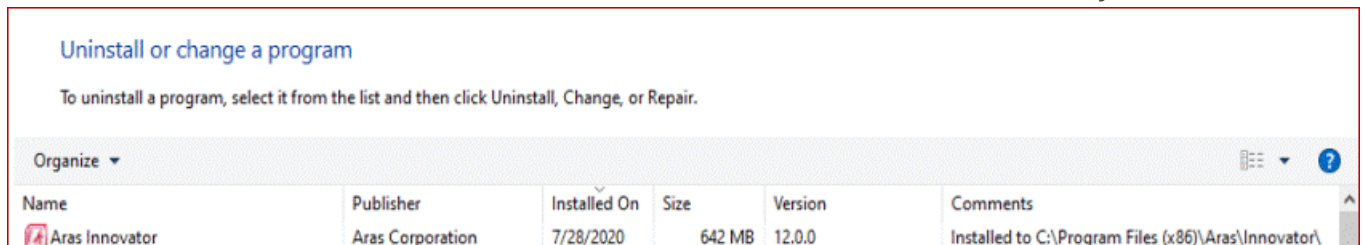
With the following contents:



Name	Type	Size
AgentService	File folder	
ConversionServer	File folder	
DB	File folder	
Innovator	File folder	
OAuthServer	File folder	
SelfServiceReporting	File folder	
Tools	File folder	
VaultServer	File folder	
ConversionServerConfig	XML File	1 KB
InnovatorServerConfig	XML File	2 KB
SelfServiceReportConfig	XML File	1 KB
VaultServerConfig	XML File	1 KB

The following steps can be used to confirm the installation folder:

1. From the Windows search field, enter **Control** to access the **Control Panel**
2. Select **Programs and Features**
3. Right-click the **Name** column and select **More...**
 - Select the **Comments** header if it is not checked
 - Click **OK**
4. Search for the **Aras Innovator** entry in the **Programs and Features** window.
5. View the value set in the **Comments** column for the Aras Innovator entry:



After the tree code installation path has been verified, back up the folder and all its contents.

4. Disconnect all users from the database.

The easiest way to prevent client sessions from committing any further changes to the database is to change the database connection string in the `InnovatorServerConfig.xml` from `<DB-Connection ...` to `<xDB-Connection` and restart the w3svc service (IIS). This expires all session and prevents all new connections to the Aras Innovator database through the existing instance.

5. Backup the database.

6. Place these files in a safe location, as they will be needed to restore if the process fails.

7. Enable database connections.

After the database backup has been completed, enable the database connection string in the `InnovatorServerConfig.xml` by changing it from `<xDB-Connection ...` to `<DB-Connection` and restarting the w3svc service (IIS).



Installing the Event Handling Framework Platform Component

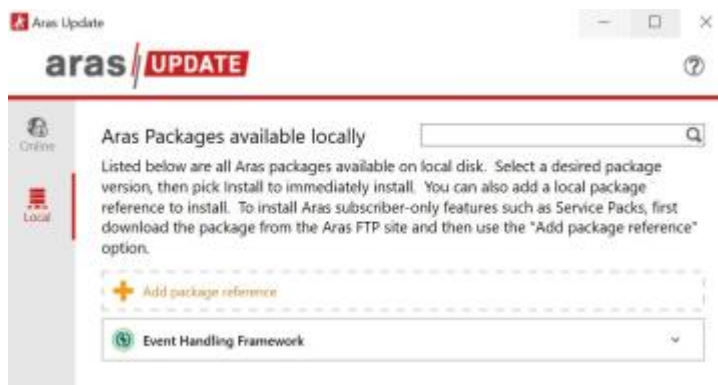
The Event Handling Framework component can be installed using either the Automated Installation option (using the Aras Update tool), or the Manual Installation option (using the Aras Import Utility).



Automated Installation Option

The following steps outline the process of Automated Installation of the Aras Event Handling Framework :

1. Copy and unzip the Aras Event Handling Framework CD Image on the local computer.
2. Enable the **Super User** login.
3. Launch the **ArasUpdate.exe** file as administrator.
4. On a default installation, the ArasUpdate.exe file can be found in the C:\Program Files (x86)\Aras\Aras Update\ directory.
5. Locate and select the **ArasEvent Handling Framework** package.



6. Click **Install**.



7. Select the **Aras Innovator Code Tree Updates** and **Aras Innovator Database Updates** options.
8. Click **Next**.
9. Select the appropriate logging option and click **Next**.

The logging option is used to record the installation attempt and can be used to troubleshoot issues.

10. Add the path to Innovator. For example, "**C:\Aras\InnovatorServer\Innovator**".
11. Input the connection information and click **Next**.

- **Innovator URL** = The connection URL for Aras Innovator
- **Database** = The target Aras Innovator database
- **Username** = root
- **Password** = Password for "root" login (Default is "innovator")
- **Event Daemon Password** = A service user password. This parameter lets the administrator set a password for the Event Daemon user

Aras Update should import everything required for Aras Event Handling Framework.

Aras Update

aras UPDATE

Enter the following parameters for each component you are going to install and then click Install button to continue.

✓ Aras Innovator Code Tree Updates

Path to Innovator

C:\Aras\14SP15\Innovator Browse...

✓ Aras Innovator Database Updates

Innovator URL

http://localhost/14SP15/

Database name of Innovator instance

14SP15

Innovator Username

root

Innovator Password

.....

Event Daemon Password

.....

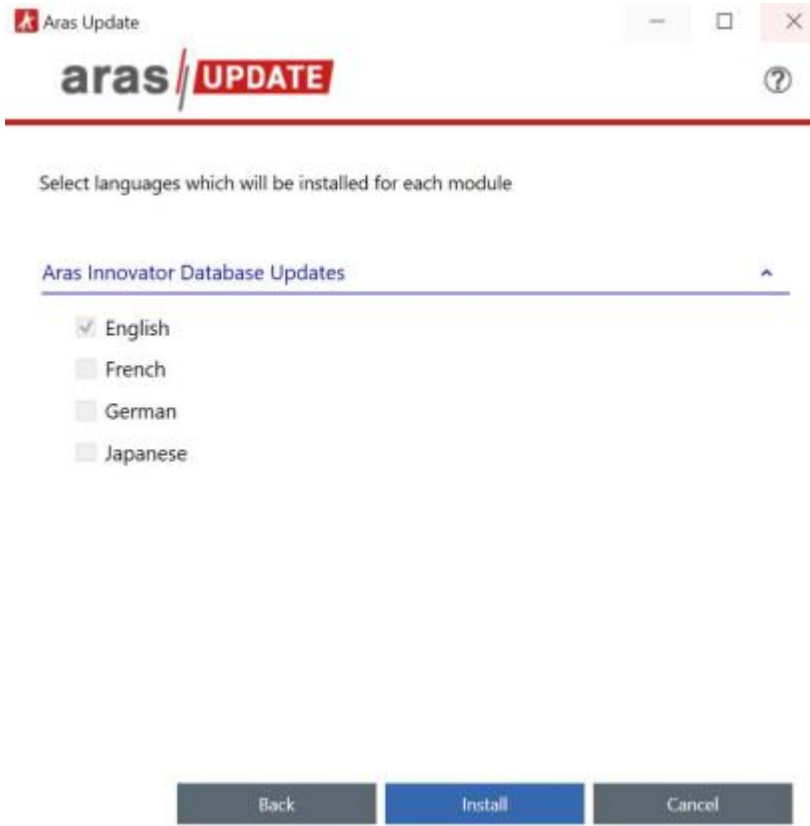
Confirm Event Daemon Password

.....

Back Next Cancel

12. Select the appropriate languages to install and click **Install**.





13. After the import is complete, disable the Super User login.

Plugin configuration

EHF plugin is responsible for generating and processing the events. Its parameters are stored in **Innovator\Server\InnovatorServer.Plugins.json** file.

The example of plugin configuration is shown below:

```
{
  "Name": "Aras.EventHandlingFramework.Plugin",
  "Enabled": true,
  "Options": {
    "EventsHandlingBackgroundServiceSettings": {
```

```
"CycleDelayInSeconds": 5,  
"EventsProcessingSettings": {  
"EventsHandlingThreadsCount": 10,  
"EventsPullBatchSize": 100  
}  
}  
}  
}
```

where:

Name – Name of EHF plugin (read-only).

Enabled – Indicates if EHF plugin is enabled.

CycleDelayInSeconds – Cycle delay for events processing.

EventsHandlingThreadsCount – Threads count for event processing.

EventsPullBatchSize – Events batch size that will be used during event processing on each cycle.

Important

The settings in the **InnovatorServer.Plugins.json** file are default and do not require any modification.



Manual Installation Option

The Following steps outline the process of Manual Installation of the Aras Event Handling Framework:

1. Download the Aras Event Handling Framework CD image from the Aras FTP site and unzip the file on the local computer.
2. Copy the `\Files\Innovator` folder to the root of your install directory, overwriting the existing `\Innovator` folder and all its contents.

It is recommended that this step be performed by an administrator on the server.

3. Go to Installed Innovator\Innovator\Server\ and open **InnovatorServer.Plugins.json** file.
4. Add the following:

```
{  
  "Name": "Aras.EventHandlingFramework.Plugin",  
  "Enabled": true,  
  "Options": {  
    "EventsHandlingBackgroundServiceSettings": {  
      "CycleDelayInSeconds": 5,  
      "EventsProcessingSettings": {  
        "EventsHandlingThreadsCount": 10,  
        "EventsPullBatchSize": 100  
      }  
    }  
  }  
}
```



}

S

```

1 {
2   "InnovatorServer.Plugins": [
3     {
4       "Name": "Aras.Plugins.DataProtectionKeysFileSystemStorage",
5       "Enabled": true,
6       "Options": {
7         "KeysDirectoryPath": "./App_Data",
8       }
9     },
10    {
11     "Name": "Aras.EventHandlingFramework.Plugin",
12     "Enabled": true,
13     "Options": {
14       "EventsHandlingBackgroundServiceSettings": {
15         "CycleDelayInSeconds": 5,
16         "EventsProcessingSettings": {
17           "EventsHandlingThreadsCount": 10,
18           "EventsPullBatchSize": 100
19         }
20       }
21     }
22   ]
23 }
24

```

5. Go to “Installed Innovator\Innovator\Server”, and open method-config.xml.

6. Add the following:

```
<name>$(binpath)/../Plugins/store/x64/net6.0/aras.eventhandlingframework.plu
```

```

<!--ReferenceAssemblies-->
<name>$(binpath)/Aras.Server.Core.dll</name>
<name>$(binpath)/Aras.TWF.Base.dll</name>
<name>$(binpath)/Aras.TWF.Base.Extensions.dll</name>
<name>$(binpath)/CommunicationManager.dll</name>
<name>$(binpath)/FileExchangeService.dll</name>
<name>$(binpath)/I18N.dll</name>
<name>$(binpath)/Newtonsoft.Json.dll</name>
<name>$(binpath)/SkiaSharp.ImageSharp.dll</name>
<name>$(binpath)/Microsoft.Data.SqlClient.dll</name>
<name>Microsoft.Extensions.Logging.Abstractions.dll</name>
<name>$(binpath)/../Plugins/store/x64/net6.0/Aras.EventHandlingFramework.Plugin/Aras.EventHandlingFramework.dll</name>
</ReferenceAssemblies-->
<support dialect="VB" template="VBscript" />
<support dialect="VBscript" template="VBscript" />
<support dialect="C#" template="CSharp" />
<support dialect="CSharp" template="CSharp" />
<!-- This next one is an example, it has not been tested -->
<template name="VBscript" line_number_offset="41">
  <!--DATA-->
  Imports Microsoft.VisualBasic
  Imports System
  Imports System.Text
  Imports System.IO
  Imports System.Collections.Generic
  Imports System.Data
  Imports System.Linq
  Imports System.Net
  Imports System.Globalization
  Imports Aras.I18N
  Imports Microsoft.AspNetCore.Mvc
  Imports Microsoft.AspNetCore.Mvc.Extensions

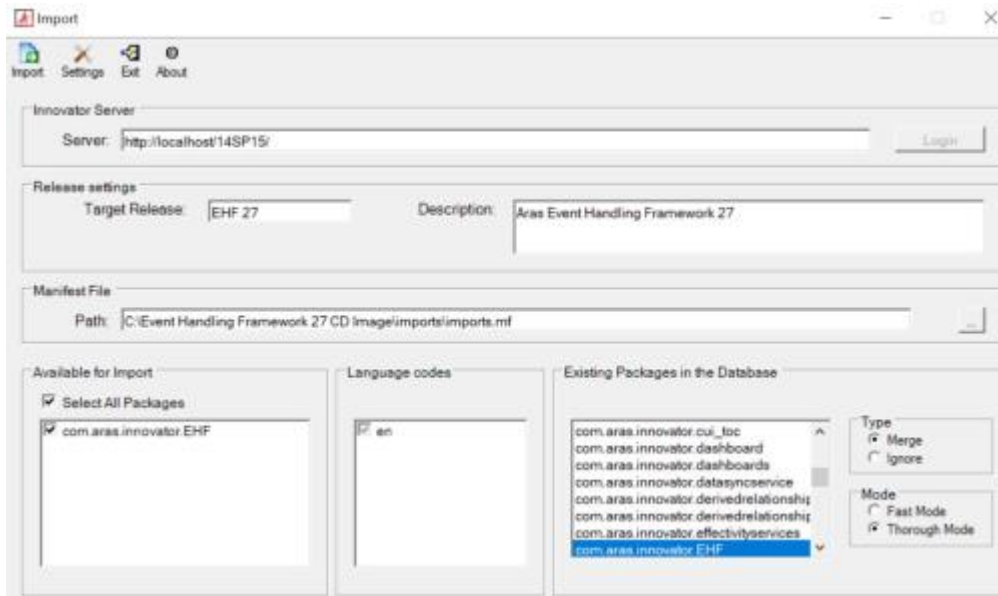
```

7. Restart IIS.

8. Import the Aras Event Handling Framework package with the Package Import Export Utilities. (For more information using this tool, please see the Aras Innovator - Package Import Export Utilities documentation.)



1. You will need to enable the “Super User” login to perform this step.
2. Browse to the \PackageImportExportUtilities\Import\ folder and run the **Import.exe** file.



3. Enter the following information:

- **Server** = The connection URL for Aras Innovator
- **Database** = The target Aras Innovator database
- **Username** = root
- **Password** = Password for “root” login (Default is “innovator”)
- **Target Release** = EHF 27
- **Description** = Aras Event Handling Framework 27
- **Manifest File Path** = The manifest is found in the unzipped patch folder in \Aras Event Handling Framework 27 CD Image\imports\imports.mf
- **Available for Import** = Check all packages
- **Type** = Merge
- **Mode** = Thorough Mode

4. Click the **Import** button.

9. Disable the “Super User” login.
10. Restart IIS.

Confirming the Installation

Use the following procedure to check if your database has been updated correctly:

1. Log in to Aras Innovator as an administrator.
2. From the TOC, select **Administration à Variables**.
3. Confirm the following sets of variables are listed:

Event Handling Framework
14.0.1

If at any time the installation fails, revert to the backups, and contact Aras Support at support@aras.com .



Post Installation

It is recommended that you review the *Aras Event Handling Framework 27 - Administration Guide* after the installation is complete.

