



Event Handling Framework

Release 39

Last Modified: 05/27/2026

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

Administration Guide	5
Introduction	6
Scope	7
Target Audience	8
Introduction to Event Handling Framework (EHF)	9
Event Handling Framework Identities	10
Subscriptions	11
Create Subscription	12
Event Types	14
Custom Event Type	19
Create Custom Event Type	20
Create Custom Event Method	22
Configure Custom Event Method Execution	24
Configure Event Publisher Identity	27
Event Handlers	29
Create Event Handlers	30
Create Custom Event Handler Method	32
Add Existing Event Types and Event Handlers to Subscription	35
Installation Guide	37
Introduction	38
Scope	39
Target Audience	40
Prerequisites	41
Checking for Eligibility	42
Installing Event Handling Framework	43
Notification and Backup	44



Installing the Event Handling Framework Platform Component	46
Automated Installation Option	47
Manual Installation Option	52
Confirming the Installation	55
Post Installation	56
SaaS Installation Guide	57
Introduction	58
Scope	59
Target Audience	60
Prerequisites	61
Checking for Eligibility	62
Installing Event Handling Framework in a SaaS Environment	63
Prerequisites	64
Installation	65
Confirming the Installation	68
Post Installation	69
Upgrade Guide	70
Overview	71
Prerequisites	72
Eligibility Verification	73
Upgrade Files Location	74
Upgrade Process	75
Notification and Backup	76
Applying the Upgrade Patch	78
Automated Upgrade Option	79
Manual Upgrade Option	82
Confirming the Application Installation	85
Release Notes	86



Platform Support Matrix	87
Enhancements, Fixed Issues, and Known Issues	88



Administration Guide



Introduction

Purpose

The Administration Guide describes the features of the Aras Event Handling Framework (EHF) 39 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 who are responsible for creating and managing all features within EHF.



Introduction to Event Handling Framework (EHF)

[Event Handling Framework Overview](#)

EHF allows the publishing of 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) published by the Event Handler(s) .



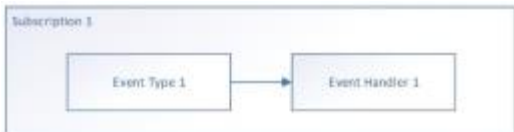
Event Handling Framework Identities

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.

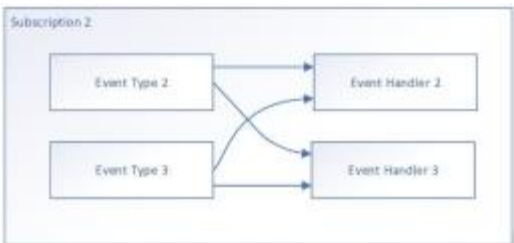


Subscriptions

The Subscription defines which **Event Handler** will process all events produced by the appropriate **Event Type**. Multiple Event Handlers and/or multiple Event Types are possible on a Subscription.



Subscription 1 contains Event Type 1 and Event Handler 1 in the above diagram. Event Handler 1 will process all events produced by Event Type 1.



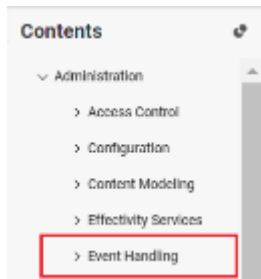
In the above diagram, **Subscription 2** contains 2 **Event Types** and 2 **Event Handlers**. Each event from **Event Type 2** or **Event Type 3** will be processed by both **Event Handler 2** and **Event Handler 3**. The same **Event Type** and **Event Handler** can be a part of multiple **Subscriptions**.



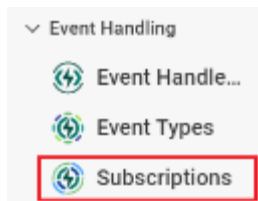
Create Subscription

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 **Event Handling**.

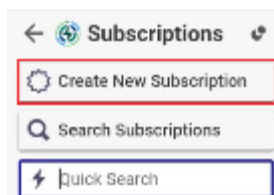


2. Select **Subscriptions**.



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. The next step is to link new Event Types to the **Event Handlers**.

Event Types

Create Item Event Types

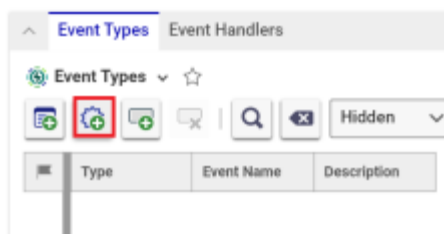
Item Event Type allows to subscribe to specific system events like “OnAfterAdd”, “OnAfterUpdate”, “OnAfterDelete”, “OnAfterCopy”, “OnAfterPromote” and “OnAfterVersion”. Every time an appropriate system event occurs, an EHF Item Event is generated and added to the publishing events queue.

Important

Since the publishing queue is in the server memory, it 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**.



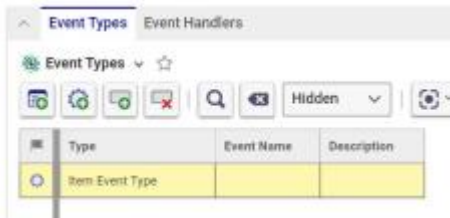
The **Select Item Type** dialog box appears.



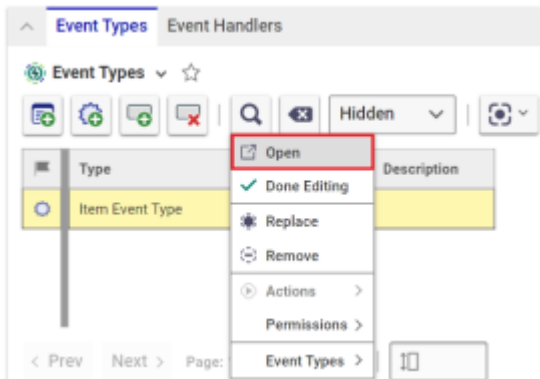
3. Select **Item Event Type** and click **OK**.



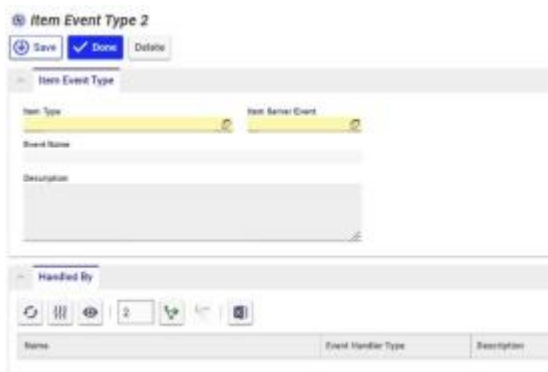
A new row appears on **EventTypes** tab.



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

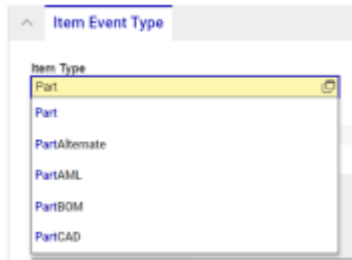


The **Item Event Type** form appears.

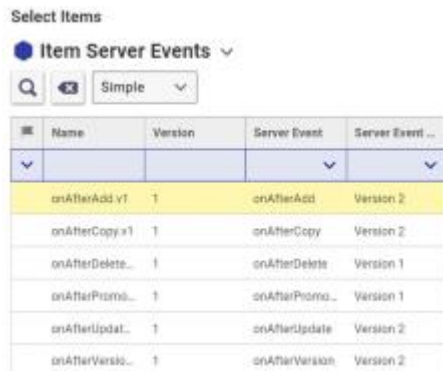


5. In the **ItemType** field, enter the **ItemType**.





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.

The server automatically generates the Event Name.

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** are visible only when the Event Type is linked to the 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 an Item Event Type will inform Aras Innovator when a specific event occurs and add the event data to the publishing queue for publication.

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>2024-08-26T12:26:31.8746280Z</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 containing the item's id 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 the 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

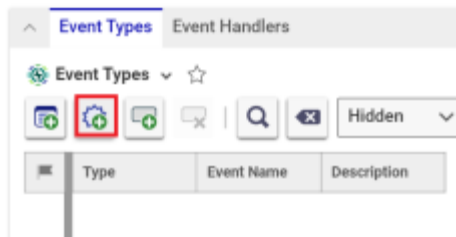
A **Custom Event Type** allows administrators to use method code to add events to the publishing queue. This capability enables publishing for events that fall outside the scope of the **Event Handling Framework's** standard **Item Event Types**.



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**.

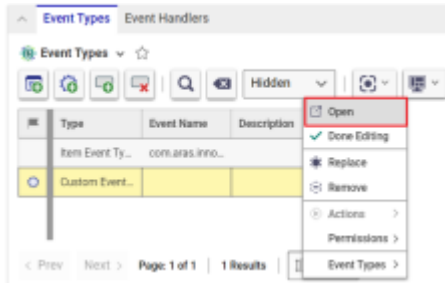


A new row appears on **EventTypes** tab.



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 a method to generate the Server Events that trigger the additional events in the publishing queue.

Important

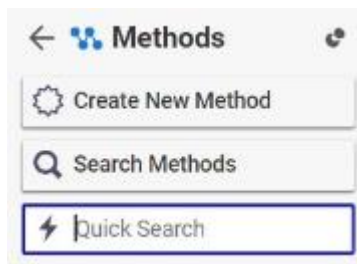
The Execution allowed field should be set to **ehf_EventPublishers** when creating a **Server Events** method.



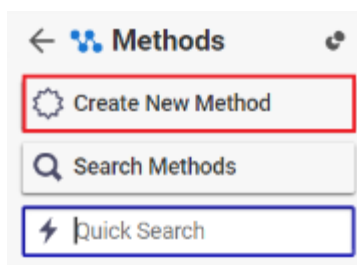
Create Custom Event Method

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

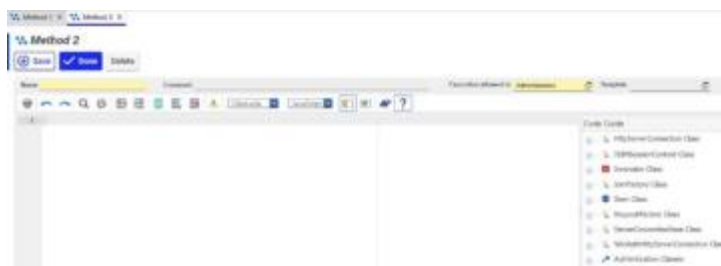
1. From the **Table of Contents**, expand **Administration** and select **Methods**. The **Methods Table of Contents** appears.



2. Click **Create New Method**.



The Method form appears.



3. Add the Method **Name** and the following properties:
 - o **Method Location:** Server-Side
 - o **Method Type:** C#
 - o **Execution Allowed To:** ehf_EventPublishers



4. Write the method code to 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;
```

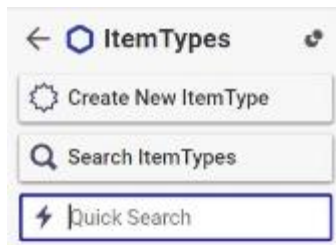
5. Click **Done** to save.



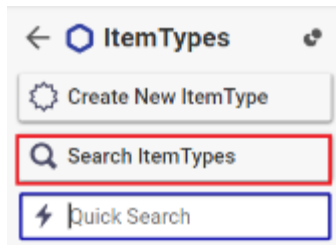
Configure Custom Event Method Execution

To publish a **Custom Event**, the **Custom Event Method** created in the “Create Custom Event Method” section 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.



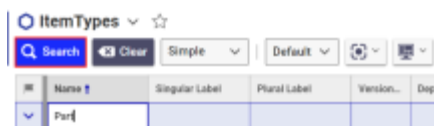
2. Click **Search ItemTypes**.



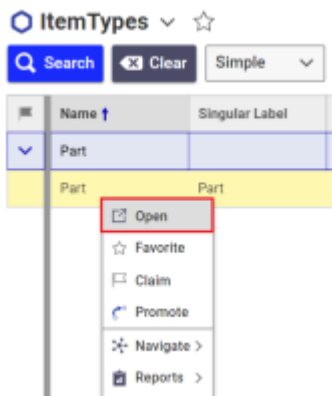
The **ItemTypes** grid appears.



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

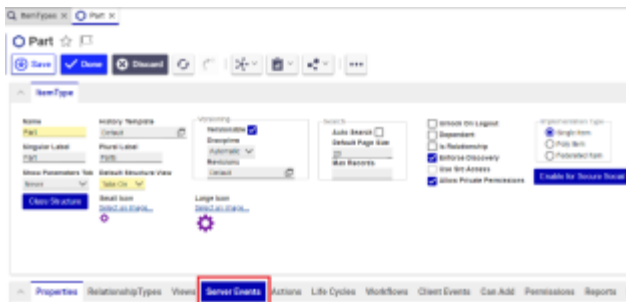


4. Right-click the ItemType in the grid and click **Open**.

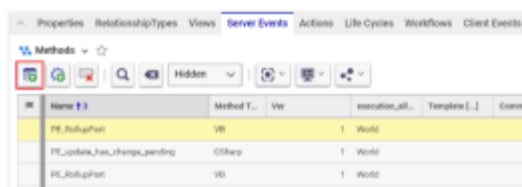


5. Click **Edit**.

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

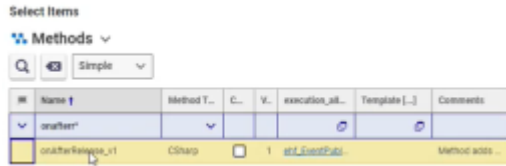


7. Click **Add Methods**.

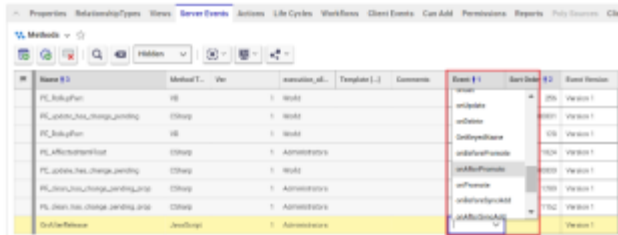


8. In the **Select Items** dialog box, search for the method created in the **Create Custom Events Method** section and click **OK**.





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**.

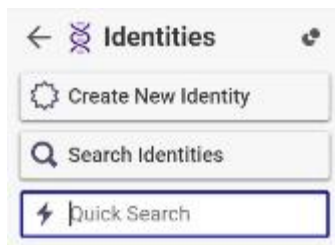


Configure Event Publisher Identity

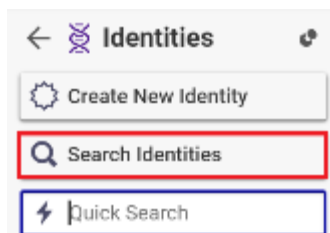
The **ehf_EventPublishers Identity** is a group Identity used to grant permissions for adding events to the publishing queue. Any users who may cause an event to be added to the publishing queue should be added to this Identity.

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

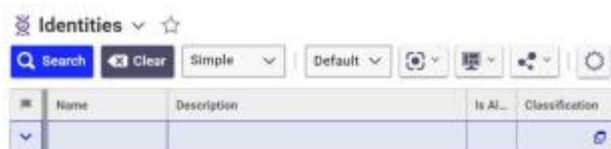
1. From the **Table of Contents**, expand **Administration** and select **Identities**. The Identities Table of Contents appears:



2. Click **Search Identities**.

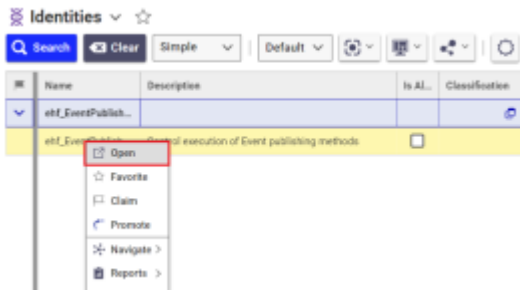


The **Identities Search** grid appears.



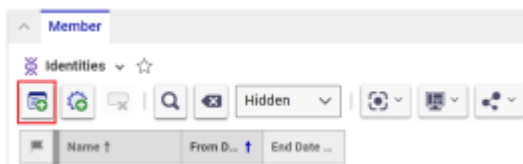
3. In the Name field, enter **ehf_EventPublishers** and click **Search**.

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**.



8. Click **Done** to save.

Event Handlers

[Ad Hoc Event Handler](#)

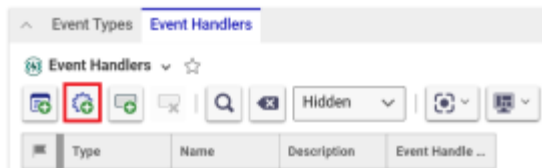
The **Event Handlers** allow the Administrator to configure how the events will be processed. The **Ad Hoc Event Handler** allows the specification of an arbitrary C-Sharp (C#) method with custom logic that will be executed each time new events are processed.



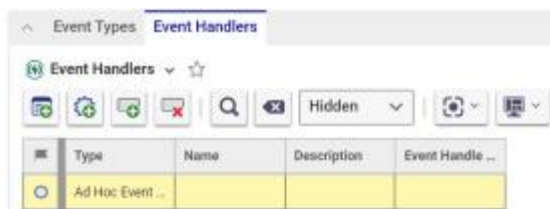
Create Event Handlers

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

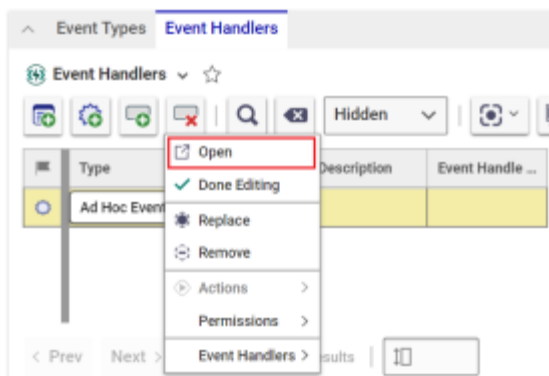
1. On the **Subscription** form, click **Edit**.
2. From the **Subscription** form, scroll down, switch to the Event Handlers tab, and click **New Event Handler**.



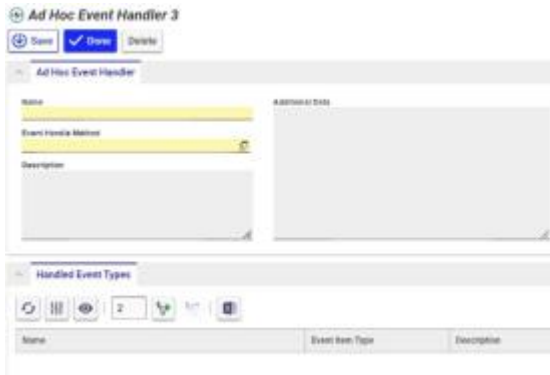
A new row appears on the **Event Handlers** tab.



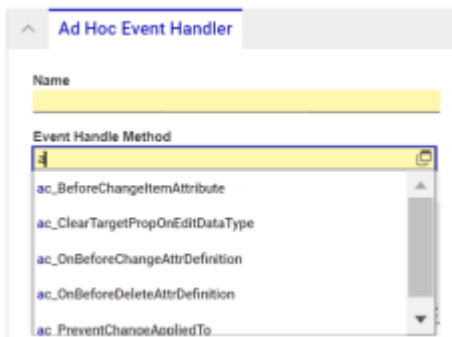
3. Right-click on the newly added row and click **Open**.



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 additional data required for the custom method.
8. Click **Done** to save.

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

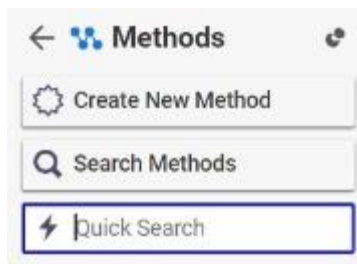


Create Custom Event Handler Method

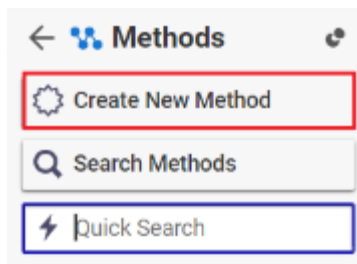
The Administrators need to create **Event Handler** Methods that will process all events produced by the appropriate **Event Type**.

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

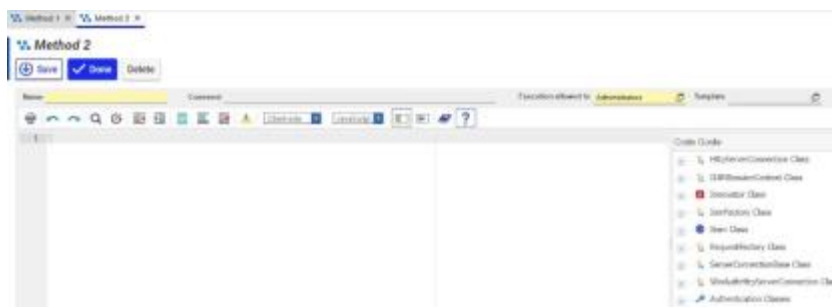
1. From the Table of Contents, expand Administration and select Methods. The Methods Table of Contents appears.



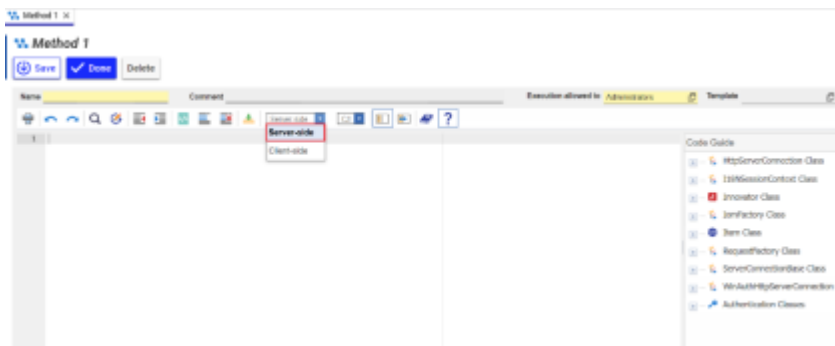
2. Click **Create New Method**.



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;
```

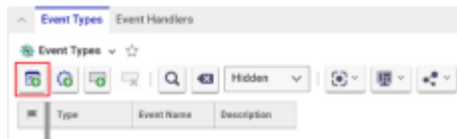
7. Click **Done** to save the changes.



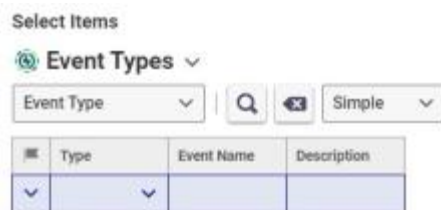
Add Existing Event Types and Event Handlers to Subscription

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

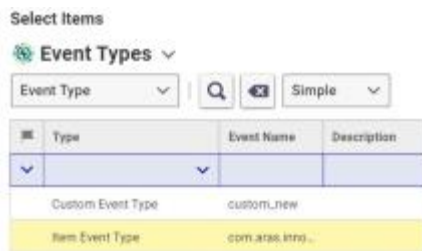
1. On the Subscription form, click **Edit**.
2. From the Subscription form, scroll down to the **Event Types** tab and click the **Add Event Types**.



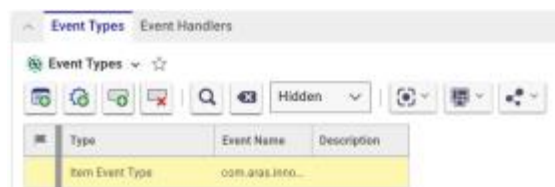
The Search Item dialog box appears.



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

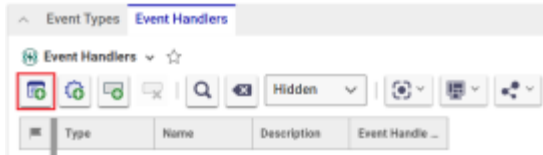


4. Click **OK**. The selected **Event Type** appears on the Event Types tab on the **Subscription** form.



5. Click **Done** to save.

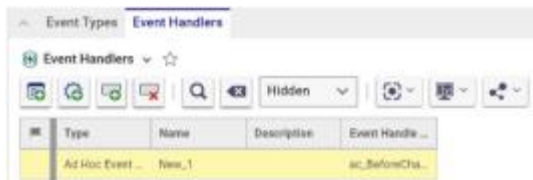
6. To add **Event Handlers**, switch to the **Event Handlers** tab and click **Add Event Handlers**.



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

This Installation Guide describes how to install Aras Event Handling Framework 39.



Scope

This 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 39, 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 check if the code tree and database are eligible for installing the Aras Event Handling Framework 39:

1. Log into **Aras Innovator** as an administrator.
2. Click the **User** icon in the top-right corner and select **About**.
3. Aras Event Handling Framework 39 is backward compatible with Aras Innovator releases 27 through 39. Verify that one of the following Aras Innovator releases appears:
 - Aras Innovator Release 27
 - Aras Innovator Release 28
 - Aras Innovator Release 29
 - Aras Innovator Release 30
 - Aras Innovator Release 31-39

From the **TOC**, select **Administration, Variables**.

VersionMajor	VersionMinor	VersionServiceUpdate	VersionLabel
14	0	15	Release 27
14	0	18	Release 28
14	0	20	Release 29
14	0	22	Release 30
14	0	25	Release 31
14	39	0	Release 39

Contact Aras Support before installing if the eligibility requirements are unmet.



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 – 39
- 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 before starting the process.

Give at least 24-hour notice and a reminder 15 minutes before taking the system offline.

2. Back up the code tree.

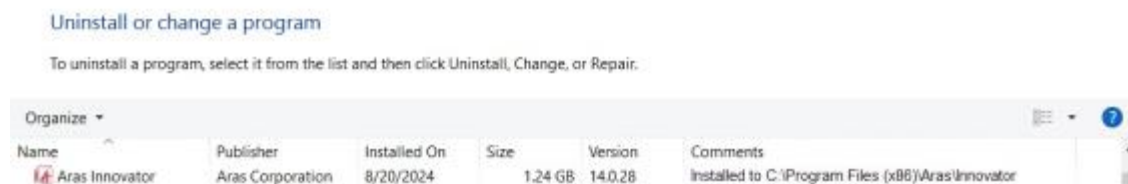
The 'Code Tree' refers to files and folders installed on the disk when Aras Innovator was first installed.

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

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

The installation folder can be confirmed with the following steps:

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:



Once the location of the code tree installation is determined, backup the folder and its contents.

3. 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 sessions and prevents all new connections to the Aras Innovator database through the existing instance.

4. Backup the database.

Place this file in a safe location. If the application fails, it will need to be restored from there.

5. Enable database connections.

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



Installing the Event Handling Framework Platform Component

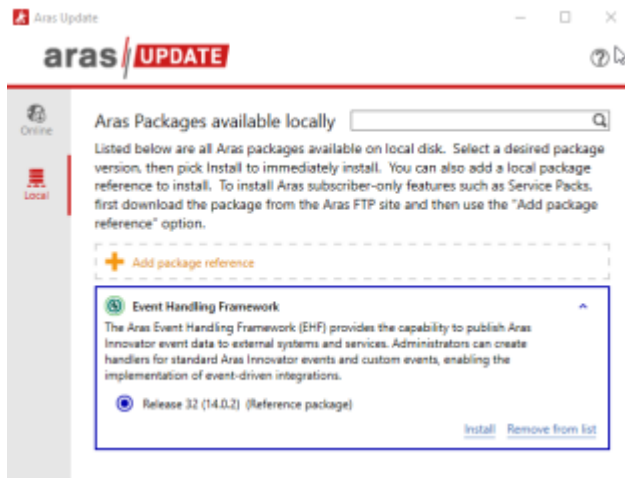
The Event Handling Framework component can be installed using either the Automated Installation option (using the **Aras Update** tool), or the Manual Installation option (using the **Aras Import** tool).



Automated Installation Option

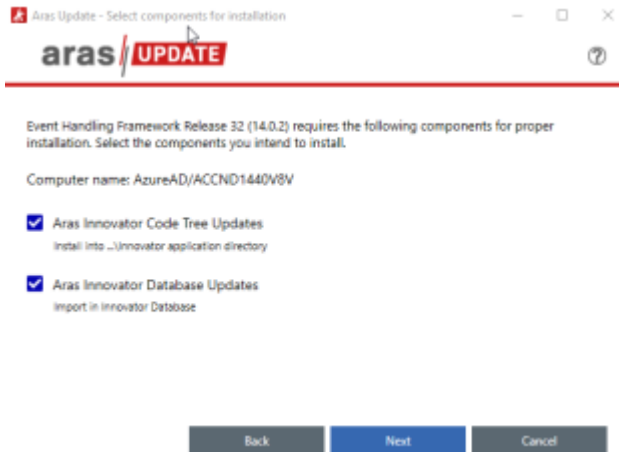
The following steps outline the process of Automated Installation of the Aras Event Handling Framework:

1. Extract the Aras Event Handling Framework CD Image on the local machine.
2. Enable the **Super User** login.
3. Launch the **ArasUpdate.exe** file as administrator.
If default options were selected during installation, the **ArasUpdate.exe** file is found in the directory C:\Program Files (x86)\Aras\Aras Update\ .
4. On the **Local** tab, click **Add package reference** and navigate to the folder from Step 1 where the **ArasEvent Handling Framework** package was extracted.

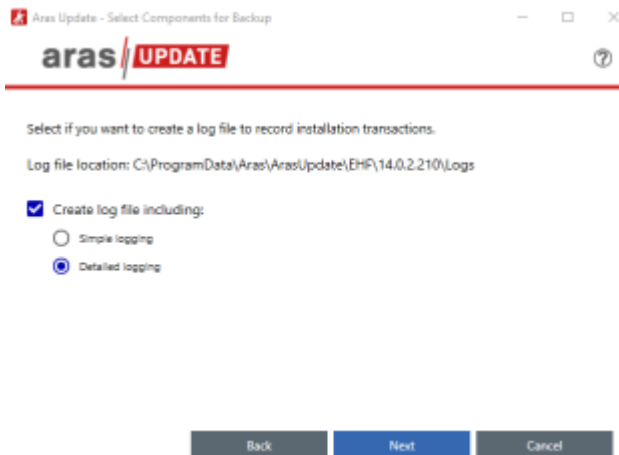


5. Click **Install**.
6. Select the **Aras Innovator Code Tree Updates** and **Aras Innovator Database Updates** options and click **Next**.





7. Select the **Detailed logging** option and click **Next**.



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

8. Fill in the **Path to Innovator**, **Innovator URL**, **Database name of Innovator Instance**, **Username** (root), **Password** for the root user, **Event Daemon Password**, and click **Next**.

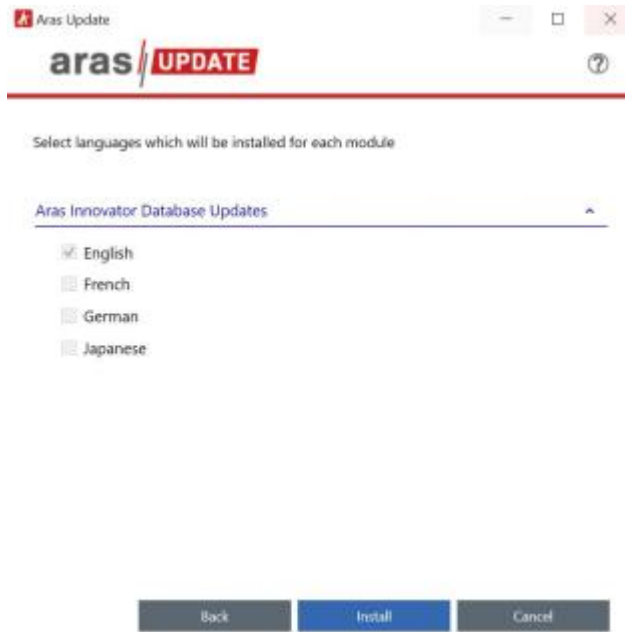


- **Innovator URL** = The connection URL for Aras Innovator
- **Database** = The target Aras Innovator database
- **Username** = root
- **Password** = Password for “root” login (Default is “innovator”)
- **Event Daemon Password** = A service user password. This parameter lets the administrator set a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

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

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





10. 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.

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 extract 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/net8.0/aras.eventhandlingframework.plu
```

```

<MethodConfig>
  <ReferencesAssembly>
    <name>$(binpath)\Aras.Server.Core.dll</name>
    <name>$(binpath)\Aras.TDF.Base.dll</name>
    <name>$(binpath)\Aras.TDF.Base.Extension.dll</name>
    <name>$(binpath)\Conversion.Base.dll</name>
    <name>$(binpath)\ConversionManager.dll</name>
    <name>$(binpath)\FileExchangeService.dll</name>
    <name>$(binpath)\IOM.dll</name>
    <name>$(binpath)\Microsoft.Team.Fdm.dll</name>
    <name>$(binpath)\Silobase.ImageSharp.dll</name>
    <name>$(binpath)\Microsoft.Data.SqlClient.dll</name>
    <name>$(binpath)\Microsoft.Extensions.Logging.Abstractions.dll</name>
    <name>$(binpath)\C:\ProgramData\Aras\Aras.EventHandlingFramework.Plugin\1.0.8\lib\net8.0\Aras.EventHandlingFramework.dll</name>
  </ReferencesAssembly>
  <Support dialect="VB" template="VBscript" />
  <Support dialect="VBscript" template="VBscript" />
  <Support dialect="C#" template="CSharp" />
  <Support dialect="CSharp" template="CSharp" />
  <!-- This new one is an example, it has not been tested -->
  <template name="VBscript" line_number_offset="41">
    <script>
      Imports Microsoft.VisualBasic
      Imports System
      Imports System.Collections
      Imports System.Collections.Generic
      Imports System.Linq
      Imports System.Net
      Imports System.Reflection
      Imports System.IO
      Imports Microsoft.AspNetCore.Http
      Imports Microsoft.AspNetCore.Mvc
    </script>
  </template>
</MethodConfig>

```

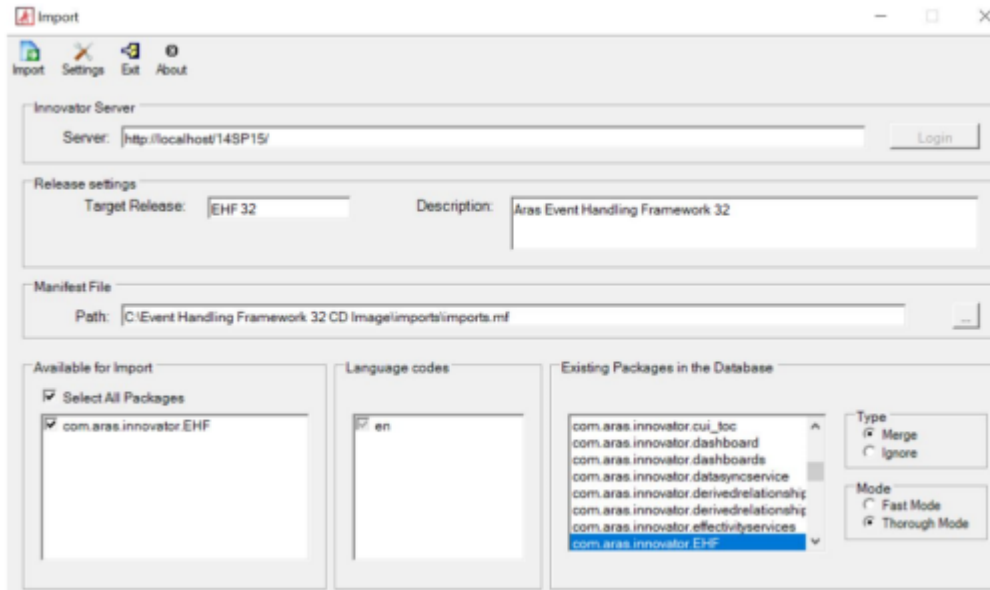
7. Restart **IIS**.

8. Use the **Package Import Export Utilities** to import the Aras Event Handling Framework package. (For more information about using this tool, please see the *Aras Innovator—Package Import Export Utilities* documentation.)

1. Enable the **Super User** login.

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 39
- **Description** = Aras Event Handling Framework 39
- **Manifest File Path** = The manifest is found in the unzipped patch folder in `\Aras Event Handling Framework 39 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

14.3.0

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



Post Installation

After the installation is complete, it is recommended that you review the *Aras Event Handling Framework 39—Administration Guide*.



SaaS Installation Guide



Introduction

Purpose

This Installation Guide describes installing Aras Event Handling Framework 39 in a SaaS environment.



Scope

This Installation Guide provides detailed information on installing the 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 39, 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 check if the code tree and database are eligible for installing the Aras Event Handling Framework 39:

1. Log into **Aras Innovator** as an administrator.
2. Click the **User** icon in the top-right corner and select **About**.
3. Aras Event Handling Framework 39 is backward compatible with Aras Innovator releases 27 through 39. Verify that one of the following Aras Innovator releases appears:
 - Aras Innovator Release 27
 - Aras Innovator Release 28
 - Aras Innovator Release 29
 - Aras Innovator Release 30
 - Aras Innovator Release 31

 - Aras Innovator Release 32-39

From the **TOC**, select **Administration, Variables**.

VersionMajor	VersionMinor	VersionServiceUpdate	VersionLabel
14	0	15	Release 27
14	0	18	Release 28
14	0	20	Release 29
14	0	22	Release 30
14	0	25	Release 31
14	0	28	Release 32
14	39	0	Release 39

Contact Aras Support before installing if the eligibility requirements are unmet.



Installing Event Handling Framework in a SaaS Environment



Prerequisites

1. Access to Azure DevOps portal
2. Local repository has been set up
3. Aras Innovator Release 27 – 39



Installation

1. Open the main folder of the Aras Event Handling Framework 39 CD Image:
 1. Copy the folder Innovator into the CodeTree folder in the repository.
 2. Navigate to .\imports\com\aras\innovator\ and copy the folder EHF into the AML-packages folder of the repository.
2. Navigate to the imports folder in the Aras Event Handling Framework 39 CD Image and copy the content of the file imports.mf to the buffer.
3. In the repository, edit the file AML-packages\imports.mf and paste the lines from step 2
4. Save and close the file
5. In the repository, navigate to the folder TransformationsOfConfigFiles and create the folders Innovator and Innovator\Server
6. Navigate to the folder Innovator\Server and create a new file method-config.xml and copy/paste the following lines:

```
<MethodConfig xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">
```

```
<ReferencedAssemblies>
```

```
<name
```

```
xdt:Transform="Insert">/app/Plugins/store/x64/net6.0/aras.eventhandlingframe
```

```
<name
```

```
xdt:Transform="Insert">/app/Plugins/store/x64/net6.0/aras.eventhandlingframe
```

```
<name
```

```
xdt:Transform="Insert">/app/Plugins/store/x64/net6.0/aras.eventhandlingframe
```

```
<name
```

```
xdt:Transform="Insert">/app/Plugins/store/x64/net6.0/aras.eventhandlingframe
```

```
<name
```

```
xdt:Transform="Insert">/app/Plugins/store/x64/net8.0/aras.eventhandlingframe
```



```

<name
xdt:Transform="Insert">/app/Plugins/store/x64/net8.0/aras.eventhandlingframe

<name
xdt:Transform="Insert">/app/Plugins/store/x64/net8.0/aras.eventhandlingframe

<name
xdt:Transform="Insert">/app/Plugins/store/x64/net8.0/aras.eventhandlingframe

</ReferencedAssemblies>

</MethodConfig>

```

- In the same folder Innovator\Server, create a new file InnovatorServer.Plugins.json and copy/paste the following lines:

```

{
  "@jdt.merge": {
    "@jdt.path": "$.*",
    "@jdt.value": [
      {
        "Name": "Aras.EventHandlingFramework.Plugin",
        "Enabled": true,
        "Options": {
          "EventsHandlingBackgroundServiceSettings": {
            "CycleDelayInSeconds": 5,
            "EventsProcessingSettings": {
              "EventsHandlingThreadsCount": 10,

```



```
"EventsPullBatchSize": 100
```

```
}
```

```
}
```

```
}
```

```
}
```

```
]
```

```
}
```

```
}
```

Note that parameter values can be customized as needed.

8. Run the build process on a local environment and test that the application is installed.
9. Create a pull request to commit the changes to the central repository. Users may need to request a code review before merging the changes, depending on the organization's policies.
10. Run the Continuous Integration pipeline to build images with the changes related to Event Handling Framework
11. Once the Continuous Integration pipeline has completed, run Deploy pipeline on packages built in previous step.



Confirming the Installation

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

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

Event Handling Framework

14.3.0



Post Installation

After the installation is complete, users should review the *Aras Event Handling Framework 39—Administration Guide*.



Upgrade Guide



Overview

This document provides instructions for upgrading the Aras Event Handling Framework platform component from Release 27 to Release 39. These changes affect the Aras Innovator code tree and the database.



Prerequisites

The following prerequisites are required to upgrade to Aras Event Handling Framework 39:

- Aras Event Handling Framework 27
- Aras Innovator Release 27 - 39



Eligibility Verification

The following steps outline the process to verify the current Aras Innovator code tree and database meet the minimum requirements to apply the Aras Event Handling Framework updates:

1. Log into **Aras Innovator** as an **Administrator**.
2. From the **Main Menu**, select **Help** and **About**.
3. Verify the dialog displays **Release 27 - 39**.
4. Close the dialog.
5. From the **TOC**, select **Administration** and **Variables**.
6. Confirm the following variables display:

Version Major	Version Minor	VersionServiceUpdate	Aras Event Handling Framework
14	0	15	14.0.1 or 14.0.2
14	0	18	14.0.1 or 14.0.2
14	0	20	14.0.1 or 14.0.2
14	0	22	14.0.1 or 14.0.2
14	0	25	14.0.1 or 14.0.2
14	0	28	14.0.2
14	30	0	14.0.2
14	34	0	14.0.2
14	35	0	14.0.2
14	36	0	14.0.2
14	37	0	14.0.2
14	38	0	14.0.2
14	39	0	14.0.2

Contact Aras to discuss upgrade options if the information displayed differs from this document's outline.



Upgrade Files Location

The upgrade files are available in the Aras Event Handling Framework CD image file `Aras Event Handling Framework 39 Upgrade Patch.zip`. These files contain the components necessary to upgrade the code tree and database.



Upgrade Process

The following steps outline the process of upgrading Aras Event Handling Framework to release 39.

Test runs in a non-production environment are recommended before executing an upgrade in the production environment.

Important

The Aras Event Handling Framework Upgrade should be applied after any Aras Innovator upgrade patch.



Notification and Backup

The following steps outline the process of sending an upgrade notification and performing a backup of the Aras Event Handling Framework platform component:

1. Notify users that the system will be down at a scheduled date and time and that they should log out before starting the process.

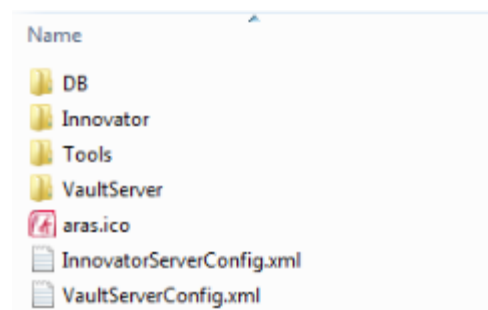
Giving at least 24 hours' notice and a reminder 15 minutes before taking the system offline is best.

2. Back up the code tree:

The 'Code Tree' refers to files and folders installed on the disk when Aras Innovator was first installed.

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

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



The installation folder can be confirmed with the following steps:

1. From the Start Menu, click Start and select Settings and Control Panel.
2. From the Control Panel, select Programs and Features.
3. Search for the **Aras Innovator** entry.

The entry should display the physical installation path.

Once the location of the code tree installation is determined, backup the folder and its contents.

3. 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 sessions and prevents all new connections to the Aras Innovator database through the existing instance.

4. Backup the database.

Place this file in a safe location. If the application fails, it will need to be restored from there.

5. Enable database connections.

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



Applying the Upgrade Patch

Aras Event Handling Framework can be upgraded using either the Automated Installation option (using the **Aras Update** tool) or the Manual Installation option (using the **Aras Import Tool**).

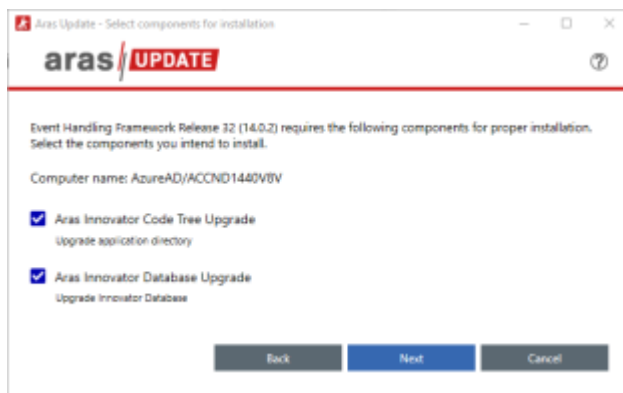


Automated Upgrade Option

1. Extract the **Aras Event Handling Framework 32 Upgrade Patch.zip** file to a folder on the local machine.
2. Enable the **Super User** login.
3. Launch the **ArasUpdate.exe** file as administrator.

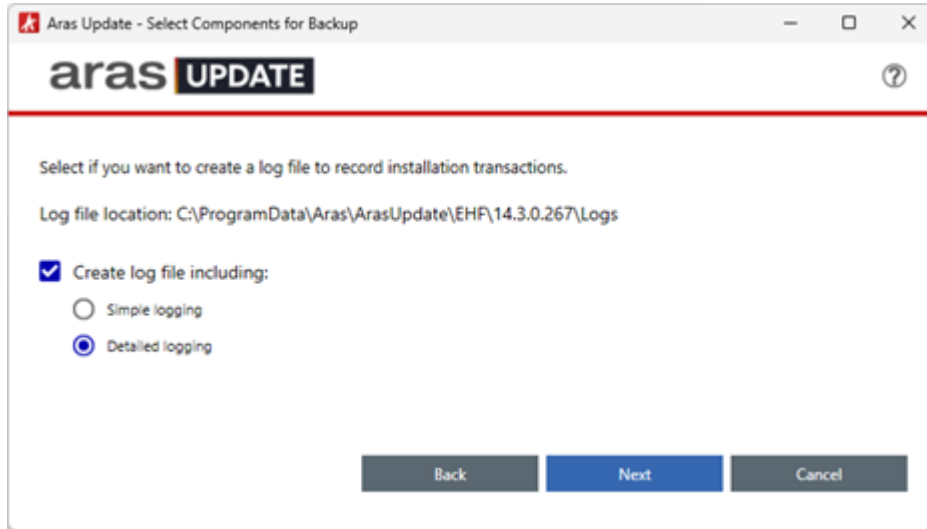
If default options were selected during installation, the **ArasUpdate.exe** file is found in the directory C:\Program Files (x86)\Aras\Aras Update\.

4. On the **Local** tab, click **Add package reference** and navigate to the folder where the upgrade file was extracted from the first step.
5. Click **Install**.
6. Select the **Aras Code Tree Upgrade** and **Aras Innovator Database Upgrade** options.



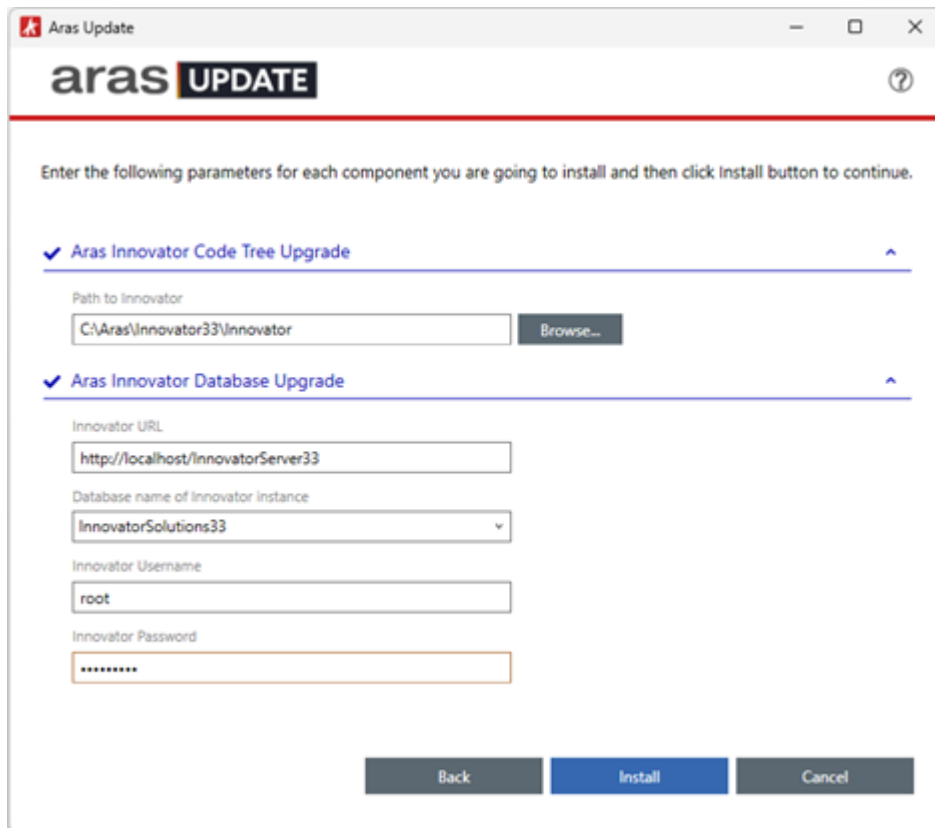
7. Select the **Detailed logging** option and click **Next**.





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

8. Input the connection information and click **Install**.

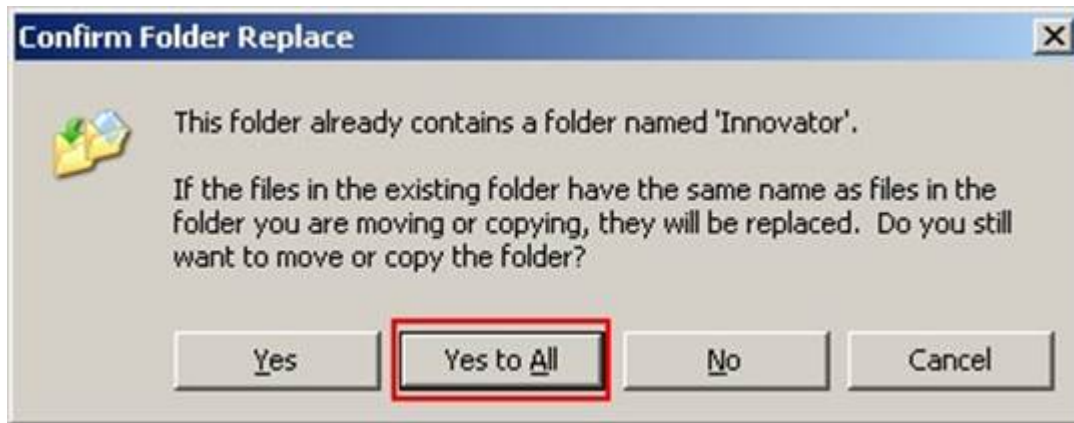


- **Server URL** = The connection URL for Aras Innovator
 - **Database** = The target Aras Innovator database
 - **Username** = root
 - **Password** = Password for “root” login (Default is “innovator”)
9. After the upgrade is complete, disable the **Super User** login.



Manual Upgrade Option

1. From the **Aras Event Handling Framework 32 CD Image** folder, locate and extract the Aras Event Handling Framework 39 Upgrade Patch.zip file.
2. Copy the **Innovator** folder from the Upgrade Patch to the **Aras Innovator Code Tree** location. This will overwrite the existing **Innovator** folder and files. It is recommended that a Systems Administrator perform this step on the server.



3. Increment the filesRevision attribute in the code tree to ensure that client machines renew their caches to get all required updates.
 1. From the root of the **Code Tree**, open the InnovatorClient.config file within the \Innovator\Client folder.
 2. Change the value of the filesRevision attribute. The value should be changed from std to 2, 3, 4, etc.


```
<cachingModule moduleEnabled="true" filesRevision="2" />
```
 3. Save the changes to the InnovatorClient.config file and close it.

Important

Incrementing this attribute sends a signal to each client machine that it needs to reload the client cache the next time it connects to Aras Innovator.

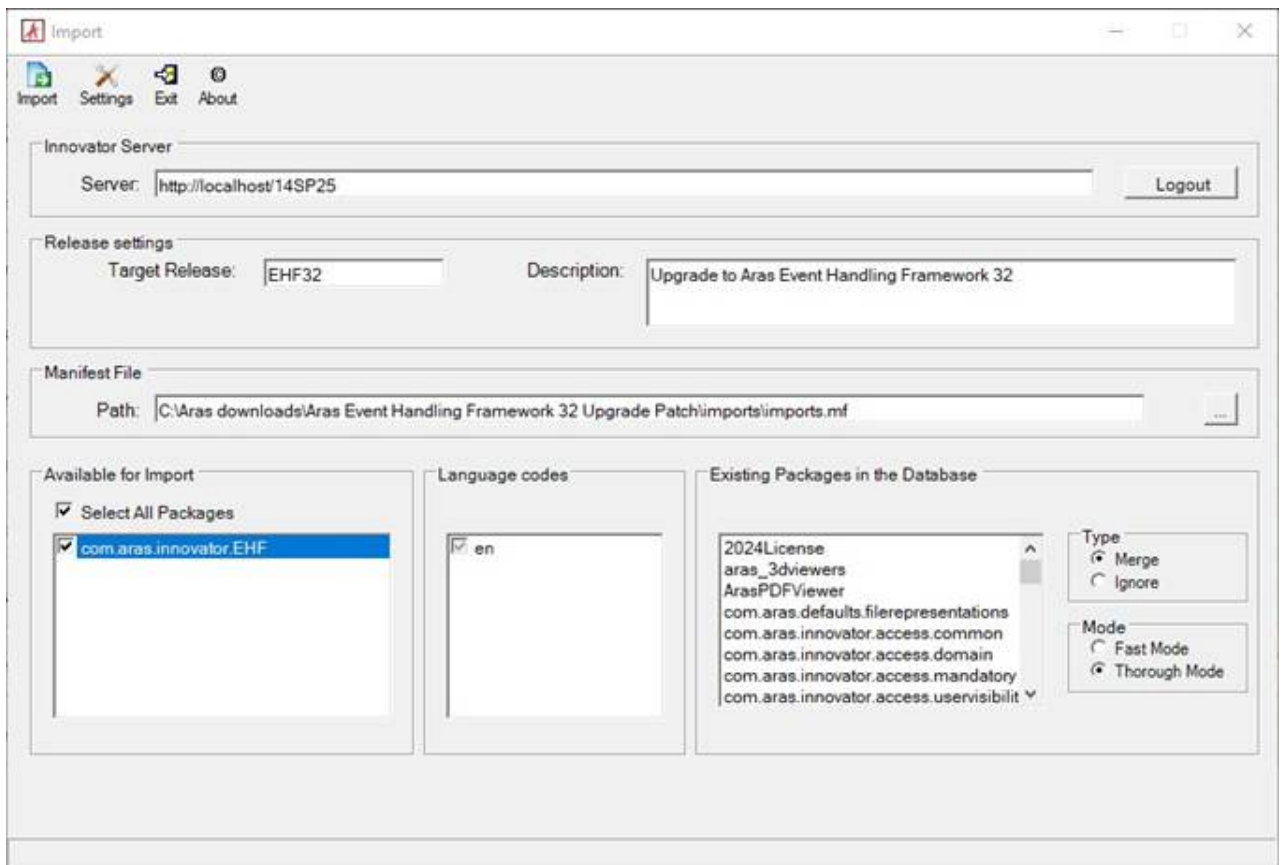
4. Restart the **World Wide Web Service (IIS)** to ensure that the server renews its cache and receives all required updates.

- Use the **Package Import Export Utilities** to import the database package from the Aras Event Handling Framework 39 Upgrade Patch.zip file. For more information on using this tool, refer to the *Aras Innovator—Package Import Export Utilities* document.

Important

This import will fail if you do not have **Aras Process Engine** installed.

- Enable the **Super User** login to perform this step.
- Browse to the \PackageImportExportUtilities\Import\ folder and run the **Import.exe** file.



- Input the connection information:



- **Server** = The connection URL for Aras Innovator
 - **Username** = root
 - **Password** = Password for “root” login (The default password used is “innovator”)
 - **Target Release** = EHF39
 - **Description** = Upgrade to Aras Event Handling Framework 39
 - **Manifest File Path** = The manifest is found in the folder \imports\import.mf
 - **Available for Import** = Select all packages
 - **Type** = Merge
 - **Mode** = Thorough Mode
4. Click **Import** to import the file.
 5. Disable the **Super User** login once complete.

Important

The **Super User** login should not be enabled in production.



Confirming the Application Installation

Use the following steps to check if the database has correctly applied the **Aras Event Handling Framework** changes to **Aras Innovator**.

1. Log in to **Aras Innovator** as an administrator.
2. From the TOC, select **Administration\Variables**.
3. Confirm the following variable is listed:

Name	Value
Event Handling Framework	14.3.0

If any part of this test fails, revert to the backups and contact Aras Support at support@aras.com .



Release Notes



Platform Support Matrix

For the software required for the Aras Event Handling Framework 39:

Aras Innovator Release Client Operating System Browser

27 - 39	Windows: 10, 11	<ul style="list-style-type: none">• Chrome• Edge• Firefox: 102 ESR, 115 ESR, 128 ESR
---------	-----------------	--



Enhancements, Fixed Issues, and Known Issues

The Aras Event Handling Framework now supports .NET 10

This release introduces compatibility with .NET 10. Aras Innovator is planning a platform-wide migration from .NET 8 to .NET 10 across Innovator and supporting components.

Why this change?

Microsoft ends .NET 8 Long Term Support (LTS) on November 10, 2026. Moving to .NET 10 keeps Innovator aligned to an LTS framework (supported through November 2028) to support customer compliance and security requirements.

