Relativity one

Event Handler Express Guide

May 16, 2025

Table of Contents

1 Relativity Event Handler Express

The Event Handler Express application facilitates the creation of simple event handlers by providing a user-friendly interface. It provides the ability to define validation rules for fields available in Relativity layouts. With this application, you create Pre Save event handlers that perform validation when the user clicks the Save button on a layout. This validation occurs before Relativity writes any data to the database.

For example, you may want to create custom layouts for your organization that you use to drive review workflows. With Event Handler Express, you can easily add custom functionality to these layouts by building event handlers. The layouts display messages to reviewers when they fail to follow the rules specified in the event handler. You can also deploy your event handlers to multiple Relativity environments, so that you can reuse your customizations.

1.1 Basic concepts

Event Handler Express simplifies the development of event handlers so that you don't manually need to write the code for them. When you build event handlers with this tool, you need to define their behavior by creating condition groups and conditions. The following list includes key concepts that you need to understand when building these event handlers:

- **Event handler** - an assembly (.dll file) that executes when a specific event occurs. You can create Pre Save event handlers that perform validation when the user clicks the Save button on a layout, but before Relativity writes any data to the database. When you deploy the event handler, the application automatically generates an assembly that executes when a user performs a specific task on an object.

    Note: Event Handler Express only supports attaching event handlers to document objects and Relativity Dynamic Objects (RDOs). It doesn't support attaching event handlers to any system objects.

- **Condition** - a validation rule. You build conditions by performing these tasks:
    - Identifying a validation type (such as current user, fixed-length text, or others).
    - Selecting a field name and an operator for evaluation purposes.
    - Choosing a valid choice or value.

        For example, you may have a layout that includes fields called Responsive and Issue Designation. You could add a condition that results in an error message when the user selects true for the Responsive field, but doesn't select an option for the Issue Designation field.

        Event Handler Expresses supports five field types, including Current User, Fixed-length Text, Single Choice, Multiple Choice, and Yes/No fields. You can use only the supported operators for each field type when building an event handler. If you add multiple conditions, you can choose to evaluate them as a series of expressions connected by the AND or OR operators. See Building your first event handler on page 18.

- **Condition group** - a set of related validation conditions. For example, a condition group might include a series of conditions that test for these items:
    - A field containing a certain value.
    - A field containing a value that is set.
    - A user belonging to a specific group.

        You can add an error message that appears when a user makes an inappropriate selection based on the

requirements specified in the condition. If you create multiple condition groups, Event Handler Express evaluates them as a series of expressions connected by the AND operator.

- **Dynamic link library (DLL) file** - a file that Event Handler Express automatically generates when you deploy an event handler to a workspace. During deployment, Event Handler Express validates the logic of the conditions set in the event handler, compiles a .dll file, and pushes it to Relativity as a resource file. The application also attaches the event handler to a specified object, so that it is immediately available on layouts in the workspace. You can download the resource file and then upload it to another Relativity installation. This process makes the event handler functionality available in additional workspaces. For more information, see Uploading an event handler to another Relativity instance on page 15.

In addition to these key concepts, review the following functionality that characterizes the event handlers developed through Event Handler Express:

- **Triggered only for updates to a single record** - Your event handlers execute when a user clicks **Save**, **Save & Next**, and **Save and Back**. They aren't executed on mass operations or inline editing.

- **Read-only fields and propagation** - Propagation doesn't occur when a field populated by an event handler condition is set to Read Only = Yes and propagation is enabled on the field.

- **Coding decisions prior to event handler deployment** - The conditions that you define in your event handlers only apply to coding decisions made after you deploy the event handler in your environment. You must re-code previously reviewed documents for the event handler conditions to take effect.

1.2 Best practices for building event handlers

Use these guidelines to optimize your event handler development through Event Handler Express.

1.2.1 Use a test environment

Confirm that your environment handlers are working properly by verifying the expected functionality in a test environment.

Note: An improperly functioning event handler could impede the document review process across your environment. Perform thorough testing of new event handlers before deploying them in a production environment. If your event handler isn't functioning properly, you can update, remove, or detach it from the object type. For more information, see Editing or deleting an event handler on page 14, or Editing Relativity objects on the Relativity Documentation site.

1.2.2 Check the logic for your conditions

Ensure that you don't have any conflicting conditions in your event handlers. For example, a conflict occurs if you have one condition that checks if Responsive is true, and a second condition that checks if Responsive is false.

Event Handler Express checks the conditions that you set to ensure that they are logical statements. However, it can't confirm the validity of your business rules. You can avoid interruptions to your review process by validating the business logic used in your event handlers before you deploy them in a workspace. The event handlers are immediately available in layouts, and execute during the review process. For more information, contact Relativity Customer Support.

1.2.3 Use unique names for choices

Ensure that the choices and the sub-choices associated with each choice that are added to your workspace all have unique names. If your workspace contains choices or sub-choices associated with each choice with identical names, errors occur when you attempt to build an event handler referencing these choices.

For example, creating a single choice field that has two "Yes" sub-choices can cause errors to occur in Event Handler Express.

1.2.4 Avoid changing field names

Do not change the names of fields referenced by an event handler. Event Handler Express does not update any changes to the field names referenced by an event handler already added to a workspace. If you modify the field name, the event handler that referenced the field no longer exists in the workspace.

Note: Event Handler Express references the Artifact Name for a layout, field, choice, or group. It doesn't use custom labels assigned to fields or groups.

1.2.5 Keep event handlers in their original workspace

The Artifact Names and IDs that event handlers rely on are specific to an individual workspace. We do not support exporting and importing event handlers from one workspace to another. Doing this may cause performance issues.

Instead of moving event handlers, install Event Handler Express in the new workspace and re-create them there.

1.2.6 Avoid using unicode characters

Event Handler Express doesn't support the use of unicode characters for event handler names.

1.2.7 Attach event handlers to documents or RDOs

Event Handler Express supports attaching event handlers to document objects and RDOs. For an object with multiple event handlers, the event handlers execute in the order that you added them to the object. We don't support attaching event handlers to system objects.

Note: The Pre Save event handler doesn't execute when you create a new instance of an RDO. It only executes when you edit an RDO instance.

1.2.8 Deploy event handlers during off hours

Deploy your event handlers during off hours to minimize disruptions to your production environment.

1.2.9 Improve performance

To improve performance, create multiple rules within one event handler rather than creating multiple event handlers.

2 Using Event Handler Express

You create a new event handler entering or selecting basic information, such as a name, object type, description, and layouts that trigger it. To each event handler, you can add one or more condition groups, which contain individual conditions used for field validation. You can then deploy your new event handler for execution on layouts in the current workspace.

Note: Event handlers created by Event Handler Express cannot be added to custom applications because the resource files are tied to the "Default" application.

2.1 Compatibility matrix

Use the following matrix to determine the appropriate version of Event Handler Express for your Relativity environment.

Event Handler Express version	Supported Relativity version
Event Handler Express 4.2.1.1	Server 2022 and forward, RelativityOne
Event Handler Express 3.0.5.1	Server 2021

2.2 Deploying and configuring Event Handler Express

Note: If you are a current RelativityOne user, and you want to install or upgrade this application, you must be a System Admin or have the proper security permissions. Contact the Support team with any issues.

To deploy this solution, you start by adding it to the Application Library as a Relativity application. How you deploy and configure it depends on whether you are installing the solution for the first time or upgrading an earlier version of the solution. Follow the appropriate set of steps below to install or upgrade the solution in your Relativity instance.

2.2.1 Installing the solution for the first time

To begin using this solution, download and add it to the Application Library as a Relativity application.

To download the application:

1. Log in to the Relativity Community and click on the **Relativity Applications** folder.

2. Locate the newest Event Handler Express .rap file that is compatible with your Relativity version and select it.

    Note: The solutions in the Relativity Applications folder are sorted from most recently modified to oldest. As a result, solutions at the top of the page are normally the newest versions of the respective solutions. If you see more than one version of the solution, select the .rap with the higher version number to ensure you're deploying the most up-to-date solution.

3. Click **Download**.

To add the solution to the Application Library:

1. Navigate to the **Application Library** tab.

2. Click **Upload Application**.

3. Click **Select File**.

4. Navigate to and select the application file included in the solution, and then click **Open**.

5. Click **Save** to upload the file to the Application Library.

To add the solution to a workspace:

1. In the Workspaces Installed section, click **Select** to install the application to one or more workspaces.

2. Select the workspaces where you want to install the application, and then click the Move selected left to right icon.

3. Click **Apply**.
   The application is installed to the selected workspaces. A list of workspaces where the application has been installed displays in the Workspaces Installed section.

2.2.2 Upgrading an earlier version of the solution

To upgrade from an earlier version:

1. Navigate to the **Application Library** tab.

2. Click the name of the **Event Handler Express** application.

3. Click **Edit**.

4. Click **Clear** to clear the application file for the earlier version of the solution.

5. Click **Select File**, select the **Event Handler Express** file, and then click **Open**.

6. Click **Save**.

7. If you receive a message asking if you want to update workspaces automatically, click **Yes**.

8. Using the **Workspaces Installed** section of the application page, ensure that the upgrade is applied to every workspace where the application has already been installed.

2.3 Prerequisites for Event Handler Express

You must complete the following prerequisites before using Event Handler Express.

2.3.1 Setting security permissions

After installing Event Handler Express, you can provide advanced users with access to the application by using the default security settings on its components. To deny access to other users, set the following permissions on its components:

- **Workspace-level security** - In the Workspace Security window, update items on the following tabs:
  - **Tab Visibility tab** - locate **Relativity Event Handler Express** in the list. Click **Deselect all (2)** to remove view permissions on the **Relativity Event Handler Express** and **Event Handler Express** tabs. For more information, see Workspace Security in the Relativity Documentation site.
  - **Object Security tab** - deny access to the **Event Handler Express** object by setting its permissions to **None**.

- **Object-level security** - In the workspace where you deployed Event Handler Express, navigate to the following tabs and remove the groups don't require access to these objects:
  - **Object Type tab** - Select the **All Object Types** view, and filter on **Event Handler Express** in the **Name** column. Remove one or more groups from the **Event Handler Express** object type. For more information, see Relativity object security on the Relativity Documentation site.

- **View tab** - Select **All Views**, and filter on **Event Handler Express** in the **Name** column. Remove one or more groups from the following views:
    - All Relativity Event Handler Express
    - Relativity Event Handler Express - All Fields
    - Relativity Event Handler Express - All Layouts
    - Relativity Event Handler Express - All Objects

Note: If you send an email with a link to Event Handler Express, the recipient needs the required permissions to access the application. Copying and pasting the link in the browser after logging in to Relativity doesn't automatically provide the recipient with access to Event Handler Express.

2.3.2 Making design decisions

Before you begin building your event handlers, consider the following design decisions:

- Identify the conditions that you want to include in your event handler. In general, these conditions determine the coding rules that Relativity applies when a reviewer makes a selection in a layout.

- List the layouts, fields, and choices that you want to reference when building your event handler and adding conditions. For additional information, see Special considerations for building event handlers.

2.4 Defining basic information for an event handler

Use the following procedure to add basic information for an event handler:

1. In Relativity, navigate to the workspace where Event Handler Express is installed.

2. Click the **Event Handler Express** tab.

    Note: You can use the **Delete** mass operation to remove an event handler from the list view, and the **Export to File** operation to download a copy of it. The **Copy** or **Edit** mass operations aren't available on this view. You must manually create a new event handler rather than copying and modifying an existing one. For more information, see Editing an event handler.

3. Click **New Event Handler Express**.

4. Complete the fields in the Event Handler Express layout. See Fields for an event handler express.

5. Perform one or more of the following tasks:

    - Add validation logic to your event handler. See Building condition groups and conditions.

    - To save your changes to the form, click **Save**. You can save your updates to the event handler and continue editing it later. However, your changes aren't deployed to the workspace.

    - To remove your updates to the form, click **Clear**. You can now start creating a new event handler. Event Handler Express deletes all the entries in the Basic Information and Condition Group sections, and displays a blank form. If you want to remove a specific condition group or condition, use 🗑 to remove the appropriate item. See Building condition groups and conditions.

2.4.1 Fields for Event Handler Express

The Event Handler Express layout contains the following fields in the Basic Information section:

- **Name**—the name of the .dll file. Ensure that you enter a name, which is unique to your Relativity environment. On the Event Handler Express tab, the item list displays this name appended with additional text used to identify the .dll for your event handler. For example, if you entered MyEventHandler in the Name field then the item list displays your event handler as MyEventHandler.REHEPreSave.

  Note: Don't use special characters, wildcards, or spaces in the Name field. In addition, don't use a number as the first character in your event handler name.

- **Object Type**—click to select a Relativity object, and then click **Set**. Event Handler Express supports attaching event handlers to document objects and RDOs. For example, you would select the document object if you want the event handler to execute when a reviewer selects choices on a layout during a review. We don't support attaching event handlers to system objects.

- **Layouts**—click to select one or more layouts that you want to trigger this new event handler. These layouts must be associated with a document object or an RDO. When a user performs a task on one of these layouts, it raises an event that triggers your event handler to validate the user's responses. If you don't select an option, the event handler runs across all layouts associated with the object type. For more information, see Executing an event handler across all layouts.

  Note: Ensure that you select layouts that contain the fields used in the conditions defined for the event handler. Relativity displays an error message when an event handler attempts to execute on a layout that doesn't contain the fields included in its conditions. Additionally, if you want to deploy all layouts, then each layout must include the fields used in the conditions.

- **Description**—information about the event handler. This information appears in the Description column of the Event Handler Express tab. It also appears in the Event Handlers section of the object type details view. The

source code for the event handler includes the description as a code comment. The maximum number of characters for this field is 100.For more information, see Editing Relativity objects on the Relativity Documentation site.

2.5 Building condition groups and conditions

In Event Handler Express, you can build a series of validation rules by adding condition groups and conditions. A condition group contains one or more conditions. When you add multiple condition groups, Event Handler Express evaluates them as a series of expressions connected by the AND operator. You can choose to evaluate a series of conditions within a condition group with the AND or OR operators. Event Handler Express initially displays Condition Group - 1 and Condition - 1, so you can begin defining your validation rules.

After you add the required conditions, you can deploy the event handler in the current workspace. Event Handler Express performs the following operations:

- It compiles a .dll.

- It adds the .dll file to Relativity as a resource file.

- It attaches it to an object in the workspace where you created the event handler.

If the .dll handler already exists in the workspace, Event Handler Express overrides the existing file, and attaches it to the object if necessary. If an event handler with the same name already exists, Event Handler Express displays an error message asking you to rename the new event handler.

Use the following procedure to build condition groups and add conditions:

1. Click **Add Condition Group** at the top of the page. A condition group appears below the Basic Information section.

2. Complete the fields for a condition group. See Fields for condition groups and conditions.

3. Click ⊕ to create a condition.

4. Complete the fields for a condition. See Fields for condition groups and conditions on the next page.

5. To add multiple conditions to a condition group, click ⊕ for each new condition that you need.

6. Repeat steps 1 through 5 until you have added all the condition groups and conditions required for the event handler.

   Note: To copy an existing condition group or condition, click ⧉. You also click on a condition's name to hide or show its details.
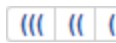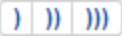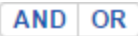
7. Perform one or more of the following tasks:

8. To associate your event handler with one or more layouts, click **Deploy Event Handler** in the Event Handler console. Your event handler now executes against layouts associated with it in your workspace.

- To save your changes to the form, click **Save**. You can save your updates to the event handler and continue editing it later, but your changes aren't deployed to the workspace.

- To remove your updates to the form, click **Clear**. Event Handler Express deletes all the entries in the Basic Information and Condition Group sections, and displays a blank form. If you want to remove a specific condition group or condition, use 🗑 to remove the appropriate item. See Building condition groups and conditions.

2.5.1 Fields for condition groups and conditions

A condition group contains the following fields:

- **Description**—information about the validation that this condition group performs. If you enter double quotation marks (") in this field, Event Handler Express removes them from the text.

- ((((((- click the option with the appropriate number of parentheses to group your conditions.

- **Validation Type**—select the **Field** or **User** as the type that you want to validate. Supported field types include Current User, Fixed-length Text, Single Choice, Multiple Choice, and Yes/No fields. For the Current User field, Event Handler Express only validates conditions for the user currently logged in to Relativity.

- **Field/Group**—the name of the field that you want to validate. If you set the Validation Type to Group, then you don't need to select an option for this field.

- **Condition**—select an operator that you want to use for validation. The field type that you selected in the **Validation Type** drop-down box must support the operator. See Supported conditions by field type.

- **Choice/Value**—select a choice that you want to use for validating your condition.

- `) )) )))` —click the option with the appropriate number of parentheses to group your conditions.

- `AND OR` —click the AND or OR operator to determine if both or only one condition needs to be met respectively. The operator that you selected appears in a red font.

- **Message**—an error message that users see when they have met all conditions in the condition group. You can add your message after you have added conditions. The maximum number of characters for this field is 255.

2.6 Supported conditions by field type

When building a condition, you must use only the operators supported by the field type selected in the Validation Type drop-down box. The following table displays ✓ for each operator supported by a specific field type.

Operator	Current User	Fixed Length Text	Multiple Choice	Single Choice	Yes/No
Is				✓	✓
Is Not				✓	
Is Set		✓	✓	✓	
Is Not Set		✓	✓	✓	✓
Contains			✓		
Has Changed Do Not Include Initially Set		✓	✓	✓	
Has Changed Include Initially Set		✓	✓	✓	
Is In Groups	✓				
Is Not In Groups	✓				

2.6.1 Additional information about operators

Event Handler Express provides operators that you can use to control whether an event handler fires on a layout that doesn't contain any initial values. In general, you create a new layout in a workspace and then deploy the event handler that you want associated with it. The fields on the layout won't contain any values when a user first begins to code with it. You may not want your event handler to fire when initial updates are made. You could use the operator called **Has Changed Do Not Include Initially Set** to add a condition on the event handler. When you set this condition, the event handler executes as follows:

- When the user sets the initial values on a layout, the event handler doesn't fire.

- When the user sets or updates any subsequent values, the event handler fires.

For example, the layout contains a required Responsiveness field that isn't set when the user initially accesses the layout. You wouldn't want the event handler to fire and display an error message the first time that a user updates the field, but only on subsequent changes.

In addition, you may want to add a condition that ensures the event handler fires when the user changes an existing value on a layout. You could use the **Has Changed Include Initially Set** operator to add this condition. When you set this condition, the event handler executes as follows:

- When the user sets the initial values on a layout, the event handler fires.
- When the user sets or updates any subsequent values, the event handler fires.

For example, you may want the event handler to fire only after the Responsiveness field on a layout has been updated to a new value. When a user changes the value from Not Responsive to Responsive, or from unset to Responsive, you may want to execute the event handler to ensure that the Issue Designation field is also set.

2.7 Editing or deleting an event handler

You can edit and delete the event handlers that you create through Event Handler Express. This section explains how editing or deleting an event handler affects your Relativity environment.

2.7.1 Editing event handlers

You can update an event handler by clicking the **Edit** link appearing next to its name on the Event Handler Express tab. When you click this link, Event Handler Express displays the basic information and condition groups used to create the event handler. You can update these sections as necessary.

Deploying your updates affects all workspaces in your environment where you have added the event handler. Event Handler Express recompiles the .dll file and overwrites the existing version. All copies of the event handler in your environment now reference the updated file. Consequently, any changes made to this file immediately modify the behavior of the event handler throughout your Relativity environment.

2.7.2 Deleting event handlers

When you deploy an event handler, Event Handler Express creates a resource file of type .dll that contains your source code. It attaches this .dll file to the object that you selected when you created the event handler. You need to remove the event handler from the Event Handler Express and the object associated with it. Additionally, you need to remove its resource file.

To delete an event handler, complete the following steps:

1. Remove the event handler from Event Handler Express. You can use the **Delete** mass operation on the Event Handler Express tab to remove the event handler from this list view. You can't edit the source code for the event handler after you delete it from this list.

2. Remove the event handler from the object that the event handler was created on. For example, you may have created the event handler on the Document object, so you now need to remove it from this object. For more information, see Creating and editing Relativity Objects on Relativity Documentation site.

3. Remove the resource file created for the event handler. You can completely remove the event handler from your Relativity environment by deleting its resource file. For more information, see Resource files on Relativity Documentation site.

Note: Deleting the resource file affects all workspaces in your environment, where you may have added the event handler. The functionality provided by the deleted event handler isn't available in any workspace after this action.
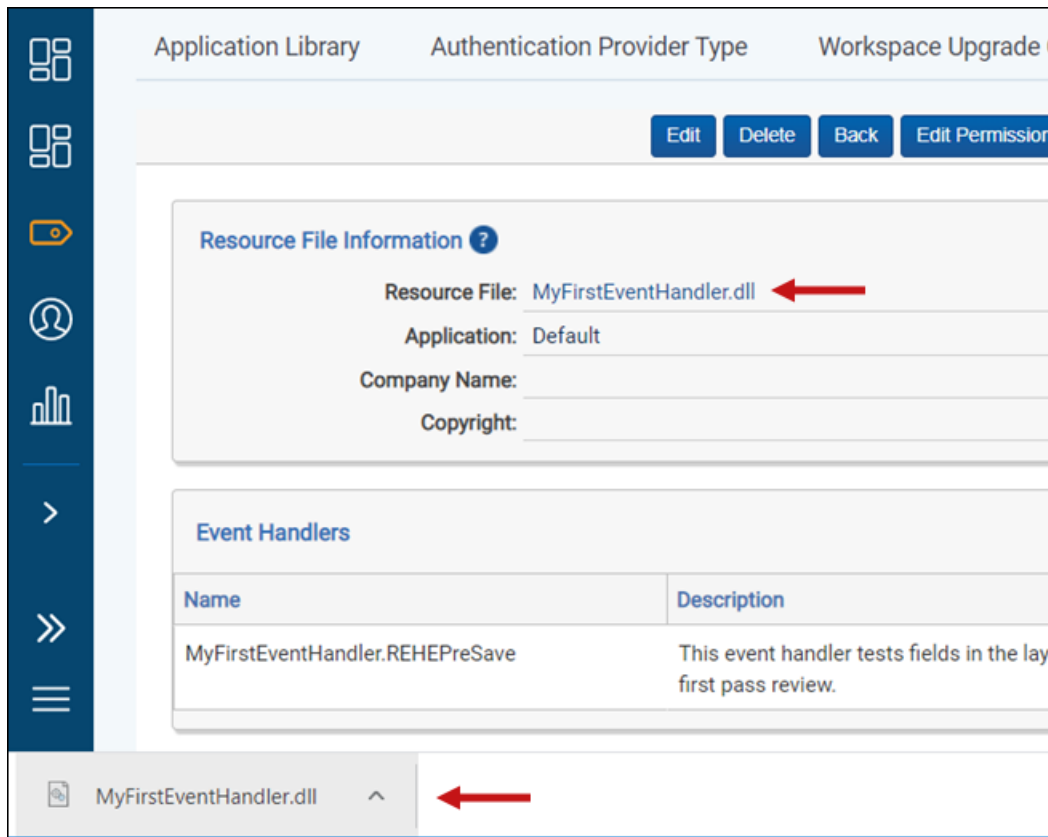
2.8 Uploading an event handler to another Relativity instance

You can reuse your custom event handlers built with Event Handler Express by uploading them to other Relativity instances. When you associate the event handler with an object type, ensure that the workspace contains layouts, objects, fields, and choices with the same names as those used to create the event handler.

> **Note:** If you are a current RelativityOne user, and you want to install or upgrade this application, you must contact the Customer Support team.

Use the following procedure to upload an event handler to another environment:

1. From Home, click the **Applications & Scripts** > **Resource Files** tabs.

2. On the Resource Files tab, locate your event handler and double-click on its name.

3. On the details view, double-click on your event handler's name in the **Resource Files** field. After your browser downloads the .dll file, save it to a location of your choice.

4. Log in to another Relativity instance.

5. On the Resource Files tab, upload the .dll for your event handler. For more information, see Resource files on Relativity Documentation site.

6. Navigate to a workspace where you want to add the event handler.

7. Click the **Workspace Admin** > **Object Type** tabs.

8. On the Object Type tab, locate the object type that you want to associate with the event handler. Event Handler Express supports attaching event handlers to document objects and RDOs.

9. Click on the name of object type to display its detail view.

10. In the Event Handlers section, click **New** to display the Select Event Handler dialog.
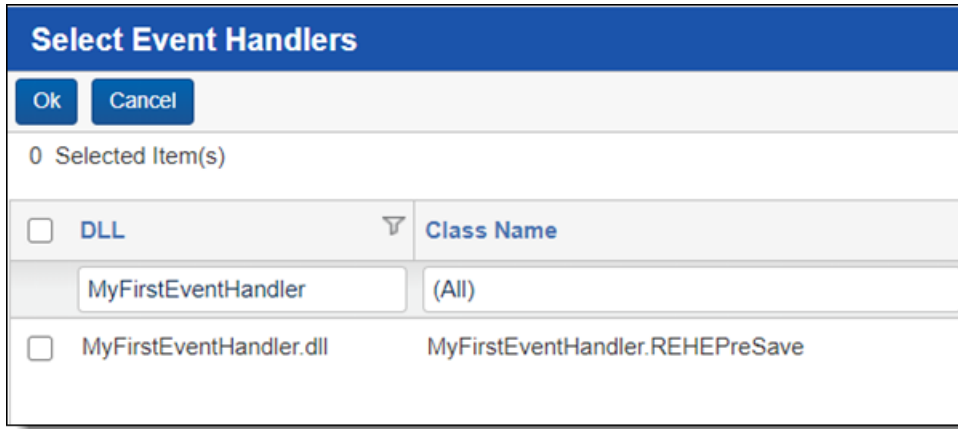
11. Select your event handler .dll, and click **OK** to add it to the Event Handlers section of the object type. The event handler now executes when a user performs a task on the layouts selected when you initially built the event
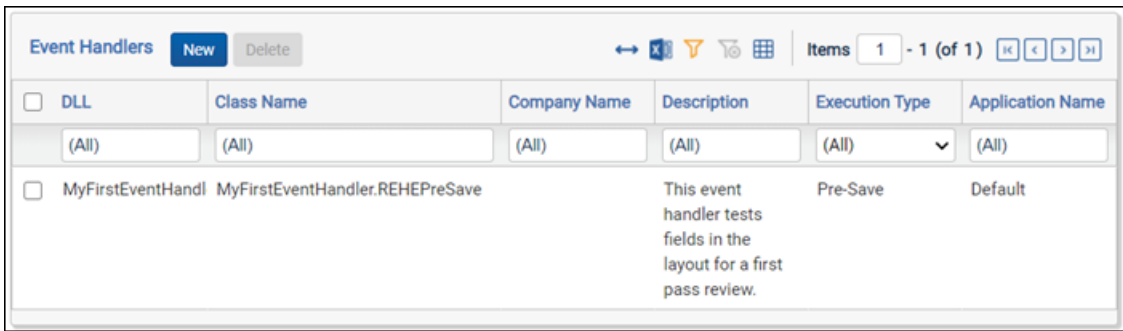
handler.

2.9 Special considerations for building event handlers

Review these guidelines to ensure that your event handlers function properly after you deploy them to a workspace.

2.9.1 Executing an event handler across all layouts

Your event handler executes across all layouts if you don't select one or more items in the Layouts option when building the event handler. If you want your event handler to run across all layouts in a workspace, ensure that they contain the fields required for the event handle to execute properly. Relativity displays an error message when an event handler attempts to execute on a layout that doesn't contain a required field. Any field that you add to a condition is a required field. See Defining basic information for an event handler.

2.9.2 Backslashes in field and choice names

When adding conditions, ensure that the names of choices and fields don't contain a character string containing a backslash preceded and followed by a space. For example, you don't want to use a multiple choice field in a condition that contains a choice with the following name:

Choice B \ 1

However, you can use choices and fields that contain a backslash preceded and followed by text. See the following choice name:

Choice B\1

2.9.3 Wildcards in field and choice names

When adding single or multiple choice fields to a condition group, you may select fields or the choices with names that contain wildcards. Ensure that fields or choices have unique names if you omitted the wildcard characters.

Event Handler Express ignores the wildcards in the field and choice names when it compiles the .dll. The generation of the .dll fails when multiple choices have the same name. For example, you could have a multiple choice field called Choice B with the following sub-choices:

- Choice A
- Choice B
    - Choice B\1
    - Choice B \ 1
    - Choice B \\ 1

Because Event Handler Express ignores the wildcard characters, it sees the same name for three sub-choices, causing the compiling event to fail.

2.10 Troubleshooting your event handler

You can use the following list to troubleshoot your event handlers and resolve common errors that prevent them from executing:

- Verify that you have associated the event handler with the correct object type.
- Ensure that the workspace where you want to use the event handler contains all the objects that it requires. These objects include layouts, fields, or choices.
- If you experience errors after adding your event handler to different workspace, compare the names of layouts, fields, and choices added to Event Handler Express with those used in the workspace where you created the event handler.
- Ensure that the conjunction operators are properly set on your event handler.

3 Building your first event handler

You can use this tutorial to learn about how to build a simple event handler with Event Handler Express. It describes how to create an event handler that checks whether the user has selected the Responsive choice on the Responsiveness field. If the user made this selection, then the event handler verifies that the user also selected at least one choice on the Issue Designation field. It displays an error message if the user hasn't made a selection for the Issue Designation field.

The tutorial also illustrates how to deploy your new event handler to the current workspace. This process makes it immediately available on its associated layout. For general information about building event handlers, see Using Event Handler Express on page 7.

3.1 Before you begin

Complete the following tasks before building your event handler:

- Obtain access to an instance of Relativity that has Event Handler Express installed. In Relativity, confirm that you have the correct permissions. For more information on event handler permissions, see Setting security permissions on page 8.

- For testing purposes, identify a workspace where you can execute your new event handler. You can create a new workspace or use an existing one. Ensure that it contains some sample documents, and add the following objects to it. You reference these objects when you build your event handler. For information about working with Relativity, see the Relativity Documentation site.

Object Type	Name	Comments
Field	Responsiveness	Create a Single Choice field on the document object.
Field	Issue Designation	Create a Multiple Choice field on the document object.
Choice	Responsive	Select Responsiveness in the Field option.
Choice	Not Responsive	Select Responsiveness in the Field option.
Choice	Not Relevant	Select Issue Designation in the Field option.
Choice	Technical Issue	Select Issue Designation in the Field option.
Layout	First Pass Review	Select Document as the Object Type. Add the Responsiveness and Issue Designation fields to the layout.

The following screen shot illustrates how the First Pass Review layout appears after you add the Responsiveness and Issue Designation fields to it.

3.2 Build an event handler
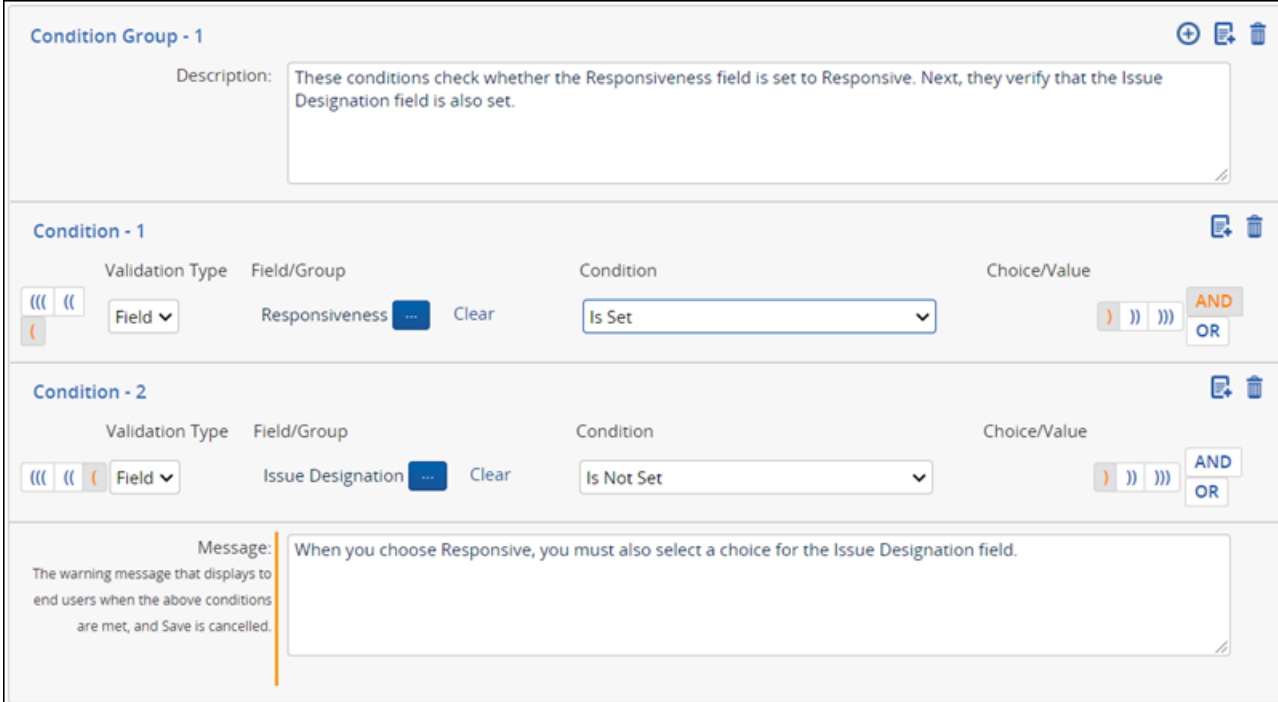
Use the following procedure to build an event handler with Event Handler Express:

1. In Relativity, navigate to the workspace where Event Handler Express is installed.

2. Click the **Event Handler Express** tab.

3. Click **New Event Handler Express**.

4. In the Basic Information section, enter the following information in these fields:

   - **Event Handler Name**—MyFirstEventHandler

   - **Object Type**—click [...] to display the Select Item dialog. Select the **Document** object, and then click **Set**.

   - **Layouts**—click [...] to display the Select Item dialog. Select the **First Pass Review** layout, and then click **Add** > **Set**.

- **Description**—This event handler tests fields in the layout for a first pass review.

5. In **Condition Group - 1**, enter the following text in the **Description** box: These conditions check whether the Responsiveness field is set to Responsive. Next, they verify that the Issue Designation field is also set.

6. To define Condition - 1, complete these steps:

   a. Click [( ] to select the left parenthesis.

   b. Select **Field** as the **Validation Type**.

   c. To set the **Field/Group**, click [...] to display the Select Item dialog. Filter on **Responsiveness** in the **Name** column, select this choice, and click **Set**.

   d. In the **Condition** drop-down menu, select **Is** as the operator.

   e. To set the **Field/Group**, click [...] to display the Select Item dialog. Select **Responsive**, and click **Set**.

   f. Click [ )] to select the right parenthesis.

7. Click **AND** to link Condition - 1 with the AND operator to Condition - 2, which you add in the following steps.

8. Click ⊕ to create a new condition. Repeat this process when you need to add another condition.

    a. Click (to select the left parenthesis.

    b. Select **Field** as the **Validation Type**.

    c. To set the **Field/Group**, click ⋯ to display the Select Item dialog. Filter on **Issue Designation** in the **Name** column, select this field, and click **Set**.

    d. In the **Condition** drop-down menu, select **Is Not Set** as the operator.

    e. Click) to select the right parenthesis.

9. In the **Message** box, enter the following: When you choose Responsive, you must also select a choice for the Issue Designation field.

10. Click **Save**.

11. Click **Deploy Event Handler** in the Event Handler console. Your event handler now executes against the First Pass Review layout associated with it in your workspace.
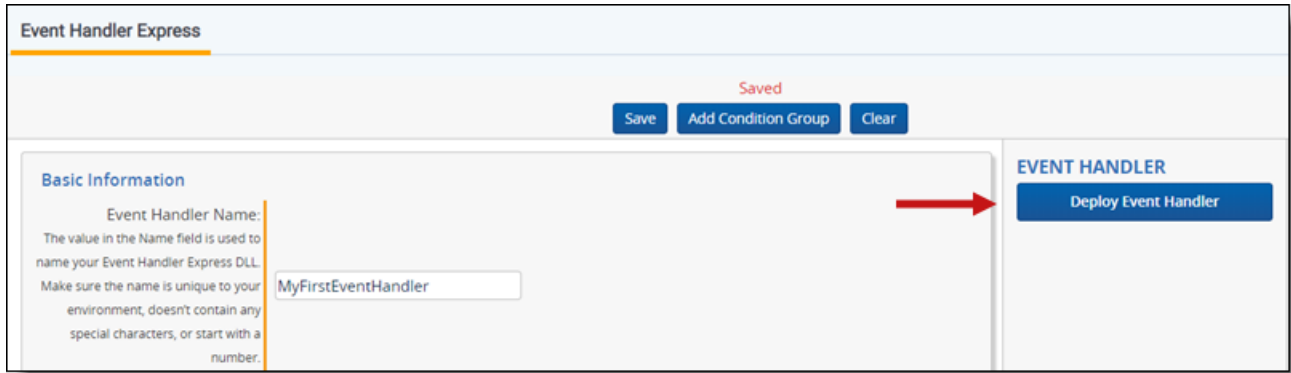
3.3 Executing your event handler in a layout

You can test your event handler by coding documents in the viewer in the workspace where you deployed the event handler.

1. In the current workspace, open a document in the viewer.

2. Select the **First Pass Review** layout and click **Edit**
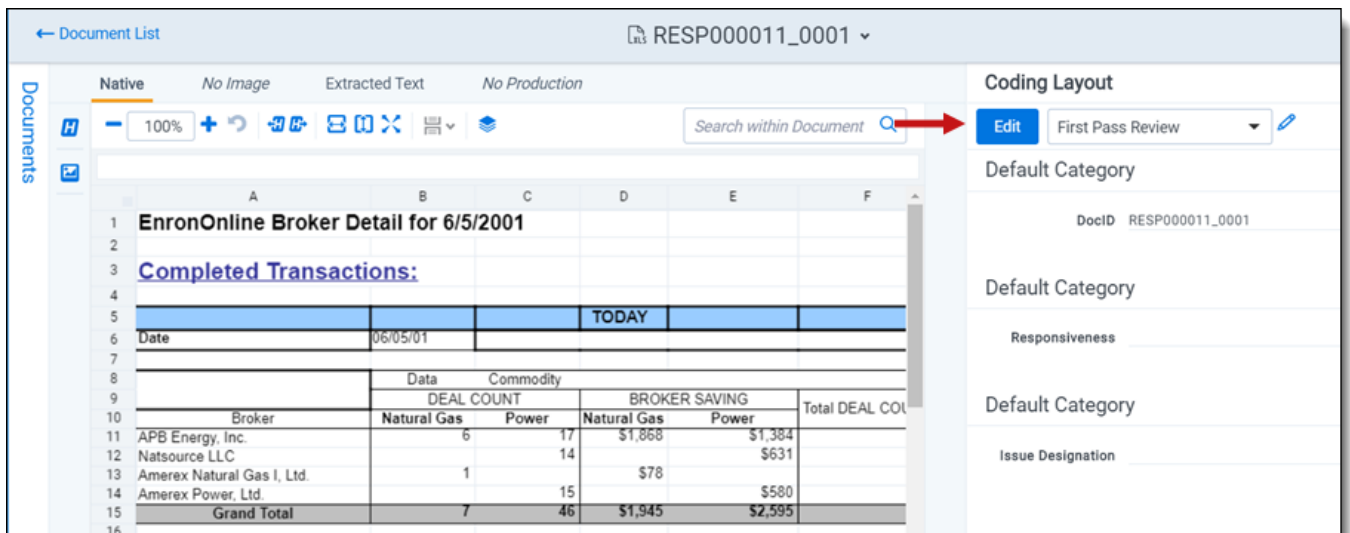
3. To test your event handler, select **Responsive** and click **Save**. Your error message should appear when you haven't selected a choice for the Issue Designation field.