# **Coding Overturn**

June 2017 – Version 5.0

## **Contents**

| 1. Coding Overturn                                   | 3 |
|--|---|
| 1.1 Supported Versions                               | 3 |
| 1.2 Category   | 3 |
| 1.3 Permissions                                      | 3 |
| 1.4 Special Considerations                           | 3 |
| 2. Deployment  | 3 |
| 2.1 Installing the Application into your Environment | 4 |
| 2.2 Installing the Application into Workspaces       | 4 |
| 3. Input and Preparation                             | 4 |
| 3.1 To add fields to monitor via Relativity Script   | 4 |
| 3.2 View Coding Overturn report                      | 5 |
| 4. Results of running                                | 6 |
| 4.1 Returned Output                                  | 6 |
| 5 Disclaimer   | 7 |

## 1. Coding Overturn

The Coding Overturn solution is a workspace level solution that provides a report on changes made to a coding decision after second level review. The report displays the original field value, the new value, the user who made the changes, their group access and the timestamp for all documents.

#### 1.1 Supported Versions

This solution is supported in Relativity versions 9.4.254.2 and higher.

#### 1.2 Category

This custom solution consists of the following components:

- Relativity scripts that run at the workspace level
- Event Handler

#### 1.3 Permissions

• Permissions for this solution have not been implemented. Users interested in using this application will need to configure permissions manually.

#### 1.4 Special Considerations

Before you deploy and run the solution, it's important to keep the following in mind: m

- Fields with special characters other than [] / . , ; & () : cannot be added to the fields to monitor.
- Coding decisions made before the solution is setup will not be included in the script results.
- Data from fields which are removed from the fields to monitor will not be included in the script results from the time they are removed.
- You cannot select the same group for the 1st Level Review Group and 2nd Level Review Group inputs for the Coding Overturn Setup script.
- You cannot select multiple/duplicate combinations of review groups for the selected field for the Coding Overturn Setup script.
- Only single-choice, multi-choice, and Yes/No fields can be monitored using this solution.
- Only actions completed using the Save or Save & Next button will be tracked.
- This solution will not work with previous versions as the table names have been altered in an effort to make this solution more generic and available as an open source project.

## 2. Deployment

To deploy and configure the solution, you must first add it to the Application Library as a Relativity application. You can then install and configure the solution in a workspace.

#### 2.1 Installing the Application into your Environment

- 1. Log in to Relativity.
- 2. Click the user drop-down menu in the upper-right corner of Relativity, then click **Home**.
- 3. Click the **Applications & Scripts** tab, then click the **Application Library** tab.
- 4. Click Upload Application.
- 5. Click Browse, navigate to and select the CodingOverturn.rap file, then click Open.
- 6. Click Save.

#### 2.2 Installing the Application into Workspaces

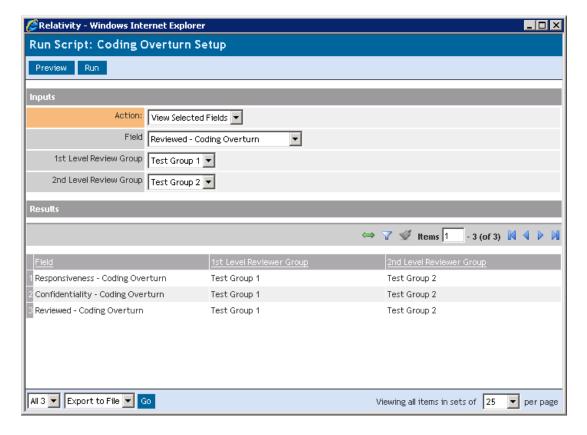
- 1. Click the user drop-down menu in the upper-right corner of Relativity, then click Home.
- 2. Click the Applications & Scripts tab, then click the Application Library tab.
- 3. Click the name of the **Coding Overturn** application.
- 4. Under Workspaces Installed, click Install.
- 5. Click the ellipsis (..) next to Workspaces.
- 6. Select the destination workspace(s) and click **Ok**.
- 7. Click Save.

## 3. Input and Preparation

#### 3.1 To add fields to monitor via Relativity Script

- 1. Go to the workspace Scripts tab in your desired workspace
- 2. Run the **Coding Overturn Setup** script
- 3. Select **View Currently Selected Fields** in the **Action** drop-down to view all fields and Review Groups which will be monitored by the event handler
- 4. Select **Add** in the **Action** drop-down along with a value in **Field, 1st Level Review Group** and **2nd Level Review Group** drop-downs to add a field to be monitored by the event handler
- 5. Select **Remove** in the **Action** drop-down along with a value in the **Field, 1st level Review Group** and **2nd Level Review Group** drop-downs to remove a field to be monitored by the event handler.

The Responsiveness-Coding Overturn, Confidentiality-Coding Overturn, Reviewed-Coding Overturn fields as well as associated Review Groups have been added in the image below.

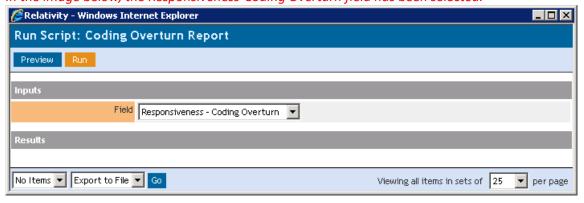


After the fields have been added, the solution will start monitoring the coding decisions made to the fields added using the Coding Overturn Setup script. Please note that the solution uses an event handler, so the fields need to be added to the Coding Overturn Setup before review begins. The solution does not account for historical coding decisions.

#### 3.2 View Coding Overturn report

- 1. In order to get a report on the coding overturn decisions made, run the Coding Overturn Report script
- 2. Go to the workspace Scripts tab in your desired workspace
- 3. Select the Coding Overturn Report script
- 4. Select the field that you would like to report on and click **Run**
- 5. The script displays one row per document per instance of a code change made.

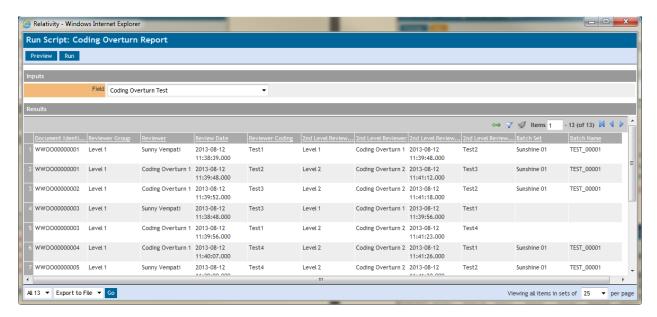
In the image below, the Responsiveness-Coding Overturn field has been selected.



## 4. Results of running

The script displays 1 row per document per instance of a code change made by a user. This could be the same user or different users as well as the same group or different group as long as the user is a member of one of the review groups identified by the **Coding Overturn Setup** script.

An example of the results from running the Coding Overturn Report is displayed below.



### **4.1 Returned Output**

The following columns will display in the result window:

| Column Heading                        | Description   |
|---------------------------------------|---|
| Document Identifier                   | The unique Document Identifier  |
| Reviewer Group                        | Name of the first group that made coding decision on selected field                                     |
| Reviewer                              | Name of the reviewer in the first group that made the coding decision on the selected field             |
| Review Date                           | The date the Reviewer made the coding decision  |
| Reviewer Coding                       | Value of the coding decision for the selected field   |
| 2 <sup>nd</sup> Level Reviewer Group  | Name of the second group that changed coding decision on selected field                                 |
| Reviewer 1                            | Name of the reviewer in the second group that made changed the coding decision on the selected field    |
| 2 <sup>nd</sup> Level Review Date     | The date the second Reviewer changed the coding decision  |
| 2 <sup>nd</sup> Level Reviewer Coding | Value of the changed coding decision for the selected field   |
| Batch Set                             | The batch set the document belongs to. This value is null if document does not belong to any batch set. |

| Column Heading | Description  |
|----------------|--|
| Batch Name     | The batch name the document belongs to. This value is null if document does not belong to any batch set. |

## 5. Disclaimer

This solution is intended for use only in the Relativity versions specified in this document and run under the guidelines presented. While each solution is carefully built and thoroughly tested to work on the versions of Relativity specified in this document, this script is not a core feature of Relativity and is not eligible for the same level of support as the Relativity platform.

In addition, custom components may not exhibit the same performance and behavior as native Relativity features. Custom solutions do not specify permission settings unless explicitly requested by the client.