# Table of Contents

1 Get started with logging .................................................................................................................. 7
   1.1 When to use logging .................................................................................................................. 7
   1.2 Where the logs are stored ......................................................................................................... 8
   1.3 Reading a log entry .................................................................................................................. 10
   1.4 Next steps ................................................................................................................................ 12

2 Configure logging ............................................................................................................................ 13
   2.1 Configuration defaults ............................................................................................................. 13
   2.2 EDDSLogging database .......................................................................................................... 14
      2.2.1 Tables ............................................................................................................................... 14
      2.2.2 Views ............................................................................................................................... 14
      2.2.3 Stored procedures ............................................................................................................ 15
   2.3 Configurations .......................................................................................................................... 16
   2.4 Sinks ......................................................................................................................................... 17
      2.4.1 File ................................................................................................................................... 18
      2.4.2 SQL Server ...................................................................................................................... 18
   2.5 Rules ......................................................................................................................................... 18
      2.5.1 Rules and sinks .................................................................................................................. 21
   2.6 Config files .................................................................................................................................. 21
   2.7 Troubleshoot logging ................................................................................................................. 24

3 Troubleshoot Relativity using logging ............................................................................................ 26
   3.1 Set up a logging rule ................................................................................................................. 26
      3.1.1 Order ............................................................................................................................... 26
      3.1.2 Machine name ................................................................................................................... 26
      3.1.3 System ............................................................................................................................. 26
      3.1.4 Subsystem ......................................................................................................................... 26
      3.1.5 Application ....................................................................................................................... 27
      3.1.6 LoggingLevel .................................................................................................................... 27
      3.1.7 Rule sinks .......................................................................................................................... 27
   3.2 Troubleshooting example .......................................................................................................... 27
4 Logging system-subsystem-application matrix .............................................................29
  4.1 Relativity systems .................................................................................................29
  4.2 Subsystems .............................................................................................................30
  4.3 Relativity applications ..........................................................................................31
5 Infrastructure considerations for logging .................................................................32
  5.1 File sink ..................................................................................................................32
  5.2 Storing database passwords .................................................................................32
  5.3 Moving EDDSLogging database .........................................................................32
1 Get started with logging

The logging framework enables you to gather runtime diagnostic information. Use it for troubleshooting application problems when you need a granular level of detail. For example, you may be asked for Relativity logs when working with a Relativity Support representative.

Using logging scenario

You experience problems with Relativity and you have contacted Relativity Support. The Support specialist first requests to look at the Errors tab. Based on the information on the Error tab, they determine that the Relativity component causing the error is an event handler. It only occurs on the rel-server-2 machine in your Relativity environment. Next, the support specialist requests that you create a logging rule to set the logging level to Debug (2) for event handlers on rel-server-2. Additional logging data is accumulated and passed to Relativity for further investigation. For more information about this scenario, see Troubleshoot Relativity using logging on page 26.

1.1 When to use logging

Relativity provides different mechanisms for diagnostic and historic information. Note the following when deciding whether to use logging, audit data, or the Errors tab:

- Logging is intended for troubleshooting and debugging. Clean up your log files as necessary.
- Audit is for defensible history that is never deleted.
- The Errors tab also contains the Error level events. Relativity logs may contain more data logged at
the Error level than is available on the Errors tab.

1.2 Where the logs are stored

By default, Relativity logs are stored in the RelativityLogs table of the EDDSLogging database. You can access the table with a database client of your choice.
Note: Make sure to use database credentials with permissions to the EDDSLogging database.

Depending on your Relativity configuration, logs can also be stored as files, in the Data Grid, or in a different SQL Server.

The following image shows the Relativity logs written out as files.
The log files directory path is defined by the RELATIVITY_LOGS environment variable. To quickly find the log files location, run this command in Windows Command Prompt:

```bash
echo %RELATIVITY_LOGS%
```

Relativity logging data destinations are referred to as sinks. For more information about sink configuration, see Configure logging on page 13.

### 1.3 Reading a log entry

By default, Relativity logs events at the Error (4) level. Event level are defined as follows:

<table>
<thead>
<tr>
<th>Level</th>
<th>Database ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verbose</td>
<td>0</td>
<td>Include all available event details.</td>
</tr>
<tr>
<td>Debug</td>
<td>1</td>
<td>Internal control flow and diagnostic state dumps to facilitate pinpointing of recognized problems</td>
</tr>
<tr>
<td>Information</td>
<td>2</td>
<td>Events of interest or that have relevance to outside observers; the default enabled minimum logging level</td>
</tr>
<tr>
<td>Warning</td>
<td>3</td>
<td>Indicators of possible issues or service/functionality degradation</td>
</tr>
<tr>
<td>Error</td>
<td>4</td>
<td>Indicating a failure within the application or connected system</td>
</tr>
<tr>
<td>Fatal</td>
<td>5</td>
<td>Critical errors causing complete failure of the application</td>
</tr>
</tbody>
</table>

Each level logs itself and every level above it in the order shown. For example, setting logging level at Warning logs Warning, Error, and Fatal. The lower the event level, the more detail is included in the event properties.

The Relativity program code defines the message and properties (metadata) for every logged event.

The message provides a verbal description of what happened and can be helpful in the initial identification of the cause of the problem.
The Properties columns of the RelativityLogs table contains the most useful details for the log entry. Logging data in a database is stored as XML. If you are using Microsoft SQL Server Management Studio, you can open it in a separate tab by clicking inside Properties column.

Pay special attention to the following properties:

- **MachineName** – the host name of a machine in the Relativity environment where the event occurred.
- **System** – systems are the top-level components of the Relativity. They are typically IIS applications, Windows services, or executables. For more information, see Logging system-subsystem-application matrix on page 29.
- **Subsystem** – subsystems are Relativity components that are called from one or more systems. Generally, they are designated based on similar functionality.
- **Application** – applications are unique identifiers to indicate what context the code is running for. Applications can be Relativity Applications.
- **ProcessID** – the ID of the operating system process where the event occurred.
- **ErrorMessage** – the .Net framework message associated with the event.
- **StackTrace** – the exception stack trace.

**Note:** If the logs are stored in a file sink or Data Grid, the data is stored in JSON format.

Log messages and properties may not always directly identify the cause of a problem in Relativity. Therefore, recognizing patterns of events and errors is critical when troubleshooting.
For more information, see Troubleshoot Relativity using logging on page 26.

1.4 Next steps

As a system admin or infrastructure manager, you may need to perform the following tasks related to logging:

- **Enable or disable logging** – sometimes it may be necessary to stop logging for your environment or for certain machines. See Configurations on page 16.

- **Change default logging level** – you may also need to change the default logging level. See Configure logging on page 13.

- **Define logging rules** – when troubleshooting Relativity, additional logging rules are usually required. See Troubleshoot Relativity using logging on page 26.

- **Change the default sink or define more sinks** – you may also need to change where logs are written to. See Sinks on page 17.

- **Perform logging data maintenance** – logging can significantly impact the performance and resource of your Relativity environment. See Infrastructure considerations for logging on page 32.
2 Configure logging

By default Relativity only logs at the ERROR level to the system database. When troubleshooting application problems, use the EDDSLogging database objects and configuration files to change the default configuration and capture the level of detail that meets your specific needs.

This page contains the following information:

2.1 Configuration defaults

Relativity logging is centrally configured in the EDDSLogging database. Logging config files provides backup for the database logging configuration.

**Note:** If both the central database configuration is unavailable (for example, due to connection failure) and the logging config files are missing or cannot be parsed because of invalid or malformed XML, no logging takes place.

When troubleshooting Relativity, you can change logging to meet your particular needs by using logging configurations, sinks, and rules.

**Note:** Logging changes take effect without an IIS reset.

By default logging is enabled for all Relativity modules. See Configurations for more information.

The following are the default logging level, sink, and log file location settings.

2.1.0.1 Default logging level
By default, Relativity logs at the Error (4) level. For information on changing the logging level, see Rules.

2.1.0.2 Default sink
By default, logs are stored in the EDDSLogging.RelativityLogs database table. For information on changing sinks, see Sinks and Rules.

2.1.0.3 Default log file location
For file logging, the log file location is set by the RELATIVITY_LOGS environment variable.

**Note:** The Relativity installer does not set RELATIVITY_LOGS. Instead you must set it manually on all machines in the Relativity environment.

You can also change this variable to a direct folder path within the LogConfig.xml file.

2.1.0.4 Restore logging defaults
To restore default logging settings for the database only, execute the ResetToDefault stored procedure in the EDDSLogging database. When this stored procedure runs, it also outputs the results from the PrintCurrentSetup stored procedure. The output is a runnable SQL script of the configuration that was just deleted. If the @WithParamNames bit parameter is set to 1, the generated script includes the @Param = [Value] notation and additional usage comments.

```sql
EXEC [EDDSLogging].[eddsdbo].[ResetToDefault] @WithParamNames = 1
```
This deletes all custom rules, configurations, and rule-sinks defined for your Relativity environment in the database.

To restore all logging setting back to default and delete all customizations including sinks, run the **ResetToDefault_WithSinks** stored procedure.

```sql
EXEC [EDDSLogging].[eddsdbo].[ResetToDefault_WithSinks] @WithParamNames = 1
```

**Note:** You can't remove the default values entirely from the database or assign them a higher order value than any other rule or configuration.

### 2.2 EDDSLogging database

Relativity uses the EDDSLogging database to store logging configuration and as the default sink for Relativity logs. In a distributed environment, it is hosted on the same SQL Server as the EDDS database. The database objects include tables, views, and stored procedures.

#### 2.2.1 Tables

- **Configuration** – stores available logging configurations. For more information, see [Configurations](#).
- **LoggingLevel** – defines the Relativity logging level values.
- **RelativityLogs** – stores the Relativity logs.
- **Rule** – stores the logging rules defined for the Relativity instance. For more information, see [Rules](#).
- **RuleSink** – stores the records that associate logging rules with sinks. For more information, see [Rules and sinks](#).
- **Sink** – stores the sinks defined for the Relativity instance. For more information, see [Sinks](#).
- **SinkType** – defines the Relativity sink types.

#### 2.2.2 Views

- **Rules** – provides quick access to the logging rules information defined in the EDDSLogging tables. Selecting from this view returns RuleID, Order, BestGuessLoggingEnabled, MachineName, System, SubSystem, Application, LoggingLevel, and JSON-formatted sink data.
  - **BestGuessLoggingEnabled** – indicates whether logging is enabled for the rule (1 – enabled, 0 – not enabled). It takes into consideration the Configuration table and applies it to the rule.
    - This field's value is not always accurate if your Configuration setting's MachineName, System, SubSystem, Application fields have "gaps" and don't match those in the rule.
    - If your rules and configuration match (MachineName, System, Application are specified without gaps in both) this field is more accurate.
    - To return the exact rule information for the log in question, use the **WhatIsMyRule** stored procedure.
2.2.3 Stored procedures

- **DeleteRule @RuleID**
  - Removes the rule and associated rule-sinks for the @RuleID passed in. See example in [Troubleshoot Relativity using logging on page 26](#).

- **EnsureConfiguration @ConfigurationID, @Order, @MachineName, @System, @SubSystem, @Application, @LoggingEnabled**
  - Ensures that a row exists in the Configuration table for the passed in parameters by inserting a new configuration if the MachineName, System, SubSystem, Application combination does not exist, or updating the Order or LoggingEnabled fields if it does. See example in [Configurations on the next page](#).
  - Ensures that the other rows' order values are updated according to the passed in @Order value. If the value already exists, the existing order values are incremented by 1 where necessary.
  - Ensures that the default configuration (the row with null values for MachineName, System, SubSystem, Application) has the highest Order value in the table.

- **EnsureRule @RuleID, @Order, @MachineName, @System, @SubSystem, @Application, @LoggingLevel, @SemiColonDelimetedListOfSinkNames**
  - Ensures that a row exists in the Rule table for the passed in parameters by inserting a new Rule when the MachineName, System, SubSystem, Application combination does not exist, or updating when the @RuleID exists and the combination doesn't already exist with a different RuleID. See example in [Rules on page 18](#).
  - Ensures that the RuleSink pairings match the @SemiColonDelimetedListOfSinkNames passed in. The param is taken literally; it deletes existing RuleSinks that were not included, and inserts RuleSinks that did not exist before.
  - Ensures that the other rows' order values are updated according to the passed in @Order value. If the value already exists, the existing order values is incremented by 1 where necessary.
  - Ensures that the default configuration has the highest Order value in the table.

- **EnsureRuleSinks @RuleID, @SemiColonDelimetedListOfSinkNames**
  - Ensures that the rule-sink pairings match the @SemiColonDelimetedListOfSinkNames passed in for the @RuleID. The parameter is taken literally; it deletes the existing rows in RuleSink table that are not included in @SemiColonDelimetedListOfSinkNames, and inserts the rows that don't exist.

- **EnsureSink @Name, @SinkTypeID, @Data**
  - Ensures that the sink exists for the passed in parameters.
  - @Name is the primary key.
  - @SinkTypeID is one of the sink types supported by Relativity (defined in the SinkType table). For more information, see [Sinks on page 17](#).
  - @Data is the JSON-formatted string defining for the sink.
Note: It is recommended that you use the other available stored procedures (EnsureSink_DataGrid, EnsureSink_EDDS, EnsureSink_File, and EnsureSink_SQLServer) for creating sinks because they create a JSON-formatted string for you.

- **EnsureSink_EDDS @Name**
  - Ensures the default EDDSLogging sink exists and creates the expected JSON-formatted value. See example in Sinks on the next page.

- **EnsureSink_File @Name, @LogFileLocation, @MaxFileSizeInMB**
  - Ensures the file sink exists and creates the expected JSON-formatted value. See example in Sinks on the next page.

- **EnsureSink_SQLServer @Name, @ConnectionString, @TableName**
  - Ensures the database sink exists and creates the expected JSON-formatted value. See example in Sinks on the next page.

- **PrintCurrentSetup @WithParamNames**
  - The script generates a dynamic representation of the stored procedures needed to recreate all logging configurations, rules, sinks, and rule-sinks defined in the EDDSLogging database.
  - If the @WithParamNames bit parameter is set to 1, the generated script includes the @Param = [Value] notation and additional usage comments.

- **ResetToDefault @WithParamNames**
  - Deletes all configuration, rules, and rule-sinks in the system and restores the original default logging settings at the Error (4) level to the RelativityLogs table. Note that sinks are not deleted.
  - Generates a SQL script to restore the deleted logging settings. Copy and save this output just in case.

- **ResetToDefault_WithSinks @WithParamNames**
  - Deletes all configuration, rules, and rule-sinks in the system and restores the original default logging settings at the Error (4) level to the RelativityLogs table.
  - Generates a SQL script to restore the deleted logging settings. Copy and save this output just in case.

- **WhatIsMyRule @MachineName, @System, @SubSystem, @Application**
  - Returns the rule view for the matching combination of machine name, system, subsystem, and application.
  - Make sure to pass in all parameters. You can find the machine name, system, subsystem, and application using the information on the Errors tab or in the Relativity logs. For more information, see Troubleshoot Relativity using logging on page 26

### 2.3 Configurations

Logging configurations in the EDDSLogging.Configuration table are used to enable or disable logging for a combination of machine/system/subsystem/application.
Configurations work similarly to logging rules, but they are used to set the LoggingEnabled flag.

By default the table contains only one row:

<table>
<thead>
<tr>
<th>ID</th>
<th>Order</th>
<th>MachineName</th>
<th>System</th>
<th>SubSystem</th>
<th>Application</th>
<th>LoggingEnabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>1</td>
</tr>
</tbody>
</table>

This enables logging for all systems, subsystems, and applications on all machines in the Relativity environment.

You can define additional configurations as necessary. For example, the following configuration disables logging for all Relativity Agents on the agent-srv1 machine:

```
EXEC [EDDSLogging].[eddsdbo].[EnsureConfiguration] @ConfigurationID = NULL, @Order = 1, @MachineName = 'agent-srv1', @System = 'Agent', @SubSystem = 'RelativityAgent', @Application = NULL, @LoggingEnabled = 0
```

Output:

<table>
<thead>
<tr>
<th>ID</th>
<th>Order</th>
<th>MachineName</th>
<th>System</th>
<th>SubSystem</th>
<th>Application</th>
<th>LoggingEnabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>agent-srv1</td>
<td>Agent</td>
<td>RelativityAgent</td>
<td>NULL</td>
<td>0</td>
</tr>
</tbody>
</table>

The following configuration enables logging for Relativity Services on rel-srv1:

```
EXEC [EDDSLogging].[eddsdbo].[EnsureConfiguration] @ConfigurationID = NULL, @Order = 2, @MachineName = 'rel-srv1', @System = 'RelativityServices', @SubSystem = NULL, @Application = NULL, @LoggingEnabled = 1
```

Output:

<table>
<thead>
<tr>
<th>ID</th>
<th>Order</th>
<th>MachineName</th>
<th>System</th>
<th>SubSystem</th>
<th>Application</th>
<th>LoggingEnabled</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>agent-srv1</td>
<td>Agent</td>
<td>RelativityAgent</td>
<td>NULL</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>rel-srv1</td>
<td>RelativityServices</td>
<td>NULL</td>
<td>NULL</td>
<td>1</td>
</tr>
</tbody>
</table>

### 2.4 Sinks

The `EDDSLogging.SinkType` table defines the following sink types:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EDDSLogging</td>
</tr>
<tr>
<td>2</td>
<td>SQLServer</td>
</tr>
<tr>
<td>3</td>
<td>FileSystem</td>
</tr>
</tbody>
</table>

The `EDDSLogging.Sink` table defines the available sinks. The entries in `EDDSLogging.Sink` are linked to the `EDDSLogging.SinkType` table byTypeID. By default, only the `EDDSLogging` sink is set up in the `Sink` table:
The data column accepts JSON-formatted strings for the Sink Type used. The easiest way to insert data is to use the stored procedures.

You can add and update sinks by using the stored procedures in the EDDSLogging database. If a sink with the specified name already exists in the table, it is updated.

### 2.4.1 File

Run the EnsureSink_File stored procedure.

```sql
EXEC [EDDSLogging].[eddsdbo].[EnsureSink_File] @Name = 'File1', @LogFileLocation = 'C:\Temp\Logs', @MaxFileSizeInMB = 10000
```

**Note:** The stored procedure attempts to escape any single backslash to make the LogFileLocation JSON serializable.

### 2.4.2 SQL Server

Run the EnsureSink_SQLServer stored procedure.

```sql
EXEC [EDDSLogging].[eddsdbo].[EnsureSink_SQLServer] @Name = 'SQL1', @ConnectionString = 'data source=localhost;initial catalog=EDDSLogging;persist security info=False;user id=EDDSdbo;password=FAKE123;workstation id=localhost;packet size=4096', @TableName = 'RelativityLogs'
```

**Note:** Do not use double-quotes in the connection string because that breaks the eddsdbo.PrintCurrentSetup script. The connection string does not need double quotes to work. The TableName does not need the database schema of the targeted table to work.

Output from running all three of the above scripts:

```
<table>
<thead>
<tr>
<th>Name</th>
<th>TypeID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Default</td>
<td>0</td>
</tr>
</tbody>
</table>
```

**Note:** Note that the data field that defines the sink (for example, the database connection string or file path) must be formatted as JSON and the stored procedures automatically do it for you.

### 2.5 Rules

Rules defined the EDDSLogging.Rule table are used to set the logging level for a combination of machine/system/subsystem/application.

By default the table contains only one row:
This sets the logging level to 4 (Error) for all systems, subsystems, and applications on all machines in the Relativity environment.

**Note:** The Order field defines the order in which Relativity logging parses the rule for a given combination of machine/system/subsystem/application. The rule with the lowest Order value is used once machine/system/subsystem/application are matched.

To change the logging level, update the **LoggingLevel** value. For example, this statement turns logging level up to **DEBUG** for everything:

```
update [EDDSLogging].[eddsdbo].[Rule] set LoggingLevel = 1
```

You can define additional rules as necessary. For example, the following rules set logging for all Relativity core agents ('Default' application) on the agent-srv1 machine to **INFORMATION** and to the **Default** sink:

```
EXEC [EDDSLogging].[eddsdbo].[EnsureRule] @RuleID = NULL, @Order = 1, @MachineName = 'agent-srv1', @System = 'Agent', @SubSystem = 'RelativityAgent', @Application = NULL, @LoggingLevel = 2, @SemiColonDelimitedListOfSinkNames = 'Default'
```

- @MachineName is 'agent-srv1'
- @System is set to 'Agent'
- @SubSystem is set to 'RelativityAgent' for core Relativity agents
- @Application is set to null
- @LoggingLevel is set to 2 (INFORMATION)
- @SemiColonDelimitedListOfSinkNames is set to 'Default'

Output:

<table>
<thead>
<tr>
<th>ID</th>
<th>Order</th>
<th>MachineName</th>
<th>System</th>
<th>SubSystem</th>
<th>Application</th>
<th>LoggingLevel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>agent-srv1</td>
<td>Agent</td>
<td>RelativityAgent</td>
<td>NULL</td>
<td>2</td>
</tr>
</tbody>
</table>

The rules also add a row to the **eddsdbo.RuleSink** table to link the sink to the rule.

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Default</td>
</tr>
<tr>
<td>2</td>
<td>Default</td>
</tr>
</tbody>
</table>
```

The following rule turns up logging to **INFORMATION** only for Relativity telemetry agents (Telemetry Host, Telemetry Metrics Transmission, and APM Transmission agent):
EXEC [EDDSLogging].[eddsdbo].[EnsureRule] @RuleID = NULL, @Order = 1, @MachineName = null, @System = 'Agent', @SubSystem = 'CustomAgent', @Application = '3d20c75d-e2a8-468d-86c3-7544c035a11d', @LoggingLevel = 2, @SemiColonDelimetedListOfSinkNames = 'Default'

- **@MachineName** is null because telemetry agents run on a single server in the Relativity environment
- **@System** is set to 'Agent'
- **@SubSystem** is set to 'CustomAgent' because the agents are part of the telemetry application
- **@Application** is set to the GUID of the telemetry application ('3d20c75d-e2a8-468d-86c3-7544c035a11d')
- **@LoggingLevel** is set to 2 (**INFORMATION**)
- **@SemiColonDelimetedListOfSinkNames** is set to 'Default'

Output:

<table>
<thead>
<tr>
<th>ID</th>
<th>Order</th>
<th>MachineName</th>
<th>System</th>
<th>SubSystem</th>
<th>Application</th>
<th>LoggingLevel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>NULL</td>
<td>Agent</td>
<td>CustomAgent</td>
<td>3d20c75d-e2a8-468d-86c3-7544c035a11d</td>
<td>2</td>
</tr>
</tbody>
</table>

The following configuration enables logging for all Relativity services on rel-srv1 to **WARNING** and to both the **Default** sink and a **File** sink:

EXEC [EDDSLogging].[eddsdbo].[EnsureRule] @RuleID = NULL, @Order = 2, @MachineName = 'rel-srv1', @System = 'RelativityServices', @SubSystem = NULL, @Application = NULL, @LoggingLevel = 4, @SemiColonDelimetedListOfSinkNames = 'Default;File1'

- **@MachineName** is 'rel-srv1'
- **@System** is set to 'RelativityServices'
- **@SubSystem** is null
- **@Application** is null
- **@LoggingLevel** is set to 4 (**WARNING**)
- **@SemiColonDelimetedListOfSinkNames** is set to 'Default;File1'

Output:

<table>
<thead>
<tr>
<th>ID</th>
<th>Order</th>
<th>MachineName</th>
<th>System</th>
<th>SubSystem</th>
<th>Application</th>
<th>LoggingLevel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>agent-srv1</td>
<td>Agent</td>
<td>RelativityAgent</td>
<td>NULL</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>rel-srv1</td>
<td>RelativityServices</td>
<td>NULL</td>
<td>NULL</td>
<td>4</td>
</tr>
</tbody>
</table>

The configuration also adds two rows to **eddsdbo.RuleSink** to link the sink to the rule.
Using the `eddsdbo.EnsureRule` stored procedure, you can remove or add sinks linked to our rule by simply adding or removing them to the `@SemiColonDelimetedListOfSinkNames` parameter. The stored procedure ensures that only the sink names that exists in the value are linked to the rule.

**Note:** When making changes to logging rules, make sure to select an appropriate logging level to avoid flooding the logs.

### 2.5.1 Rules and sinks

The entries in `EDDSLogging.RuleSink` table associate rules with sinks. You can use the table to set up logging to multiple sinks from a single rule.

By default the table contains only one row:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Default</td>
</tr>
<tr>
<td>2</td>
<td>Default</td>
</tr>
<tr>
<td>3</td>
<td>Default</td>
</tr>
<tr>
<td>4</td>
<td>File1</td>
</tr>
</tbody>
</table>

This instructs the system to log everything to the `EDDSLogging.RelativityLogs` table. The entries in `EDDSLogging.RuleSink` are linked to the `EDDSLogging.Sink` table by name.

### 2.6 Config files

The logging config files provide a fallback in case of a database connection failure. The files are named `LogConfig.xml` and are located next to the main configuration file for each Relativity system. The following example shows `LogConfig.xml` in a production environment, right next to `web.config`. 

```xml
<configuration>
  <RelativityLogging>
    <Rules>
      <Rule name="Default" />
    </Rules>
    <Sinks>
      <Sink name="EDDSLogging.RelativityLogs" />  
    </Sinks>
  </RelativityLogging>
</configuration>
```
2.6.0.1 Default config file

The following is the LogConfig.xml file installed by default for all modules.

```xml
<?xml version="1.0" encoding="utf-8" ?>
<!-Information on how to use this file can be found here https://platform.relativity.com/9.3/Content/Logging/Local_config_files.htm -->
<kCuraLogging masterLoggingConfiguration="EDDS">
    <rules enabled="true">
        <rule system="*" loggingLevel="Error" sink="File1"/>
    </rules>
    <sinks>
        <fileSink name="File1" logFileLocation="%RELATIVITY_LOGS%" maxFileSizeInMB="10000" />
    </sinks>
</kCuraLogging>
```

2.6.0.2 Config file reference

<kCuraLogging masterLoggingConfiguration>

- Required attributes:
  - `masterLoggingConfiguration="Local"` – use the settings in this file.
  - `masterLoggingConfiguration="EDDS"` – connect to the primary SQL Server in your Relativity environment to access the EDDSLogging database. If the connection fails, use the settings in this file.

<rules enabled>

- Required attributes:
  - `enabled="true"` – if the settings in the file are used and not the database, logging is turned on and the rules in this file apply.
○ **enabled="false"** – if the settings in the file are used and not the database, logging is turned off and the rules in this file are ignored.

### <rule system subSystem application loggingLevel sink>

- **General information:**
  - Each rule is read in a top-down manner. Once a logger has found a matching rule, it stops and uses the logging level and sink specified in that rule.

- **Required attributes (If these are not specified, the rule fails to initialize):**
  - **loggingLevel** – values are **Verbose, Debug, Information, Warning, Error, and Fatal**.
    - **Verbose** - logs all messages. Should only be used with Relativity Support supervision.
    - **Debug** - logs information that is diagnostically helpful to developers. Should only be used by Relativity developers.
    - **Information** - logs informational messages that highlight the progress of the application (where it’s at in the code and what it’s currently doing).
    - **Warning** - logs anything that can cause application exceptions that the application is able to recover and continue.
    - **Error** - logs all application exceptions from which the application cannot recover.
    - **Fatal** - logs any error that would force a shutdown of the service or application to prevent data loss or further data loss.
  - **sink** – value is the exact name of the sink that is defined in the **sinks** element. Multiple sinks can be included and are semicolon (;) delimited, for example:
    ```xml
    <sinks>
      <sink>MyFileSink</sink>
      <sink>MyFileSink;MySQLSink;</sink>
    </sinks>
    ```

  **Note:** If multiple sinks are specified, the same information logged to each sink.

- **Optional attributes (If these are not specified or their value="*", then the rule applies to all items of that type):**
  - **system** – values are predefined names of the System this rule applies to.
  - **subSystem** – values are predefined names of the SubSystem this rule applies to.
  - **application** – values are the GUID of the Application this rule applies to.

If you have multiple `<rule>` elements in the `<rules>` section, the `<rule>` elements are evaluated in the order they are found, so you must put more specific rules first. For example, this is how you set logging level to **Verbose** for the Processing application (specified by GUID) while logging is set to **Error** for the rest of Relativity:

```xml
<rules enabled="true">
  <rule application="ed0e23f9-da60-4298-af9a-ae6a9b6a9319" loggingLevel="Verbose" sink="File1"/>
  <rule system="*" loggingLevel="Error" sink="File1"/>
</rules>
```

### <sinks>

- **General information:**
  - Each sink type can be included more than once.
  - If a name is not unique the first sink found with that name is used.
<fileSink name logFileLocation maxFileSizeInMB>

- Required attributes:
  - **name** – user defined string that links to a rule’s **sink** attribute.
  - **logFileLocation** – absolute path to the location where the log files are saved. You can use the RELATIVITY_LOGS environment variable to specify the default directory portion of log file path.

**Note:** If you use a relative path for **logFileLocation**, the output is relative to the location where you are running your application and not to the path of the executable or the **bin** directory. It is strongly recommended to only use an absolute path.

- **maxFileSizeInMB** – the max size for a log file can be. Once the max is reached, a new file is created, and the name is suffixed with a number.

</fileSink>

<eddsLogSink name>

- General information:
  - This sink attempts to find primary Relativity SQL Server with the **EDDSLogging** database using the connection string that is typically provided in web or app config files of the application.
  - Writes the logs into the **RelativityLogs** table on the **EDDSLogging** database.

- Required attributes:
  - **name** – user-defined string that links to a rule’s **sink** attribute.

</eddsLogSink>

<sqlSink name connectionString tableName>

- Required attributes:
  - **name** – user-defined string that links to a rule’s **sink** attribute.
  - **connectionString** – full connection string of the database that contains the table to log to.
  - **tableName** – name of the table to log to. Table name does not need the schema name.

</sqlSink>

### 2.7 Troubleshoot logging

Logging failures can occur when logging configuration is missing or cannot be processed due the central database connection failure or malformed XML in the config files. If logging for a Relativity component fails to initialize, Windows records error messages in the Event Viewer. Look for the messages in **Server Manager > Diagnostics > Event Viewer > Applications and Services Logs > kCura**.
Troubleshoot Relativity using logging

When troubleshooting Relativity, use logging to capture the details to identify the root cause of a problem. Rules allow you to drill down to the root cause by increasing the logging level in the affected area of the application and avoid capturing unnecessary information.

3.1 Set up a logging rule

Logging uses rules to determine how to treat Relativity events. We recommend that you target your logging rule to filter out unrelated events.

Logging applies rules based on the machine name, system, subsystem, and application, in order of the Order field, to determine the logging level and the sinks to log to.

3.1.1 Order

Rules are processed in order. Logging uses the first valid rule it finds that meets all conditions. If there are two valid rules for a given log message, the first rule is used and the second is ignored.

Example

Rules A and B are both valid for a log event. Rule A has an order value of 1 while Rule B has an order value of 4. Rule A has a Logging Level of Error (4) while Rule B has a logging level of Verbose (0).

Because A has a lower order, it is processed first, and thus only Error or above is logged.

3.1.2 Machine name

The Machine column filters the rule to only apply to logging on a specific machine name. A NULL value indicates that there is no filter for machine name so the rule applies to all machines in the Relativity environment.

Example

Rule A has a MachineName of Server1 and is valid only on Server1. Logging on Server2 ignores Rule A.

3.1.3 System

System column filters the rule to only apply to logging for that system. A NULL value indicates that there is no filter for system, so the rule applies to all systems.

Example

Rule A has a System value of Relativity.Rest and is valid only for Relativity.Rest. Logging for the Relativity.Services system ignores processing Rule A.

3.1.4 Subsystem

The SubSystem column filters the rule to only apply to logging for that subsystem. A NULL value indicates that there is no filter for subsystem, so the rule applies to all subsystems.

Example

Rule A has a subsystem value of EventHandler and is valid only for event handlers. Logging for the CustomPage subsystem ignores processing Rule A.
3.1.5 Application
The Application column filters the rule to only apply to logging for that application. A NULL value indicates that there is no filter for the application.

Example
Rule A has an Application value of 51b4e374-ef1b-43a0-b5a8-e2841ac3efe1 and is valid only for Analytics (51b4e374-ef1b-43a0-b5a8-e2841ac3efe1). Logging for Imaging (c9e4322e-6bd8-4a37-ae9e-c3c9be31776b) ignores processing Rule A.

3.1.6 LoggingLevel
The LoggingLevel column contains the level at which the rule applies filtering for log events. The logger writes a log event only when it is equal to or higher than the logging level of the rule.

Example
Rule A has a LoggingLevel of Error (4) and logs error- and fatal-level events (4 and 5). Log events that are Warning (3), Information (2), Debug (1), and Verbose (0) are ignored.

3.1.7 Rule sinks
A rule is not complete without a sink. A sink is a destination to which log messages are written. Logging ignores rules without any sinks. The RuleSink table associates rules with sinks. Rules can share a sink and have multiple sinks.

3.2 Troubleshooting example
By default, Relativity logging is set to the Error level (capturing the same details as the Errors tab). The logging database settings are as follows:

EDDSLogging.Rule Table

<table>
<thead>
<tr>
<th>ID</th>
<th>Order</th>
<th>MachineName</th>
<th>System</th>
<th>SubSystem</th>
<th>Application</th>
<th>LoggingLevel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>4</td>
</tr>
</tbody>
</table>

EDDSLogging.RuleSink Table

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Default</td>
</tr>
</tbody>
</table>

This means that if an error occurs anywhere in Relativity, it is logged to the EDDSLocking.RelativityLogs table (the default sink).

Note that the error in the example below initially doesn't provide a lot of troubleshooting information:

```xml
<properties>
  <property key="ErrorMessage">The save could not be completed because an error occurred during a Relativity Event Handler save action. Please contact your system administrator</property>
</properties>
```
<property key="StackTrace">SomeApplication.Eventhandler.PreSaveMyObject.Execute().....</property>
<property key="CastArtifactID">1234567</property>
<property key="CaseName">Important Case</property>
<property key="ErrorArtifactID">1068612</property>
<property key="MachineName">Server1</property>
<property key="AppDomain">4</property>
<property key="Application" /></property>
<property key="SubSystem" />
<property key="System">Relativity</property>
<property key="ProcessId">29336</property>
</properties>

But the **MachineName** property does tell you that the error occurred on **Server1**, and the System property indicates that it occurred in the **Relativity** system. Use the information to set up a rule to capture additional details.

The **ErrorMessage** property tells you that the error happened during an event handler save action, so you know you can add subsystem of **EventHandler** to the new rule. The **StackTrace** property mentions **SomeApplication**, and you can add this application's identifier to the rule to make it even more specific.

Use the **EnsureRule** stored procedure to add the rule:

EXEC [EDDSLogging].[eddsdbo].[EnsureRule] @RuleID = NULL, @Order = 1, @MachineName = 'Server1', @System = 'Relativity', @SubSystem = 'EventHandler', @Application = '99999999-9999-9999-9999-999999999999', @LoggingLevel = 2, @SemiColonDelimetedListOfSinkNames = 'Default'

The logging database settings are now:

**EDDSLogging.Rule Table**

<table>
<thead>
<tr>
<th>ID</th>
<th>Order</th>
<th>MachineName</th>
<th>System</th>
<th>SubSystem</th>
<th>Application</th>
<th>LoggingLevel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Server1</td>
<td>Relativity</td>
<td>EventHandler</td>
<td>99999999-9999-9999-9999-999999999999</td>
<td>2</td>
</tr>
</tbody>
</table>

**EDDSLogging.RuleSink Table**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Default</td>
</tr>
<tr>
<td>2</td>
<td>Default</td>
</tr>
</tbody>
</table>

You have created a new rule using the information from the initial error message to increase our logging level for the component you believe to be causing the problem to **Information (2)**. This rule provides more information for troubleshooting. If you aren't getting enough information, you can continue to drill down and increase the logging level to **Debug (1)** or **Verbose (0)**.

After you identify the root cause of the problem, remove the new rules you created. This ensures that unnecessary information is not collected and no system performance degradation occurs through additional use of the CPU, disk, database, and network resources. Use the **DeleteRule** stored procedure to delete the rule:

EXEC [EDDSLogging].[eddsdbo].[DeleteRule] @RuleID = 2
4 Logging system-subsystem-application matrix

You can identify Relativity system components that log messages based on the system-subsystem-application designation. When troubleshooting, use the system-subsystem-application matrix to configure logging to target specific Relativity components.

4.1 Relativity systems

Systems are the top-level components of Relativity. They are typically IIS applications, Windows services, or executables. Logging systems are as follows:

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent</td>
<td>Agent windows service running as kCura EDDS Agent Manager.</td>
</tr>
<tr>
<td>Invariant</td>
<td>Relativity Invariant processing and conversion engine.</td>
</tr>
<tr>
<td>Relativity</td>
<td>Relativity web site under IIS installed by default as /Relativity.</td>
</tr>
<tr>
<td>RelativityDistributed</td>
<td>Relativity web site for downloads installed by default as /Relativity.Distributed.</td>
</tr>
<tr>
<td>RelativityRest</td>
<td>Relativity RESTful web services installed by default as /Relativity.Rest.</td>
</tr>
<tr>
<td>RelativityServices</td>
<td>Relativity web services installed by default as /Relativity.Services.</td>
</tr>
<tr>
<td>RelativityWebAPI</td>
<td>Relativity web services installed by default as /RelativityWebAPI.</td>
</tr>
<tr>
<td>ServiceHost</td>
<td>Windows service for running hosted services. Installed by default as kCura Service Host Manager.</td>
</tr>
<tr>
<td>WebProcessing</td>
<td>Agent windows service running as kCura EDDS Web Processing Manager.</td>
</tr>
</tbody>
</table>

The following image presents the Relativity components hosted in IIS. They correspond to the logging systems listed in the table above.
### 4.2 Subsystems

Subsystems are Relativity components called from one or more systems. Generally, they are designated based on similar functionality.

Logging subsystems are as follows:

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container</td>
<td>Web API for dynamic endpoints for RESTful services.</td>
</tr>
<tr>
<td>CustomAgent</td>
<td>Background process of an agent that is part of an application or a resource file.</td>
</tr>
<tr>
<td>CustomPage</td>
<td>Custom Pages are IIS applications deployed under <code>/Relativity/CustomPage/&lt;GUID&gt;</code></td>
</tr>
<tr>
<td>EventHandler</td>
<td>Event handlers that run custom code build on the Relativity platform.</td>
</tr>
<tr>
<td>IdentityServer</td>
<td>Authentication server that is used for managing tokens for user login.</td>
</tr>
<tr>
<td>Kepler</td>
<td>Web API for dynamic endpoints for RESTful services.</td>
</tr>
<tr>
<td>RelativityAgent</td>
<td>Background process of an agent that is part of the Relativity.</td>
</tr>
<tr>
<td>InvariantAPI</td>
<td>Invariant API hosted in the Invariant Queue Manager service for processing and conversion requests.</td>
</tr>
</tbody>
</table>
### Subsystem Description

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvariantAdminAPI</td>
<td>Invariant Admin API hosted in the Invariant Queue Manager service for managing processing workers.</td>
</tr>
</tbody>
</table>

### 4.3 Relativity applications

Applications are unique identifiers of the context the code is running in. Applications are identified by GUID.

Logging applications are as follows:

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>51b4e374-ef1b-43a0-b5a8-e2841ac3efe1</td>
<td>Analytics</td>
</tr>
<tr>
<td>62284add-91f5-4f35-a582-bbca439ad8c</td>
<td>Analytics Core</td>
</tr>
<tr>
<td>81ceb2f0-747a-4e8b-aad5-7c40d864d96d</td>
<td>Assisted Review</td>
</tr>
<tr>
<td>62ab9904-2f5e-442e-9d5e-259636eee79c</td>
<td>Binders</td>
</tr>
<tr>
<td>6a8c2341-6888-44da-b1a4-5bdce0d1a383</td>
<td>Data Grid Core</td>
</tr>
<tr>
<td>293c50ad-7b6d-45d0-9121-7f3826e1cca5</td>
<td>Data Grid for Audit</td>
</tr>
<tr>
<td>3e86b18f-8b55-45c4-9a57-9e0cbd7baf46</td>
<td>Default (Resource File Default)</td>
</tr>
<tr>
<td>5725cab5-ee63-4155-b227-c74cc9e26a76</td>
<td>Document Viewer</td>
</tr>
<tr>
<td>5975ec2c-ef13-4358-a062-aff4891b2343</td>
<td>Fact Manager</td>
</tr>
<tr>
<td>c9e4322e-6bd8-4a37-ae9e-c3c9be31776b</td>
<td>Imaging</td>
</tr>
<tr>
<td>3e258ac2-2b63-498e-a895-9515ecea8df4</td>
<td>Layout Builder</td>
</tr>
<tr>
<td>e5fddd9f-b55b-454c-8d96-b8285d0de2be</td>
<td>Lists</td>
</tr>
<tr>
<td>901c560a1-bdbd-4b48-a5e9-fe3f17211149</td>
<td>Mass Operation Handler</td>
</tr>
<tr>
<td>835d537-688a-4dde-b057-5ef2fe4db2a2b</td>
<td>OCR</td>
</tr>
<tr>
<td>60a1d0a3-2797-4fb3-a260-614cbfd3fa0d</td>
<td>Performance Dashboard</td>
</tr>
<tr>
<td>ed0e23f9-da60-4298-af9a-ae6a9b6a9319</td>
<td>Processing</td>
</tr>
<tr>
<td>51b19ab2-3d45-406c-a85e-f98c01b033ec</td>
<td>Production</td>
</tr>
<tr>
<td>98f31698-90a0-4ead-87e3-dac723fed2a6</td>
<td>Relativity Legal Hold</td>
</tr>
<tr>
<td>2ff16b11-a4ca-4f02-8bbb-1f07f23fe713</td>
<td>Relativity List Page</td>
</tr>
<tr>
<td>3d20c75d-e2a8-468d-86c3-7544c035a11d</td>
<td>Relativity Telemetry</td>
</tr>
<tr>
<td>3b884f04-f8ec-45f3-8427-2f8beed9bfc</td>
<td>Review Manager</td>
</tr>
<tr>
<td>f807ac5a-6f0c-4ef3-9c7d-db2b051a5f4</td>
<td>Search Terms Report</td>
</tr>
</tbody>
</table>

Custom Relativity applications can also use logging and be identified by GUID.
5 Infrastructure considerations for logging

Logging can be a resource-intensive operation. As a system admin or an infrastructure manager, you must consider the impact of logging on your IT infrastructure.

5.1 File sink

Consider this when configuring a file sink for logging:

- Because Relativity application threads can't have an exclusive lock on a network share, we recommend you don't select a network share as a logging sink.
- Granular logging (WARNING and lower) can fill disk space quickly. We recommend you don't select your primary (C:) drive as a destination for file logging.

5.2 Storing database passwords

When you configure an alternative SQL Server database sink for logging (other than the RelativityLogs table in the EDDSLogging database), the password in the connection string is stored as unencrypted (clear) text. This may not be allowed in certain high-security environments.

5.3 Moving EDDSLogging database

Relativity logging relies on the EDDSLogging database for centralized configuration storage. It also uses it the database as the default sink. In a distributed database environment, EDDSLogging must be hosted on the primary database server (the server that hosts the EDDS database). If you move the EDDS database to a different database server, you must also move the EDDSLogging database to the same server.
Proprietary Rights

This documentation ("Documentation") and the software to which it relates ("Software") belongs to Relativity ODA LLC and/or Relativity’s third party software vendors. Relativity grants written license agreements which contain restrictions. All parties accessing the Documentation or Software must: respect proprietary rights of Relativity and third parties; comply with your organization’s license agreement, including but not limited to license restrictions on use, copying, modifications, reverse engineering, and derivative products; and refrain from any misuse or misappropriation of this Documentation or Software in whole or in part. The Software and Documentation is protected by the Copyright Act of 1976, as amended, and the Software code is protected by the Illinois Trade Secrets Act. Violations can involve substantial civil liabilities, exemplary damages, and criminal penalties, including fines and possible imprisonment.

©2019. Relativity ODA LLC. All rights reserved. Relativity® is a registered trademark of Relativity ODA LLC.