

# Cúram 8.1.3.0 iFix1

## CONTENTS

[Introduction](#)  
[System Requirements](#)  
[Download](#)  
[Security Bulletins](#)  
[Installation](#)  
[Improvements, Resolved Issues and Third Party Updates](#)  
[Known Issues](#)  
[Notices](#)

## Introduction

Welcome to the Cúram 8.1.3.0 iFix1 release.

For product documentation, see [Merative Support Docs](#).

[Back to top](#)

---

## System Requirements

For information about the supported software for this release, see the [Prerequisites and supported software](#).

[Back to top](#)

---

## Download

This release is available to download from [Merative Support](#). You must log in to download software, request access if needed. Select Cúram Support and Software Download, enter your credentials and download the software.

[Back to top](#)

---

## Security Bulletins

Security Bulletins are now available from [Merative Support](#). You must log in to access Security Bulletins, request access if needed. Select Cúram Support and Software Download, enter your credentials, and open Knowledge Articles to see the Security Bulletins.

[Back to top](#)

---

## Installation

For full installation instructions, see the Development Environment Installation Guide at [Product Documentation and PDFs](#).

The basic installation steps are as follows:

- Review the release notes and documentation for any prerequisite steps.
- Download and extract the contents of the .zip file to a local drive location.
- Ensure that all files in your Cúram installation are writable.
- Run the Cúram installer, which you can find in the INSTALLER folder.
- Run the appropriate build targets for your installation.

## Upgrading

If you are upgrading from a previous version, the Cúram Upgrade Helper contains documentation and tools to help you to upgrade your Cúram application codebase and database. The Cúram Upgrade Guide describes the recommended process for application and database upgrades. The Upgrade Helper contains tools to assist you with implementing the upgrade, including tools to produce a schedule of required migrations for your upgrade, tools to provide information about database schema changes, and tools to generate initial SQL scripts for applying changes to your database. Download the appropriate version of the Cúram Upgrade Helper from [Merative Support](#). You must log in to download software, request access if needed. Select Cúram Support and Software Download, enter your credentials and download the software.

[Back to top](#)

---

## Improvements, Resolved Issues and Third Party Updates

[Curam Enterprise Framework](#)

### Curam Enterprise Framework

[Application Development Environment](#)

### Application Development Environment

## Rest

### Rest

#### DT037101 , WorkItem:SPM-137537 - Internal error codes returned from the Rest Infrastructure are not documented

##### Issue Description:

The 'Code' attribute of the failure response message body is used for both standard HTTP response codes and internal Curam error codes. Consumer applications normally expect HTTP response codes and do not know how to handle internal error codes.

##### User Interface Impact: No

##### Steps to Reproduce:

Make a Rest invocation using an incorrectly formatted date.

E.g: Send the following message with an incorrectly formatted 'endDate'

```
{
  "specialCautionType": { "tableName" : "SpecialCautionType", "value" : "SCSC12", "description" : "Escape Threat", "parentCodeTable"
  "specialCautionCategory": { "tableName" : "SpecialCautionCategory", "value" : "SCC03", "description" : "Behavioral Alert" },
  "specialCautionid": "100",
  "versionNo": 0,
  "startDate": "2021-03-18",
  "endDate" : "202-03-20",
  "Comments" : "test comments",
  "concernroleid": "101"
} to the Rest path, E.g: https://10.0.0.229:10101/Rest/v1/persons/101/special_cautions
```

**Issue:** The resulting error message will contain internal code -150212 to indicate that 'endDate' could not be decoded:

```
{
  "code" : -150212,
  "field" : "endDate",
  "message" : "There was an issue while parsing a 'endDate' attribute, which contains a value of '202-03-20': This is not in the supported range.",
  "level" : "error"
}
```

##### Resolution:

All negative values for the 'code' fields can be interpreted as 'HTTP 400 The data request is invalid'. If needed, the full set of internal code values and their associated message can be obtained from the following property files:

- EJBServer/components/Rest/rest/properties/RestErrorMessages.properties
- CuramCDEJ/doc/defaultproperties/curam/omega3/i18n/RuntimeMessages.properties

#### DT037090, WorkItem:SPM-139456 - The error object defined in the Swagger specification is incorrect

##### Issue Description:

The error object definition in the Swagger specification is incorrect and inconsistent with the actual error response returned by REST APIs.

There are two key problems:

1. The error object definition contains an incorrectly named Error attribute instead of errors.
2. The error object definition is missing field and stack\_trace attributes, which may not always contain values but should be included for consistency and accuracy.

##### User Interface Impact: No

##### Steps to Reproduce:

To reproduce the issue you need to invoke a REST API.

##### Using Swagger UI or a REST client (e.g., Postman):

1. Log in as a caseworker.
2. Open a new browser tab. The authentication cookies from logging in will be reused by the REST client or Swagger UI tool.
3. Invoke a REST API and pass an invalid parameter to force an error (e.g., /Rest/v1/persons/12345xxx).

##### Using a command-line tool (e.g., cURL):

1. Programmatically call the /Res/jsecuritycheck URL.
2. Retrieve the LTPA token from the response cookie header.
3. Add the LTPA token to the cookie request header for all API calls.
4. Call an existing REST API with an invalid parameter (e.g., /Rest/v1/persons/12345xxx).

Issue: The API response contains an errors attribute, along with field and stacktrace attributes (if curam.trace is set to traceon), however, the Swagger specification defines the error object as having an Error attribute instead of errors and does not include field or stack\_trace.

##### Resolution:

The Swagger specification has been updated to:

1. Rename the incorrectly named Error attribute to errors.
2. Add definitions for the field and stacktrace (will now always appear but will not have a value unless curam.trace is set to traceon) attributes, with descriptions indicating that these fields may not always contain values.

```
"errors": [
  {
```

```
    "code": 404,  
    "field": null,  
    "message": "HTTP 404 Not Found",  
    "level": "error",  
    "message_id": null,  
    "stackTrace": null  
  }  
}  
}
```

#### DT037091, WorkItem:SPM-139457 - The Swagger specification for the REST apis is missing the description for the mandatory referer request header

##### Issue Description:

The Swagger specification for REST APIs does not include descriptions for the mandatory Referer header, which is required for all API requests.

**User Interface Impact:** No

##### Steps to Reproduce:

1. Open the Swagger specification.
2. Expand the details for any GET, PUT, POST, or DELETE API request.
3. Issue: The Referer header is not listed in the Parameters section, even though it is mandatory for all API requests.

##### Resolution:

The Swagger specification has been updated to include the Referer header as a mandatory parameter with an appropriate description for all API requests.

```
{  
  "name": "Referer",  
  "in": "header",  
  "description": "The Referer header is mandatory for all API requests.",  
  "type": "string",  
  "required": true  
}
```

#### DT037104, WorkItem:SPM-139458 - The swagger spec generator in the REST infrastructure incorrectly always assumes the same request and responses body for a PUT request

##### Issue Description:

The Swagger spec generator in the REST infrastructure incorrectly assumes that all PUT requests have the same request and response body. This results in the object definition for the request body being reused for the response body, regardless of whether the input and output structs are different in the actual API.

**User Interface Impact:** No

##### Steps to Reproduce:

1. Open the Swagger specification.
2. Expand a PUT API with differing input and output structs.
3. View the definitions section for the request and response bodies.
4. Issue: The Swagger spec generator reuses the request body object for the response body, even if the API has different input and output structs.

##### Resolution:

The Swagger spec generator has been updated to detect when a PUT API has differing input and output structs and generate separate object definitions for the request and response bodies when required.

#### DT037089, WorkItem:SPM-139459 - REST API error messages return HTML code for a single quote (") instead of the single quote character

##### Issue Description:

Certain REST API error messages contain parts which are subjected to HTML escaping, causing characters to be converted to their unreadable HTML-escaped representation.

**User Interface Impact:** No

**Steps to Repo:** N/A

Issue: There was an issue while parsing a 'concern/roleid' attribute, which contains a value of '12345xxx': The field (")concern/roleid(") must be a whole number.

##### Resolution:

The REST infrastructure has been updated to ensure that error messages are displayed correctly for consumers of the API.

Resolved line should resemble this example. There was an issue while parsing a 'concern/roleid' attribute, which contains a value of '12345xxx': The field 'concern/roleid' must be a whole number.

[Back to top](#)

---

## Known Issues

Please refer to the [Known Issues documented for 8.1.3.0](#).

[Back to top](#)

---

## Notices

Before using this information and the product it supports, read the information in ["Notices"](#)

Copyright

© Merative US L.P. 2025

Merative is a trademark of Merative US L.P. in the United States and other countries.

[Back to top](#)

---