Cúram 8.1.3

Verification Guide

Note

Before using this information and the product it supports, read the information in

Edition

This edition applies to Cúram 8.1, 8.1.1, 8.1.2, and 8.1.3.

© Merative US L.P. 2012, 2024

Merative and the Merative Logo are trademarks of Merative US L.P. in the United States and other countries.

Contents

1 Verification guide

The verification engine interprets evidence that is based on verification rules. Client information can be verified by documents such as birth certificates or bank statements. Caseworkers can manage client evidence verifications in the participant manager. Verification can be administratively configured.

The following list outlines the key characteristics of verification:

- Verification is the process of checking the accuracy of the information that is given by clients who are applying for services from a social enterprise organization.
- Verification consists of three components: an administration component, a case component, and a participant component.
- Six verification elements are configured in the administration component of Cúram verification: verification categories, verifiable data items, verification item utilizations, verification requirements, verification requirement usages, and dependent data items. The structure of the elements, as specified in the administration component, is the template for all verification processing in the application.
- In the verification case component, caseworkers can manage deadlines for verification requirements and can provide verification information for captured evidence.
- The effect of modifying evidence on a verification requirement depends on whether the applicable evidence is in an `Active` or `In Edit` state.
- Evidence cannot be activated unless the mandatory verification requirements are met for a piece of evidence. A case cannot be activated until all mandatory verification requirements are met for the evidence that is associated with the case or a current verification waiver exists for the mandatory verification.

For more information about verifications, see the *Evidence* and *Cúram Workflow Overview Guide* related links.

Related information

1.1 Understanding verification

Verification is the process of checking the accuracy of the information that is given by clients who are applying for services from a social enterprise organization.

To verify client information, which is known as evidence, clients can provide written or verbal evidence. The following list outlines examples of evidence verifications that might be required by a social enterprise organization:

- An original copy of a birth certificate.
- A fax from a doctor to certify a patient's inability to work.
- A telephone call from a parole officer to certify that a person met the required parole obligations.

The challenges of verification

The main challenges in verifying evidence are variations in verification requirements and the inefficiencies of verification rules. Cúram helps to address these challenges.

Variations in verification requirements

The following list outlines three reasons that verification requirements can vary:

- Verification requirements can vary by jurisdiction. For example, different verification requirement might apply to states and counties.
- Verification requirements often vary between agencies or between programs, products, or both.
- Verification requirements can change as a result of frequent changes to social welfare legislation.

Time requirements and inefficiencies

Currently, agencies implement verification rules by converting legislation into sets of rules that are coded directly into the application. As a result, changes to verification processing requires rebuilding and redeploying the application. For these reasons, defining and maintaining an organization's requirements can be both time-consuming and inefficient.

Using a flexible verification module

Cúram verification can address the difficulties that are associated with defining and maintaining an organization's requirements by providing a flexible verification module. So, a user can define both the evidence that requires verification and how to verify that evidence. Configure verifications at run time, that is, the application does not need to be rebuilt or redeployed to change verification requirements. By using a flexible verification module and by configuring verifications at run time means that caseworkers can efficiently manage verification processes that were complex and difficult to implement and maintain.

Components of verification

Verification consists of three components: an administration component, a case component, and a participant component.

Administration component

Use the administration component to customize aspects of verification. For example, the following list outlines three ways that the verification component can be customized:

- Customize restricted access to verifiable data.
- Customize specialized processing that is triggered by changes to verified evidence.
- Determine whether a verification is mandatory.

The verification settings can either be applied to one product or can be reused for multiple products. Verification settings can also be applied to an application case and participant evidence.

Case and participant components

- **Record verifications for evidence**
  Use the case and participant components of the verification process so that caseworkers can record verifications for evidence. To implement the process, the verification engine interprets the rules that were defined during verification administration. The rules include how to identify whether there are any verification requirements for a selected piece of evidence. During the maintenance of the evidence, the verification engine ensures that any rules that relate to the verification are implemented. For example, if two verification items are needed to satisfy a verification requirement then the relevant evidence cannot be activated unless two items are provided.
- **Fulfill verification requirements**
  Evidence and case list pages are provided to assist caseworkers to fulfill verification requirements. Caseworkers can also view verifications that are related to participant evidence from the participant manager. Caseworkers can use the pages to view either the full list of verifications or the outstanding, that is, unsatisfied, verifications. Caseworkers can also run other functions, for example caseworkers can add attachments, such as graphics files, to verification items.

Related information

1.2 Verification administration

Administrators can use the administration component of verification to define the verification rules that are associated with case and participant evidence.

The verification engine tree structure

The administration component of Cúram uses a tree view that displays verification elements based on the element's hierarchical relationships. The following list outlines the six verification elements:

- Verification categories.
- Verifiable data items.
- Verification item utilizations.
- Verification requirements.
- Verification requirement usages.
- Dependent data items.

Verification categories

Use a verification category to arrange evidence data into logical groups by grouping verifiable data items elements.

For example, an organization might define a list of evidence that relates to personal information: social security number, date of birth, place of birth, and income. The related evidence can then be grouped into a Personal verification category. Other verification categories might include employment information, financial information, or child support information.

Verifiable data items

Verifiable data items are pieces of evidence that require verification.

Mandatory evidence entity attributes

The piece of evidence that requires verification corresponds to a single attribute within a specific evidence entity, for example, an income amount on the income entity. The following list outlines the two attributes that must be entered for the evidence entity:

- The name of the entity. The name is stored in the evidence type field for an entity.
- The specific name of the attribute to be verified.

Security identifier (SID)

Verifiable data items also provide organization's with a built-in application security function. Organization's can use verifiable data items to enter a security identifier (SID) that can restrict users' rights to access sensitive verifications. If a user's security profile does not contain the SID that is entered in the field, then that user cannot access the verification. For more information about application security function, see the *Configuring the system* related link.

Related information

Verification item utilizations

Verification item utilizations define the verification items to use for a particular verifiable data item. A passport and a birth certificate are examples of verification items.

A verification item defines what can be used to verify the information that is provided by a participant. A passport, a birth certificate, a pay slip, or a medical certificate are examples of the verification item that can be used to verify the information. For specific verifiable data items provided by the participant, the data item might be verified in various ways. In such cases, multiple verification item utilizations are available for verifiable data items. To verify a date of birth, for example, participants can provide a birth certificate or a passport.

The following list provides an overview of seven configuration settings for verification item utilizations and specifies how the settings affect the runtime function.

Configuration settings for verification item utilization

The following list provides an overview of seven configuration settings for verification item utilizations and specifies how the settings affect the run time function.

- **From and To dates**
  You can define a set time period during which a verification item can be used to verify a verifiable data item. The property is set by defining a time period on the verification item utilization. After the time period, the verification item can't be used to verify the verifiable data item. After the time period, one of the defined alternative verification item utilizations must be used or a new verification item utilization must be configured for the verifiable data item.
- **Usage type**
  The usage type property defines how a verification item is used when multiple evidence records exist for a client for a particular evidence type that requires verification. Two values

can be set for a usage type: shared and unique. By default, the usage type for a verification item is shared.

When the usage type is set as shared and when the caseworker captures a verification item against the first evidence record, the document is applied by default to the other evidence records of this evidence type. The functionality applies where multiple evidence records exist for a client for a particular evidence type that requires verification. For example, a hospital receipt might be used to verify more than one medical expense. A client might have an asthma condition and an arthritis condition that are being treated in the same hospital. So, a receipt from the hospital might contain information about the amount that the client must pay for both conditions.

When the usage type is set as unique and when the caseworker captures a verification item against the first evidence record, the document is applied only to that record. The functionality applies where multiple evidence records exist for a client for a particular evidence type that requires verification. For example, a client has two part-time jobs and must verify the earnings evidence from both jobs by providing separate pay slips, that is, one pay slip for each job. In the scenario, set the verification item as unique so that when the client produces one pay slip the pay slip is not applied to the other earnings verification record.

- **Expiry and warning days**
  When the expiry day is set on verification item utilizations, the expiry date is calculated when proof is added for the verification. The date is determined based on the value configured for the Expiry Date From field, which can be Item Added, Item Received, or Verification Added. When the expiry date is reached, the system initiates a workflow event. If warning days are specified, the system notifies the case owner when the warning date is reached. Expiry date processing and due date processing for verification uses workflow functionality. For more information about expiry date and due date workflow processing, see the *Managing deadlines* related link.

> **Note:** Verification items for participant information do not expire. The time limits that apply to participant information do not apply to cases.

- **Level**
  The level property is used to indicate the level that is achieved by the verification item utilization. Levels are in the range 1 - 5 in ascending order. A level 1 item cannot satisfy a level 5 requirement. For example, a photocopy of a birth certificate might be considered a level 1 verification item, but the original birth certificate might be considered a level 5 verification item. As items are added, the verification engine compares the level setting of verification items against the level setting of the verification requirement to evaluate if the data item is verified.

- **Mandatory**
  The mandatory property is used to indicate that a verification item is required to verify a particular verifiable data item. If a verifiable data item uses associated verification items that must be supplied, the verifiable data item is not considered verified, regardless of the other added items, until all the mandatory verification items are supplied.

- **Security identifiers (SIDs) to add or remove items**
  The security identifiers (SIDs) to add or remove items are two properties that specify the SIDs that a user must provide to either add or remove a particular verification item for a verifiable data item. If a SID is not provided for either of these properties, then any user can perform the

action that is associated with that property. For example, if no SID is provided for the remove items SID property, then any user can remove a verification item.

- **Client-supplied**

  The client-supplied property is used to indicate whether a verification item is provided by a client for a particular verifiable data item. The property can be used, for example, during communications between the organization and the client to ensure that a client is not asked to supply a verification item that might be sourced elsewhere. No system processing is associated with the client-supplied property. The property is only used for informational purposes for the user.

Related information

Verification groups

Use verification groups in scenarios where a user must submit a combination of verification items to verify a piece of evidence.

For example, the following list outlines either of the verification items that clients can provide to verify citizenship evidence:

- A passport or a driver's license and a utility bill.
- A photocopy of a passport, a utility bill, and a bank statement.

In such a scenario, create three different verification groups with the same level. The verification requirement for the citizenship evidence is satisfied when all the verification items from any of the groups are submitted.

Verification group level

Each verification group is associated with a level that is used to indicate the level that is achieved when all the verification items of a group are provided. For example, if a level 5 is associated with a verification group, the verification engine considers a verification requirement of level 5 satisfied when all the verification items that are defined in a group are provided.

The user can also define verification item utilization settings for each of the verification items in a group. For more information about the settings for verification item utilizations, see the *Verification item utilizations* related link.

Related information

Verification configuration properties

You can configure verifications through the system administration properties. For example, for conditional verification you can prevent duplicate verifications from displaying in the user interface.

The VerificationFilteringEnabled property

The following table describes the `VerificationFilteringEnabled` property.

Table 1: The `VerificationFilteringEnabled` property.

Application property	Description
Property	`VerificationFilteringEnabled`
Default value	**True**
Description	The property controls the filtering of duplicate verifications in the user interface. The default value allows filtering behavior in the user interface component.

The preventingDuplicateVerificationInsertion property

The following table describes the `preventingDuplicateVerificationInsertion` property.

Table 2: The `preventingDuplicateVerificationInsertion` property.

Application property	Description
Property	`preventingDuplicateVerificationInsertion`
Default value	**True**
Description	The property prevents the insertion of duplicate verifications when verifications are created. The default value prevents new duplicate verifications from being inserted into the database.

The curam.verification.submittedDocuments.display.enabled property

The following table describes the
`curam.verification.submittedDocuments.display.enabled` property.

Table 3: The `curam.verification.submittedDocuments.display.enabled` property.

Application property	Description
Property	`curam.verification.submittedDocuments.display.enabled`
Default value	**No**
Description	The property determines whether documents that are submitted to verify evidence display with the associated verification. The default value does not display documents that are submitted to verify evidence with the associated verification.

Verification requirements

A verification requirement provides the rules of verification for a piece of data, that is, a verifiable data item.

Many variables are included in the verification requirement rules. The rules include where and how the rules apply at run time, for example, whether the verification engine must apply the rules to participant level data or to a specific case. Date of birth is an example of a verifiable data item. For certain organizations, the rules might verify the piece of data one time and so, the verification engine applies the rules within participant manager. For other organizations, the rules might require that date of birth is verified at a program level and, so, the verification engine applies the

rules to a specific case. For more information about verification requirements, see the *Verification requirement usages* related link.

Related information

Verification requirement properties

You can configure seven properties on a verification requirement. Level and minimum items are two examples of the properties to configure.

- **Due date and warning date**

  Use various properties to set a due date on a verification. The due date property specifies the number of days after a particular event that a verification must fall due. Administrators can also specify whether the number of due days is calculated from the date the case was created or from the date evidence was inserted or received.

  The warning date property specifies the number of days prior notice that a case owner receives before the verification due date. If no warning date is specified, a case owner does not receive a warning before the verification due date. Due date processing for verification requirements uses workflow functionality.

  For more information about expiry date and due date workflow processing, see the *Managing deadlines* related link.

- **Level**

  The level property indicates the level of verification to achieve to consider data verified. The system doesn't consider evidence verified unless a verification item with the appropriate level is received. For example, if a verification requirement specifies that a level 5 verification item, for example an original birth certificate, then providing a level 1 item, for example a photocopy of a birth certificate, does not satisfy the verification requirement. Alternatively, providing a combination of verification items that form a verification group of level 5 satisfies the verification requirement.

- **From and To dates**

  Use the From and To date properties to indicate the time period that the verification requirement is effective. The From and To date properties interact with the effective dates of verification item utilizations and the effective dates of evidence to determine the verifications that a caseworker can perform. For example, a requirement to verify that an income amount might be defined as effective from January to December. However, one verification item might be defined as effective from January to July 31, for example, a pay slip, while another is defined to be effective from 1 July to December, for example, a tax return.

  For the month of July only, both pay slip and tax return display on the verification list for the caseworker to select to add the proof. The date that the caseworker selects the proof for the income evidence determines the verification items that are required to satisfy the verification requirement.

  > **Note:** The To Date of the verification item utilization is intended only to control the items that are displayed to a caseworker when adding proof. The To date is not used in the calculation that determines if the item used to verify has expired. Instead, Expiry Days and Expiry Date From are used to verify that a piece of evidence has expired.

- **Minimum items**

  Use the minimum items property to specify the minimum number of verification items that must be provided before the system considers data verified. For example, if the minimum item specified is 2 then the system considers the verification requirement satisfied if at least two verification items or verification groups are provided. When all the verification items that are specified in a verification group are provided, the verification engine considers it a single item. Clients can also provide a combination of verification items and groups to satisfy the minimum number of verification items of a verification requirement.

- **Mandatory**

  Use the mandatory property to indicate whether the verification requirement is mandatory. A mandatory verification requirement means that evidence and cases that are associated with the verification might not be activated until the rules that are defined for the verification are met. When the mandatory property is not set, the verification requirement is optional and, so, the system can activate evidence that is associated with the verification even if the evidence is not verified.

- **Client-supplied**

  Use the client-supplied property to indicate whether a verification item is provided by a client for a particular verifiable data item. For example, you can use the property during communications between the organization and the client to ensure that a client is not asked to supply a verification item that might be sourced elsewhere. No system processing is associated with the client-supplied property. The property is only used for informational purposes for the user.

- **Reverification**

  Use the reverification property so that the user can specify the verification engine's response to changes to `Active` evidence. The reverification property does not apply to participant evidence. The following list provides the names of the settings for the property and specifies how the settings affect the property:

  - **Reverify always**

    If a caseworker changes `Active` evidence, no previously met verification requirements are transferred to the new `In Edit` evidence. The new `In Edit` record must then be reverified.

  - **Reverify if changed**

    If a caseworker changes `Active` evidence and the value that the caseworker entered for the verifiable data item or any dependent data items remains the same, the existing verification information on the `Active` record is copied to the new `In Edit` record. If the value that the caseworker entered for the data item or for any dependent data items changes, then no verification information is copied from the `Active` record.

  - **Never reverify**

    If a caseworker changes `Active` evidence, the verification information on the `Active` record is always copied to the `In Edit` record.

Related information

Conditional verifications

Define the conditional verifications as verifications that are based on a set of conditions rather than verifications that are based only on added or modified evidence.

Verification engine

When evidence is added or modified, the verification engine checks the current specified conditions. However, the verification engine creates an outstanding verification record only when a condition that is defined is met. The verification engine does not create an outstanding verification record each time a verifiable data item is added or modified. The conditions can range from conditions against the value of the verifiable data item to more complex conditions where the values of a set of dependent evidences determine whether verification is required.

Examples

For example, a verification might be required only when the value of earnings is more than $200 per week or only where the alternate ID is of type SSN. A set of dependent evidences is another, more complex, example. Eligibility for an income assistance program, for example, that requires verification of a household income evidence type when the income is more than $1,150 per month. The household income evidence type consists of multiple income evidence types, for example dividends, pension, wages, and salaries. The verification is configured for the income amount of the household income evidence type. However, the verification engine reevaluates whether the household income requires verification when the income of any dependent evidence types, that is, dividends, pension, wages, and salaries, changes.

Rule classes

The verification engine can permit creating a conditional verification so that the user can associate a rule class. The organization must provide the rule classes that define the conditions for the verifiable data item. To use conditional verifications that suit specific business scenarios, the organization must provide three items: a rule class, a display rule class, and a display UIM.

- **Rule class**
  A mandatory rule class is required that defines the conditions for the verification to trigger for the verifiable data item.
- **Display rule class**
  If required, a rule class that defines how the results of the verification are displayed.
- **Display UIM**
  If required, a UIM page reference to display the results of the conditional verifications in the verifications page.

Workflow events for verification requirements

Your organization can use five optional workflow events to suit specific business scenarios. The five optional workflow events are in addition to due date workflow processing.

Workflow events and triggers

The following list outlines the workflow events and describes what triggers each event.

- **Due date event**

  Reaching the verification due date triggers the due date event.
- **Expiry date event**

  Specifying an expiry date triggers the expiry date event.
- **Add event**

  Creating a verification for the requirement triggers the add event.
- **Update event**

  Updating the verification by adding or removing a verification item triggers the update event.
- **Value changed event**

  Changing the value of the verifiable evidence triggers the value changed event.

Use the workflow events to integrate the verification process with workflow functionality.

> **Note:** If your organization wants to enact workflows by using the events, a software developer must customize application code to support the implementation. For more information about workflows, see the *Cúram Workflow Overview Guide* related link.

Related information

Verification requirement usages

The verification engine permits different types of cases to use an individual verification requirement. A verification requirement usage means that administrators can associate specific case types with specific verification requirements. In practice, an administrator can specify different evidence verification requirements for different types of cases.

For example, a client's income amount is captured at the integrated case level. If there is a requirement to verify the income amount, multiple cases within the integrated case can use the requirement.

The following list outlines the scenarios that verification requirement usages can accommodate:

- Applying verification rules to groups of cases, that is, all the cases within an integrated case
- Applying verification rules separately to individual cases

A verification requirement usage is also available for participant evidence and permits an administrator to specify different evidence verification requirements for participant evidence.

Dependent data items

Dependent data items are specific pieces of evidence that directly influence the verification of a related data item.

Example

If your organization wants to verify why a household member was absent from the household, the length of the absence might be an important fact to record for the verification. For example, the **Absence Reason** is the verifiable data item. The To and From dates of the absence are the dependent data items. The verification engine treats any change to a dependent data item the same way it treats a change to the verifiable data item.

For a dependent data item, the properties that you must store include a unique name and the name of a specific data item. The **Data Item** that a caseworker enters for the dependent data item must reference an attribute from the evidence type that is specified in the parent verifiable data item.

1.3 Verification for caseworkers during the evidence lifecycle

The verification engine uses the rules that are specified in the verification administration component to perform verification processing for evidence. The system calls the verification engine as part of maintaining case evidence and of maintaining participant data for evidence. The system also calls the verification engine when verifications are added or modified.

> **Note:** During all of the verification processes for caseworkers, the verification engine considers security settings that are implemented within the verification setting for a piece of evidence. For example, if the caseworker does not have the security privileges to add a verification item then the caseworker cannot view or change the verification item.

Capturing evidence

The verification engine handles case evidence and participant evidence in different ways.

Case evidence

When evidence is captured for a case, the verification engine is called to determine whether any of the evidence data requires verification. If a piece of data requires verification, the verification engine checks to see whether verifications are required for the case type where the evidence is captured. For shared evidence that is captured for a case, the verification engine determines whether the application case, the integrated case, or its product deliveries, where product deliveries apply, requires verifying the evidence. All product deliveries that are not closed are considered. If the evidence requires verification requirements, a list of the requirements is displayed as a message to the caseworker.

Participant evidence

When participant evidence is captured, it can be verified separately to any case usage of the evidence. When participant evidence is captured, it is automatically activated. So, any mandatory verification that is defined about the participant evidence exists against the active evidence. For the caseworker, the verifications are displayed in the verifications list at the person and the case level. Case level processing does not affect participant level verifications. Even where there are outstanding mandatory participant verifications, the verification engine permits activating the cases for that participant. Eligibility and entitlement processing is not affected. Where verifying participant data, for example verifying a person's date of birth or SSN, must affect the case level processing, associate the evidence with the case, and configure case level verifications.

Deferring verifications

You can separate evaluating verification from the maintaining evidence function. As a result, evaluating verifications can be deferred.

Typically, verifications are evaluated when evidence is captured. Evaluating evidence when it is captured applies to all evidence maintenance functions. By default, evaluating verification requirements is tightly linked to evidence maintenance functions. Caseworkers cannot defer verifications, but the function can be built into a business process by using APIs provided with the product. The feature supports implementing a business process that involves multiple discrete steps rather than implementing a single transaction. Implementing a business process as many discrete steps means that recovery is easier. When problems arise, the system only rollbacks one discrete step rather than the entire business process. When evidence is inserted and verification is evaluated as two discrete steps in a business process, if problems arise in running verifications the most rollback that is required is evaluating verifications. The evidence is still captured and stored on the system.

Unevaluated evidence

When verifications are not evaluated for evidence, the evidence is in an unevaluated state. The unevaluated state is displayed to users on various evidence list pages. The state uses a special icon against the evidence that specifically represents unevaluated evidence. Unevaluated evidence means that when evidence was inserted the verification engine was not called. So, the system cannot determine whether any of the evidence data requires verification. For evidence in an unevaluated state, a user cannot evaluate verifications. The expectation is that the business process that programmatically deferred verifying the evidence is resumed by a suitable recovery technique and programmatically processes unevaluated evidences before the business process completes.

Viewing verification requirements

Caseworkers can use multiple ways to view data that requires verification.

The following list outlines the verifications that a caseworker can view in the evidence area of a case:

- Verifications that are associated with the case.
- Verifications that are associated with a particular evidence type.
- Verifications that are associated with a particular piece of evidence.

Verifications lists are also displayed on the **Person** home page so that a caseworker can see verifications that are configured on participant evidence. The following list outlines the information that is provided by the listed verification requirements:

- General information, for example the name of the verifiable data item.
- Information about whether a verification requirement is mandatory.
- Information about whether a verification requirement is satisfied.
- Information whether items were received for the verification requirement when the verification requirement is outstanding.

Caseworkers can use the information that is provided by the listed verification requirements to determine whether verification items must be added, modified, or removed for a particular piece of evidence.

Lists of verification requirements

Six pages list verification requirements. The lists provide information about evidence types, evidence objects, integrated case verifications, product delivery verifications, participant verifications, and application case verifications.

Evidence type list

Evidence type list pages can display all verifications that are specific to the applicable evidence type for the current case. This list displays the verification requirements that are defined for a specific evidence type. Verification items might be provided for a particular verification requirement, but the items are applied to the evidence. So, the items might be used to satisfy other verifications that are required for that evidence, for example on other cases.

Evidence object

Lists can display verifications that are specific to a specific piece of evidence. If verifications are defined for a specific piece of evidence and the evidence object changes over time, the evidence object might need to be reverified as the evidence is corrected or changed. As the group of verifications relate to the same evidence object, it might be useful to look at the group of verifications together.

Integrated case verifications

The integrated case verifications list displays the verification requirements that are associated with a specific integrated case. The following list outlines the two parts of the integrated case verifications list:

- A list of current verifications.
- A list of outstanding verifications.

The overall list contains only verification requirements that are defined for the integrated case. The overall list includes verifications that are associated with superseded evidence. The list does not contain any verification requirements that are defined for product deliveries that are present within the integrated case. Verification requirements that are associated with deleted evidence are only displayed if the application property to display deleted evidence is configured.

Product delivery verifications

The product delivery verifications list displays all verification requirements that are associated with a specific product delivery. The following list outlines the two parts of the product delivery verifications list:

- A list of current verifications.
- A list of outstanding verifications.

The overall list contains all verification requirements that are defined for the product delivery. The overall list includes verifications that are associated with superseded evidence. Verification

requirements that are associated with deleted evidence are only displayed if the application property to display deleted evidence is configured.

Participant verifications

Caseworkers can view the verification requirements for participant data in the participant manager for the applicable evidence type page. Caseworkers can also add verification items from the evident type pages. From the participant manager home page, caseworkers can view lists of verifications and can view outstanding verifications for all participant evidence types. Verification requirements that are associated with canceled or superseded evidence is not displayed in the list.

Application case verifications

The application case verifications list displays all verification requirements that are associated with a specific application case. The following list outlines the two parts of the application case verifications list:

- A list of current verifications.
- A list of outstanding verifications.

The overall list contains only verification requirements that are defined for the application case. Verification requirements that are associated with canceled or superseded evidence are not displayed in the list.

Verifying evidence

Verifying evidence is the process of adding verification items that satisfy the verification rules for evidence. Caseworkers can verify case evidence and participant evidence.

Managing deadlines

Your organization can configure an expiration period on a verification item after which the item is no longer valid. Your organization can also specify the number of days after a particular event occurred that the verification is due.

The following list outlines the options for the due date event:

- The date on which the evidence that is associated with a verification was entered.
- The date on which the evidence that is associated with the verification was received, that is, the receipt date that is present on the evidence descriptor.
- The date on which the case for which the evidence is being recorded was created.
- **Deadline**
  When an administrator creates a verification, the due date is calculated by adding the number of due days that are defined to the date on which the specified event occurred. An administrator can also specify a warning date. A warning date indicates the number of days before the due date on which the caseworker is notified of the outstanding verification.
  If a verification is satisfied before the associated deadline is reached, the deadline is then monitored unless the status of the verification changes.

- **Expiry date**
  The following list outlines the options for calculating the expiry date when an administrator adds a verification item to a verification requirement:

  - Add the number of expiry days to the date the verification is added.
  - Add the number of expiry days to the date the item is added.

  If expiry dates are specified, the system initiates a workflow event. If warning dates are specified, the system sends a notification of the upcoming verification expiry date to the case owner. When the expiry date is reached, the system initiates the date the expiry date event must be administered.

> **Note:** The system does not maintain due date functionality for participant verifications because the criterion that is used to define due date applies only to cases. For example, the date on which the case was created.

Modify due date and workflow

Caseworkers can modify the due date that is associated with a verification requirement. The business processing that occurs in response to the deadline management functionality is defined by a sample workflow.

Modify due date

The caseworker modifies the due date that is associated with a verification requirement. In such cases, due dates can be modified only if the verification due date is defined as modifiable within the verification administration component. When the caseworker modifies a due date, the caseworker can increase or decrease the number of days before the verification item is due.

Workflow

The business processing that occurs in response to the deadline management functionality is defined by a sample workflow that is run when a verification with a deadline is created. A similar sample workflow is run when a verification item with an expiry date is created. When a verification due date elapses where the verification is not satisfied, the corresponding processing varies by program type and by jurisdiction. So, the processing that is run within the sample workflow is not mandated. An agency might instead define its own workflow process to meet agency specific-verification processing requirements. The following list outlines the main activities that are run within the sample due date workflow for a verification requirement:

1. The system notifies the caseworker. If warning days are specified, a communication is sent to the client ahead of the deadline date.
2. The system notifies the caseworker when the due date is reached.
3. The system closes the case when the deadline is reached.

The following list outlines the main activities that are run within the sample expiry date workflow for a verification item:

1. The system notifies the caseworker. If warning days are specified, a communication is sent to the client ahead of the deadline date.
2. The system notifies the caseworker when the due date is reached.

3. The system expires the item. The item can't be used to verify the requirement when the verification item is mandatory or when the verification item is required to meet the minimum items for the requirement. The verification status is then set to **Not Verified**.

Bypassing mandatory verifications

When a verification is defined as mandatory, typically that verification must be satisfied before the evidence can be activated and used as part of eligibility and entitlement calculations. However, there are programs where a client is allowed a specified time period during which the program proceeds into the delivery stage while the client provides the required verification documentation. If the client does not provide the required verification documentation during the specified period, then delivery of the program stops.

Verification waiver

To permit the verification waiver functionality, define the verification as mandatory but allow the mandatory verification to be waived for the specified time period by creating the applicable verification waiver entries. For example, expedited food stamps means that clients can receive a benefit earlier than standard food stamps and for the first month the verifications are not mandatory.

You cannot manually create verification waiver entries in the interface. Instead, create the entries in the context of the business process or program that handles the waiver period. The verification engine provides the necessary APIs to support a business process to create the entries. Where there are verification waiver entries, the verification engine checks for current entries as part of the business rules that check for outstanding mandatory verifications.

While a mandatory verification is bypassed, the application continues to indicate the verification as outstanding but indicates that the verification is bypassed. For each verification, the system maintains a history of verification waiver entries. The following list outlines what users can view the history to determine:

- Whether a piece of evidence bypassed verification
- If the evidence bypassed verification, users can view the duration of time for which the verification was bypassed.

Customizing the verification waiver

A verification waiver includes an optional productID value. Use the productID value to create verification waiver entries to customize the verification waiver that activating product delivery to a specific product uses. As a result, you can define evidence as common evidence at integrated case level and shared across product delivery cases, but still provide product-specific behavior for mandatory verifications against the evidence. You must also turn on the functionality by using the application property `curam.miscapp.considerproductidforicwaivers`. If you do not set `curam.miscapp.considerproductidforicwaivers` to **Yes**, the activate product delivery functionality identifies the general verification waiver entries only and ignores any product-specific verification waivers.

When evidence is activated, the system identifies mandatory verifications against the evidence. For each mandatory verification, the verification engine checks if any waiver records exist for the current period. Where waiver records exist for the current period, the evidence is activated. For

the business rule, it does not matter whether the verification waiver is configured to be product-specific or general. When there is as a waiver record, the evidence is activated.

When a product delivery is activated within the integrated case, the system identifies mandatory verifications against evidences that are in the `Active` state on the integrated case. For each mandatory verification, the verification engine checks whether to waive the verification. If the application property `curam.miscapp.considerproductidforicwaivers` is set to **No**, the verification engine tries to identify a verification waiver. The verification engine ignores a waiver even where a productID is specified. If a waiver is found, the product delivery is activated. So, where products in an integrated case share evidence and the evidence uses a general verification waiver, each product moves to the delivery stage.

If the application property `curam.miscapp.considerproductidforicwaivers` is set to **Yes**, the verification engine checks for product-specific verification waiver records for the outstanding mandatory verification on the integrated case. If a record is found that matches the product that is being activated, then the system can activate the product delivery. When `curam.miscapp.considerproductidforicwaivers` is set to **Yes** and all products must be waived, the system must create a product-specific verification waiver record for each individual product. So, where products in an integrated case share the same evidence with a mandatory verification that is configured on the integrated case but the evidence includes one or more product-specific verification waivers, some products can move to delivery. However, other products must wait for the verifications from the client. As a result, the functionality gives customers a more granular level of control.

The business process or program that handles the waiver periods is the business process or program that must create the verification waiver records. Any product that is configured to permit the bypassing of mandatory verifications must also ensure that the product rules are modified to ensure that bypassed evidence is only used for the time period that is specified on the verification waiver table.

Satisfying verification rules and adding a verification item
To meet the verification requirements that are defined for evidence, the verification items that caseworkers provide meet specific criteria. The add verification item process declares that a verification item is provided to confirm the accuracy of entered evidence.

Satisfying verification rules

The following list outlines the rules that the verification items that caseworkers provide must meet to satisfy the verification requirements that are defined for evidence:

1. The level of a verification item or a verification group must be at least the same level as the level that is defined for the verification requirement.
2. If a minimum number of items are defined for the verification requirement, then that number of items, at least, must be provided. When all verification items of a group are provided, the verification engine views the group as one item.
3. If a specific verification item is defined as mandatory, then the item must be provided unless the verification is bypassed. The verification engine checks all product delivery cases that are not closed or suspended. A hook point is provided to implement custom conditions that suit specific business needs to exclude mandatory verification requirements from activating an evidence.

4. The items that are provided for a verification requirement must be valid for the date range that is specified in the verification requirement.

To satisfy a verification requirement, all rules must be met. For example, if a verification requirement is defined as level 5 and, for example, requires an original copy of a birth certificate, and requires two items. Then, the verification requirement cannot be satisfied by one level 1 item, for example a photocopy of a birth certificate. To fully satisfy the requirement, at least two verification items must be provided and both verification items must be level 5 items.

When a verification item is added that meets the verification requirement of more than one evidence item, the verification item can be propagated forward. The following list outlines the only circumstances in which the items are propagated forward across each instance of the evidence:

• The reverification mode for the requirement is set to Never Reverify.
• The reverification mode is set to Reverify if Changed and the evidence was not changed.

Adding a verification item

Use the add verification item process to declare that an item of verification is provided to confirm the accuracy of entered evidence. When the caseworker adds a verification item, only items that are valid for the period that is defined in the verification requirement are displayed to the caseworker.

The caseworker can add the verification item and the caseworker can apply the verification item to multiple evidences on a case that requires verification. The evidence to which the verification item is applied can be the same evidence type or a different evidence type. The verification item can also be within the context of one participant or across multiple participants on the case.

During the process, the caseworker can also add an attachment that relates to the verification item. Caseworkers can also add attachments to verification item provisions to provide an electronic record of a verification. Attachments can be in graphic or written form.

Reviewing documents submitted for verifications

To confirm the accuracy of evidence entered by a citizen, organizations can allow citizens to upload documents as part of the add verification item process.

In an organization's business process, an organization can use the to include the online submission of documents. When information that is provided by citizens must be verified with supporting documentation, citizens can upload documents by using the . For more information, see the *Verify* related link.

By using the submitted document review feature for citizen verifications, caseworkers can view and manage the uploaded documents. For more information about configuring the caseworker view of submitted documents, see the *Verification configuration properties* related link.

Viewing submitted documents

When a citizen uses the IBM® Universal Access Responsive Web Application to upload documents, the documents are awaiting review. The documents are displayed with the outstanding verifications for which the documents were uploaded. For any verifications with submitted documents that are awaiting review, a new submitted document icon is displayed

against the verifications to notify the caseworker. When all the submitted documents are reviewed for the verification, the icon is not displayed.

Managing submitted documents

Depending on the suitability of the documents that citizens submit as valid proof, the documents can be used to verify evidence. Caseworkers can manage submitted documents by accepting or rejecting the documents as valid proof for verifying evidence.

The following table outlines the four options that caseworkers can use to manage submitted documents.

Table 4: Options that caseworkers can use to manage submitted documents.

Option	Explanation
Accept	Caseworkers click to accept an individual document as valid proof to verify evidence.
Reject	Caseworkers click to reject an individual document as invalid proof to verify evidence.
Accept All	Caseworkers click to accept multiple documents as valid proof to verify evidence.
Reject All	Caseworkers click to reject multiple documents as invalid proof to verify evidence.

Related concepts

Verification configuration properties on page 14
You can configure verifications through the system administration properties. For example, for conditional verification you can prevent duplicate verifications from displaying in the user interface.

Related information

Modifying and removing evidence

Modifying and removing evidence are the two types of evidence changes that can affect verification.

How evidence changes affect verification

The effect of modifying evidence on a verification requirement depends on whether the applicable evidence is `Active` or `In Edit`. However, the effect of removing evidence on a verification requirement does not depend on whether the evidence is activated.

The processing for evidence changes to verifiable data items also applies to any dependent data items. For example, evidence might contain a date of birth verifiable data item with a dependent place of birth data item. Here, any changes to the date of birth dependent data item triggers the same processing that applies for the date of birth verifiable data item.

Modifying `In Edit` evidence

The effect of modifying `In Edit` evidence that requires verification depends on whether the verification items are provided. If no verification items are provided, then no verification processing is required. For example, the `In Edit` evidence for a person's date of birth might require verification. However, if the caseworker did not provide a verification item, for example a birth certificate, then modifying the evidence does not trigger verification processing.

If the caseworker provides a verification item for the `In Edit` evidence, a message is displayed. The message lists each verifiable data item that is affected by modifying the evidence. For example, the message might read: "The changes that you have made may affect the verification information that is recorded for the following item(s): Date of Birth. Please review this verification information." Here, "Date of Birth" refers to the name of the verifiable data item.

If verification is provided, then the verification engine raises a workflow event for every verification requirement that contains a value-changed workflow event as defined in the administration component. The event runs regardless of whether the data meets any or all of the verification requirements. For each verification requirement, the value changed event is raised only one time.

Modifying `Active` evidence

When a caseworker modifies an active evidence record, the system creates a new `In Edit` evidence record. From the perspective of the verification engine, creating a new `In Edit` record is identical to creating a new `In Edit` record when evidence is first added. The verification information that is recorded for the new `In Edit` record is independent of the information that is recorded for the `Active` record. Effectively, a new piece of data is recorded.

However, the reverification mode that is defined for a verification requirement determines whether verification information from the previously active evidence record is copied to the newly created `In Edit` record. The following list outlines the three reverification modes:

• Reverify Always.
• Reverify If Changed.
• Never Reverify.

The reverification modes do not apply to modifications to active participant evidence. For more information about the reverification modes, see the *Verifications requirements* related link.

If information is copied to the `In Edit` record, the new verification information for the record is maintained separately from any verification that was associated with the previous `Active` evidence record. There is no link between the previous verification information and new verification information.

Removing evidence

When a caseworker removes evidence, the associated verifications are not affected. However, the caseworker cannot modify or change any verification that is associated with the removed evidence.

Related information

Activating evidence and cases

Evidence cannot be activated unless the mandatory verification requirements are met for a piece of evidence. A case cannot be activated until all mandatory verification requirements are met for the evidence that is associated with the case or a current verification waiver exists for the mandatory verification.

Activating evidence

When a caseworker tries to activate evidence, the system calls the verification engine to verify whether there are outstanding mandatory verification requirements. If all mandatory verification requirements are satisfied, then the verification engine does not stop activating the evidence.

If there are mandatory verification requirements that are not satisfied, then the verification engine stops activating the evidence. A message is displayed to the caseworker. The message indicates that mandatory verification requirements must be satisfied before the evidence can be activated.

A hook is available that supports bypassing mandatory verifications for a specified time period. By using the hook, evidence can be activated and used in eligibility and entitlement calculations, even though there are mandatory verifications for the evidence. For more information about the hook, see the *Bypassing mandatory verifications* related link.

Activating cases

When a caseworker tries to activate a case, the system calls the verification engine to check that all mandatory verifications that are associated with active evidence are satisfied or that a current verification waiver exists for the mandatory verification. If the system finds evidence with unsatisfied mandatory verification requirements, the verification engine stops activating the case and displays a message to the caseworker. The message indicates that mandatory verification requirements must be satisfied before the case can be activated.

Related information

1.4 Verification customization points

The options to customize verification points include using a hook point to skip mandatory verifications and using conditional verifications to determine whether verification is applicable for an evidence.

Supporting extra case types for verifications

To add a case type to support verifications, include the new type of case in the `CT_VerificationTypeCode` code table. The new case type corresponds to the relatedItemID of the verification requirement usage entity, for example product delivery or integrated case.

Code table values

The possible values for each code in the `CT_VerificationTypeCode` code table correspond to a **relatedItemType** field in the verification requirement usage entity. For example, by

selecting an integrated case as verification type code then the possible values are the recorded individual integrated case types. The possible values for the case type are the same as the value of `caseType` inserted in the case header table when creating the corresponding case.

An entry must also be created in the `Curam-config.xml` for the domain definition that is used for `RelatedTypes` so that it displays in the search option in the administration section. When a verification requirement is configured and applied in the administration section, the verification engine processes the same verification requirement.

Extend verifications support for a new case type

To extend verifications support for a new case type, four steps are required. By extending verifications support, customers can customize the case functionality to better suit the customers' specific requirements.

The following list outlines the steps that are required to extend verifications support for a new case type:

1. Create a `VerificationType` code table entry. The entry displays an extra item in the **Applies To** drop-down in the verification requirement usage page. The page maps to the relatedItemId value in the `VerificationRequirementUsage` entity.
2. Create a search page to display when the new item is selected in the drop-down. The selection maps to the relatedItemType value in the `VerificationRequirementUsage` entity. Create a user interface (UI) page and configure the page as a search page in the `Curam-config.xml`. For more information about performing the steps, see the *Cúram web client reference* related link.
3. Implement the `VerificationRelatedTypeHook`. The implementation must be bound with the newly created `VerificationType` code table entry.
4. At run time, the customer must, by using the caseID of the case that is created in the application, be able to determine the relatedItemType attribute. The customer must implement the logic in the implementation of another hook, `VerificationRelatedItemHook`. The `VerificationRelatedItemHook` hook must be bound with the `CaseTypeCode` for the new case type.

Related information

Hook to consider extra case participants

Use the hook point `curam.verification.impl.VerificationHookForProductDelivery` where an extra case participant must be considered for checking verifications on a product delivery case activation.

The `curam.verification.impl.VerificationHookForProductDelivery` hook point

When the verification engine is processing a product delivery case, the verification engine's default behavior is to check for the verifications on the primary client and case members of the case. Where an extra case participant must be considered for checking verifications on a product delivery case activation, use the hook point `curam.verification.impl.VerificationHookForProductDelivery`. Ensure that implementing the hook point is bound to the productid of the product delivery case.

The `ConcernRoleIDList.getOtherParticipantsForProductDelivery(CaseKey pdCaseKey)` is the only method to implement. The verification engine starts the method when the verification engine checks verifications for the product delivery case. Implementing the method returns extra case participants. Ensure that checks are performed on the extra case participants that are returned.

Skipping mandatory verifications

Customers can use the `skipVerificationCheck()` hook to perform custom processing to determine whether the case evidence must be verified before the system activates evidence.

The `skipVerificationCheck()` hook

The `VerificationHook` class supports the custom implementation of hook points at various stages during the evidence verification processing. The `VerificationHook` class is applied per product. The custom implementation is configured by using the Guice dependency injection mechanism.

Customers can use the `skipVerificationCheck()` hook to perform custom processing to determine whether the case evidence must be verified before the system activates evidence. Evidence is activated where the product delivery cases satisfy the conditions that are provided in the hook implementation. Evidence is activated even if there are any mandatory outstanding verifications for any of the products that use the evidence. Customers can use the hook point to apply customers' own custom conditions.

Implementing verification waivers

Depending on the rules that govern the product or the verification, you can use verification waivers to bypass a mandatory verification for a set time period.

Verification waivers

Even if you use mandatory verification for an evidence, you can use verification waivers to activate evidence for a set time period. To use verification waivers, you must implement the `curam.verification.sl.infrastructure.impl.EvidenceVerificationWaiver` interface. You must bind the implementation by using Google Guice MapBinder with a key. The key must be in a specific format for case evidence and for participant evidence.

The following sections list the format for case evidence and for participant evidence. Ensure that you use a period, that is, `.`, to separate codes.

When the evidence is activated and there is a waiver, the evidence is activated. The evidence is activated regardless of the waiver's duration and even if a verification requirement is mandatory but no verification items exist. Ensure that product rules are updated to accommodate the waivers for eligibility and entitlement. If you are using Cúram Express Rules (CER), then you can use propagator mechanisms to update the rules object.

In the sample component, a sample implementation of the interface is provided for Sporting Grant.

Case evidence

The following table outlines when to use the format `[CaseTypeCode].[CaseSubTypeCode].[EvidenceTypeCode]` for case evidence.

Table 5:

Format	Explanation
`[CaseTypeCode]`	A code from the code table `CaseTypeCode`.
`[CaseSubTypeCode]`	The code depends on the case types. The following list outlines the example code tables from which the codes are taken: • `ProductCategory`: integrated case type when the case type is integrated case • `ProductType`: product type when the case type is product delivery case • `InvestigateConfigType`: investigation configuration type when the case type is investigation case • `ScreeningNameCode`: screening name code when the case type is screening case
`EvidenceTypeCode`	A code from the code table `CaseEvidenceTypeCode`.

Participant evidence

The following table outlines when to use the format `[ConcernRoleType].[EvidenceTypeCode]` for participant evidence.

Table 6:

Format	Explanation
`[ConcernRoleType]`	A code from the code table `ConcernRoleType`.
`[EvidenceTypeCode]`	A code from the code table `CaseEvidenceTypeCode`.

Implementing custom conditional verification rules

Use conditional verifications to determine whether verification is applicable for an evidence. Conditional verifications are based on the conditions that are associated to the evidence that requires the verification.

Conditional verifications

Rule-class implementations support conditional verifications. Verification for a piece of evidence is determined by the set of conditions that are listed in the rule classes. When evidence is added or modified, the verification engine checks the specified conditions. The verification engine creates an outstanding verification only when a defined condition is met each time a verifiable data item is added or modified. The conditions can vary from conditions against the value of the verifiable data item to more complex conditions where the values of a set of dependent evidences determine whether verification is required.

Rule artifacts that are supplied by the verification framework

To facilitate integration between the verification framework and the rule implementations that are supplied by other components, the verification framework supplies the core rule artifacts. The artifacts contain abstract rule classes to which the rule implementations of other components must conform.

Rule sets

The rule set `VerificationRuleSet` is included in the verification framework. The `VerificationRuleSet` rule set contains all the framework's artifacts, for example the rule classes and the data container classes.

Rule classes

The following list outlines the rule classes that are included in the `VerificationRuleSet` rule set:

- `VerificationDeterminator`
- `VerificationDeterminatorResult`
- `VerificationDeterminatorParams`

`VerificationDeterminator`

Use the rule class `VerificationDeterminator` for the business logic that determines whether conditional verification is required for particular evidence types. Components that create rule implementations must adhere to the specification by directly or indirectly extending the class. The following table outlines the attributes that are available in the `VerificationDeterminator` rule class.

Rule attribute name	Type	Purpose
determine	`VerificationDeterminatorResult`	The implementation contains the business logic that determines the output of conditional verification. A **TRUE** value indicates to the evidence framework that verifications do not apply to the evidence. A **FALSE** value indicates to the evidence framework that verifications must be explicitly added.
verificationDeterminatorParams	`VerificationDeterminatorParams`	The attribute is populated by the conditional verifications framework. The attribute contains the values for all the input parameters for a specific instance.

`VerificationDeterminatorResult`

The rule class `VerificationDeterminatorResult` is a data container that stores the results of business logic in the `VerificationDeterminator`. The following list outlines the two attributes of the class:

- Result is a Boolean that states if verification is required for an evidence.

- Reason is a code table value from `VerificationSkippedReason` that contains the values of reason for which the conditional verification does not apply.

The rule implementation must create or populate the attribute so that the verification framework, after the framework examines the state of the attribute, can apply the appropriate business decisions.

VerificationDeterminatorParams

While the verification framework determines whether conditional verification is required, the framework supplies various input parameters to the rule implementation classes for various calculations. Example parameters include the evidence that is being edited and the associated case identifier for the evidence. The following table outlines the complete details of the input parameters.

Property name	Data type	Description
verifiableDataItemName	String	The property represents the name of the verifiable data item, for example person income and date of birth. The value is taken from the `VerifiableItemName` code table.
evidenceDescriptorID	Number	The unique identifier of the applicable evidence record.
caseID	Number	The unique identifier of the case with which the evidence is associated.

Propagator

The following list outlines the states of evidence to which verifications apply:

- `Active` evidence.
- `In Edit` evidence.
- Identical shared `In Edit` evidence.

Use the appropriate propagators to populate the rule objects for the Cúram Express Rules (CER). When evidence is created, modified, or deleted on a case, conditional verification rules are evaluated. So, ensure that your rules run when a limited set of evidence exists. Otherwise, technical errors in the evaluation of the conditional verifications might occur. If such errors occur, the errors display as runtime issues for users.

Notices

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the Merative website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of Merative

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of Merative.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

Merative reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by Merative, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

MERATIVE MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Merative or its licensors may have patents or pending patent applications covering subject matter described in this document. The furnishing of this documentation does not grant you any license to these patents.

Information concerning non-Merative products was obtained from the suppliers of those products, their published announcements or other publicly available sources. Merative has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-Merative products. Questions on the capabilities of non-Merative products should be addressed to the suppliers of those products.

Any references in this information to non-Merative websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those

websites are not part of the materials for this Merative product and use of those websites is at your own risk.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

The licensed program described in this document and all licensed material available for it are provided by Merative under terms of the Merative Client Agreement.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to Merative, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. Merative, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. Merative shall not be liable for any damages arising out of your use of the sample programs.

Privacy policy

The Merative privacy policy is available at https://www.merative.com/privacy.

Trademarks

Merative ™ and the Merative ™ logo are trademarks of Merative US L.P. in the United States and other countries.

IBM®, the IBM® logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

Adobe™, the Adobe™ logo, PostScript™, and the PostScript™ logo are either registered trademarks or trademarks of Adobe™ Systems Incorporated in the United States, and/or other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft™, Windows™, and the Windows™ logo are trademarks of Microsoft™ Corporation in the United States, other countries, or both.

UNIX™ is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.