IBM Cúram Social Program Management
Version 7.0.2

Cúram Verification Guide

Note

Before using this information and the product it supports, read the information in "Notices" on page 19

Edition

This edition applies to IBM® Cúram Social Program Management v7.0.2 and to all subsequent releases unless otherwise indicated in new editions.

Contents

Chapter 1. Cúram Verification Guide

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

Introduction

Purpose

The purpose of this guide is to describe the business processes that underpin Cúram Verification ™. In order to best understand these concepts, the guide should be read in full. The guide is not intended to be used as a training manual or user guide.

Audience

This guide is intended for business analysts employed by your organization. It is assumed that this audience has a strong knowledge of the organization's business requirements.

Prerequisites

Readers should be familiar with the application, specifically with the processes related to evidence maintenance and case processing. The `Cúram Integrated Case Management Guide` and the `Cúram Evidence Guide` should be read prior to reading this guide.

In addition, some understanding of Workflow and Administration functionality is useful in order to understand how Cúram Verification interacts with workflow processing and parts of the Administration component (e.g., application security). This information can be found in the `Cúram Workflow Overview Guide` and the `Cúram System Configuration Guide`, respectively.

Chapters in this Guide

The following list describes the chapters within this guide:

Understanding Verification
This chapter defines verification, reveals the challenges faced by organizations attempting to implement verification, and describes the benefits offered by Cúram Verification.

Verification Administration
This chapter describes the administration component of Cúram Verification, which is used to set up a structure of verification elements. This structure is the template for all verification processing within the application.

Verification for Caseworkers
This chapter describes the impact of verification on case maintenance, from initial evidence capture to adding and changing both evidence and the verification information recorded for that evidence.

Verification Customization Points
This chapters describes the customization points offered in Verifications, which allow customers to adapt Verification processing to suit their specific requirements.

Understanding Verification

What is Verification?

Verification is the process of checking the accuracy of the information given by clients seeking services from a Social Enterprise organization. The verification of client information (or "evidence") can take a number of forms; it can be provided by documents, e.g., birth certificates or bank statements, or by verbal means, e.g., telephone calls. Some examples of evidence verifications that might be required by a Social Enterprise organization are:

- An original copy of a birth certificate.
- A fax from a doctor's certifying a person's inability to work.
- A telephone call from a parole officer certifying that someone has met his or her parole obligations.

The Challenges of Verification

The process of verifying the evidence gathered by an organization has many challenges. Verification requirements can vary by jurisdiction; for example, states and counties may have different verification requirements. In addition, verification requirements often vary between agencies or programs/products. Finally, verification requirements can change as a result of frequent changes to social welfare legislation.

Currently, agencies implement verification rules by translating legislation into sets of rules that are coded directly into the application. This means that any changes to verification processing require the rebuilding and redeploying of the application. For these reasons, the definition and maintenance of an organization's requirements can be both time consuming and inefficient.

Cúram Verification addresses these difficulties by providing a flexible verification module that allows a user to define both the evidence that requires verification and the means by which that evidence can be verified. Verifications can be configured at runtime, which means that the application does not need to be rebuilt or redeployed in order to change verification requirements. Using these methods, Cúram Verification allows caseworkers to efficiently manage verification processes that have previously been complex and very difficult to implement and maintain.

Cúram Verification

Cúram Verification consists of three components; an administration component, a case component, and a participant component. The administration component provides the ability to customize many aspects of verification functionality, such as: limited access to verifiable data, specialized processing triggered by changes to verified evidence, and determining whether or not a verification is mandatory. These verification settings can either be applied to one product or can be re-used for multiple products. They can also be applied to an application case and participant evidence. “Verification Administration” on page 2 provides information on using the administration component of Cúram Verification to configure evidence verification requirements.

The case and participant components of Cúram Verification allows caseworkers to record verifications for evidence. In order to do this, the Cúram Verification engine interprets the rules defined during verification administration, identifying whether or not there are any verification requirements for a selected piece of evidence. During the maintenance of this evidence, the Cúram Verification engine will ensure that any rules pertaining to the verification are implemented. For example, if two verification items are needed to satisfy a verification requirement, then the evidence in question cannot be activated unless two items are provided.

Evidence and case list pages are provided to assist caseworkers in fulfilling verification requirements. Caseworkers can also view verifications related to participant evidence from the participant manager. These pages allow caseworkers to view either the full list of verifications or outstanding (unsatisfied) verifications. Caseworkers can also perform additional functionality such as adding attachments, e.g., graphics files, to verification items. “Verification for Caseworkers” on page 8 provides information on using the case management component of Cúram Verification.

Verification Administration

Introduction

The administration component of Cúram Verification allows administrators to define the verification rules that are associated with case and participant evidence. The following sections describe the elements used in verification administration and relate these components to real-world examples of evidence verification.

Verification Engine Tree Structure

The administration component of Cúram Verification provides a tree view that displays verification elements according to their hierarchical relationships. These elements - categories, verifiable data items, verification item utilizations, verification requirements, verification requirement usages, and dependent data items - are described in the following sections.

Verification Categories

A *verification category* is a means of arranging evidence data into logical groups by grouping elements called *verifiable data items*. (Verifiable data items are fully defined in the following section). For example, an organization might define a list of evidence that relates to personal information: social security number, date of birth, place of birth, income. This related evidence can then be grouped into a "Personal" verification category. Other verification categories could include employment, financial, or child support information.

Verifiable Data Items

In practical terms, the *verifiable data item* can be defined as a piece of evidence that requires verification. This piece of evidence corresponds to a single attribute within a specific evidence entity, e.g., an income amount on the income entity. In order to create a verifiable data item, two attributes must be entered for the evidence entity; the name of the entity (this is stored in the "evidence type" field for an entity) and the exact name of the desired attribute to be verified.

Verifiable data items also provide built-in application security functionality by allowing organizations to enter a security identifier (SID) that can restrict a user's ability to access sensitive verifications. If a user's security profile does not contain the SID entered in this field, then that user will not be able to access the verification. For more information on how application security functionality works, see the `Cúram System Configuration Guide`.

Verification Item Utilizations

Verification item utilizations defines which *verification items* (e.g. passport, birth cert) are to be used for a particular verifiable data item. A verification item defines what can be used to verify the information provided by a participant, for example a passport, birth certificate, pay slip or medical certificate. For some verifiable data items, it may be possible to provide a number of options as to how the data item is verified, in which case such verifiable data items will have a number of verification item utilizations. For example, date of birth can be verified by providing a birth certificate or a passport.

There are a number of configuration settings for verification item utilizations and these impact the runtime functionality. The following list gives an overview of these setting.

From and To Dates

You can define a set period of time during which a verification item can be used to verify a verifiable data item. This is set by defining a period of time on the verification item utilization. Once past this period of time the verification item can no longer be used to verify the verifiable data item; therefore either one of the alternative verification item utilizations defined must be used or a new verification item utilization configured for the verifiable data item in question.

Usage Type

This property defines how a verification item should be utilized when multiple evidence records exist for a client for a particular evidence type which requires verification. The following values can be set for a Usage Type – Shared and Unique. By default, the usage type for a verification item is set as Shared.When the usage type is set as Shared, if multiple evidence records exist for a client for a particular evidence type which requires verification; once the caseworker captures a verification item against the first evidence record, this document is applied by default to the other evidence records of this evidence type. For example, a hospital receipt could be used to verify more than one medical expense. A client may have an asthma condition as well as an arthritis condition that are being treated in the same hospital and so a receipt from the hospital might contain information about the amount that the client was responsible for paying for both conditions.When the usage type is set as Unique, if multiple evidence records exist for a client for a particular evidence type which requires verification; once the caseworker captures a verification item against the first evidence record, this document is applied only to that record. For example, a client has two part time jobs and must verify the Earnings evidence from both the jobs by providing two separate pay slips (one per job). In this scenario, the verification item can be set as Unique so that when the client produces one pay slip, it is not applied to the other Earnings verification record.

Expiry and Warning Days

By setting expiry days on verification item utilizations, the expiry date will be calculated when a verification is added at runtime and when the expiry date is reached a workflow event is initiated. If warning days have been specified, the case owner will be notified when the warning date is reached. Note that expiry date processing and due date processing for verification uses Workflow functionality. For more information on expiry date and due date workflow processing, see section 4.4.1 Deadline Management

Note: Verification items for participant information do not expire. Participant information is not subject to the same time limits as cases are.

Level

Indicates the level achieved by the verification item utilization. Levels range from 1 to 5 in ascending fashion, 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. The Verification Engine will compare the level setting of verification items as they are added against the level setting of the verification requirement in evaluating if the data item is verified.

Mandatory

This property indicates that a particular verification item is mandatory in order to verify a particular verifiable data item. If a verifiable data item has any associated verification items that must be supplied, then regardless of what other items have been added, the verifiable data item is not considered verified until all of the mandatory verification items have been supplied.

SIDs (For Adding or Removing Items)

These two properties specify the SIDs that a user must have in order to either add or remove a particular verification item for a given verifiable data item. If a SID is not supplied for either of these properties, then any user can perform the action associated with that property. For example, if no SID is provided for the Remove Items SID property, then any user will be able to remove a verification item.

Client-Supplied

This property indicates whether or not a verification item is provided by a client for a particular verifiable data item. This property could be used during communications between the organization and the client to ensure that a client is not asked to supply a verification item that should be sourced elsewhere. Note there is no system processing associated with this property, it is used for informational purposes only for the user.

Verification Groups

Verification groups can be used in scenarios where a user has to submit a varying combination of verification items to verify a piece of evidence. For example, Citizenship evidence can be verified by providing a either a passport or (a driver's license and a utility bill) or (photo-copy of a passport, a utility bill and a bank statement). In this scenario, three different verification groups can be created with the same level. The verification requirement for the Citizenship evidence can be satisfied when all the verification items from any of the groups are submitted.

Each verification group has a level associated with it. This indicates the level 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 will consider a verification requirement of level 5 to be satisfied when all the verification items 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 on verification item utilization settings, refer section Verification Item Utilizations.

Verification Configuration Properties

You can configure verifications through system administration properties. For example, you might want to prevent duplicate verifications from being displayed in the user interface for conditional verifications. The following list outlines the properties that affect verifications:

VerificationFilteringEnabled
The VerificationFilteringEnabled property controls filtering of duplicate verifications in the user interface. The default value is `true`, which enables filtering behavior in the user interface component.

preventingDuplicateVerificationInsertion
The preventingDuplicateVerificationInsertion property prevents insertion of duplicate verifications when verifications are created. The default value is `true`, which prevents new duplicate verifications from being inserted into the database.

Verification Requirements

A *verification requirement* provides the rules of verification for a piece of data (verifiable data item). There are many variables included in these rules including where and how the rules apply at runtime. For example whether the verification engine needs to apply the rules to participant level data or to a specific case. Again using date of birth as an example of a verifiable data item, for some organizations the rules may be to verify this piece of data once and therefore verification engine applies the rules within participant manager. For other organizations the rules may require that date of birth is verified at a program level and therefore the verification engine applies the rules to a specific case - see 3.5.4 Verification Requirement Usages for further information.

Verification Requirement Properties

The following is an overview of the properties that can be set on a verification requirement.

Due Date and Warning Date
A number of properties exist for setting a due date on a verification. The "due days" property specifies the number of days after a particular event that a verification should fall due. Administrators can also specify whether the number of due days should be calculated from the date the case was created or from the date evidence was inserted or received. The "warning days" property specifies how many day's prior notice a caseworker will receive before the verification due date. If no warning date is specified, a caseworker will not receive a warning before the verification due date. Note that due date processing for verification requirements uses workflow functionality. For more information on expiry date and due date workflow processing, see "Deadline Management" on page 10.

Level
This property indicates the level of verification that must be achieved in order to consider data verified. Evidence will not be considered verified unless a verification item with the appropriate level is received. For example, if a verification requirement specifies a level 5 verification item (such as an original birth certificate) then providing a level 1 item (a photocopy of a birth certificate) will not satisfy the verification requirement. Alternatively, a combination of verification items that form a verification group of level 5 can be provided to satisfy the verification requirement.

From and To Dates
These properties indicate the period during which this verification requirement is effective. Note that these properties interact with the effective dates of verification item utilizations and the effective dates of evidence in order to determine the verifications that a caseworker can perform. For example, a requirement to verify an income amount might be defined as effective from January to December. However, one verification item may be defined as effective from January to July (e.g., a payslip), while another is defined to be effective from July to December (e.g., a tax return). The date that the income

evidence is active determines which verification item is necessary to satisfy the verification requirement.

Minimum Items

This property specifies the minimum number of verification items that must be provided before data can be considered verified. For example, if the minimum item specified is 2, then the verification requirement will be considered satisfied if at least two verification items or verification groups are provided. When all the verification items specified in a verification group are provided, the Verification Engine will consider it to be a single item. A combination of verification items and groups can also be provided to satisfy the minimum number of verification items of a verification requirement.

Mandatory

This property indicates whether or not the verification requirement is mandatory. A mandatory verification requirement means that evidence and cases associated with the verification may not be activated until the rules defined for the verification have been met. When the mandatory property is not set, the verification requirement is optional and therefore the evidence associated with the verification can be activated even if the evidence has not yet been verified.

Client-Supplied

This property indicates if it is the case participant's responsibility to supply the verification items. This property could be used during communications between the organization and the client to ensure that a client is not asked to supply a verification item that should be sourced elsewhere. Note there is no system processing associated with this property, it is used for informational purposes only for the user.

Re-verification

This property allows users to specify the Cúram Verification engine's response to changes to "Active" evidence. The following list provides the names and impact of the settings for this property. Note that re-verification property does not apply to participant evidence.

Reverify Always

If a caseworker changes "Active" evidence, no previously met verification requirements are carried over 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 entered for the verifiable data item or any dependent data items has not changed, the existing verification information on the "Active" record is copied to the new "In Edit" record. If the value entered for the data item or any dependent data items has changed, 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.

Conditional Verifications

The Conditional Verifications feature is where verification is based on a set of conditions as opposed verification based on added or modified evidence only. The Verification Engine will check the conditions specified, at the time of adding or modifying evidence but will create an outstanding verification record only when a condition that has been defined is met and not every 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 or not verification is required.

For example, a verification may be required only when the value of Earnings amount is more than $ 200 per week or a verification may be required only where the alternate ID is of type SSN. Or to give a more complex example involving a set of dependent evidences; eligibility for an income assistance program requires verification of Household Income evidence type when the income is more than $1150 per month. The Household Income evidence type is made of multiple income evidence types such as Dividends, Pension and Wages and Salaries. Though the verification is set up for the income amount of the Household Income evidence type; the Verification Engine re-evaluates whether the Household Income requires verification when the income of any dependent evidence types, Dividends, Pension and Wages and Salaries, changes.

The Verification Engine allows a conditional verification to be created by allowing the user to associate a rule class. The organization must provide their own rule classes that define the conditions for the verifiable data item. To use conditional verifications that suit specific business scenarios, your organization has to provide the following:

Rule Class
A rule class that defines the conditions for which verification should be triggered for the verifiable data item must be provided.

Display Rule Class
If required, a rule class that defines how the results of the verification should be displayed can be provided.

Display UIM
If required, a UIM page reference for displaying the results of the conditional verifications in the verifications page can be provided.

Workflow Events for Verification Requirements

In addition to due date workflow processing, Cúram Verification provides a number of optional workflow events that your organization can further extend to suit specific business scenarios. The following list provides the names of these workflow events and describes what triggers each event.

Due Date Event
This event is triggered when the verification due date has been reached.

Expiry date event
This event is always triggered if an expiry date has been specified.

Add Event
This event is triggered when a caseworker creates a verification for this requirement.

Update Event
This event is triggered when the verification is updated by the addition or removal of a verification item.

Value Changed Event
This event is triggered when the value of the verifiable evidence is changed.

These workflow events allow the verification process to be integrated with workflow functionality. Note that if your organization wishes to enact workflows using these events, a software developer must customize application code in order to support this. For more information on workflow, see the `Cúram Workflow Overview Guide`.

Verification Requirement Usages

The Cúram Verification engine allows an individual verification requirement to be used by many different types of cases. A *verification requirement usage* allows administrators to associate specific case types with specific verification requirements. In practical terms, this enables an administrator to 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, this requirement can be used by multiple cases within the integrated case. Verification requirement usages are beneficial because they allow verification rules to be applied to groups of cases (i.e., all the cases within an integrated case), or separately applied to individual cases.

A verification requirement usage also exists for participant evidence. This enables an administrator to specify different evidence verification requirements for participant evidence.

Dependent Data Items

Dependent data items are specific pieces of evidence that have a direct influence on the verification of a related data item. Although these pieces of evidence do not require verification, it may be important to record them for the verification of a related data item. For example, if your organization wishes to verify the reason that a household member was absent from the household, the length of the absence may be an important fact to record for the verification. In this example, the "Absence Reason" is the verifiable data item, and the "To" and "From" dates of the absence are dependent data items. The Cúram Verification engine treats any change to a dependent data item in the same manner as a change to the verifiable data item.

The properties that must be stored for a dependent data item include a unique name and the name of a specific data item. The "Data Item" that is entered for the dependent data item should reference an attribute from the evidence type specified in the parent verifiable data item.

Verification for Caseworkers

Introduction

The Cúram Verification engine is called as part of maintenance of case evidence, and as part of maintaining participant data which is used as evidence. It is also called whenever verifications are added or modified. The Cúram Verification engine uses the rules specified in the verification administration component to perform verification processing for evidence.

The following sections describe the processes that are performed by the Cúram Verification engine throughout the life cycle of a piece of evidence. These processes are performed during the addition and modification of evidence, as well as the addition and modification of verification information. In addition, the following sections describe the ways in which caseworkers can access verification details at various stages in the evidence life cycle.

Security Settings: Note that during all of these processes documented in the following sections, the Cúram Verification engine takes into consideration any security settings implemented within the verification settings for a piece of evidence. For example, if a caseworker does not have the security privileges to add a verification item, then that caseworker will not be able to see or effect that verification item.

Capturing Evidence

When evidence is captured for a case, the Cúram Verification engine is called in order to determine if any of the evidence data requires verification. If a piece of data requires verification, the Cúram Verification engine checks to see whether or not verifications are required for the case type where the evidence has been captured. In the case of shared evidence captured for a case, the Cúram Verification engine determines if either the application case, integrated case or its product deliveries (if any exist) require that the evidence be verified. All non-closed product deliveries are considered. If the evidence has any verification requirements, a list of these requirements is returned to the caseworker via an informational message.

When participant evidence is captured, it can be verified in isolation of any case usage of the evidence. Participant evidence is automatically activated when captured and therefore any mandatory verification defined about the participant evidence will exist against this active evidence. The caseworker will see these verifications listed in Verifications listed at the Person and Case level. Note that participant level verifications do not impact the case level processing. Even if there are outstanding mandatory participant verifications present, the Verification Engine will allow the cases for that participant to be activated and will not impact the eligibility and entitlement processing. Where verification of participant data (e.g. verification of a person's date of birth or SSN) must impact the case level processing, the recommended approach is to associate that evidence with the case and set up case level verifications.

Deferring Verifications

The default behavior, as mentioned above, is that verifications are evaluated when evidence is captured and the same is true for all evidence maintenance functions; evaluation of verification requirements by default is tightly linked to evidence maintenance functions. It is however possible to separate evaluation of verifications and evidence maintenance functionality and therefore defer evaluation of verifications.

The ability to defer verifications, is not something that is available to a user to perform but rather can be built into a business process using APIs provided out of the box. This feature supports implementing a business process as many discrete steps rather than implementing as one single transaction. Implementing a business process as many discrete steps provides easier recovery when problems arise as it will only rollback one discrete step rather than all of the business process. With evidence insertion and evaluation of verification as two discrete steps in a business process, if any problems arise in the execution of verifications, the most that would rollback is the evaluation of verifications and the evidence would still be captured and stored on the system.

When verifications are not evaluated for evidence, the evidence is in an "unevaluated" state. This state is presented to a user on various evidence list pages, using a special icon against the evidence that is used specifically to represent unevaluated. Unevaluated evidence means the Cúram Verification Engine was not called when evidence was inserted and the system does not know if any of the evidence data requires verification. There are no actions available for a user to evaluate verifications for evidence that is in an "unevaluated" state. The expectation is that the business process which programmatically deferred verification of the evidence, would be resumed by appropriate recovery technique, and programmatically process unevaluated evidences before it reaches completion.

Accessing Verification Requirements

Caseworkers can view data requiring verification in numerous ways. Within a case, when in the evidence area a caseworker can view verifications associated with the case or can view verifications associated with a particular evidence type or can view verification associated with a particular piece of evidence. Verifications lists are also provided at the Person home page so that a caseworker can see verifications configured on participant evidence. Each of these lists provides the following information about the listed verification requirements:

- general information, e.g., the name of the verifiable data item;
- an indication of whether or not a verification requirement is mandatory; and
- an indication of whether or not a verification requirement has been satisfied.
- an indication of whether or not items have been received for the verification requirement when the verification requirement is outstanding.

This information gives caseworkers the ability to easily determine if verification items need to be added, modified, or removed for a particular piece of evidence. The following sections describe the pages that provide lists of verification requirements.

Evidence Type list

Evidence Type list pages provide the ability to list all verifications specific to the evidence type in question on the current case. This list displays the verification requirements defined for a specific evidence type. Note that while verification items may have been provided for a particular verification requirement, they are applied to the evidence and thus may be used to satisfy other verifications required for that evidence e.g. on other cases.

Evidence Object

We can list verifications specific to a particular piece of evidence. As a particular evidence object changes over time if verifications are defined for it and therefore may need to be re-verified as the evidence is corrected or changed over time, it may be useful to look at these group of verifications together given they are relate to the same evidence object.

Integrated Case Verifications

This list displays all verification requirements associated with a specific integrated case. The list is split into two parts - a list of current verifications and a list of outstanding verifications. The overall list contains only verification requirements that are defined for the integrated case. This overall list includes verifications that are associated with superseded evidence. It does not contain any verification requirements that are defined for product deliveries that are present within the integrated case.

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

Product Delivery Verifications

This list displays all verification requirements associated with a specific product delivery. The list is split into two parts - a list of current verifications and a list of outstanding verifications. The overall list contains all verification requirements that are defined for the product delivery. This overall list includes verifications that are associated with superseded evidence. Verification requirements associated with deleted evidence are only displayed if the application property to display deleted evidence is configured to be displayed.

Participant Verifications

The verification requirements for participant data can be viewed in the participant manager which is accessed the evidence type page. Verification items may also be added from these pages. Caseworkers can view lists of verifications and outstanding verifications for all participant evidence types from the participant manager homepage. This list does not display verification requirements that are associated with canceled or superseded evidence.

Application Case Verifications

This list displays all verification requirements associated with a specific application case. The list is split into two parts - a list of current verifications and a list of outstanding verifications. The overall list contains only verification requirements that are defined for the application case. In addition, the list does not display verification requirements that are associated with canceled or superseded evidence.

Verifying Evidence

Verifying evidence is the process of adding verification items that satisfy the verification rules for evidence. The following sections describe the functionality that Cúram Verification provides to caseworkers to manage the task of verifying case and participant evidence.

Deadline Management

The organization can set up an expiration period on a verification item after which the item will no longer be valid. An organization can also specify the number of days after a particular event has occurred that the verification is due. The due day event may be one of the following:

- The date on which the evidence associated with a verification was entered;
- The date on which the evidence associated with the verification was received (receipt date, present on the evidence descriptor); or,
- The date on which the case for which the evidence is being recorded was created.

When a verification is created, the due date is calculated by adding the number of due days 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 prior to the due date on which the caseworker is notified of the outstanding verification. If a verification is satisfied before the associated deadline has been reached, the deadline will not be monitored further unless the status of the verification changes.

When a verification item is added to a verification requirement, the expiry date is calculated by adding the number of expiry days to either the date the verification is added or the date the item is added. A workflow event is always initiated if expiry dates have been specified. If warning dates have been specified, a notification will be sent to the case owner of the encroaching verification expiry. When the expiry date is reached the administrable expiry date event is kicked off.

Note: Due date functionality is not maintained for participant verifications. This is because the criterion that can be used to define due date only apply to cases e.g. date on which the case was created.

Modify Due Date

This process allows caseworkers to modify the due date associated with a verification requirement. Note that due dates can only be modified if the verification due date has been defined as "modifiable" within the verification administration component. Modifying a due date allows caseworkers to 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 enacted in response to the creation of a verification which has a deadline. A similar sample workflow is enacted in response to the creation of a verification item which has an expiry date. The processing that is undertaken when a verification due date elapses without the verification being satisfied varies by both program type and jurisdiction. Therefore the processing executed within the sample workflow is not mandated and an agency may instead define its own workflow process in order to meet agency specific verification processing requirements. The following are the principal activities executed within the sample "Due Date" workflow for a verification requirement:

1. The case worker is notified and a communication sent to the client in advance of the deadline date, if warning days are specified.
2. The case worker is notified when the due date is reached
3. The case is closed when the deadline has been reached.

The following are the principal activities executed within the sample "Expiry Date" workflow for a verification item:

1. The case worker is notified and a communication sent to the client in advance of the expiry date, if warning days are specified.
2. The case worker is notified when the expiry date is reached.
3. The item is expired and can no longer be used to verify the requirement when the verification item is mandatory or is required to meet the minimum items for the requirement. The verification status is then set to "Not Verified".

Bypassing Mandatory Verifications

Under normal circumstances, when a verification is defined as mandatory, that verification must be satisfied before the evidence can be activated and used as part of eligibility and entitlement calculations. However programs exist where a client is given a certain period of time during which the program will proceed into delivery stage while the client provides the necessary verification documentation. If the client does not provide the necessary verification documentation during this period, then delivery of the program stops. From a verifications perspective, this is achieved by defining the verification as mandatory but allowing the mandatory verification to be waived for this period of time by creating appropriate "Verification Waiver" entries. For example, expedited food stamps allows clients to get a benefit earlier than standard food stamps and for the first month the verifications are not mandatory.

There is no user interface to manually create "Verification Waiver" entries, these should be created within the context of the business process or program that understands the waiver period. Verification Engine provides the necessary APIs to support a business process creating these and once the "Verification Waiver" entries exists, the Verification Engine will check for current entries as part of the business rules that checks for outstanding mandatory verifications.

While a mandatory verification is bypassed, the application will continue to present this verification as being outstanding but will indicate that it is "Bypassed". Also for each verification a history of "Verification Waiver" entries is maintained. This allows a user to determine if a verification was bypassed for a piece of evidence at any point in time and if so, the duration of time it was bypassed.

A "Verification Waiver" contains an optional productID value, which allows verification waiver entries to be created that will tailor verification waiver used by the activation of product delivery to a specific product. This allows evidence to be defined as common evidence at Integrated Case level, shared across product delivery cases yet provide product specific behaviour for mandatory verifications against this evidence. This functionality must also be turned on using the application property, *curam.miscapp.considerproductidforicwaivers* , otherwise activate product delivery will look at the general verification waiver entries only and ignore any product specific verification waivers.

On activating evidence, the system looks for mandatory verifications against this evidence. For each mandatory verification, the Verification Engine checks if any waiver records exists for the current period, and if so, it will allow the evidence to be activated. Therefore for this business rule, it does not matter whether the verification waiver is setup to be product specific or general, as the existence of any waiver record will allow the evidence to be activated.

On activating a product delivery within that Integrated Case, the system looks for any mandatory verifications against evidences that are in the Active state on the Integrated case. For each mandatory verification that exists, the Verification Engine checks if the verification should be waived. If application property *curam.miscapp.considerproductidforicwaivers* is set to No, the Verification Engine will look for a verification waiver (will ignore even if a productID is specified) and if one exists, it will allow the product delivery to be activated. Therefore where Products in an Integrated Case share the same evidence and the evidence has a general verification waiver, this would result in each Product being allowed to proceed into delivery stage.

However if application property *curam.miscapp.considerproductidforicwaivers* is set to YES, the Verification Engine will look for product specific verification waiver records for the outstanding mandatory verification on the Integrated Case and if one exists that matches the product being activated, then the product delivery can be activated. When this application property is set to YES and all products must be waived, a product specific verification waiver record must be created for each individual product. Therefore where Products in an Integrated Case share the same evidence with a mandatory verification setup on the Integrated Case, but the evidence has one or more product specific verification waivers setup, here in contrast some products can proceed to delivery while others must wait for the verifications from the client so it provide a more granular level of control to customers.

As noted above, the verification waiver records must be created by the business process or program that understand the waiver periods. Likewise any product that allows mandatory verifications to be bypassed must also ensure that the product rules are modified to ensure that bypassed evidence is only used for the period of time specified on the "Verification Waiver" table.

Satisfying Verification Rules

The verification requirements defined for evidence cannot be satisfied unless caseworkers provide verification items that meet a number of rules. The following is a list of these rules:

1. The level of a verification item or a verification group must be at least the same level as that defined for the verification requirement.
2. If a minimum number of items has been defined for the verification requirement, then at least this many items must be provided. Note when all the verification items of a group are provided, the Verification Engine will consider this as one item.
3. If a particular verification item is defined as mandatory, then that item must be provided unless the verification is bypassed. The Verification Engine will consider all product delivery cases that are not closed or suspended. Note that 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 provided for a verification requirement must be valid for the date range specified in the verification requirement.

All of these rules must be met in order for a verification requirement to be satisfied. For example, if a verification requirement is defined to be "Level 5" (e.g., requiring an original copy of a birth certificate) and requires two items, then that verification requirement cannot be satisfied by a one "Level 1" item (e.g., a photocopy of a birth certificate). In order for the requirement to be fully satisfied, at least two verification items must be provided, both of which must be "Level 5".

Verification items may be propagated forward when verifying evidence when the verification item added meets the verification requirement of more than one evidence item. The items are propagated forward across each instance of the evidence in the following circumstances only: If the re-verification mode for the requirement is set to Never Reverify, or if the re-verification mode is set to Reverify if Changed and the evidence has not been changed.

Adding a Verification Item

The Add Verification Item process is used to declare that an item of verification has been provided in order to confirm the accuracy of entered evidence. When adding a verification item, the caseworker is only presented with a list of items that are valid for the period defined in the verification requirement.

The verification item can be added and applied to multiple evidences on a case which require verification. The evidence to which the verification item is applied to could either be the same or different evidence type and also could be within the context of one participant or across multiple participants on the case.

During this process, the caseworker can also add an attachment relating to the verification item. Attachments can be added to verification item provisions in order to provide an electronic record of a verification. Attachments can be in the form of graphics or documentation.

The Impact of Evidence Changes on Verification

There are two types of evidence changes that can impact verification: modification and removal of evidence. The effect that an evidence modification has on a verification requirement depends on whether or not the evidence is question is "Active" or "In Edit". The effect that evidence removal has on a verification requirement, however, does not depend on whether or not the evidence has been activated.

Note that the processing used 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 that has a dependent data item called "place of birth". In this case, any changes to the "place of birth" dependent data item will trigger the same processing that is used for the "date of birth" verifiable data item.

The following sections describe the impact that different types of evidence changes can have on verification processing.

Modifying "In Edit" Evidence

The impact of modifying "In Edit" evidence that requires verification depends on whether or not the verification items have been provided. If no verification items have been provided, then no verification processing is necessary. For example, the "In Edit" evidence for a person's date of birth may require verification. However, if the caseworker has not provided a verification item (e.g., a birth certificate), modifying this evidence does not trigger verification processing.

If the caseworker has provided a verification item for the "In Edit" evidence, an informational message is returned to the caseworker. The informational message lists each verifiable data item that has been impacted by the modification of the evidence. For example, the message might say: `The changes that you have made may affect the verification information recorded for the following item(s): Date of Birth. Please review this verification information.` In this example, the text "Date of Birth" refers to the name of the verifiable data item.

If verification has been provided, then the Cúram Verification engine raises a workflow event for every verification requirement that contains a Value Changed workflow event, as defined in the administration component. Note that this event occurs regardless of whether or not the data actually meets any or all verification requirements. The Value Changed event is only raised once for each verification requirement.

Modifying Active Evidence

Modifying a currently active evidence record results in the creation of a new "In Edit" evidence record. From the perspective of the Cúram Verification engine, the creation of a new "In Edit" record in this manner is identical to the creation of 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 recorded for the "Active" record, as in effect a new piece of data is being recorded.

However, the re-verification mode defined for a verification requirement determines whether or not verification information from the previously active evidence record is copied forward to the newly created "In Edit" record. The three re-verification modes are: "Reverify Always", "Reverify If Changed", and "Never Reverify". These re-verification modes are described in full in "Verification Requirements" on page 5. They do not apply to modifications made to active participant evidence.

If information is copied forward to the "In Edit" record, the new verification information for this record is maintained separately from any verification that was associated with the previous "Active" evidence record. In other words, no link exists between the previous verification information and new verification information.

Removing Evidence

Removing evidence has no impact on the associated verifications. However, any verification associated with the removed evidence may not be modified or changed in any manner.

Activating Evidence

Evidence cannot be activated unless all mandatory verification requirements have been met for a piece of evidence. When a caseworker attempts to activate evidence, the Cúram Verification engine is called to verify whether or not there are any outstanding mandatory verification requirements. If all mandatory verification requirements have been satisfied, then the Cúram Verification engine does not prevent the activation of the evidence.

If there are mandatory verification requirements that have not been satisfied, then the Cúram Verification engine prevents the activation of the evidence and returns an informational message to the caseworker stating that mandatory verification requirements must be satisfied before the evidence can be activated.

As described in “Bypassing Mandatory Verifications” on page 11, a hook exists which supports bypassing mandatory verifications for a period of time, which allows evidence to be activated and used in eligibility and entitlement calculations even though mandatory verifications exist for the evidence.

Activating Cases

A case cannot be activated until all mandatory verification requirements have been met for the evidence associated with that case or a current verification waiver exist for the mandatory verification. When a caseworker attempts to activate a case, the Cúram Verification engine is called in order to check that all mandatory verifications associated with active evidence have been satisfied or a current verification waiver exist for the mandatory verification.. If evidence with unsatisfied mandatory verification requirements is found, the Cúram Verification engine prevents the activation of the case and returns an informational message to the caseworker stating that mandatory verification requirements must be satisfied before the case can be activated.

Conclusion

Summary

The following is a summary of the main concepts covered in this guide:

- Verification is the process of checking the accuracy of the information given by clients seeking services from a Social Enterprise organization. Cúram Verification implements simple and effective methods of verification that allow organizations control and flexibility over verification processing.
- Cúram Verification consists of three components; an administration component, a case component and a participant component. The administration component allows an organization to define the data items that require verification and the conditions under which these verification requirements must be satisfied. The case component of Cúram Verification allows caseworkers to record verified data that is received for case evidence. The participant component of Cúram Verification allows caseworkers to record verified data that is received for participant evidence.
- The elements of Cúram Verification that are set up in the administration component include: categories, verifiable data items, verification items, verification item utilizations, verification groups, verification requirements, verification requirement usages, and dependent data items. Conditional Verifications can be set for a Verification Requirement such that a verification record is created only when a custom condition that has been defined is met and not every time a verifiable data item is added or modified.

- The structure of these elements, as laid out in the administration component, is the template for all verification processing in the application. For example, these elements can be customized to determine whether or not certain aspects of verification are mandatory or optional, and to determine if there should be any specific deadlines set on the provision of verifications.
- In the case component of Cúram Verification, caseworkers have the ability to manage deadlines for verification requirements and to provide verification information for captured evidence.
- Changing evidence has an impact on verification, depending on whether that evidence is "active" or "in edit". The effect of a change to "active" evidence also depends on the re-verification modes that are specified for verification items in the administration component.
- Evidence cannot be activated unless all verification requirements that have been defined as mandatory for that evidence are satisfied or a current verification waiver exist for the mandatory verification.
- A case cannot be activated unless all evidence verification requirements that have been defined as mandatory are satisfied or a current verification waiver exist for the mandatory verification.

Additional Information

Additional information on the topics covered in this guide are covered in several related documents:

Cúram Evidence Guide
This guide provides an overview of evidence.

Cúram Workflow Overview Guide
This guide provides an overview of workflow functionality.

Verification Customization Points

Introduction

This appendix provides an overview of the verification customization points.

Supporting Additional Case types for Verifications

If a customer wishes to add a new case type to support verifications then the new type of case should be included in the CT_VerificationTypeCode code table. This would correspond to the relatedItemID of the Verification Requirement Usage entity. For example, Product Delivery, Integrated Case.

The various possible values for each code in the VerificationTypeCode code table would correspond to a relatedItemType field of the Verification Requirement Usage entity. For example, if an Integrated case is selected as verification type code, the possible values are the individual Integrated Case types recorded. The various possible values for this case type would be same as the value of caseType inserted in the case header table when the corresponding case is created.

There must also be an entry in the Curam-config.xml for the domain definition used for RelatedTypes, so that it appears in the search pop-up in the administration section.

When a verification requirement is configured and applied on the administration section the same will be processed by the Verification Engine.

A set of technical steps are provided below which outlines what must be done to extend support. This therefore allows customers to customize the case functionality to better suit their own custom requirements.

The following steps outline how a customer would extend Verifications support for a new case type:

1. Create a new VerificationType code table entry to display an additional item in the 'Applies To' drop down in the Verification Requirement Usage page (that maps to the relatedItemId value in the VerificationRequirementUsage entity).
2. Create a new search page that will be displayed on selecting this new item in the drop down, with a list of available options to select. This selection maps to the relatedItemType value in the

VerificationRequirementUsage entity. This would require creating a new UI page and configuring that as a search page in the curam-config.xml. Please refer to CuramWebClientReferenceManual guide for more details on this.

3. VerificationRelatedTypeHook has to be implemented and the implementation should be bound with the new code table item added in step 1.
4. At runtime, the customer must be able to determine the relatedItemType attribute(mentioned in step 2), given the caseID of the case created in the application. This logic must be implemented by the customer in the implementation of another hook, VerificationRelatedItemHook and this should be bound with the CaseTypeCode for the new case type.

Hook To Consider Additional Case Participants

The default behavior of the verification engine while processing a product delivery case is to check for the verifications on the primary client and case members of the case. If any additional case participant has to be considered for checking verifications on a product delivery case activation, a new hook point *curam.verification.impl.VerificationHookForProductDelivery* should be implemented. Any implementation should be bound to the product ID of the product delivery case.

The only method to be implemented here is *ConcernRoleIDList.getOtherParticipantsForProductDelivery(CaseKey pdCaseKey)*. The verification engine will invoke this method on checking verifications for the product delivery case. This method implementation will return additional case participants for which verifications should be checked.

Skipping Mandatory Verifications

The *VerificationHook* class supports custom implementation of hook points at various stages during the evidence verification processing on a per product basis. The custom implementation should be configured using the Guice dependency injection mechanism. The *skipVerificationCheck()* hook provided here allows the customers to perform custom processing to decide whether the case evidence should be verified before activating the evidence. While activating the evidence, even if there are any mandatory outstanding verifications for any of the products using this evidence, then the evidence will be activated if those product delivery cases satisfy the conditions that are provided in the hook implementation. The customers can plug in their own custom conditions via the hook point.

Implementing Verification Waivers

Verification waivers can be used to bypass a mandatory verification for a set period of time, depending on rules that govern the product or the verification. With this functionality, even if mandatory verification exists for an evidence it can be activated for a set period of time. To use this functionality,the *curam.verification.sl.infrastructure.impl.EvidenceVerificationWaiver* interface must be implemented. It is necessary to bind the implementation using Google Guice MapBinder with a key.

The key should be in a specific format for the following evidences:

For Case Evidence, the format is [CaseTypeCode].[CaseSubTypeCode].[EvidenceTypeCode] where

- [CaseTypeCode] is a code from the code table "CaseTypeCode".
- [CaseSubTypeCode] depends on the type of case and the codes are from code tables such as

  "ProductCategory" - Integrated Case Type, when Case Type is Integrated Case

  "ProductType" - Product Type when Case Type is Product Delivery Case

  "InvestigateConfigType" - Investigation Configuration Type
  when Case Type is Investigation Case

  "ScreeningNameCode" - Screening Name Code when Case Type is Screening Case
- [EvidenceTypeCode] is a code from the code table "CaseEvidenceTypeCode"

For Participant Evidence, the format is [ConcernRoleType].[EvidenceTypeCode] where

- [ConcernRoleType] is a code from the code table "ConcernRoleType"

- [EvidenceTypeCode] is a code from the code table "CaseEvidenceTypeCode"

These codes should be separated by a period i.e., "."

On activation of the evidence if a waiver exists (irrespective of duration) the evidence will be activated even if a verification requirement is mandatory and no verification items exists. Product rules must be updated to take care of these waivers for Eligibility and Entitlement. If using CER rules propagator mechanisms can be used to update the rules object.

A sample implementation of the interface has been provided for "Sporting Grant" in the sample component.

Implementing Custom Conditional Verification Rules

Conditional Verifications is a feature where a user can determine if verification is applicable for an evidence based on the conditions associated to the evidence requiring the verification. This functionality is supported through rule-class implementations and verification for a piece of evidence is determined based on the set of conditions mentioned in the rule classes. The Verification Engine will check these specified conditions at the time of adding or modifying the evidence. But it will create an outstanding verification only when a condition that has been defined is met every 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 or not verification is required.

Rule Artefacts supplied by Verification framework

To facilitate integration between Verification framework and the rule implementations supplied by other components, the framework supplies core Rule Artefacts. These artefacts contain abstract rule classes that other components' rule implementations must adhere to. This section identifies and details such low level Rule Artefacts which will be supplied as part of Verification framework.

Rule Sets

The rule set *VerificationRuleSet* is available as part of Verification framework. This rule set holds all the framework's artefacts such as the rule classes and the data container classes.

Rule Classes

The following rule classes are available as part of *VerificationRuleSet*. The purpose of these rule classes are explained in the corresponding sections.

- **VerificationDeterminator**

  The business logic that determines whether conditional verification is required for particular evidence type goes in this rule class. Components creating rule implementations must adhere to the specification by directly/indirectly extending this class. The following attributes are available in this rule class.

Rule Attribute Name	Type	Purpose
determine	VerificationDeterminator Result	The implementation will contain the business logic that determines the output of conditional verification. A value of 'TRUE' indicates to the evidence framework that verifications are not applicable for the evidence, whereas 'FALSE' denotes that verifications need to be explicitly added.

Rule Attribute Name	Type	Purpose
verificationDeterminator Params	VerificationDeterminator Params	This attribute is populated by the Conditional verifications framework and contains the values for all the input parameters for a particular instance.

- **Verification Determinator Result**

  This rule class is a data container whose purpose is to store the results of business logic in the Verification Determinator. Currently this class has two attributes,

  – result - a boolean that states whether verification is required or not for a given evidence

  – reason - a codetable value from VerificationSkippedReason, which contains the values of reason for which the conditional verification is not applicable

  It is the responsibility of the rule implementation to create or populate these attribute so that the verification framework, after examining the state of the attribute, can take appropriate business decisions.

- **Verification Determinator Params**

  While determining whether conditional verification is required or not, the framework will supply various input parameters to the rule implementation classes for various calculation purposes such as the evidence that is getting currently edited, the associated case identifier for the evidence etc. Please refer the following table for complete details of the input parameters.

Property Name	Data Type	Description
verifiableDataItemName	String	Represents the name of the 'Verifiable Data Item' such as 'Person Income', 'Date Of Birth' etc. The value comes from the code table 'VerifiableItemName'
evidenceDescriptorID	Number	The unique identifier of the evidence record in question
caseID	Number	The unique identifier of the case with which the evidence is associated

Propagator

Verifications are applicable to active evidences as well as to evidences which are in 'In Edit' state and 'Identical Shared In Edit'/'Non Identical Shared In Edit' state. Appropriate propagators should be used to populate the rule objects for the CER rules.

Note that as conditional verification rules are evaluated whenever evidence is created, modified or deleted on a case it is important to ensure that your rules will still execute when a limited set of evidence exists. Otherwise technical errors in the evaluation of the conditional verifications may occur, and if they do they will appear as runtime issues for users.

Notices

This information was developed for products and services offered in the United States.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM 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 IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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.

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 IBM, 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. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Privacy Policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies or other similar technologies that collect each user’s name, user name, password, and/or other personally identifiable information for purposes of session management, authentication, enhanced user usability, single sign-on configuration and/or other usage tracking and/or functional purposes. These cookies or other similar technologies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “ Copyright and trademark information ” at http://www.ibm.com/legal/copytrade.shtml.

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.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, 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 names may be trademarks of their respective owners. Other company, product, and service names may be trademarks or service marks of others.

Part Number:

(1P) P/N: