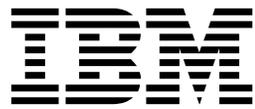


IBM Cúram Social Program Management  
Version 7.0.0

*Cúram Incremental Modernization and  
Transformation (IMT) Web Services  
Cookbook*



**Note**

Before using this information and the product it supports, read the information in "Notices" on page 31

**Edition**

This edition applies to IBM Cúram Social Program Management v7.0.0 and to all subsequent releases unless otherwise indicated in new editions.

Licensed Materials - Property of IBM.

© **Copyright IBM Corporation 2012, 2016.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© Cúram Software Limited. 2011. All rights reserved.

---

# Contents

**Figures . . . . . v**

**Tables . . . . . vii**

## **Cúram Incremental Modernization and Transformation Web Services Cookbook 1**

|  |    |
|--|----|
| Introduction . . . . .                 | 1  |
| Purpose . . . . .                      | 1  |
| Audience . . . . .                     | 1  |
| Prerequisites . . . . .                | 1  |
| Chapters in this Guide . . . . .       | 1  |
| Register Person . . . . .              | 2  |
| The Register Person Service . . . . .  | 2  |
| Incoming Parameters . . . . .          | 2  |
| Minimum Requirements . . . . .         | 2  |
| Optional Incoming Parameters . . . . . | 3  |
| Response Message . . . . .             | 6  |
| Response Parameters . . . . .          | 6  |
| Claim Intake . . . . .                 | 7  |
| The Claim Intake Service . . . . .     | 7  |
| Incoming Parameters . . . . .          | 8  |
| Minimum Requirements . . . . .         | 8  |
| Optional Incoming Parameters . . . . . | 8  |
| Response Messages . . . . .            | 9  |
| Response Parameters . . . . .          | 9  |
| Evidence Maintenance . . . . .         | 10 |
| Evidence Maintenance Service . . . . . | 10 |
| Creating Evidence . . . . .            | 10 |
| Incoming Parameters . . . . .          | 10 |
| Reading Evidence . . . . .             | 12 |

|   |    |
|---|----|
| Incoming Parameters . . . . .             | 12 |
| Response Message . . . . .                | 13 |
| Activating Evidence . . . . .             | 16 |
| ActivateForUsers . . . . .                | 16 |
| ActivateChanges. . . . .                  | 17 |
| ActivateAllChanges. . . . .               | 20 |
| Verification . . . . .                    | 22 |
| The Verification Services . . . . .       | 22 |
| Verification Result . . . . .             | 22 |
| Incoming Parameters . . . . .             | 22 |
| Response Message . . . . .                | 23 |
| Verification Check . . . . .              | 24 |
| Incoming Parameters . . . . .             | 24 |
| Response Message . . . . .                | 25 |
| Determination . . . . .                   | 26 |
| The Determination Service . . . . .       | 26 |
| Incoming Parameters . . . . .             | 27 |
| Incoming Parameter Descriptions . . . . . | 27 |
| Response Message . . . . .                | 27 |
| Response Parameter Descriptions . . . . . | 28 |
| Triage . . . . .                          | 28 |
| The Triage Service . . . . .              | 28 |
| Incoming Parameters . . . . .             | 29 |
| Incoming Parameter descriptions . . . . . | 29 |
| Response Message . . . . .                | 29 |
| Response Parameter Descriptions . . . . . | 30 |

**Notices . . . . . 31**

|   |    |
|---|----|
| Privacy Policy considerations . . . . . | 32 |
| Trademarks . . . . .                    | 33 |



---

## Figures

|   |    |   |    |
|---|----|---|----|
| 1. Inbound Example : Register Person . . . . .    | 6  | 15. Inbound Example : ActivateChanges: Option B     | 19 |
| 2. Response Example : Register Person . . . . .   | 7  | 16. Response Example : ActivateChanges . . . . .    | 20 |
| 3. Error Response Example : Register Person       | 7  | 17. Error Response Example : ActivateChanges        | 20 |
| 4. Inbound Example : Claim Intake . . . . .       | 9  | 18. Inbound Example : ActivateAllChanges            | 21 |
| 5. Response Example : Claim Intake . . . . .      | 10 | 19. Response Example : ActivateAllChanges           | 21 |
| 6. Error Response Example : Claim Intake          | 10 | 20. Error Response Example : ActivateAllChanges     | 22 |
| 7. Inbound Example : Create Evidence . . . . .    | 12 | 21. Inbound Example : VerificationResult . . . . .  | 23 |
| 8. Inbound Example : Read Evidence . . . . .      | 13 | 22. Response Example : VerificationResult . . . . . | 24 |
| 9. Response Example : Read Evidence . . . . .     | 15 | 23. Inbound Example : VerificationCheck . . . . .   | 25 |
| 10. Error Response Example : Read Evidence        | 15 | 24. Response Example : VerificationCheck . . . . .  | 26 |
| 11. Inbound Example : ActivateForUsers . . . . .  | 16 | 25. Inbound Example : Determination . . . . .       | 27 |
| 12. Response Example : ActivateForUsers . . . . . | 17 | 26. Response Example : Determination. . . . .       | 28 |
| 13. Error Response Example : ActivateForUsers     | 17 | 27. Error Response Example : Determination          | 28 |
| 14. Inbound Example : ActivateChanges: Option A   | 18 |   |    |



---

## Tables

|  |    |  |    |
|--|----|--|----|
| 1. Fields . . . . .                                    | 2  | 24. Parameter Descriptions. . . . .          | 18 |
| 2. Parameter Descriptions . . . . .                    | 3  | 25. Response Parameters . . . . .            | 19 |
| 3. Additional Parameters . . . . .                     | 3  | 26. Parameter Descriptions. . . . .          | 19 |
| 4. Response Parameters. . . . .                        | 6  | 27. Minimum Parameter Requirements . . . . . | 20 |
| 5. Parameter Descriptions . . . . .                    | 6  | 28. Parameter Descriptions. . . . .          | 20 |
| 6. Minimum Requirements. . . . .                       | 8  | 29. Response Parameters . . . . .            | 21 |
| 7. Parameter Descriptions . . . . .                    | 8  | 30. Parameter Descriptions. . . . .          | 21 |
| 8. Additional Parameters . . . . .                     | 8  | 31. Minimum Requirements . . . . .           | 22 |
| 9. Response Parameters. . . . .                        | 9  | 32. Parameter Descriptions. . . . .          | 23 |
| 10. Parameter Descriptions . . . . .                   | 9  | 33. Response Elements . . . . .              | 23 |
| 11. Minimum Requirements . . . . .                     | 11 | 34. Parameter Descriptions. . . . .          | 23 |
| 12. Parameter Descriptions. . . . .                    | 11 | 35. Minimum Requirements . . . . .           | 24 |
| 13. Additional Parameters . . . . .                    | 11 | 36. Parameter Descriptions. . . . .          | 24 |
| 14. Minimum Requirements: Option A. . . . .            | 12 | 37. Response Elements . . . . .              | 25 |
| 15. Minimum Requirements: Option B. . . . .            | 12 | 38. Parameter Descriptions. . . . .          | 25 |
| 16. Parameter Descriptions. . . . .                    | 13 | 39. Incoming Parameters . . . . .            | 27 |
| 17. Response Parameters . . . . .                      | 13 | 40. Parameter Descriptions. . . . .          | 27 |
| 18. Parameter Descriptions. . . . .                    | 14 | 41. Response Parameters . . . . .            | 27 |
| 19. Minimum Parameter Requirements . . . . .           | 16 | 42. Parameter Descriptions. . . . .          | 28 |
| 20. Parameter descriptions. . . . .                    | 16 | 43. Minimum Requirements . . . . .           | 29 |
| 21. Response Parameters . . . . .                      | 16 | 44. Parameter Descriptions. . . . .          | 29 |
| 22. Minimum Parameter Requirements: Option A . . . . . | 17 | 45. Response Parameters . . . . .            | 29 |
| 23. Minimum Parameter Requirements: Option B . . . . . | 18 | 46. Parameter Descriptions. . . . .          | 30 |



---

# Cúram Incremental Modernization and Transformation Web Services Cookbook

The required input and response parameters for Cúram web service are described. Examples of the expected XML messages that are sent and received from Cúram web services are provided. The following web services are available as part of Cúram: register person, claim intake, evidence maintenance, verification, determination, and triage.

---

## Introduction

### Purpose

This guide is intended as a reference handbook for developers working on Cúram Web Services. The guide lists the required input and response parameters for each service, and gives examples of the expected XML messages sent and received.

Cúram Web Services are a means of providing services that are normally available within the Cúram system only, to external systems. The web services currently available are:

- Register Person
- Claim Intake
- Evidence Maintenance
- Verification
- Determination
- Triage

### Audience

This guide is intended for developers working on Cúram Web Services.

### Prerequisites

To best make use of this guide, the reader should have experience in developing the functionality which is available in the application.

### Chapters in this Guide

The following list describes the chapters within this guide:

#### Register Person

Register Person gathers the information required to create a person in the Cúram system.

#### Claim Intake

Claim intake gathers the information required to create a case within the Cúram system.

#### Evidence Maintenance

Evidence is the data used to determine entitlement for benefits and services. This chapter describes the requirements for creating, reading and activating evidence.

### Verification

Verification confirms the accuracy of information given by clients seeking assistance from SEM agencies.

### Determination

Determination takes information gathered in the Cúram system as part of intake, and applies it against enterprise-specific and program-specific rules to create eligibility decisions.

**Triage** Triage applies an initial level of review to a basic set of information, to determine a client's need or likely benefit from a program or service.

---

## Register Person

### The Register Person Service

Register Person gathers the information required to create a person in the Cúram system. This service equates to the Register Person business process currently available in the Cúram system. In addition it can operate off a list to allow for registration of more than one client in a single web service call. This will optimize performance as it prevents the overhead involved in calling the service multiple times.

When this service is complete, the person details will be stored in the Cúram system for later reference and use by subsequent services. For example, the person record may be referenced by the Claim Intake service, to allow a product delivery case to be created in the Cúram system against a previously registered person. Person details may be used by the Determination web service.

Certain data items are mandatory as part of the Register Person business process currently available in the Cúram system, and therefore must also be populated in the web service. Otherwise the web service will require exception handling.

### Incoming Parameters

#### Minimum Requirements

The parameters are used to populate the internal struct: `core.facade.PersonRegistrationDetails`:

Table 1. Fields

| Intake Element     | Map to Parameter              | Schema Type      |
|--------------------|-------------------------------|------------------|
| firstname          | firstForeName                 | xs:string        |
| surname            | surname                       | xs:string        |
| gender             | sex                           | bt:codetablecode |
| dateOfBirth        | dateOfBirth                   | bt:date          |
| dateOfRegistration | registrationDate              | bt:date          |
| maritalStatus      | currentMaritalStatus          | bt:codetablecode |
| nationality        | nationality                   | bt:codetablecode |
| countryOfBirth     | countryOfBirth                | bt:codetablecode |
| addressLayout      | addressData.addressLayoutType | bt:codetablecode |
| addressLine1       | addressData.addressLayoutType | bt:codetablecode |

## Incoming Parameter Descriptions:

Table 2. Parameter Descriptions

| Parameter                     | Domain              | Description   |
|-------------------------------|---------------------|---|
| firstForeName                 | FIRST_FORENAME      | The first name of the person to be registered. Type: string                             |
| surname                       | SURNAME             | The surname of the person to be registered. Type: string                                |
| sex                           | GENDER_CODE         | The gender of the person to be registered. Code table: Gender                           |
| dateOfBirth                   | CÚRAM_DATE          | The date of birth of the person to be registered. Format: ddMMyyyy                      |
| registrationDate              | CÚRAM_DATE          | The date of the persons registration. Format: ddMMyyyy                                  |
| currentMaritalStatus          | MARITAL_STATUS_CODE | The marital status of the person to be registered. Code table: MartialStatus            |
| nationality                   | NATIONALITY_CODE    | The nationality of the person to be registered. Code table: Nationality                 |
| birthCountry                  | COUNTRY_CODE        | The country of birth of the person to be registered. Code table: Country                |
| addressData.addressLayoutType | ADDRESS_DATA        | The address layout type for the incoming address details. Code table: AddressLayoutType |
| addressData.addressLine1      | ADDRESS_DATA        | The first line of the address of the person to be registered. Type: string              |

## Optional Incoming Parameters

Table 3. Additional Parameters

| Intake Element       | Map to Parameter         | Schema Type      |
|----------------------|--------------------------|------------------|
| address.addressLine2 | addressData.addressLine2 | bt:addressdata   |
| address.addressLine3 | addressData.addressLine3 | bt:addressdata   |
| address.addressLine4 | addressData.addressLine4 | bt:addressdata   |
| address.addressLine5 | addressData.addressLine5 | bt:addressdata   |
| address.city         | addressData.city         | bt:string        |
| address.county       | addressData.county       | bt:codetablecode |
| address.country      | addressData.country      | bt:codetablecode |
| address.postalCode   | addressData.postalCode   | bt:codetablecode |
| address.statecode    | addressData.statecode    | bt:codetablecode |
| address.comments     | addressData.comments     | bt:string        |
| address.statusCode   | addressData.statusCode   | bt:codetablecode |

Table 3. Additional Parameters (continued)

| Intake Element               | Map to Parameter                 | Schema Type      |
|------------------------------|----------------------------------|------------------|
| address.zipCode              | addressData.zipCode              | bt:codetablecode |
| addressType                  | addressType                      | bt:codetablecode |
| addressIndicator             | addressIndicator                 | bt:boolean       |
| mailingAddress.addressLayout | mailingAddressData.addressLayout | bt:codetablecode |
| mailingAddress.addressLine1  | mailingAddressData.addressLine1  | int1:addressdata |
| mailingAddress.addressLine2  | mailingAddressData.addressLine2  | int2:addressdata |
| mailingAddress.addressLine3  | mailingAddressData.addressLine3  | int3:addressdata |
| mailingAddress.addressLine4  | mailingAddressData.addressLine4  | int4:addressdata |
| mailingAddress.addressLine5  | mailingAddressData.addressLine5  | int5:addressdata |
| mailingAddress.city          | mailingAddressData.city          | bt:string        |
| mailingAddress.county        | mailingAddressData.county        | bt:codetablecode |
| mailingAddress.country       | mailingAddressData.country       | bt:codetablecode |
| formattedAddress             | formattedAddressData             | bt:addressData   |
| othername                    | otherForename                    | bt:string        |
| type                         | type                             | bt:codetablecode |
| title                        | title                            | bt:codetablecode |
| initials                     | initials                         | bt:string        |
| suffix                       | nameSiffix                       | bt:string        |
| ssn                          | socialSecurityNumber             | bt:string        |
| motherBirthSurname           | motherBirthSurname               | bt:string        |
| preferredName                | preferredName                    | bt:string        |
| verifiedDateOfBirth          | dateOfBirthVerified              | bt:boolean       |
| dateOfDeath                  | dateOfDeath                      | bt:date          |
| verifiedDateOfDeath          | dateOfDeathVerified              | bt:boolean       |
| specialInterest              | specialInterest                  | bt:codetablecode |
| phoneType                    | phoneType                        | bt:string        |
| phoneCountry                 | phoneCountryCode                 | bt:codetablecode |
| phoneAreaCode                | phoneAreaCode                    | bt:int32         |
| phoneNumber                  | phoneNumber                      | bt:int32         |
| phoneExtension               | phoneExtension                   | bt:int32         |
| contactPhoneNumber           | contactPhoneNumber               | bt:int32         |
| contactPhoneCountry          | contactPhoneCountryCode          | bt:codetablecode |
| contactPhoneArea             | contactPhoneAreaCode             | bt:int32         |
| contactName                  | contactName                      | bt:string        |
| contactPhoneExtension        | contactPhoneExtension            | bt:int32         |
| contactEmail                 | contactEmailAddress              | bt:string        |
| contactEmailType             | contactEmailType                 | bt:codetablecode |
| contactTitle                 | contactTitle                     | bt:string        |
| publicOffice                 | publicOfficeID                   | bt:int16         |
| preferredPOfficeContact      | preferredPublicOfficeContact     | bt:string        |

Table 3. Additional Parameters (continued)

| Intake Element           | Map to Parameter            | Schema Type         |
|--------------------------|-----------------------------|---------------------|
| preferredPOfficeName     | preferredPublicOfficeName   | bt:string           |
| preferredLanguage        | preferredLanguage           | bt:codetablecode    |
| placeOfBirth             | birthPlace                  | bt:string           |
| concernID                | concernID                   | bt:long             |
| ethnicOrigin             | ethnicOriginCode            | bt:codetablecode    |
| exceptionMethod          | commExceptionMethodCode     | bt:codetablecode    |
| exceptionReason          | commExceptionReasonCode     | bt:codetablecode    |
| exceptionFromDate        | commExceptionFromDate       | bt:date             |
| exceptionToDate          | commExceptionToDate         | bt:date             |
| foreignResidencyCountry  | foreignResidencyCountryCode | bt:codetablecode    |
| foreignResidencyReason   | foreignResidencyReasonCode  | bt:codetablecode    |
| foreignResidencyFromDate | foreignResidencyFromDate    | bt:date             |
| foreignResidencyToDate   | foreignResidencyToDate      | bt:date             |
| citizenshipCountry       | citizenshipCountryCode      | bt:codetablecode    |
| citizenshipReason        | citizenshipReasonCode       | bt:codetablecode    |
| citizenshipFromDate      | citizenshipFromDate         | bt:date             |
| citizenshipToDate        | citizenshipToDate           | bt:date             |
| preferredCommMethod      | preCommMethod               | bt:codetablecode    |
| relatedClientID          | relatedConcernRoleID        | bt:clientIdentifier |
| paymentFrequency         | paymentFrequency            | bt:frequencyPattern |
| nextPaymentDate          | nextPaymentDate             | bt:date             |
| paymentMethod            | methodOfPmtCode             | bt:string           |

```

<root>
  <register>
    <registerPerson id="123252">
      <firstname>MARY</firstname>
      <surname>McConnell</surname>
      <gender>SX1 </gender>
      <dateOfBirth>06061975</dateOfBirth>
      <dateOfRegistration>12112007</dateOfRegistration>
      <maritalStatus>MS1 </maritalStatus>
      <nationality>NT7</nationality>
      <birthCountry>PK</birthCountry>
      <address>
        <addressLayout>US</addressLayout>
        <addressLine1>PineWood</addressLine1>
        <addressLine2>The hills</addressLine2>
        <addressLine3>HillView</addressLine3>
        <addressLine4>The Rise</addressLine4>
        <addressLine5>Malahide</addressLine5>
        <city>Ballymun</city>
        <countryCode>US</countryCode>
      </address>
    </registerPerson>
  </register>
</root>

```

Figure 1. Inbound Example : Register Person.

This figure displays an example of the inbound register person xml message.

## Response Message

### Response Parameters

The parameters are contained within the internal struct: `core.facade.PersonRegistrationResult`

Table 4. Response Parameters

| Map from Parameter | Reponse Element | Type |
|--------------------|-----------------|------|
| clientID           | clientID        | long |

### Response Parameter Descriptions:

Table 5. Parameter Descriptions

| Parameter    | Domain | Description   |
|--------------|--------|---|
| id attribute | n/a    | An optional identification, if found in the original inbound details, is included within the response message. This allows the third party to easily match inbound and outbound person registration data. Type: int |

Table 5. Parameter Descriptions (continued)

| Parameter | Domain          | Description  |
|-----------|-----------------|--|
| clientID  | CONCERN_ROLE_ID | The client identifier of the related client.<br>Type: long |

```
<receiveDocumentReturn>
  <response>
    <registerPerson id="123252" success="true">
      <clientID>5449355549118300160</clientID>
    </registerPerson>
  </response>
</receiveDocumentReturn>
```

Figure 2. Response Example : Register Person.

This figure displays an example of the register person response XML message.

```
<receiveDocumentReturn>
  <response>
    <registerPerson success="false" id="123252">
      <exception>
        <message>An error occurred during the person
          registration process.</message>
        <exceptionMessage>The codetable Gender does not
          contain the value passed: SX100.
        </exceptionMessage>
      </exception>
    </registerPerson>
  </response>
</receiveDocumentReturn>
```

Figure 3. Error Response Example : Register Person.

This figure displays an example of the register person error response XML message.

## Claim Intake

### The Claim Intake Service

Claim intake gathers the information required to create a case within the Cúram system. This service equates to the Create Product Delivery business process and therefore we do not include details of the creation of Integrated Case, Service Plan or any other case type. The objective is to make a product delivery case available for the subsequent storage of evidence and execution of business services, such as verification and determination.

Certain configuration data is required in order for a product delivery case to be successfully created. This configuration data must be created in advance of using Claim Intake services. Therefore it is assumed that the product and the product provider configuration data are available for this service. Also it is assumed that

primary client for whom the claim is being captured has already been registered on the Cúram system using the Register Person service.

## Incoming Parameters

### Minimum Requirements

The parameters are used to populate the internal struct: `core.facade.CreateCaseDetails`

Table 6. Minimum Requirements

| Intake Element   | Map to Parameter         | Schema Type                          |
|------------------|--------------------------|--------------------------------------|
| clientID         | clientID                 | bt:clientIdentifier                  |
| productID        | productID                | bt:productProviderIdentifier         |
| providerID       | productProviderID        | bt:productProviderIdentifier         |
| providerLocation | providerLocation         | bt:providerLocation                  |
| deliveryPattern  | productDeliveryPatternID | bt:providerDeliveryPatternIdentifier |
| receivedDate     | receivedDate             | bt:date                              |
| currency         | currencyType             | bt:codetablecode                     |

### Incoming Parameter Descriptions:

Table 7. Parameter Descriptions

| Parameter        | Domain                      | Description   |
|------------------|-----------------------------|---|
| clientID         | CONCERN_ROLE_ID             | The client's identification. Type: long.                          |
| productID        | PRODUCT_ID                  | The case product's identification. Type: long.                    |
| providerID       | PRODUCT_PROVIDER_ID         | The case's product provider's identification. Type: long.         |
| providerLocation | PROVIDER_LOCATION           | The case's provider's location. Type: int.                        |
| deliveryPattern  | PRODUCT_DELIVERY_PATTERN_ID | The Identification of the product's delivery pattern. Type: long. |
| receivedDate     | Cúram_DATE                  | The date of receipt. Format: ddMMyyyy                             |
| currency         | CURRENCY_CODE               | The currency type. Code table: Currency                           |

### Optional Incoming Parameters

Table 8. Additional Parameters

| Intake Element | Map to Parameter | Schema Type |
|----------------|------------------|-------------|
| objective      | objectiveCode    | bt:string   |
| caseStartDate  | caseStartDate    | bt:date     |

```

<root>
  <claimIntake>
    <claimIntakeGroup>
      <clientID>8232580118833266688</clientID>
      <productID>2111</productID>
      <providerID>123</providerID>
      <providerLocation>2701</providerLocation>
      <deliveryPattern>111</deliveryPattern>
      <receivedDate> 13112007 </receivedDate>
      <currency> USD</currency>
    </claimIntakeGroup>
  </claimIntake>
</root>

```

Figure 4. Inbound Example : Claim Intake.

This figure displays an example of the inbound Claim Intake xml message.

## Response Messages

### Response Parameters

The parameters are contained within the internal struct: core.facade.CreatedCaseIDKey

Table 9. Response Parameters

| Map from Parameter | Reponse Element | Type |
|--------------------|-----------------|------|
| caseID             | caseID          | long |
| clientID           | clientID        | long |

### Response Parameter Descriptions:

Table 10. Parameter Descriptions

| Parameter | Domain/Attribute | Description   |
|-----------|------------------|---|
| caseID    | CASE_ID          | The case identifier of the created Product Delivery. Type: long |
| clientID  | CONCERN_ROLE_ID  | The client identifier of the related client. Type: long         |

```

<receiveDocumentReturn>
  <response>
    <claimIntake success="true">
      <caseID>3278620528725721088</caseID>
      <clientID>-5728578726015270912</clientID>
    </claimIntake>
  </response>
</receiveDocumentReturn>

```

Figure 5. Response Example : Claim Intake.

This figure displays an example of the response XML message.

```

<receiveDocumentReturn>
  <response>
    <claimIntake success="false">
      <exception>
        <message>An error occurred during the claim intake
          procedure for the client .</message>
        <exceptionMessage>Record not found.</exceptionMessage>
      </exception>
    </claimIntake>
  </response>
</receiveDocumentReturn>

```

Figure 6. Error Response Example : Claim Intake.

This figure displays an example of the error response xml message.

---

## Evidence Maintenance

### Evidence Maintenance Service

Evidence is the data used to determine entitlement for benefits and services. Therefore the presence of this data is required to support other Cúram services in the entitlement area, for example Triage and Determination. Services offered for evidence maintenance are simple in nature. They assume that any approvals are performed in advance by the calling system, and the management of evidence relationships, evidence hierarchies and so on is dealt with in the calling system.

### Creating Evidence

#### Incoming Parameters

The Create Evidence service equates to the generic Insert Evidence pattern for case evidence in the evidence framework. In addition, it can operate off a list to allow for insertion of multiple evidence records in a single service call.

Following creation of the evidence record(s) in the Cúram system, they will be activated immediately without initiating evidence approvals. If the activation cannot successfully complete, for example if the evidence must be verified, then the evidence record will be left in an in-edit status. In-edit evidence records can be verified at a later date using the Activate Evidence service when any issues are resolved and the evidence is verified.

The parameters are used to populate the internal struct:  
 Cúram.core.sl.infrastructure.struct.EvidenceDescriptionInsertDtIs

Table 11. Minimum Requirements

| Intake Element | Map to Parameter | Schema Type        |
|----------------|------------------|--------------------|
| caseID         | caseID           | sem:caseIdentifier |
| evidenceType   | evidenceType     | sem:codetablecode  |
| receivedDate   | receivedDate     | sem:date           |
| effectiveDate  | effectiveDate    | sem:date           |
| dataObjects    | see below        | see below          |

Each Evidence Create schema has an object structure defined for the incoming data. The dataObjects structure is:

```
<dataItem name="{data item name}"
  >{value}</dataItem>
```

- Data Item name: The name of the attribute within the struct that is passed to the entity object.
- Value: The value to populate the struct field with. This will be passed to the entity object.

**Note:** DataItem to struct mapping controls all data type conversions and checks.

#### Incoming Parameter Descriptions:

Table 12. Parameter Descriptions

| Parameter     | Domain        | Description   |
|---------------|---------------|---|
| caseID        | CASE_ID       | The numeric identification of the evidence records related case. Type: long |
| evidenceType  | EVIDENCE_TYPE | The evidence type of the evidence record created. Codetable: evidenceType   |
| receivedDate  | CÚRAM_DATE    | The date of receipt of the evidence creation. Format ddMMyyyy               |
| effectiveDate | CÚRAM_DATE    | The date from when the created evidence is effective. Format ddMMyyyy       |

#### Optional Incoming Parameters:

Table 13. Additional Parameters

| Intake Element | Map to Parameter | Schema Type               |
|----------------|------------------|---------------------------|
| participantID  | participantID    | sem:participantIdentifier |

The following figure displays an example of the inbound Create Evidence xml message:

```

<root>
  <evidence>
    <evidenceData>
      <evidenceDetails>
        <caseID>8034421735228964864</caseID>
        <evidenceType>ET500</evidenceType>
        <receivedDate>01010001</receivedDate>
        <effectiveDate>01010001</effectiveDate>
      </evidenceDetails>
      <dataObjects>
        <dataItem name="sportingActivityID">
          -1333065489701666816</dataItem>
        <dataItem name="caseParticipantRoleID">
          -900719925474099200</dataItem>
        <dataItem name="sportingActivityType">SA5</dataItem>
        <dataItem name="sportingAwardType">SAT2</dataItem>
        <dataItem name="paymentAmount">100.00</dataItem>
        <dataItem name="comments"/>
        <dataItem name="startDate">01010001</dataItem>
        <dataItem name="endDate">01010001</dataItem>
        <dataItem name="versionNo">1</dataItem>
      </dataObjects>
    </evidenceData>
  </evidence>
</root>

```

Figure 7. Inbound Example : Create Evidence

## Reading Evidence

### Incoming Parameters

The Read Evidence service equates to the generic View Evidence pattern for case evidence in the evidence framework. In addition once again it can operate off a list to allow for retrieval of more than one evidence record in a single service call. The service is simply to retrieve evidence data from the Cúram system.

The parameters with both Minimum requirements tables are used to populate the internal struct: Cúram.core.sl.infrastructure.struct.EIEvidenceKey

Table 14. Minimum Requirements: Option A

| Intake Element | Map to Parameter | Schema Type            |
|----------------|------------------|------------------------|
| evidenceID     | evidenceID       | sem:evidenceIdentifier |
| evidenceType   | evidenceType     | sem:codetablecode      |

Table 15. Minimum Requirements: Option B

| Intake Element       | Map to Parameter     | Schema Type                      |
|----------------------|----------------------|----------------------------------|
| evidenceDescriptorID | evidenceDescriptorID | sem:evidenceDescriptorIdentifier |

## Incoming Parameter Descriptions:

Table 16. Parameter Descriptions

| Parameter            | Domain                 | Description  |
|----------------------|------------------------|--|
| evidenceID           | EVIDENCE_ID            | The evidence identification of the evidence record to be read. Type: long            |
| evidenceType         | EVIDENCE_TYPE          | The evidence type of the evidence record to be read. Codetable: evidenceType         |
| evidenceDescriptorID | EVIDENCE_DESCRIPTOR_ID | The evidence descriptor identification of the evidence record to be read. Type: long |

The following figure displays an example of the inbound Read Evidence xml message.

```
<root>
  <evidence>
    <evidenceRead>
      <evidenceID>-1333065489701666816</evidenceID>
      <evidenceType>ET500</evidenceType>
    </evidenceRead>
    <evidenceRead>
      <evidenceID>-7259802599321239552</evidenceID>
      <evidenceType>ET500</evidenceType>
    </evidenceRead>
  </evidence>
</root>
```

Figure 8. Inbound Example : Read Evidence

## Response Message

Table 17. Response Parameters

| Map from Parameter   | Response Element     | Type      |
|----------------------|----------------------|-----------|
| caseID               | caseID               | long      |
| evidenceType         | evidenceType         | string    |
| evidenceDescriptorID | evidenceDescriptorID | long      |
| effectiveDate        | effectiveDate        | date      |
| dataObjects          | See below            | See below |

Each Evidence Read response has a dataObjects element made up of dataItem child elements. The dataItem structure is:

```
<dataItem name="{data item name}"
  >{value}</dataItem>
```

- Data Item name: The name of the attribute within the struct that is passed to the entity object..
- Type: The type of field that will be populated.

- The value to populate the struct field with. This will be passed to the entity object

| Type             | Description   |
|------------------|---|
| string           | A string.class type value.  |
| Int              | An int.class type value.  |
| Short            | A short.class type value.   |
| Double           | A double.class type value.  |
| Float            | A float.class type value.   |
| Long             | A long.class type value.  |
| Date             | A Cúram.util.type.Date.class type value.                                |
| Date             | A Cúram.util.type.Date.class type value, format: ddMMyyyy.              |
| DateTime         | A Cúram.util.type.DateTime.class type value, format: ddMMyyyy hh:mm:ss. |
| Money            | A Cúram.util.type.Money.class type value.                               |
| FrequencyPattern | A Cúram.util.type.FrequencyPattern.class type value.                    |

#### Response Parameter Descriptions:

Table 18. Parameter Descriptions

| Parameter            | Domain/Attribute       | Description   |
|----------------------|------------------------|---|
| caseID               | CASE_ID                | The case identifier for the case to which the evidence record belongs. Type: long |
| evidenceType         | EVIDENCE_TYPE          | The evidence type code for the evidence record created. Codetable: EvidenceType   |
| evidenceDescriptorID | EVIDENCE_DESCRIPTOR_ID | The evidence descriptor identifier for the evidence record created. Type: long    |
| effectiveDate        | CÚRAM_DATE             | The date from when the created evidence is effective. Format ddMMyyyy             |

The following figure displays an example of the Read Evidence response xml message:

```

<receiveDocumentReturn>
  <response>
    <evidenceRead success="true">
      <caseID>8034421735228964864</caseID>
      <evidenceType>ET500</evidenceType>
      <evidenceDescriptorID>
        684547143360315392
      </evidenceDescriptorID>
      <effectiveDate>11092007</effectiveDate>
      <dataObjects>
        <dataItem name="sportingActivityID" type="long">
          -1333065489701666816</dataItem>
        <dataItem name="caseParticipantRoleID" type="long">
          -900719925474099200</dataItem>
        <dataItem name="sportingActivityType"
          type="string">SA5</dataItem>
        <dataItem name="sportingAwardType"
          type="string">SAT2</dataItem>
        <dataItem name="paymentAmount"
          type="money">100.00</dataItem>
        <dataItem name="comments" type="string"/>
        <dataItem name="startDate" type="date">
          11022007</dataItem>
        <dataItem name="endDate" type="date">
          10092008</dataItem>
        <dataItem name="versionNo" type="int">1</dataItem>
      </dataObjects>
    </evidenceRead>
  </response>
</receiveDocumentReturn>

```

Figure 9. Response Example : Read Evidence

```

<receiveDocumentReturn>
  <response>
    <evidenceRead success="false">
      <exception>
        <message>The evidence read operation failed.
          Please contact the administrator.</message>
        <exceptionMessage>
          Index: 0, Size: 0
        </exceptionMessage>
      </exception>
    </evidenceRead>
  </response>
</receiveDocumentReturn>

```

Figure 10. Error Response Example : Read Evidence.

This figure displays an example of the Read Evidence error response xml message.

## Activating Evidence

The Activate Evidence service equates to the generic Apply Changes evidence pattern for case evidence. Like the other evidence services however, it can take a list of evidence descriptor IDs as input and activate each one.

### ActivateForUsers

**Incoming Parameters:** The parameters are used to populate the internal struct: `Cúram.core.struct.CaseKey`.

Table 19. Minimum Parameter Requirements

| Intake Element | Map to Parameter | Schema Type        |
|----------------|------------------|--------------------|
| caseID         | caseID           | sem:caseIdentifier |

### Incoming Parameter Descriptions

Table 20. Parameter descriptions

| Parameter | Domain  | Description  |
|-----------|---------|--|
| caseID    | CASE_ID | The case identifier of the case to which the evidence to be activated is related. Type: long |

```
<root>
  <evidence>
    <activateForUsers>
      <caseID>8034421735228964864</caseID>
    </activateForUsers>
    <activateForUsers>
      <caseID>8435421537284500864</caseID>
    </activateForUsers>
  </evidence>
</root>
```

Figure 11. Inbound Example : ActivateForUsers.

This figure displays an example of the inbound ActivateForUsers xml message.

### Response Message:

Table 21. Response Parameters

| Map from Parameter | Response Element | Type    |
|--------------------|------------------|---------|
| caseID             | caseID           | long    |
| activated          | activated        | boolean |

### Response Parameter Descriptions:

| Parameter | Domain/Attribute | Description   |
|-----------|------------------|---|
| caseID    | CASE_ID          | The case identifier of the case for which all evidence records were to be activated. Type: long |

| Parameter | Domain/Attribute | Description   |
|-----------|------------------|---|
| activated | n/a              | The indicator as to whether the activation attempt was a success or failure. Type boolean |

```
<receiveDocumentReturn>
  <response>
    <activateUserChanges success="true">
      <caseID>-6737385042546262016</caseID>
      <activated>>true</activated>
    </activateUserChanges>
  </response>
</receiveDocumentReturn>
```

Figure 12. Response Example : ActivateForUsers.

This figure displays an example of the ActivateForUsers response xml message.

```
<receiveDocumentReturn>
  <response>
    <activateAllChanges success="false">
      <caseID>77912273553500958080</caseID>
      <activated>>false</activated>
      <exception>
        <message>The evidence activate all changes operation
          failed.Please contact the administrator.
        </message>
        <exceptionMessage>For input string:
          "77912273553500958080"</exceptionMessage>
      </exception>
    </activateAllChanges>
  </response>
</receiveDocumentReturn>
```

Figure 13. Error Response Example : ActivateForUsers.

This figure displays an example of the ActivateForUsers error response xml message.

### ActivateChanges

**Incoming Parameters:** The parameters are used to populate the internal struct: `Cúram.core.sl.infrastructure.struct.EvidenceEvidenceDescriptionInsertDtls`

Table 22. Minimum Parameter Requirements: Option A

| Intake Element | Map to Parameter | Schema Type            |
|----------------|------------------|------------------------|
| caseID         | caseID           | sem:caseIdentifier     |
| evidenceID     | evidenceID       | sem:evidenceIdentifier |
| evidenceType   | evidenceType     | sem:evidenceType       |

The parameters are used to populate the internal struct: `Cúram.core.sl.infrastructure.struct.EvidenceEvidenceDescriptionInsertDtls`

Table 23. Minimum Parameter Requirements: Option B

| Intake Element       | Map to Parameter     | Schema Type                      |
|----------------------|----------------------|----------------------------------|
| caseID               | caseID               | sem:caseIdentifier               |
| evidenceDescriptorID | evidenceDescriptorID | sem:evidenceDescriptorIdentifier |

### Incoming Parameter descriptions

Table 24. Parameter Descriptions

| Parameter            | Domain                 | Description  |
|----------------------|------------------------|--|
| caseID               | CASE_ID                | The case identifier of the case to which the evidence to be activated is related. Type: long |
| evidenceID           | EVIDENCE_ID            | The evidence identifier of the evidence record to be activated. Type: long                   |
| evidenceType         | EVIDENCE_TYPE          | The evidence type code of the evidence record to be activated. Codetable: EvidenceType       |
| evidenceDescriptorID | EVIDENCE_DESCRIPTOR_ID | The evidence descriptor identifier of the evidence record to be activated. Type: long        |

```

<root>
  <evidence>
    <activateChanges>
      <caseID>8034421735228964864</caseID>
      <evidenceID>501</evidenceID>
      <evidenceType >PET10</evidenceType>
    </activateChanges >
  </evidence>
</root>

```

Figure 14. Inbound Example : ActivateChanges: Option A.

This figure displays an example of the inbound ActivateChanges xml message.

```

<root>
  <evidence>
    <activateChanges>
      <caseID>8034421735228964864</caseID>
      <evidenceDescriptorID>6719370644036780032
      </evidenceDescriptorID>
    </activateChanges >
  </evidence>
</root>

```

Figure 15. Inbound Example : ActivateChanges: Option B.

This figure displays an example of the inbound ActivateChanges xml message.

**Response Message:**

Table 25. Response Parameters

| Map from Parameter   | Reponse Element      | Type    |
|----------------------|----------------------|---------|
| caseID               | caseID               | long    |
| evidenceDescriptorID | evidenceDescriptorID | long    |
| activated            | activated            | boolean |

**Response Parameter Descriptions:**

Table 26. Parameter Descriptions

| Parameter            | Domain/Attribute       | Description  |
|----------------------|------------------------|--|
| caseID               | CASE_ID                | The case identifier of the case to which the evidence record activated is related. Type: long      |
| evidenceDescriptorID | EVIDENCE_DESCRIPTOR_ID | The evidence descriptor identifier of the evidence record activated. Type: long                    |
| activated            | n/a                    | The Boolean indicator as to whether the activation attempt was a success or failure. Type: boolean |

```

<receiveDocumentReturn>
  <response>
    <activateChanges success="true">
      <caseID>7791227355350958080</caseID>
      <evidenceDescriptorID>
        4332462841530417152
      </evidenceDescriptorID>
      <activated>true</activated>
    </activateChanges>
  </response>
</receiveDocumentReturn>

```

Figure 16. Response Example : ActivateChanges.

This figure displays an example of the ActivateChanges response xml message.

```

<receiveDocumentReturn>
  <response>
    <activateChanges success="false">
      <caseID>7791227355350958080</caseID>
      <activated>false</activated>
      <exception>
        <message>The evidence activate changes operation
          failed. Please contact the administrator.</message>
        <exceptionMessage>Record not found.</exceptionMessage>
      </exception>
    </activateChanges>
  </response>
</receiveDocumentReturn>

```

Figure 17. Error Response Example : ActivateChanges.

This figure displays an example of the ActivateChanges error response xml message.

## ActivateAllChanges

### Incoming Parameters:

Table 27. Minimum Parameter Requirements

| Intake Element | Map to Parameter | Schema Type        |
|----------------|------------------|--------------------|
| caseID         | caseID           | sem:caseIdentifier |

### Incoming Parameter Descriptions

Table 28. Parameter Descriptions

| Parameter | Domain  | Description  |
|-----------|---------|--|
| caseID    | CASE_ID | The case identifier of the case to which the evidence to be activated is related. Type: long |

```

<root>
  <evidence>
    <activateAllChanges>
      <caseID>8034421735228964864</caseID>
    </activateAllChanges>
    <activateAllChanges>
      <caseID>8435421537284500864</caseID>
    </activateAllChanges>
  </evidence>
</root>

```

Figure 18. Inbound Example : ActivateAllChanges.

This figure displays an example of the inbound ActivateAllChanges xml message.

**Response Message:**

Table 29. Response Parameters

| Map from Parameter | Reponse Element | Type    |
|--------------------|-----------------|---------|
| caseID             | caseID          | long    |
| activated          | activated       | boolean |

**Response Parameter Descriptions:**

Table 30. Parameter Descriptions

| Parameter | Domain/Attribute | Description  |
|-----------|------------------|--|
| caseID    | CASE_ID          | The case identifier of the case for which all evidence records were to be activated. Type: long    |
| activated | n/a              | The Boolean indicator as to whether the activation attempt was a success or failure. Type: boolean |

```

<receiveDocumentReturn>
  <response>
    <activateAllChanges success="true">
      <caseID>7791227355350958080</caseID>
      <activated>true</activated>
    </activateAllChanges>
  </response>
</receiveDocumentReturn>

```

Figure 19. Response Example : ActivateAllChanges.

This figure displays an example of the ActivateAllChanges response xml message.

```

<receiveDocumentReturn>
  <response>
    <activateAllChanges success="false">
      <caseID>77912273553500958080</caseID>
      <activated>false</activated>
      <exception>
        <message>The evidence activate all changes operation
          failed.Please contact the administrator.
        </message>
        <exceptionMessage>For input string:
          "77912273553500958080"</exceptionMessage>
      </exception>
    </activateAllChanges>
  </response>
</receiveDocumentReturn>

```

Figure 20. Error Response Example : ActivateAllChanges.

This figure displays an example of the ActivateAllChanges error response xml message.

---

## Verification

### The Verification Services

Verification confirms the accuracy of information given by clients seeking assistance from SEM agencies. Two services are provided for verification of evidence. One service, Verification Result, is required for verifying evidence. This can operate off a list, to allow a single service to verify multiple evidence records.

The second service, Verification Check, checks for outstanding verifications on a case. In Cúram, when there are outstanding verifications on a case, we prevent the case progressing through to delivery of payments. Therefore this service enables you to do a similar validation from a third party system that is using Cúram for capture and verification of evidence.

Both verification web services must assume that:

- The Cúram Verification administration component is present.
- Verification requirements are defined in the administration component.
- The Cúram system contains the evidence to be verified.

### Verification Result

#### Incoming Parameters

The verification result service is used to return the current verification status of a given list of evidence records.

The parameters are used to populate the struct: EvidenceDescriptorKey

Table 31. Minimum Requirements

| Intake Element      | Map to Parameter     | Schema Type                     |
|---------------------|----------------------|---------------------------------|
| evidenceDescriptorD | evidenceDescriptorID | bt:evidenceDescriptorIdentifier |

## Incoming Parameter Descriptions:

Table 32. Parameter Descriptions

| Parameter            | Domain                 | Description   |
|----------------------|------------------------|---|
| evidenceDescriptorID | EVIDENCE_DESCRIPTOR_ID | The evidence descriptor identifier for the evidence record for which the verification results are being queried. Type: long |

```

<root>
  <verification>
    <verificationResult>
      <evidenceDescriptorID>
        810647932926689280
      </evidenceDescriptorID>
      <evidenceDescriptorID>
        6719370644036780032
      </evidenceDescriptorID>
    </verificationResult>
  </verification>
</root>

```

Figure 21. Inbound Example : VerificationResult.

This figure displays an example of the inbound VerificationResult xml message.

## Response Message

The parameters are contained within the struct:

Cúram.verification.sl.infrastructure.struct.EvidenceVerificationDetails

Table 33. Response Elements

| Map from parameter   | Reponse element      | Type    |
|----------------------|----------------------|---------|
| evidenceDescriptorID | evidenceDescriptorID | long    |
| verificationStatus   | verificationStatus   | boolean |

## Response Parameter Descriptions:

Table 34. Parameter Eescriptions

| Parameter            | Domain/Attribute       | Description   |
|----------------------|------------------------|---|
| evidenceDescriptorID | EVIDENCE_DESCRIPTOR_ID | The evidence descriptor identifier of the evidence record. Type: long                                   |
| verificationStatus   | EVIDENCE_STATUS_CODE   | The code table value to show the verification status of the evidence recored. Codetable: EvidenceStatus |

```

<receiveDocumentReturn>
  <response>
    <verificationResult success="true">
      <evidenceDescriptorID>810647932926689280
    </evidenceDescriptorID>
    <verificationStatus>ES2</verificationStatus>
    </verificationResult>
    <verificationResult success="true">
      <evidenceDescriptorID>820647932926347586
    </evidenceDescriptorID>
    <verificationStatus>ES6</verificationStatus>
    </verificationResult>
  </response>
</receiveDocumentReturn>

```

Figure 22. Response Example : VerificationResult.

This figure displays an example of the VerificationResult response xml message.

## Verification Check

### Incoming Parameters

The verification check service is used to return a list of outstanding evidence records, for a given case..

The parameters are used to populate the struct: EvidenceDescriptorKey.

Table 35. Minimum Requirements

| Intake Element | Map to Parameter | Schema Type       |
|----------------|------------------|-------------------|
| caseID         | caseID           | bt:caseIdentifier |

### Incoming Parameter Descriptions:

Table 36. Parameter Descriptions

| Parameter | Domain       | Description  |
|-----------|--------------|--|
| caseID    | CASE_ID Long | The case identifier for the case which will be queried for all its evidence and their current verification status. |

```

<root>
  <verification>
    <verificationCheck>
      <evidenceDescriptorID>810647932926689280
    </evidenceDescriptorID>
      <evidenceDescriptorID>6719370644036780032
    </evidenceDescriptorID>
    </verificationCheck>
  </verification>
</root>

```

Figure 23. Inbound Example : VerificationCheck.

This figure displays an example of the inbound VerificationCheck xml message.

## Response Message

The parameters are contained within the struct:  
verification.sl.infrastructure.struct.EvidenceVerificationDetails

Table 37. Response Elements

| Map from parameter   | Reponse element      | Type    |
|----------------------|----------------------|---------|
| caseID               | caseID               | long    |
| verificationRequired | verificationRequired | boolean |
| evidenceDescriptorID | evidenceDescriptorID | long    |

## Response parameter descriptions:

Table 38. Parameter Descriptions

| Parameter            | Domain/Attribute       | Description  |
|----------------------|------------------------|--|
| caseID               | CASE_ID                | The case identifier for the case to which all the evidence records are connected to. Type: long.                             |
| verificationRequired | n/a                    | A value to state whether the case has any evidence records with outstanding verifications, ie "Not verified". Type: boolean. |
| evidenceDescriptorID | EVIDENCE_DESCRIPTOR_ID | The evidence descriptor identifier of any evidence records that where found to be in a state of "Not verified". Type: long.  |

```

<receiveDocumentReturn>
  <response>
    <verificationCheck success="true">
      <caseID>-4377498837804122112</caseID>
      <verificationRequired>true</verificationRequired>
      <evidenceDescriptorID>
        810647932926689280
      </evidenceDescriptorID>
      <evidenceDescriptorID>
        6719370644036780032
      </evidenceDescriptorID>
    </verificationCheck>
  </response>
</receiveDocumentReturn>

```

Figure 24. Response Example : VerificationCheck.

This figure displays an example of the VerificationCheck response xml message.

---

## Determination

### The Determination Service

Determination takes information gathered in the Cúram system as part of intake, and applies it against enterprise-specific and program-specific rules to create eligibility decisions. Determination is different for each program. It requires a ruleset that has been defined to evaluate a participant's eligibility for benefit from a particular product.

Determination is a full eligibility test. It requires a full set of business rules and the set of data on which these rules operate. The outcome of running these business rules is that the client has either been submitted for eligibility testing or failed during the submittal process.

At minimum, the following information is needed for determination.

- Configuration data covering the product and its associated set of business rules to define eligibility for this product must exist in the Cúram system.
- The participant must exist in the Cúram system. This can be achieved using the Register Person service.
- The case must exist in the Cúram system. This can be achieved using the Claim Intake service.
- All evidence used in determination must exist in the Cúram system, and be available for the case. This can be achieved using a combination of the Evidence maintenance services.
- The decisions created by determination will be stored in the Cúram system, but only the submitted for approval verification will be returned to the calling service.

## Incoming Parameters

A request for determination must include the case identifier which is used to identify the case to be submitted for approval via the populating of the struct core.SubmitForApprovalKey.

The fromDate and toDate are used to determine if the case's current certification date values are currently before the fromDate or after the toDate. If not the case the certification records are modified to incorporate the inbound dates.

Table 39. Incoming Parameters

| Intake Element | Map to Parameter                            | Schema Type       |
|----------------|---|-------------------|
| caseID         | caseID                                      | bt:caseIdentifier |
| fromDate       | ProductDeliveryCertDiaryDtls.periodFromDate | periodFromDate    |
| toDate         | ProductDeliveryCertDiaryDtls.periodToDate   | periodToDate      |

## Incoming Parameter Descriptions

Table 40. Parameter Descriptions

| Parameter | Domain     | Description   |
|-----------|------------|---|
| caseID    | CASE_ID    | The case's identification.<br>Type: long  |
| fromDate  | CÚRAM_DATE | The date from which the determination period is submitted for the case.<br>Format: ddMMyyyy |
| toDate    | CÚRAM_DATE | The date to which the determination period is submitted for the case.<br>Format: ddMMyyyy   |

```

<root>
  <determination>
    <submitforapproval>
      <caseID>-4377498837804122112</caseID>
      <fromDate>23112007</fromDate>
      <toDate>30112007</toDate>
    </submitforapproval>
  </determination>
</root>

```

Figure 25. Inbound Example : Determination.

This figure displays an example of the inbound Determination xml message.

## Response Message

Table 41. Response Parameters

| Map from Parameter | Reponse Element | Type    |
|--------------------|-----------------|---------|
| caseID             | caseID          | long    |
| submitted          | submitted       | boolean |

## Response Parameter Descriptions

Table 42. Parameter Descriptions

| Parameter | Domain/Attribute | Description  |
|-----------|------------------|--|
| caseID    | CASE_ID          | The case identifier of the case was submitted for approval. Type: long                 |
| submitted | n/a              | The indicator as to whether the determination was submitted for approval. Type boolean |

```
<receiveDocumentReturn>
  <response>
    <submittedForApproval success="true">
      <caseID> -4998995586381250560 </caseID>
      <submitted> true </submitted>
    </submittedForApproval>
  </response>
</receiveDocumentReturn>
```

Figure 26. Response Example : Determination.

This figure displays an example of the Determination response xml message.

```
<receiveDocumentReturn>
  <response>
    <submittedForApproval success="false">
      <caseID> -4998975586381250560 </caseID>
      <exception>
        <message>The determination submit procedure failed.
          Please contact the administrator.
        </message>
        <exceptionMessage>
          Record not found.
        </exceptionMessage>
      </exception>
    </submittedForApproval>
  </response>
</receiveDocumentReturn>
```

Figure 27. Error Response Example : Determination.

This figure displays an example of the Determination error response xml message.

---

## Triage

### The Triage Service

Triage applies an initial level of review to a basic set of information, to determine a client's need or likely benefit from a program or service.

Triage is not a full eligibility test. Triage requires a small set of business rules and consequently, a small amount of information to process the rules. The results provided by Triage are indicative, not final they are suggestive of what a client may be entitled to.

Triage is different for each program and therefore this service must be tailored to each solution. The Triage web service provides an easy means of routing to the solution specific service. Each solution requires a ruleset to be executed which evaluates the indicative entitlement for their product.

There a number of requirements that must be met before the Triage service can be successfully run:

- The ruleset must be configured and available in the Cúram system.
- The evidence over which the ruleset runs must exist in the Cúram system. This can be achieved using a combination of the Evidence maintenance services.
- The participant must exist in the Cúram system. This can be achieved using the Register Person service.

## Incoming Parameters

The parameters are used to populate the struct:  
Cúram.core.sl.struct.CaseIDandTriageTypeKey

Table 43. Minimum Requirements

| Intake Element | Map to Parameter | Schema Type       |
|----------------|------------------|-------------------|
| caseID         | caseID           | bt:caseIdentifier |
| trriageType    | tirageType       | bt:trriageType    |

## Incoming Parameter descriptions

Table 44. Parameter Descriptions

| Parameter   | Domain           | Description   |
|-------------|------------------|---|
| caseID      | CASE_ID          | The evidence descriptor identifier of the evidence record. Type: long                   |
| trriageType | TRIAGE_TYPE_CODE | The code table value to show the type of triage operation to run. Codetable: triageType |

## Response Message

The parameters are contained within the struct: core.sl.struct.TriageResult

Table 45. Response Parameters

| Map from parameter | Reponse element | Type    |
|--------------------|-----------------|---------|
| caseID             | caseID          | long    |
| qualified          | qualified       | boolean |
| amount             | amount          | double  |

## Response Parameter Descriptions

Table 46. Parameter Descriptions

| Parameter | Domain/Attribute | Description  |
|-----------|------------------|--|
| caseID    | CASE_ID          | The case identifier for the case to which the triage relates. Type: long                               |
| qualified | n/a              | The indicator as to whether the case has qualified for a payment, via the triage process. Type boolean |
| amount    | CÚRAM_MONEY      | The monetary amount due in payment to the case. Type: double   |

---

## 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 10504-17*

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, Nihombashi*

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*

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-IBM products. 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](http://www.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.







Printed in USA