

IBM Cúram Social Program Management
Version 7.0.10

*Cúram Universal Access Configuration
Guide*



Note

Before using this information and the product it supports, read the information in [“Notices” on page 38](#)

Edition

This edition applies to IBM® Cúram Social Program Management v7.0.10 and to all subsequent releases unless otherwise indicated in new editions.

Licensed Materials - Property of IBM.

© **Copyright International Business Machines Corporation 2012, 2020.**

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

© .

Contents

Tables.....	V
Chapter 1. Configuring Universal Access.....	1
Prerequisites.....	2
Configuring Triage.....	6
Configuring Triage Script and Rule set.....	6
Configuring Triage Results.....	6
Writing Rule Sets For Triage.....	7
Configuring Service Areas and PDF Forms.....	11
Configuring Service Areas.....	11
Configuring PDF Forms.....	12
Searching For A Local Office.....	12
Configuring Programs.....	12
Introduction.....	12
Configuring a New Program.....	12
Defining Local Offices For A Program.....	16
Defining PDF Mappings For A Program.....	16
Defining Program Evidence Types.....	17
Configuring Screenings.....	17
Introduction.....	17
Configuring A New Screening.....	17
Configuring Eligibility and Screening Details.....	17
Configuring Screening Display Information.....	18
Defining Programs For A Screening.....	19
Configuring Authenticated Screening.....	19
The screening auto-save property.....	19
Displaying A Confirmation Page On Quit.....	19
Prepopulating The Screening Script.....	19
Writing Rule Sets For Screening.....	19
Configuring Applications.....	21
Introduction.....	21
Configuring A New Application.....	21
Configuring An Application Script.....	22
Configuring A Submission Script.....	22
Defining A PDF Form.....	22
Configuring Client Registration.....	22
Configuring Submission Confirmation Page Details.....	23
Associating Programs With Applications.....	23
Defining Mappings For An Application.....	23
Configuring Withdrawal Reasons.....	23
Mandating Authentication Before Applying.....	23
Optional Authenticated Application.....	23
Displaying A Confirmation Page On Quit.....	23
Mandating Authentication Before Submission.....	24
Enabling Applications Link.....	24
Prepopulating The Application Script.....	24
The application auto-save property.....	24
Configuring Online Categories.....	24
Configuring Online Categories.....	24
Associating Screenings And Applications.....	24

Configuring the Citizen Account.....	25
Configuring The Citizen Account Home Page.....	25
Configuring Contact Information.....	27
Configuring Appeal Requests.....	28
Configuring the number of communications on the Notices tab.....	29
Configuring Life Events.....	29
Introduction.....	29
Configuring a Life Event.....	29
Categorizing Life Events.....	31
Defining Remote Systems.....	31
Configuring Authentication Factors.....	31
Configuring An Authentication Factor.....	32
Activating Second Level Authentication.....	32
Configuring Languages.....	33
Configuring A New Language.....	33
Configuring High Contrast.....	33
Enable High Contrast Selector.....	33
Configuring Remote Systems.....	33
Configuring A Remote System.....	33
Configuring Motivations.....	34
Configuring a New Motivation.....	34
Configuring the Motivation Results Page.....	34
UIM Support in IBM Cúram Universal Access.....	35
Notices.....	38
Privacy Policy considerations.....	39
Trademarks.....	39

Tables

1. Information Message for Browser preference.....	3
2. UIM in Universal Access.....	36

Chapter 1. Configuring Universal Access

Various configuration options are available for IBM Cúram Universal Access. These are listed below.

Prerequisites

Cookies and JavaScript need to be enabled in the browsers to access the **Citizen Portal** application. The following chart lists the browser prerequisites for the application to work and the following errors that are displayed if these per-requisites are not met

T
a
b
l
e
1
.
I
n
f
o
r
m
a
t
i
o
n
M
e
s
s
a
g
e
f
o
r
B
r
o
w
s
e
r
p
r
e
f
e
r
e
n
c
e
(
c
o
n
t
i
n
u
e
d
)

**B
n
t
w
s**

Configuring Triage

Triage results show services that meet a client's needs based on answers that are provided in the triage script. You must define a triage question script and rule set to produce triage results. You can configure the triage results, specifying pre-selections and how providers of services are displayed in Google maps.

Configuring Triage Script and Rule set

Triage uses an IEG (Intelligent Evidence Gathering) script combined with a CER (Cúram Eligibility Rule set) rule set to derive triage results. The IEG script can be defined in the IEG section of the Cúram Administration Application. The `curam.citizenworkspace.triage.script.id` and `curam.citizenworkspace.triage.datastore.schema` system properties are used to specify the triage script and associated data store schema for UA. The `curam.citizenworkspace.triage.ruleset.name` system property is used to specify the CER rule set to run when triage is performed in UA. For more information on writing rule sets for triage, see [“Writing Rule Sets For Triage” on page 7](#).

Configuring Triage Results

This sections outlines the configuration options available for the triage results page. The sections below detail triage configurations.

Configuring Services List

When the results page opens initially a number of services may be preselected and the locations of the providers of the preselected services are displayed on the map. Preselecting a number of services will prevent potentially hundreds of providers being displayed on the map which may confuse the citizen. The `curam.citizenworkspace.triage.num.services.selected` system property is provided to dictate the number of services which are preselected, for example, if the preconfigured number is set to 10 and more than 10 services are returned, then the first 10 services are preselected; if the preconfigured number is set to 10 and 10 services or less are returned then all the services are preselected. The default value for this property is 10.

Configuring the Map

A number of configuration options are provided which dictate how the map is displayed when the triage results page is first opened. These include the central point, the zoom level and a geocoding bias.

Central Point

In order for the map to display providers specific to an agency's location, a system property is provided which allows the map to be opened displaying a particular area. Two system properties, `curam.citizenworkspace.triage.map.default.center.latitude` and `curam.citizenworkspace.triage.map.default.center.longitude` are provided which dictate the central point at which the map will open.

Zoom Level

The initial zoom level of the Google Maps display can be configured using the system property `curam.citizenworkspace.triage.map.default.zoom.level`. Zoom levels between 0 (the lowest zoom level in which the entire world can be seen on one map) to 21+ (where individual buildings can be seen) are possible. The default value for this property is 12.

The zoom level of the map also effects whether icons displayed are useful or not i.e. displaying icons when an entire country is displayed on the map will not help a citizen find an appropriate provider close to them. The system property `curam.citizenworkspace.triage.map.min.zoom.level.display.pins` is provided which dictates the zoom level at which icons will no longer be displayed. This can be set to a value between 0 and +21. For example, if this is set to 0 (world view), icons are displayed at all zoom levels. If it is not set, icons will always be displayed. The default value for this setting is 8. If a citizen zooms to a level that no longer allows icons to be displayed, a message can be displayed to the citizen informing them that they must zoom in to see icons. The

curam.citizenworkspace.triage.map.show.message.min.zoom.passed system property is provided to dictate if this message is displayed or not i.e. it works in conjunction with curam.citizenworkspace.triage.map.min.zoom.level.display.pins. The default value for this property is 'YES'.

Geocoding Bias

The citizen is also provided with the ability to search for providers in a particular area, for example, if the citizen specifies 'Springfield' all providers of the services listed in the Springfield area will be displayed on the map. An additional area can be defined known as the geocoding bias which dictates the north/south/east/west boundaries of the map within which providers should be returned i.e any providers whose location is beyond this boundary are not displayed. In the example above, there may be numerous locations called 'Springfield' and so the geocoding bias can be used to dictate the area of the map that should be displayed to a citizen. The geocoding bias can be defined in the Triage section of the UA Administration Application. The administrator can navigate a map to zoom and select a bias area.

Caching Provider Information

Provider information is cached to improve performance. A number of configuration settings are provided which relate to the caching of provider information. The cache is an in-memory representation of the configured geocode bias area segmented into a 2-d array. System properties curam.citizenworkspace.num.provider.cache.rows and curam.citizenworkspace.num.provider.cache.cols determine the number of rows and columns that the array contains.

The system property curam.citizenworkspace.triage.expire.provider.cache is used to determine whether the client-side provider cache expires. It is recommended that this is always set to 'YES' so that the provider information displayed matches what is stored on the data base for that provider. Setting this property to 'NO' could result in provider data being out of sync with the providers that are mapped. The system property curam.citizenworkspace.triage.provider.cache.lifespan works in conjunction with the curam.citizenworkspace.triage.expire.provider.cache property. It is used to determine the period (in minutes) that the provider cache will live for. Once the cache has been populated for this period it is automatically reloaded to ensure that it is consistent with the data on the database. The default for this property is 1440 minutes (60 [minutes]* 24 [hours]).

Defining Google Maps License Keys

A license for Google Maps is required in order for the triage map to display. The system property curam.citizenworkspace.google.map.license.key is used to store the license key. A free key can be requested on <http://code.google.com/apis/maps/signup.html>.

A security modal is displayed before the triage map is launched because mixed content is displayed i.e. secure (Cúram) information and non secure information (Google Maps) which results in a warning being displayed to the user. To avoid this, a premier account with Google Maps is required and HTTPS must be switched on for this account. Two system properties are provided to store this information. curam.citizenworkspace.enable.https.google.map is used to dictate that HTTPS is being used. curam.citizenworkspace.google.client.id is used to store the Google Maps client ID (as opposed to the map key stored in curam.citizenworkspace.google.map.license.key which is used in the standard HTTP map).

Prepopulating The Triage Script

When triage is performed from within the citizen account, information already known about the citizen performing the triage can be pre-populated. The system property curam.citizenaccount.prepopulate.triage is provided which dictates whether the IEG script is pre-populated or not. The default value of this property is true which means that the script will be pre-populated with information already known about the citizen.

Writing Rule Sets For Triage

All triage rule sets will need to use the triage interface so that they can be executed within UA. The interface is detailed below:

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
Copyright 2010-2011 Curam Software Ltd.
All rights reserved.

This software is the confidential and proprietary
information of Curam Software, Ltd. ("Confidential
Information"). You shall not disclose such Confidential
Information and shall use it only in accordance with
the terms of the license agreement you entered into
with Curam Software.
-->
<RuleSet
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation=
    "http://www.curamsoftware.com/CreoleRulesSchema.xsd"
  name="TriageInterfaceRuleSet">

  <!-- This class must be extended by all rule sets invoked by
  the Citizen Portal screening results processing. -->
  <Class name="AbstractTriageResult" abstract="true">

    <Attribute name="alertMessage">
      <type>
        <javaclass name="List">
          <ruleclass name="AbstractAlertMessage"/>
        </javaclass>
      </type>
      <derivation>
        <abstract/>
      </derivation>
    </Attribute>

    <Attribute name="services">
      <type>
        <javaclass name="List">
          <ruleclass name="AbstractService"/>
        </javaclass>
      </type>
      <derivation>
        <abstract/>
      </derivation>
    </Attribute>

    <Attribute name="programs">
      <type>
        <javaclass name="List">
          <ruleclass name="AbstractProgram"/>
        </javaclass>
      </type>
      <derivation>
        <abstract/>
      </derivation>
    </Attribute>

    <Attribute name="screenings">
      <type>
        <javaclass name="List">
          <ruleclass name="AbstractScreening"/>
        </javaclass>
      </type>
      <derivation>
        <abstract/>
      </derivation>
    </Attribute>

    <Attribute name="intakeApplications">
      <type>
        <javaclass name="List">
          <ruleclass name="AbstractIntakeApplication"/>
        </javaclass>
      </type>
      <derivation>
        <abstract/>
      </derivation>
    </Attribute>

    <Attribute name="needs">
      <type>
        <javaclass name="List">
          <ruleclass name="AbstractNeed"/>
        </javaclass>
      </type>
    </Attribute>
  </Class>
</RuleSet>

```

```

        </type>
        <derivation>
          <abstract/>
        </derivation>
      </Attribute>
    </Class>

<!-- Abstract Need -->
<Class abstract="true" name="AbstractNeed">
  <Attribute name="description">
    <type>
      <javaclass name="curam.creole.value.Message"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>

  <Attribute name="identifier">
    <type>
      <javaclass name="String"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>

  <Attribute name="appropriate">
    <type>
      <javaclass name="Boolean"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>

  <Attribute name="priority">
    <type>
      <javaclass name="Number"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>
</Class>

<!-- Abstract Screening -->
<Class abstract="true" name="AbstractScreening">
  <Attribute name="description">
    <type>
      <javaclass name="curam.creole.value.Message"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>

  <Attribute name="appropriate">
    <type>
      <javaclass name="Boolean"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>

  <Attribute name="identifier">
    <type>
      <javaclass name="String"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>
</Class>
<!-- End Abstract Screening -->

<!-- AbstractIntakeApplication -->
<Class abstract="true" name="AbstractIntakeApplication">
  <Attribute name="description">
    <type>

```

```

        <javaclass name="curam.creole.value.Message"/>
    </type>
    <derivation>
        <abstract/>
    </derivation>
</Attribute>

<Attribute name="appropriate">
    <type>
        <javaclass name="Boolean"/>
    </type>
    <derivation>
        <abstract/>
    </derivation>
</Attribute>

<Attribute name="identifier">
    <type>
        <javaclass name="String"/>
    </type>
    <derivation>
        <abstract/>
    </derivation>
</Attribute>
</Class>
<!-- End AbstractIntakeApplication -->

<Class name="AbstractAlertMessage" abstract="true">
    <Attribute name="alertRequired">
        <type>
            <javaclass name="Boolean"/>
        </type>
        <derivation>
            <abstract/>
        </derivation>
    </Attribute>
    <Attribute name="message">
        <type>
            <javaclass name="curam.creole.value.Message"/>
        </type>
        <derivation>
            <abstract/>
        </derivation>
    </Attribute>
</Class>

<!-- Abstract Service -->
<Class abstract="true" name="AbstractService">
    <Attribute name="description">
        <type>
            <javaclass name="curam.creole.value.Message"/>
        </type>
        <derivation>
            <abstract/>
        </derivation>
    </Attribute>

    <Attribute name="appropriate">
        <type>
            <javaclass name="Boolean"/>
        </type>
        <derivation>
            <abstract/>
        </derivation>
    </Attribute>

    <Attribute name="identifier">
        <type>
            <javaclass name="String"/>
        </type>
        <derivation>
            <abstract/>
        </derivation>
    </Attribute>

    <Attribute name="groupInd">
        <type>
            <javaclass name="Boolean"/>
        </type>
        <derivation>
            <abstract/>
        </derivation>
    </Attribute>

```



```

</Attribute>

<Attribute name="needIdentifier">
  <type>
    <javaclass name="String"/>
  </type>
  <derivation>
    <abstract/>
  </derivation>
</Attribute>

</Class>

<!-- Abstract program -->
<Class abstract="true" name="AbstractProgram">
  <Attribute name="description">
    <type>
      <javaclass name="curam.creole.value.Message"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>

  <Attribute name="appropriate">
    <type>
      <javaclass name="Boolean"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>

  <Attribute name="programTypeReference">
    <type>
      <javaclass name="String"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>

  <Attribute name="needIdentifier">
    <type>
      <javaclass name="String"/>
    </type>
    <derivation>
      <abstract/>
    </derivation>
  </Attribute>

</Class>

</RuleSet>

```

The ruleset specified by the `curam.citizenworkspace.triage.ruleset.name` application property must include a rule class, typically named "Triage" which extends the `AbstractTriageResult` rule class outlined above.

This usage of the `AbstractTriageResult` rule class guarantees that the required attributes are available during the rules execution.

Configuring Service Areas and PDF Forms

You can define a service area by configuring the counties or ZIP codes that are associated with the service area. You can also specify a PDF form that a citizen can use to apply for programs.

Configuring Service Areas

Service areas are defined in the Service Areas section of the UA Administration Application. When defining a service area a name must be specified, for example, Washington. Once the service area is created, counties and zip codes can then be associated with the service area. These counties and zip codes represent the areas covered by the service area. Once defined, service areas can be associated

with a local office which represents the office that services the service areas associated with it. Local offices are used to identify where a citizen can apply in person for a particular program and/or where they can send a particular application. For more information on associating service areas with local offices where a citizen can apply in person for a program, see [“Defining Local Offices For A Program” on page 16](#) . For more information on associating a service area with a local office where a citizen can send an application, see [“Associating a Local Office with a PDF Form” on page 12](#).

Configuring PDF Forms

PDF Forms are defined in the PDF Forms section of the UA Administration Application. When defining a PDF Form a name, PDF Form and language must be specified. The agency can add a version of the form for each language (locale) configured for UA. Once defined, these forms are accessible from the Print and Mail Form page of UA.

Associating a Local Office with a PDF Form

A local office can be associated with a PDF Form. Associating a local office with a PDF Form allows an administrator to define the local office and associated service areas where a citizen should send their completed application to.

Searching For A Local Office

A search page is provided in UA which allows a citizen to search for a local office where they can send an application to or apply in person. The citizen can either search by county or by zip code. The system property `curam.citizenworkspace.page.location.search.type` is provided which dictates which of these options is made available to a citizen. If this property is set to 'Zip' the citizen will be able to search for a local office using a zip code. If this property is set to 'County' the citizen will be able to select from a list of counties in order to return a list of local offices.

Configuring Programs

An overview of the configurations that are available for defining programs in UA. You can define different types of programs in Universal Access. After you define a program, you can configure display and system processing information, local offices, mappings to PDFs, and evidence types.

Introduction

UA allows agencies to define different types of programs. Once programs have been defined, they can be associated with screenings and applications which allow clients to screen and apply for these programs via UA. There are three main aspects to configuring a program:

- Configuring programs and associated display and system processing information
- Configuring local offices where an application for a program can be mailed to
- Configuring mappings which allow information gathered during application intake to be mapped to a PDF form
- Configuring evidence types which allows for expedited authorization of programs which may need to be processed before other programs within a multi program application

Configuring a New Program

Programs are configured on the **New Program** page. Details and specifications of the program are required to be defined when the program is created.

The sections that follow detail program configurations.

Defining a Name and Reference

A name and reference must be defined when a new program is created. The name that is defined is displayed in the internal application. The reference is used to reference the program in code.

A name and reference must be defined when creating a new program. The name that is defined is displayed to the citizen in UA and in the internal application. The reference is used to reference the program in code.

Defining an Intake Processing System

An intake processing system must be specified for each program. If a case processing system is not specified, the system is not able to respond to the submission of an application for that program.

Two options are available: **Cúram** or select from pre-configured remote systems. Select **Cúram** if intake is managed by a Cúram-based system. Select a remote system if intake is managed by a non-Cúram-based system (the program application then is sent to the remote system by using the ProcessApplicationService web service).

If **Cúram** is specified as the intake system, an application case type must be selected. An application case of the specified type is created in response to a submission of an application for the program. An indicator is provided which dictates whether a **Reopen** action is enabled on the programs list on an application case for denied and withdrawn programs of a particular type. A workflow can be specified that is initiated when the program is reopened. For more information on configuring application cases, see the *IBM Cúram Intake Configuration Guide*.

When an application case type is selected, the program can be added manually to that type of application case by a worker in the internal application as part of intake processing. A configuration is provided that dictates if the program is a coverage type. Coverage types are automatically evaluated by program group rules in the context of healthcare reform applications, such as insurance affordability. As such, coverage types cannot be applied for directly by a citizen or manually added to an application case by a worker and authorized. If the program is a coverage type, select the 'Yes' option. The program will be filtered out of the list of programs available to be added to online and internal applications in administration and the list of programs available to be manually added to an application case by a worker. If the program is not a coverage type, select the 'No' option. The program will be available to be manually added to online and internal applications in administration and to an application case by a worker.

A remote system must be configured in the administration application before it can be selected as the case processing system. For more information about remote systems, see the "Configuring Remote Systems".

Defining Case Processing Details

A case processing system must be specified for each program.

Two options are available; **Cúram** or select from pre-configured remote systems. Select **Cúram** if the program eligibility is determined and managed by using a Cúram-based system. Select a remote system if eligibility is determined and managed by a non-Cúram-based system.

If **Cúram** is selected as the case processing system, more options are available to allow for program level authorization to be configured. Program level authorization means that if an application case contains multiple programs, each program can be authorized individually, and a separate case is used to manage the clients on an ongoing basis.

Defining the Integrated Case Strategy

The Integrated Case Strategy must be defined in order for the system to identify whether a new or existing integrated case needs to be used when program authorization is successful. The integrated case is used to host any product deliveries created as a result of the authorization.

The Integrated Case Strategy must be defined in order for the system to identify whether a new or existing integrated case should be used when program authorization is successful. The integrated case is used to host any product deliveries created as a result of the authorization. If a new integrated case is created, all of the application case clients are added as case participants to the integrated case. If an existing integrated case is used, any additional clients on the application case are added as case participants to the integrated case. Any evidence captured on the application case that is also required on

the integrated case is copied to the integrated case upon successful authorization. The configuration options for the integrated case strategy are as follows:

New

A new integrated case of the specified type is always created when authorization of the program is successful.

Existing (Exact Client Match)

If an integrated case of the specified type exists with the same clients as those cases present on the application case, the existing case is used automatically. If multiple integrated cases that meet these criteria exist, the worker is presented with a list of the cases and must select one to proceed with the authorization. If no existing cases match the criteria, a new integrated case is created.

Existing (Exact Client Match) or New

If one or more integrated cases of the specified type exist with the same clients as those cases present on the application case, the user is presented with the option to select an existing case to use as the ongoing case, or to create a new integrated case. If no existing cases match the criteria, a new integrated case is created.

Existing (Any Client Match) or New

If one or more integrated cases of the specified type exist, where any of the clients of the application case are case participants, the user is presented with the option to select one of the existing cases to use as the ongoing case, or to create a new integrated case. If no existing cases match the criteria, a new integrated case is created.

Specifying the Integrated Case Type

The administrator must specify the type of integrated case to be created/used upon successful program authorization as defined by the Integrated Case strategy listed previously.

Specifying a Client Selection Strategy

The Client Selection Strategy is used to define how clients are added from the application case to the product delivery created as a result of authorization of a program. If a product delivery type is specified, a client selection strategy must be selected. The configuration options are as follows:

All Clients

All of the application clients are added to the product delivery case. The application case primary client is set as the product delivery primary client. Once the primary client is set, it cannot be changed. All other clients are added to the product delivery as members of the case members group.

Rules

A rule set is used to determine the clients to be added to the product delivery(if a product delivery is configured). At least one client must be determined by the rules for authorization to proceed.

User Selection

The user selects the clients who should be added to the product delivery. The user must select both the primary client and any other clients to be added to the case member group on the product delivery.

Specifying a Client Selection Ruleset

A Client Selection Ruleset must be selected when the Client Selection Strategy is 'Rules'.

Specifying a Product Delivery Type

The Product Delivery Type is used to specify the product delivery used to make a payment to a client(s) in respect of a program. The drop down displays all active products configured on the system.

Note: This field applies to both program and application authorization processing i.e. Program and application authorization can result in the creation of the product delivery type specified.

Submitting a Product Delivery Automatically

The Submit Product Delivery indicator is used to dictate if the product delivery created as a result of program authorization should be submitted automatically for approval. If selected, the product delivery

created as a result of authorization of this program is submitted automatically for a supervisor for approval.

Note: This field applies to both program and application authorization processing i.e. Program and application authorization can result in the automatic submission of a product delivery.

Configuring Timers

Many organizations impose time limits within which an application for a program must be processed. Cúram provides organizations with the ability to configure application timers for each program.

For example, a government agency may have a requirement that food assistance applications must be authorized within 30 business days of the date of application.

A number of configuration options are available, including the duration of the timer, whether the timer is based on business or calendar days, a warning period, and timer extension and approval. All configuration options are outlined in the following list.

Duration

The length of the timer in days. This value, along with the fields **Start Date** and **Use Business Days** (and the configured business hours for the organization) are used to calculate the expiry date for the timer. This value is used as a number of business days if **Use Business Days** is set. If **Use Business Days** is not set, this value is used as calendar days.

Start Date

Specifies whether the timer needs to start on the application date or the program addition date. The options available are **Application Date** and **Program Addition Date**.

Note: In most cases, these dates are the same (the programs will be added at the same time as the application is made) but when a program is added later to the application, after initial submission, this information might not be the case.

Warning Days

Allows for the specification of a number of warning days. The warning days are used to warn the user that the timer deadline is approaching. If configured, the Warning Reached workflow also is enacted when the warning date is reached and the timer is still running (for example, the program is not completed).

End Date Extension Allowed

An indicator to dictate if a user can extend the timer by a number of days.

Extension Approval Required

An indicator to dictate if a timer extension requires approval from a supervisor. If approval is required for the extension, the case supervisor must review and either approve or reject the extension. After the extension is approved, or if approval is not required, the timer expiry date is updated to reflect the extension.

Use Business Days

An indicator to dictate that the timer should not decrement over non-working days. If this indicator is set, the system uses the **Working Pattern Hours** for the organization to determine the non-working days when it is calculating the expiry date for the timer.

Resume Timer

An indicator to dictate whether the program timer needs to be resumed when the program is reopened.

Resume From

If a timer is resumed, the **Resume From** field dictates the dates from which a program can be resumed. The values include the date that the program was completed (denied or withdrawn), and the date that the program was reopened.

Timer Start

Allows for the specification of a workflow that is enacted when the timer starts.

Warning Reached

Allows for the specification of a workflow that is enacted when the warning period is reached.

Deadline Not Achieved

Allows for the specification of a workflow that is enacted if the timer deadline is not achieved; that is, the program is not being withdrawn, denied, or approved by the timer expiry date.

Configuring Multiple Applications

The Multiple Applications indicator dictates if a citizen can apply for this program while they have a previous application pending. If set to true, the citizen can have multiple pending applications for the given program i.e. they can submit an application for this program while they already have a pending application in the system. If it is set to false, this program will not be offered for application in UA if the currently logged in citizen has pending applications for this program.

This configuration is not applicable to Health Care Reform Applications.

Defining A PDF Form

Defining a PDF form for a program enables a citizen to print an application for the specified program and either post it to the agency or bring it, in-person, to a local office. When a PDF Form is specified for a program, the PDF form will be displayed on the Print Out and Mail section of the Your Next Step page displayed when a citizen has completed a screening. PDF Forms must be defined before they can be associated with a program. When they are defined, they will be displayed on the Print and Mail Application Form page that is accessible from UA.

Defining a URL

If a URL is defined, a 'More Info' link is displayed beside the program name in UA allowing a citizen to find out more information about the selected program.

Defining Description and Summary Information

When a program is displayed to a citizen on the Select Programs page in UA, a description can be displayed which gives a description of the program. The Online Program Description field allows for the definition of this description.

A description summary of the program can also be defined using the Online Program Summary field. This is a high level description of the program displayed on the 'Your Next Steps' page displayed when a citizen has completed a screening.

Defining Local Office Application Details

It may be possible for a program to be applied for at a local office. If this is the case, the Citizen Can Apply At Local Office indicator can be used to dictate that local office information is displayed in UA for a particular program. Additional information can also be defined, for example, a citizen may need to bring proof of identity if the citizen wants to apply at the local office. An administrator can define this information in the Local Office Application Information field.

Defining Local Offices For A Program

A citizen may be able to apply for a program in person at a local office. A local office must be first defined in the LocalOffice code table in system administration. Associating a local office with a program allows an administrator to define the local offices and their associated service areas where a particular program can be applied for in person. This information is displayed to a citizen on the Your Next Step page which is displayed to a citizen when they have performed a screening. Note: Service areas must be defined before they can be associated with a local office.

Defining PDF Mappings For A Program

The information entered during an online application can be mapped to a PDF Form which can be subsequently printed by the citizen. In order for the application data to be mapped to the PDF Form for all programs a citizen is applying for, a mapping configuration of type PDF Form Creation must exist for each of the programs. Note: this PDF Form is the form specified for the Online Application the program is associated with.

Defining Program Evidence Types

Evidence Types can be associated with a program. This can be used by an organization to provide support for applications for multiple programs where a particular program needs to be authorized much more quickly than other programs for which the client may have applied, for example, expedited food assistance. Using this type of configuration, only the evidence required for the program to be authorized will be used and copied to the ongoing cases. This allows benefits for the authorized program to be delivered to the client, while the worker continues to gather the evidence required for the other programs applied for.

Configuring Screenings

An overview of the configurations available for defining screenings in UA administration. You can define different types of UA screenings that a citizen can complete to identify programs that they might be eligible to receive. For each screening, you can configure the available programs and eligibility requirements. You can then configure the script, rules, and data schema to collect and process citizen information, and define what information is displayed to the citizen.

Introduction

UA administration allows an agency to define different types of screenings. Once defined, a citizen can perform a screening to identify programs that they may be eligible to receive. There are four main aspects to configuring a screening:

- configuring information about a screening to be displayed to a citizen in UA
- configuring the script, rules and schema used to collect and process the information specified by a citizen to identify their eligibility
- configuring the programs for which a citizen can check their eligibility when performing a screening
- configuring additional screening system properties

Configuring A New Screening

Screenings are configured on the New Screening page. The sections below detail screening configurations.

Defining A Name

A name must be defined when creating a new online screening. The name defined is the name of the screening displayed to a citizen in the UA portal.

Defining Program Selection

The Program Selection indicator dictates whether a citizen or user will be provided with the ability to select specific programs that they would like to screen for or whether they will be brought directly into a screening script i.e. they will be screened for all programs associated with the screening.

Defining a More Info URL

If a More Info URL is defined, a 'More Info' link is displayed beside the name of the screening in UA allowing a citizen to find out more information about the selected screening.

Configuring Eligibility and Screening Details

Two types of screening are supported by UA - filtered screening and eligibility screening. Eligibility screening involves collecting answers to a set of questions, storing this information and processing it to identify eligibility. Filtered screening is used to reduce the number of programs for which a citizen can screen. The objective is to reduce the number of programs that a citizen might screen for by asking a short set of questions and using the answers to filter out the programs that they would not be eligible for.

This section outlines how eligibility and filtered screening is configured.

Configuring Eligibility Screening Details

An IEG script must be specified for the screening which is used to collect the answers to a set of questions. A data store schema must also be specified to store the data entered in the script. On saving the screening, an empty template for both the script and schema will be created by the system based on the Question Script and Schema specified. It will then be possible to update these from the Screening tab by selecting hyperlinks provided on the page. Clicking on the Question Script link will launch the IEG Editor which will allow the question script to be edited. Clicking on the Schema link will launch the Datastore Editor which will allow the schema to be edited.

To process the data in the data store and to produce an eligibility result, a CER rule set must be specified. Once specified on creation of the screening, an empty rules template will be created by the system. It will then be possible to update the ruleset from the Screenings tab by selecting the hyperlink provided on the page. Clicking on the link will launch the CER Editor which will allow the ruleset to be edited. For more information about writing screening rule sets, see [“Writing Rule Sets For Screening” on page 19](#)

For more information about using the CER Editor, see the *IBM Cúram Working With Cúram Express Rules Guide*. For more information about the IEG Editor see the *Working With Curam Intelligent Evidence Gathering guide*.

Configuring Filtered Screening Details

In order for filtered screening to be available before performing eligibility screening, filtered screening details must be specified for a screening. As with eligibility screening, a Filter Script (IEG) and associated data store schema must be defined to collect and store the answers to questions, and Filter Rules (CER rule set) must be specified to process the data and produce a filtered screening result. Once specified on the New Online Screening page, the system will automatically create an empty template for the scripts and ruleset which can be subsequently updated by selecting the associated hyperlinks on the Screening page.

Reusing Rule Sets Across Screenings

A system property is provided which dictates whether CER rule sets can be reused across different screenings and also whether the same rule set can be used for eligibility and filtered screening. The system property `curam.citizenworkspace.screening.ruleset.reuse.enabled` can be used to control this. If this property is switched on, rule sets can be reused. If it is switched off, rule sets cannot be reused. Note: the `ScreeningRulesLinkDAO.readActiveByRuleSet` method cannot be used when this property is enabled.

Configuring Screening Display Information

Four pieces of information can be configured for each screening which is displayed in UA. A Summary, Description, Am I Eligible Text and How To Apply text can be defined.

Summary Information

Summary information allows an administrator to define a high level description of the screening.

Am I Eligible Text

Am I Eligible Text allows an administrator to define the text to be displayed on the Am I Eligible page in UA which is displayed to show a citizen the results of a completed screening.

Description

Descriptions allow an administrator to define a description of the screening to be displayed in UA.

How To Apply Text

How To Apply Text allows an administrator to define the text displayed on the Your Next Step page in UA.

Defining Programs For A Screening

In order to screen for particular programs, programs must be associated with the screening. Any program, as described in "Configuring Programs", can be associated with a screening. When associating programs with a screening, an order can be assigned which dictates the display order of the selected program relative to other programs associated with the screening.

Configuring Authenticated Screening

The agency can configure the system to specify whether, before screening, a citizen can choose to be authenticated or not. The system property `curam.citizenworkspace.authenticated.screening` is used to dictate if authentication is switched on or not. If this property is switched on, a citizen will be given the option to login to UA before starting a screening. If this is switched off, the citizen will be taken directly to the screening selection page.

The screening auto-save property

The screening auto-save property dictates if screenings are auto-saved in the citizen account.

By default, this property is set to true. All screenings, irrespective of type, are automatically saved. Each screening is auto-saved when the citizen clicks **Next** when progressing through the IEG script. If this property is set to false, screenings are not automatically saved in the citizen account.

Displaying A Confirmation Page On Quit

The agency may wish to display a confirmation page to a citizen when they choose to quit a screening. The system property `curam.citizenworkspace.display.confirm.quit.screening` can be used to dictate if a confirmation page is displayed or not. If this property is switched on a confirmation page will be displayed to a citizen when quit is selected while performing a screening. If this property is switched off a confirmation page is not displayed when a citizen selects to quit a screening.

Prepopulating The Screening Script

When screening is performed from within the citizen account, information already known about the citizen performing the screening can be pre-populated. The system property `curam.citizenaccount.prepopulate.screening` is provided which dictates whether the IEG script is pre-populated or not. The default value of this property is true which means that the script will be pre-populated with information already known about the citizen.

Writing Rule Sets For Screening

This section describes how to develop screening rule sets. It begins by outlining the type of data store schema required for developing effective screening rule sets.

How to add a data store schema for use in screening

Developers are free to create new data store schema for use with their Universal Access screening and intake intelligent evidence gathering (IEG) scripts. However, some constraints exist on the format of these schemas. In some cases, requirements dictate that clients can screen for a program and then follow that screening with an application for benefits.

In many cases, such applications are processed by Cúram and mapped to Cúram cases and evidence by using the Cúram Data Mapping Engine (CDME). In these circumstances, it is advisable to use the `CitizenPortal.xsd` as a basis for the schema for screening. This process is used because the same data store schema also needs to be used for intake. In particular, the CDME features do not work correctly if a schema is used that removes or changes the data type of any of the attributes or entities in the `CitizenPortal.xsd` schema.

All schema that follows the pattern of the `CitizenPortal.xsd` schema are guaranteed to be safe for future upgrades; for instance, later releases. This assurance means that upgrades do not add any new mandatory attributes or entities. Upgrades do not change any existing attributes or entities that currently are required to support existing Cúram data mapping engine functions.

The Screening Rules Interface

Please note that all screening rule sets will need to use the screening rules interface so that they can be executed within Universal Access. The interface is detailed below:

```
<?xml version="1.0" encoding="UTF-8"?>

<RuleSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://www.curamsoftware.com/
  CreoleRulesSchema.xsd"
  name="ScreeningInterfaceRuleSet">

  <!-- This class must be extended by all rule sets invoked by
  the Citizen Portal screening results processing. -->
  <Class name="AbstractScreeningResult" abstract="true">

    <Initialization>
      <Attribute name="calculationDate">
        <type>
          <javaclass name="curam.util.type.Date"/>
        </type>
      </Attribute>
    </Initialization>

    <!-- The programs supported by this Screening Ruleset. -->
    <Attribute name="programs">
      <type>
        <javaclass name="List">
          <ruleclass name="AbstractProgram"/>
        </javaclass>
      </type>

      <derivation>
        <!-- Subclasses of AbstractScreeningResult must override
        this attribute to create a list of the Programs
        supported by the rule set. -->
        <abstract/>
      </derivation>
    </Attribute>
  </Class>

  <!-- This class must be extended by all programs supported
  in the rule set. -->
  <Class name="AbstractProgram" abstract="true">

    <!-- Identifies the program as configured in the Citizen
    Portal administration application. -->
    <Attribute name="programTypeReference">
      <type>
        <javaclass name="String"/>
      </type>
      <derivation>
        <abstract/>
      </derivation>
    </Attribute>

    <!-- Whether the claimant is eligible for this program. -->
    <Attribute name="eligible">
      <type>
        <javaclass name="Boolean"/>
      </type>
      <derivation>
        <abstract/>
      </derivation>
    </Attribute>

    <!-- The localizable explanation as to why the claimant is
    or is not eligible for this program. May contain HTML
    formatting/hyperlinks/etc. -->
    <Attribute name="explanation">
      <type>
        <javaclass name="curam.creole.value.Message"/>
      </type>
      <derivation>
        <abstract/>
      </derivation>
    </Attribute>
  </Class>
</RuleSet>
```

Screening rule sets must include a class that extends the `AbstractScreeningResult` rule class outlined above.

This usage of the `AbstractScreeningResult` rule class guarantees that the required attributes are available during the rules execution.

Configuring Applications

An overview of the configurations available for defining applications in UA administration. You can define different types of UA applications that a citizen can complete to apply for programs that they might be eligible to receive. For each application, you can configure the available programs and an application script and data schema. Configure the remaining applications details, including application withdraw reasons.

Introduction

UA administration allows an agency to define different types of applications. Once defined, a citizen can submit an application for programs to the agency.

There are five main aspects to configuring an application:

- configuring information about an application and associated display information
- configuring the script and schema used to collect and store the information specified during the application process
- configuring the programs for which an application can be used to apply
- configuring reasons that can be selected if a citizen withdraws an application
- configuring additional application system properties

Configuring A New Application

Applications are configured on the New Application page. The sections below detail application configurations.

Defining A Name

A name must be defined when creating a new application. The name defined is the name of the application displayed to a citizen in the online portal.

Defining Program Selection

The Program Selection indicator dictates whether a citizen or user will be provided with the ability to select specific programs that they would like to apply for or whether they will be brought directly into an application script i.e. they will apply for all programs associated with the application.

Defining a URL

If a URL is defined, a 'More Info' link is displayed beside the name of the application in UA allowing a citizen to find out more information about the selected application.

Configuring Summary Information

Summary information allows an administrator to define a high level description of the application to be displayed in UA.

Configuring Description Information

Descriptions allow an administrator to define a description of the application to be displayed in UA.

Configuring An Application Script

An IEG script must be defined for the application which is used to collect the answers to the application questions. A script name must be specified in the Question Script field. A data store schema must be specified to store the data entered in the script. A schema name must be specified in the Schema field. On saving the application, an empty template for both the script and schema will be created by the system based on the Question Script and Schema specified. It will then be possible to update these from the Application tab by selecting hyperlinks provided on the page. Clicking on the Question Script link will launch the IEG Editor which will allow the question script to be edited. Clicking on the Schema link will launch the Datastore Editor which will allow the schema to be edited.

Configuring A Submission Script

A submission script can be defined for an application in order to submit an application to the agency. This is used to define additional information which does not form part of the application script to be captured, for example, a TANF typically requires information regarding the citizen's ability to attend an interview.

A submission script (IEG) can be specified in the Submission Script field. On saving the application, an empty template for both the submission script and will be created by the system based on the Submission Script specified. It will then be possible to update this from the Application tab by selecting the hyperlink provided on the page. Clicking on the link will launch the IEG Editor which will allow the question script to be edited.

Defining A PDF Form

Defining a PDF form for an application identifies the agency-designed form that is displayed to a citizen when an online application is completed. The data collected during the online application is copied by the system into this PDF, which allows a citizen to print it. The PDF Form can be selected from the PDF Forms drop down menu. If a PDF Form is not specified for an application, a default generic PDF Form can be used. This default template is accessible in the XSL Templates section of the Cúram System Administration Application.

The data passed to the XSL Template reads directly from the data store. Rather than displaying the datastore labels in the PDF(which is not user-friendly), a property file can be defined that can be used to specify user-friendly names for entities and attributes and to hide entities and attributes that should not be displayed in the PDF. This property file must then be uploaded to Application Resources in the Intelligent Evidence Gathering section of the administration application.

The name of the property file needs to be as follows: <application schema name>PDFProps. The contents of the property file should follow these guidelines:

To Name an Entity

<Entity Name=<Name To Be Displayed in the PDF>, for example, Application=Intake Application

To Hide an Entity

<Entity Name.hidden=true, for example, ScreeningType.hidden=true

To Hide an Attribute

<Entity Name.Attribute Name.hidden=true, for example, Application.user Name.hidden=true

To Specify a Label For an Attribute

<Entity Name.Attribute Name=PDF Label, for example, Submission.dig FirstName=First Name

Configuring Client Registration

The Client Registration field allows an administrator to dictate whether clients on online applications are registered as prospects or persons.

To determine whether to register the client as a prospect or a person, the system checks the client registration configuration in the following two scenarios:

- If Person Search and Match is configured and no match can be found for the client
- If Person Search and Match is not configured, i.e., the clients on an application are always registered without the system automatically searching and matching them.

If the Client Registration field is not set, the system checks the system property Register as Prospect Person to identify if a client is registered as a prospect or a person.

Configuring Submission Confirmation Page Details

Additional information can be configured such that it is displayed on the Submission Confirmation page which is displayed when a citizen submits an online application. The Title and Text fields can be used to define a title and text to be displayed on the confirmation page.

Associating Programs With Applications

In order to apply for particular programs, programs must be associated with the application. Any program, as described in "Configuring Programs" can be associated with an application. When associating programs with an application, an order can be assigned which dictates the display order of the selected program relative to other programs associated with the application.

Defining Mappings For An Application

As mentioned earlier an application can be processed by a Cúram based system or a remote system. If the application is processed by a Cúram based system the information entered in an application is mapped to the evidence tables associated with the application case defined for the programs associated with the application. The mappings are configured for an application by creating a mapping using the Cúram Data Mapping Editor. A mapping configuration must be specified in order for the appropriate evidence entities to be created and populated in response to an online application submission.

For more information about the Cúram Data Mapping Editor, see the *IBM Cúram Data Mapping Editor Guide*.

Configuring Withdrawal Reasons

A citizen may withdraw the application for all or any one of the programs for which the citizen has applied. When withdrawing an application, a withdrawal reason must be specified. Withdrawal reasons can be defined for a particular application in the Intake Application section of the UA Administration Application. Before associating a withdrawal reason with an application, withdrawal reasons must be defined in the WithdrawalRequestReason code table.

Mandating Authentication Before Applying

The agency can configure the system to specify whether, before starting an application, a citizen must create an account or login to make an application. The system property `curam.citizenworkspace.authenticated.intake` is used to dictate if authentication is switched on or not. If this property is switched on, a citizen must create an account or login to UA before starting an application. If this is switched off, the citizen will be taken directly to the application selection page.

Optional Authenticated Application

The agency can configure the system to specify whether, before applying, a citizen can choose to be authenticated or not. The system property `curam.citizenworkspace.intake.allow.login` is used to dictate if authentication is switched on or not. If this property is switched on, a citizen will be given the option to login to UA before starting an application. If this is switched off, the citizen will be taken directly to the application selection page.

Displaying A Confirmation Page On Quit

The agency may wish to display a confirmation page to a citizen when they choose to quit the application process. The system property `curam.citizenworkspace.display.confirm.quit.intake` can be used to dictate

if a confirmation page is displayed or not. If this property is switched on, a confirmation page will be displayed to a citizen when quit is selected while making an application. If this property is switched off, a confirmation page is not displayed when a citizen selects to quit an application. This property is only used when the property `curam.citizenworkspace.intake.allow.login` is set to 'NO'.

Mandating Authentication Before Submission

The agency may wish to mandate that a citizen logs in before they submit an application. The system property `curam.citizenworkspace.intake.submit.intake.mandatory.login` can be used to dictate that a citizen logs in before they submit an application. If this property is switched on, a citizen must create an account or login before they can submit an application. If this is switched off, the citizen can submit an application without logging in.

Enabling Applications Link

The system property `curam.citizenworkspace.intake.enabled` can be used to dictate if a citizen can start the application process from the UA Home page. If this indicator is switched on, the applications (Apply For Benefits) link is displayed on the UA Home page. If this is switched off the applications link is not displayed on the UA Home page.

Prepopulating The Application Script

When a citizen is making an application from within the citizen account, information already known about the citizen performing the application can be pre-populated. The system property `curam.citizenaccount.prepopulate.intake` is provided which dictates whether the IEG script is pre-populated or not. The default value of this property is true which means that the script will be pre-populated with information already known about the citizen.

The application auto-save property

The auto-save Intake property dictates if applications are auto-saved in the citizen account.

By default, this property is set to true. All applications irrespective of type are automatically saved. Each application is auto-saved when the citizen clicks **Next** as the citizen progresses through the IEG script. If this property is set to false, applications are not automatically saved in the citizen account.

Configuring Online Categories

An overview of the configurations available for defining online categories in UA administration. Online categories group different types of applications or screenings together to make it easier for a citizen to find the ones that they need. You must define online categories for screenings and applications to be displayed. After you define online categories, you must associate each screening and application to a category.

Configuring Online Categories

When defining an online category a name must be specified. A URL can also be defined and if a URL is defined a 'More Info' link is displayed beside the name of the online category in UA allowing a citizen to find out more information about the selected category. An order can be assigned to a category which dictates the display order in UA of the selected category relative to other categories.

Associating Screenings And Applications

Screenings and applications must be associated with an online category in order for them to be displayed in UA. When associating a screening with an online category, an order can be applied which dictates the display order in UA of the screening relative to other screenings within the same category. When associating an application with an online category an order can be applied which dictates the display order in UA of the application relative to other applications within the same category.

Configuring the Citizen Account

An overview of the configuration options available in the citizen account. Although customization is required to modify some citizen account information, you can configure information on the citizen account home page and the contact information tab.

Messages can originate as a result of transactions in a Cúram system or a remote system. The majority of the configuration options apply to all messages but there are a number of configuration options which do not apply to messages originating from a remote system.

Configuring The Citizen Account Home Page

Configure aspects of the Citizen Account Home Page, including the welcome page displayed to the client, the text display in the client's Messages panel, and the campaigns displayed in the Outreach panel.

Many aspects of the citizen account home page are configurable:

- the text displayed in the participant messages in the messages panel
- system messages displayed in the message panel
- the display order of the messages displayed in the messages panel
- the campaigns displayed in the Outreach panel
- the welcome message displayed to a citizen

This section outlines the citizen account configuration options in more detail.

Configuring Messages

The **Messages** section of the **home page** displays messages that aim to predict why the client logged in. For example, a message can be displayed that informs a client of when their next benefit payment is due or the amount of the last payment.

Messages can be displayed which relate to meetings a client is invited to, activities that are scheduled for a client, application acknowledgments, and similar. Messages can be displayed as a result of transactions in a Cúram system or they can originate from remote systems by way of a web service.

The links that follow outline the aspects of the **Messages** pane, which are configurable.

Account Messages

Adding a message or changing a dynamic element of an existing Cúram message requires customization. The text that is defined for existing messages that are provided in the initial application configuration can be updated by using a set of properties that exist for each type of message

These properties are:

- `CitizenMessageMyPayments` - contains the messages that relate to payments.
- `CitizenMessageApplicationAcknowledgement` - contains the messages that relate to application acknowledgments.
- `CitizenMessageVerificationMessages` - contains the messages that relate to verification messages.
- `CitizenMessageMeetingMessages` - contains the messages that relate to meetings.
- `CitizenMessagesReferral.properties` - contains the messages that relate to referrals.
- `CitizenMessagesServiceDelivery` - contains the messages that relate to service deliveries.
- `CitizenAppealRequestMessage` - contains the messages that relate to appeal requests.

All of these property files are stored in the **Application Resources** section of the **administration** application. To update the message, each file needs to be downloaded, updated, and uploaded again. The icons that are displayed in the Citizen Account for each type of message can be configured in the **Account Messages** section of the **administration** application.

Adding a message that originates from a remote system requires that a code table entry to be added to the `ParticipantMessageType` code table and an associated entry in the **Account Messages** listing in the administration application. Messages then can be sent by way of the `ExternalCitizenMessageWS` web service.

For more information about customizing messages, see the *Customizing Universal Access* documentation.

System Messages

System Messages can be used when an agency wants to send a message to all clients who have a Citizen Account. For example, if an agency wants to provide information and help line numbers to clients who were affected by a natural disaster, such as a flood, hurricane, or earthquake. System Messages can be configured in the *Cúram Universal Access (UA) Administration* application by using the **New System Message** page.

The **Title** and **Message** fields allow an administrator to define the title of the message and the message body that is displayed to a client in the **My Messages** pane. The message can be defined as a priority by using the **Priority** field, which means that the message appears at the top of the messages listing.

Note: If multiple priority messages exist, the effective date of the message and the message type is used to dictate the order. For more information, see the **Ordering and Enabling/Disabling Messages** topic.

The **Effective Date and Time** field allows an administrator to define an effective date for the message, such as when the message is displayed in the Citizen Account. The **Expiry Date and Time** field allows an administrator to define an expiry date for the message, for instance, when the message no longer is to be displayed in the Citizen Account.

When the message is saved initially, it has a status of **In-Edit**. Before the message is displayed in the Citizen Account, it must be published. After it is published, the message is active and is displayed in the Citizen Account based on the effective and expiry dates defined.

Ordering and Enabling or Disabling Messages

Messages can be ordered by type such that messages of a particular type are displayed first. For example, all meeting messages can be configured to be displayed before messages related to a client's activities.

The Account Messages section allows an administrator to order messages by type. Ordering of messages within a particular type works with the priority assigned to messages and the effective date of messages. That is, if a certain type of message is configured to display first in the messages listing, priority messages within this type are displayed first. Then, the effective date is used to dictate the ordering of the remaining messages. If multiple priority messages exist within a particular type, the effective date is used to dictate the order of the messages.

Types of messages can be enabled and disabled. The **Account Messages** section of the *Cúram Universal Access (UA) Administration* application provides a list of all message types and allows an administrator to enable or disable messages of a particular type. An administrator also is able to define the icons to appear in the Citizen Account for each message type.

Enabling messages of a particular type means that all messages of that type are displayed in the Citizen Account. For example, all meeting messages (invitations, updates, and cancellations) are displayed in the Citizen Account. Disabling messages of a particular type means that all messages of that type are not displayed in the Citizen Account. For example, all meeting messages (invitations, updates, and cancellations) are not displayed in the Citizen Account.

Configuring The Duration Of Messages

System properties are provided which dictate the length of time a type of message is displayed in the Citizen Account. For example, a payment message can be configured to be displayed for 10 days in the Citizen Account. These configuration options apply only to messages that originate as a result of transactions on a *Cúram* system.

The following system properties are provided:

- `curam.citizenaccount.payment.message.expiry.days` - this property allows an administrator to dictate the number of days from the effective date that a payment message is displayed in the Citizen

Account. A payment message is displayed for this duration unless another payment message is created which replaces it. The default value is 10.

- `curam.citizenaccount.intake.application.acknowledgement.message.expiry.days` - this property allows an administrator to dictate the number of days from the effective date that an application acknowledgment message is displayed in the Citizen Account. An acknowledgment message is displayed for this duration unless another acknowledgment message is created which replaces it. The default value is 10.
- `curam.citizenaccount.meeting.message.effective.days` - this property allows an administrator to dictate the number of days from the effective date that a meeting message is displayed in the Citizen Account. A meeting message is displayed for this duration unless another meeting message is created which replaces it. The default value is 10.

Switching Off Messages

An agency might not want to display messages to a client in the Citizen Account. To cater for this choice, the system property `curam.citizenaccount.generate.messages` is provided which enables an agency to switch all messages *on* or *off*. The default value is *true*, which means that messages are generated and displayed in the Citizen Account.

Configuring Outreach

Outreach allows agencies to define targeted campaigns which can be displayed in the citizen account. The campaigns are based on information that the agency knows about the citizen. Outreach campaigns can be configured in the Outreach section of the UA Administration Application. Outreach campaigns use CER rule sets to dictate whether a campaign should be displayed or not and as such the appropriate rule set should be created before creating the campaign in the Rules and Evidence section of the Cúram Administration Application. A new outreach campaign can be defined in the New Outreach Campaign page.

A name must be specified which uniquely identifies the campaign in the administration application.

An expiry date can be defined which dictates when the campaign ends i.e. the campaign will no longer be displayed to a citizen when the expiry date is effective.

A CER rule set must be specified. This rule set will dictate whether the campaign is appropriate to a particular citizen.

An image can be displayed in the outreach panel on the citizen account home page. This image can be specified along with the image alt text which will be displayed in the tool tip displayed when a citizen hovers over the image. The images can also allow a citizen to link to a website. This link is defined in the CER rule set associated with the campaign.

For more information on creating outreach campaign rule sets, see the *IBM Cúram Universal Access Customization Guide*.

Configuring Last Logged In Information

The text displayed in the welcome message and last logged on information can be updated using the following properties stored in the CitizenAccountHome properties file stored in the Application Resource section of the Cúram Administration Application. The following properties are provided:

- `citizenaccount.welcome.caption` - allows the welcome message to be updated.
- `citizenaccount.lastloggedon.caption` - allows the last logged on text to be updated.
- `citizenaccount.lastloggedon.date.time.text` - allows the date time text to be updated.

Configuring Contact Information

Configure contact information for citizens (such as phone numbers and email addresses) and contact information for case workers (such as business phone number and business email address).

Contact Information displayed in the citizen account displays contact details (phone numbers, addresses and email addresses) stored for the logged in citizen in Cúram and also case worker contact details

(business phone number, mobile phone number, pager, fax and email) of the case owners of cases associated with the logged in citizen in a Cúram system and on remote systems.

This section outlines the configuration options available for each type of contact information.

Configuring Citizen Contact Information

The following system property is provided that dictates whether client contact information is displayed to a citizen.

Note: This property applies only to citizens registered on a Cúram system.

- `curam.citizenaccount.contactinformation.show.client.details`. If this property is set to `true`, the client's address, phone number, and email address information is displayed. If this property is set to `false`, the citizen contact information is not displayed. The default value for this property is `true`.

Configuring Worker Contact Information

The following system properties are provided to dictate agency worker contact information is displayed to a citizen and if displayed, additional system properties are provided to dictate the type of contact information displayed:

- `curam.citizenaccount.contactinformation.show.caseworker.details` - dictates if worker contact details are displayed in the citizen account. If this property is set to `true`, worker contact details of cases associated with the logged in citizen are displayed. If this property is set to `false`, worker contact information is not displayed. The default value for this property is `true`.
- `curam.citizenaccount.contactinformation.show.businessphone` - if worker contact details are displayed, this property dictates if the worker's business phone number is displayed. The default value of this property is `true`.
- `curam.citizenaccount.contactinformation.show.mobilephone` - if worker contact details are displayed, this property dictates if the worker's mobile number is displayed. The default value of this property is `true`.
- `curam.citizenaccount.contactinformation.show.emailaddress` - if worker contact details are displayed, this property dictates if the worker's email address is displayed. The default value of this property is `true`.
- `curam.citizenaccount.contactinformation.show.faxnumber` - if worker contact details are displayed, this property dictates if the worker's fax number is displayed. The default value of this property is `true`.
- `curam.citizenaccount.contactinformation.show.pagernumber` - if worker contact details are displayed, this property dictates if the worker's pager is displayed. The default value of this property is `true`.
- `curam.citizenaccount.contactinformation.show.casemember.cases` - dictates if worker contact information is displayed for cases where the client is a case member. If this property is set to `true`, cases where the client is a case member are displayed. If this property is set to `false`, then only cases where the client is the primary client are displayed. Note: this property only applies to cases originating from a Cúram system. The types of product deliveries and integrated cases to be displayed can be configured in the Product section of the Cúram Administration Application. For more information on administering this see the *IBM Cúram Integrated Case Management Configuration Guide*.

Configuring Appeal Requests

Complete the following steps to enable a citizen to request an appeal from their citizen account.

Procedure

1. Configure an Appeals entry in the Citizen Account navigation menu. Link the menu item to the shipped `CitizenAccount_listOnlineAppealRequest` page.
 - a) Configure an Appeals entry in the Citizen Account application banner menu.
For example, in the help menu.
2. Create a custom IEG script and data store schema to capture the appeal information.

- a) Implement `curam.citizenworkspace.pageplayer.impl.AppealDatastorePrepopulator` to populate the data store.
3. Set the values of the `curam.citizenworkspace.appeals.datastore.schema` and the `curam.citizenworkspace.appeals.datastore.script.id` properties to the values of the script and data store schema that you created.
4. Create an XSL template to generate a PDF of the appeal information.

Configuring the number of communications on the Notices tab

The Notices tab lists communications either sent to the citizen by the agency or sent to the agency by the citizen. By default, up to 20 communications are displayed in the Notices tab. You can configure the maximum number of communications that are displayed.

Before you begin

For solutions that use a custom Citizen Portal, such as Income Support and Health Care Reform, you must have enabled the Notices tab.

Procedure

Edit the `curam.citizenaccount.max.communication` system property and specify the maximum number of communications to display in the Notices tab.

Configuring Life Events

An overview of the configurations available for defining life events in UA administration. Different types of citizen life events enable submitting, maintaining, and sharing citizen information. For each life event, you must define how information is collected, stored, and displayed. You can configure life event information categories, mappings to dynamic evidence, and information sharing with internal and external sources.

Introduction

UA administration allows an agency to define different types of life events, for example, Having a Baby, Getting Married, and so on. Once defined, life events are displayed in the citizen account in UA allowing citizens to submit information to the agency as well as providing a citizen with useful information and resources relating to a particular life event. Life events can also be made available in other channels, for example, life events can be submitted online by an agency worker in the internal application. Configurations allow for different information to be displayed depending on where the life event is initiated from, for example, the Having a Baby life event question script displayed to a citizen can be different to the Having a Baby life event question script displayed to an agency worker.

There are four main aspects to configuring life events:

- configuring the name of the life event and a schema to store data submitted using a life event
- configuring channel specific information such as an IEG question script to collect data, a rule set to display recommendations, display information and mapping to dynamic evidence types
- configuring categories to group life events together
- associating remote systems where life event information can be sent to

Configuring a Life Event

The New Life Event page is used to configure a life event in UA Life Event Administration. This section provides more detail on the configuration options.

Defining a Name and Display Name

A name must be specified and uniquely identifies the life event. This name is only displayed in the administration application. A schema must be specified if the life event allows for a citizen or user to submit information to the agency. The schema defines where the information submitted by a citizen or

user in the life event script is stored. For more information on defining data store schema see the *IBM Cúram Working With Intelligent Evidence Gathering Guide*.

Defining a Question Script, Answer Script and Schema

An IEG script must be defined for the life event if the life event allows a citizen or user to submit information to the agency. The IEG script is used to collect the answers to a set of questions related to the life event. A script name can be specified in the Question Script field. A schema must also be specified if the life event allows for a citizen or user to submit information to the agency. The schema defines where the information submitted by a citizen or user in the life event script is stored. A schema can be specified in the Schema field. An answer script must be specified to allow a citizen to review the answers they have provided to the questions during submission of the life event. An answer script can be specified in the Answer Script field.

On saving the life event, empty template scripts and a schema will be created by the system based on the Question Script, Answer Script and Schema specified. It will then be possible to update these from the Life Event tab by selecting hyperlinks provided on the page. Clicking on the Question Script and Answer Script links will launch the IEG Editor which will allow the scripts to be edited. Clicking on the Schema link will launch the Datastore Editor which will allow the schema to be edited. Existing schema, question scripts and answer scripts can be utilized by selecting them on the Edit Life Event page.

Note: if a life event has been configured to send information to remote systems the Finish Page field in the script properties(accessed by selecting Edit->Configure Script Properties in the IEG Editor) should be set to `cw/DisplayRemoteSystems.jspx`.

For more information on defining IEG scripts and schema see the *IBM Cúram Working with Intelligent Evidence Gathering Guide*.

Defining a Channel Type

The channel type defines the channel in which a life event is used, for example, 'Online' or 'Internal'.

Defining a Display Name

The display name represents the name of the life event that will appear to a citizen or agency worker. For example, a change of job life event could be displayed as 'Lost My Job' to a citizen but 'Client Loses Job' to a case worker.

Defining a Common Life Event

UA displays a list of common life events in the citizen account. Common life events represent life events that may happen frequently in people's lives, for example, having a baby, changing jobs. Setting the common indicator means that the selected life event will appear in the common life events listing in the citizen account.

Defining Client Consent

For certain life events, a client's consent might be required before information is sent to a remote system or agency. The **Citizen Consent Enabled** field allows an administrator to dictate if a client's consent is required or not. This provision means that the client has the opportunity to select the agencies they would like to send their life event information to.

If this indicator is specified, a list of remote systems are displayed to a client on completion of the life event script. If this indicator is switched off, the client is not presented with a list of remote systems. If only one remote system is associated with the life event, the **Citizen Consent If One Choice Only** field is provided to dictate if the client is presented with the remote systems list. The client must specify their consent to send information to this remote system by selecting it on completion of the question script.

Defining a Launch Button Label

The launch button label used to launch the life event question script can be specified in the Launch Button Label field. If this is not specified, a default value of 'Launch Life Event' is used.

Defining A Display Description

A high level definition describing the life event can be defined. This can then be displayed in the channel in which the life event is being displayed. Rich text is supported.

Defining Additional Information

Additional information can be specified which relates to the life event. This can be used to display links to useful websites or information which the agency deems relevant to a particular life event.

Defining Submission Text

The text to be displayed to a citizen post submission of a life event can now be configured. If text is not configured, default text is displayed. The following default text is displayed if a rule set has not been defined for the life event: 'Your information has been submitted'. If a rule set has been defined, the following default text is displayed: 'Your information has been submitted. Based on the information you have given us, we have identified services and programs that may be of use to you. View your results'.

Mapping Life Event Information to Evidence Entities

Information gathered in the life event script is stored in the data store schema defined for the life event. In order to pass this information into the Cúram system it must be mapped to dynamic evidence entities. Dynamic Evidence entities must be firstly defined in the Rules and Evidence section of the administration application. Once defined, these entities must be specified as Social Record Evidence Types in the Universal Access section of the administration application. An indicator is also provided which will allow an administrator to dictate if a particular evidence type will be visible to a citizen in the portal(My Information in the citizen account) or not. Once the social record evidence entities are defined, the Data Mapping Editor(accessible from the Mappings tab on the life event) can be used to map the data from the data store to the appropriate evidence entities.

When a life event is submitted from the portal the information gathered is mapped (using the mappings defined above) to evidence entities associated with a new case type called a social record case. The Evidence Broker can then be used to pass the information from this case to the appropriate ongoing client cases.

For more information on Dynamic Evidence, see the *IBM Cúram Dynamic Evidence Configuration Guide*. For more information on data mapping, see the *IBM Cúram Data Mapping Editor Guide*. For more information on the brokering of evidence, see the *IBM Cúram Evidence Broker Guide*.

Categorizing Life Events

Life event administration allows for similar life events to be categorized or grouped together, for example, changing jobs, changing address and changing income life events could be categorized within an employment category. Categorizing life events makes it easier for a citizen or user to find the life event they need. Categories are defined in life event administration and can then be associated with a life event. When defining a category, a name and description must be specified. Life events can then be associated with the category. An icon can also be specified for a category which is then displayed in the citizen account beside the category name

Defining Remote Systems

Life event information can be submitted to remote/external systems. In order for life event information to be sent to a remote system, a remote system must be associated with a life event. The remote system must have the Life Event Service web service associated with it. This is used to transmit life event information to the remote system. Remote Systems can be configured in the Remote Systems section of the administration application.

Configuring Authentication Factors

An overview of the configurations available for defining authentication factors in UA administration that are used to apply second-level authentication to linked citizens when they log in to the UA. You can use

authentication factors to apply second-level authentication to linked citizens when they log in to UA. You can define multiple authentication factors that are required in addition to a user name and password, for example, SSN, date of birth. When you activate second-level authentication, a citizen must submit this additional information during login.

Configuring An Authentication Factor

The **Authentication Factors** page is used to define an authentication factor. Use the links included to understand how to configure an authentication factor.

The links to the topics that follow detail authentication factor configurations.

Configuring Help Text

Help text allows text to be defined for a client when the authentication factors are displayed to a client as they log in to Cúram Universal Access (UA). This text can be used to describe to a client what they need to do log in successfully.

Configuring A Title And Description

The title allows an administrator to define what is displayed to a client in Cúram Universal Access (UA) to identify the data that needs to be entered, for example, *Please enter your date of birth* or *Date of Birth*.

The description allows an administrator to specify a description of the authentication factor. This description is visible only in the UA administration application.

Configuring A Name

To authenticate the data that is submitted by a client, authentication handlers must be created. For example, to authenticate the date of birth that is submitted by a client, an authentication handler exists that contains a class to authenticate the data submitted by a client against the date of birth stored on the Cúram (or external) system.

The name that is defined for each factor identifies the handler that is used to authenticate the data. Three authentication handlers are provided in the initial product configuration - date of birth, Social Security number (SSN), and personal identification number (PIN). For more information on creating authentication handlers, see *Customizing Cúram Universal Access*.

Configuring Domain Name

The domain name is used to define the type of data entry field that is displayed to a client to capture information that is related to an authentication factor. For example, if a date of birth needs to be entered then the field that is displayed to a client must have a date selection icon beside it that allows the client to select a date.

Configuring A Display Order

The display order allows an administrator to dictate the order in which the authentication factors are displayed to a client relative to other authentications factors. For example, date of birth can be displayed first, followed by Social Security number (SSN).

Activating An Authentication Factor

An authentication factor can be activated/deactivated at any time. Activating an authentication factor means that the citizen will be requested to submit this information in order to successfully login.

Deactivating an authentication factor means that the authentication factor is not displayed to a citizen in the UA and hence the citizen does not have to provide this information at login.

Activating Second Level Authentication

Second level authentication can be activated/deactivated at any time. Activating second level authentication means that all active authentication factors will be displayed to a citizen and the citizen will have to submit appropriate data in order to successfully login. Deactivating second level

authentication means that no authentication factors will be displayed to a citizen. A citizen will only have to submit a user name and password in order to successfully login.

Configuring Languages

You can configure the available languages on the UA home page.

Configuring A New Language

The New Language page allows the user to define the languages that UA is available in. Once defined, the languages are displayed on the UA home page. Languages should only be defined for the languages for which the application has been localized.

The Display Name allows an administrator to define the information that will be displayed on the UA Home page, for example, English, Deutsch. The Language drop down allows a user to specify the language that the application is available in.

Configuring High Contrast

Configure high contrast to assist people with vision impairment using the UA portal.

High contrast is an accessibility feature that assists people with vision impairment. It allows for the size and color of fonts and the background to be changed to aid easy viewing of web pages. The ability for a citizen to switch to high contrast is available on the UA home page. This is configurable in the system administration application.

Enable High Contrast Selector

Use the Enable High Contrast Selector property to dictate whether a citizen can switch to high contrast or not on the UA home page.

The Enable High Contrast Selector dictates whether a citizen can choose to switch to high contrast or not. If this property is set to true, the option to turn on high contrast is displayed on the UA home page and can be enabled by the citizen before performing any of the processes available in the portal. If this property is set to false, the ability to turn on high contrast is not displayed on the home page and cannot be enabled by the citizen.

Configuring Remote Systems

An overview of the configuration options available for defining remote systems. Applications and life events data can be sent through web services for processing by a remote system. To enable remote processing, specify a remote system and the required web services. Remote systems can be configured allowing applications and life event data to be sent to them for processing via associated web services.

Configuring A Remote System

Remote systems are configured in the Universal Access section of the Administration application. A name and Root URL must be specified. The Root URL represents the root Uniform Resource Locator (URL) of the remote system. It consists of the Protocol (http or https), Host Name (for example, shell) and Port (for example, 9082). An example for root URL is http://shell:9082/. A Display Name can be configured which is used to display the name of the agency associated with the remote system. This is used in the citizen account when a list of remote systems are displayed to a citizen. It should be used to represent a more meaningful agency name to a citizen rather than using the remote system name. The Source User Name represents the user name that the remote system uses when invoking inbound Cúram web services.

Adding a Service to a Remote System

A target system can have multiple services associated with it. A URL must be defined for every service associated with a remote system. The URL is used for identifying and interacting with the service in the remote system. The URL is based on the combination of the root URL of the remote system, consisting of the system host name and port, and the extension URL for the associated service. For example, a URL `http://shell:9082/ProcessApplicationService` for a process application web service on a remote system is the combination of the root URL (`http:// shell:9082/`) of the remote system and the extension URL (`ProcessApplicationService`) of the associated process application service. The Invoking User Name and Password define the user name and password required to communicate with the web service on the remote system.

Configuring Motivations

A motivation allows a customer to define their own processes and make them available from the citizen account. A motivation consists of an IEG script, an associated data store schema, a data rule set and a display rule set. Running a motivation results in a configurable results page.

Configuring a New Motivation

The following options are available when configuring a motivation:

Name and Reference

A name and reference must be specified when creating a new motivation. The name uniquely identifies the motivation. The reference is used to reference the motivation in code.

Script and Schema

An IEG script must be defined for the application which is used to collect the answers to the application questions. A script name must be specified in the Script field. A data store schema must be specified to store the data entered in the script. A schema name must be specified in the Schema field. On saving the motivation, an empty template for both the script and schema will be created by the system based on the Script and Schema specified. It will then be possible to update these from the Motivation tab by selecting hyperlinks provided on the page. Clicking on the Question Script link will launch the IEG Editor which will allow the question script to be edited. Clicking on the Schema link will launch the Datastore Editor which will allow the schema to be edited.

Ruleset

To process the data in the data store and to produce a result, a CER rule set must be specified. Once specified on creation of the motivation, an empty rules template will be created by the system. It will then be possible to update the ruleset from the Motivation tab by selecting the hyperlink provided on the page. Clicking on the link will launch the CER Editor which will allow the ruleset to be edited. For more information on writing motivation rule sets, see the *IBM Cúram Universal Access Customization Guide*. For more information on using the CER Editor, see the *Working With CER Guide*. For more information on the IEG Editor, see the *Working With Curam Intelligent Evidence Gathering guide*.

Result Schema

The running of a motivation results in a set of results. These results (the result of a ruleset execution) are stored in a data store. The Result Schema allows for a schema to be specified for these results which allows for the extension of the information generated as a result of running rules.

Configuring the Motivation Results Page

When a motivation is run, a results page is displayed. This results page is displayed through a combination of display rules and application configuration. This section describes how to supply the configurable elements of the results page.

Configuring the Household Member Tooltip

As a result of running a motivation and performing actions from the results page, each household member will have a status per program e.g. 'Potentially Eligible'. This information is displayed for each household member in a tooltip on the results page. A sysadmin property (`curam.citizenworkspace.motivationresults.householdmemberstatuses.householdmembertooltip.exclude`) is provided which dictates the types of household member states to be **excluded** from the household member tool tip.

Configuring My Items

As a result of running a motivation and performing actions from the results page each household member will have a status per program e.g. Potentially Eligible. These states are stored in the `MotivResultMemberStatus` code table. A sysadmin property `curam.citizenworkspace.motivationresults.householdmemberstatuses.myitemstooltip.include` is provided which dictates the types of household member states which should be included for display in the My Items tooltip.

Configuring Result Actions

The actions to be displayed for a given result state can be configured, for example, if the status of a result is 'Ready To Enroll' an action of 'Enroll' can be configured for display to the citizen. For each action, a Name and Display State must be specified. A URI can be specified which defines the page to open when a citizen selects an action. Another motivation can also be initiated from a result action. The Type field defines whether another motivation is initiated or whether the page specified in the URI field should be initiated. If another motivation is to be initiated, the Motivation Reference must be specified.

Configuring Result Categories

The results of a motivation can be categorized together, for example, the set of programs that a citizen is eligible for can be categorized together and displayed on the results page. Help Text and a set of actions applicable to a category can be defined.

Configuring Icons

Person icons are displayed on a result to represent the state of the person. A base icon and a secondary icon can be configured. The base icons represent eligibility and ineligibility for a female adult, female child, male adult and a male child. A secondary icon can be used to place a secondary icon on top of the base icon. For example, a tick can be placed on top of the base icon to represent an enrolled person. Icons can be configured in the `MotivationResultsPersonStatusIconMapping` property file.

Configuring Additional Result Information

Additional information can be displayed on the results panel if required. An application resource property, `MotivationResultsProgramDetailsURLMapping`, is provided which contains a result reference and a link to a UIM fragment allowing customers to define the page to be displayed. If no link is defined, nothing is displayed.

UIM Support in IBM Cúram Universal Access

UIM Screens can be used in the Universal Access user interface. However, only a subset of UIM features supports the alternative look and feel in Universal Access.

Table 1 summarizes the supported UIM elements in Universal Access, and the contexts they are supported in.

Table 2. UIM in Universal Access

UIM Element Name	Parent UIM Element Name	Main Content Panel Context	Modal Dialog Context	DETAILS_ROW Context	Notes®
ACTION_SET	PAGE	Support for up to two items divided by a separator	Supported	Unsupported	The buttons are displayed in the Universal Access look and feel.
ACTION_SET	CLUSTER	Supported	Supported	Unsupported	The buttons are displayed in the Universal Access look and feel.
ACTION_SET	LIST	Unsupported	Unsupported	Unsupported	
ACTION_SET[@TYPE="LIST_ROW_MENU"]	LIST	Unsupported	Unsupported	Unsupported	The ACTION_CONTROL element can be used in the column of a LIST
ACTION_CONTROL	CONTAINER	Supported	Supported	Supported	
PAGE_TITLE	-	Supported	Supported	n/a	
DESCRIPTION	-	Supported	Supported	n/a	
CLUSTER	-	Supported	Supported	Supported	The collapsible behavior is not supported. Nesting of clusters is not supported.
LIST	-	Supported	Supported	Supported	The collapsible behavior is not supported. Paginated lists, scrollable lists and nested lists are not supported.
DETAILS_ROW	LIST	Supported	Supported	Supported	Nesting of expandable lists is not supported.

Table 2. UIM in Universal Access (continued)

UIM Element Name	Parent UIM Element Name	Main Content Panel Context	Modal Dialog Context	DETAILS_ROW Context	Notes®
FIELD	-	Support for the following input types only: <ul style="list-style-type: none"> • Text input • Text area • Code table drop-downs • Date picker • Password 	Support for the following input types only: <ul style="list-style-type: none"> • Text input • Text area • Code table drop-downs • Date picker • Password 	Unsupported	Code-table hierarchy and all other items not defined in the cell for Main Content Panel or Modal Dialog are not supported

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-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.

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

Microsoft, Windows, 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: