

IBM Cúram Social Program Management
Version 7.0.5

*Authoring Scripts using Intelligent
Evidence Gathering (IEG)*



Note

Before using this information and the product it supports, read the information in [“Notices” on page 92](#)

Edition

This edition applies to IBM® Cúram Social Program Management v7.0.5 and to all subsequent releases unless otherwise indicated in new editions.

Licensed Materials - Property of IBM.

© **Copyright International Business Machines Corporation 2012, 2018.**

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

© .

Contents

- List of Figures..... V**
- List of Tables..... vi**

- Chapter 1. Authoring Intelligent Evidence Gathering scripts..... 1**
 - Introduction..... 1
 - Purpose..... 1
 - Audience..... 1
 - Prerequisites..... 1
 - Defining IEG Script Structures..... 1
 - Overview..... 2
 - Organizing Script Pages into Sections..... 2
 - Subscripts..... 3
 - IEG Script Element Reference..... 4
 - Display elements..... 4
 - Meta-display elements..... 33
 - Flow-control elements..... 37
 - Structural, administrative and other elements..... 39
 - Operations Supported for IEG Expressions..... 43
 - Bracketing of Terms..... 43
 - Operator Precedence..... 43
 - Data Types and Supported Operations..... 44
 - Custom functions in expressions..... 45
 - IntakeProgramType and ScreeningProgramType in Expressions..... 46
 - Controlling the Flow of Your IEG Script..... 47
 - Natural Flow of an IEG Script..... 47
 - Controlling the Flow using Sections..... 47
 - Controlling the Flow using Conditions..... 48
 - Controlling the Flow using Loops..... 50
 - Control-Questions..... 53
 - Looping through People..... 53
 - Customizing Links on Summary Pages..... 55
 - Configuring IEG..... 57
 - Using the layout element to customize IEG pages..... 57
 - Using configuration properties to customize IEG pages..... 62
 - Configuration properties to customize IEG widgets..... 80
 - IEG Administration..... 80
 - Listing all Scripts..... 80
 - Downloading an Existing Script..... 80
 - Removing an Existing Script..... 81
 - Running a Script..... 81
 - Validating a Script..... 81
 - Uploading a New Script..... 81
 - Read-only Mode..... 82
 - Command Line Development Options..... 82
 - Migrating Superseded IEG Scripts..... 82
 - Migration process..... 82
 - Compliance for IEG..... 87
 - Customizing IEG Scripts..... 87
 - Public API..... 90
 - Identifying the API..... 90
 - Outside the API..... 90

Model Customization.....	90
Notices.....	92
Privacy Policy considerations.....	93
Trademarks.....	93

List of Figures

- 1. Script XML Overview.....3
- 2. Subscript element..... 3
- 3. Datastore Schema Required for Relationship Page..... 26
- 4. Datastore schema for relationship Attributes..... 27
- 5. Validation XML..... 39
- 6. set-attribute XML.....43
- 7. Intake and Screening Program Schema..... 46
- 8. Intake Program Expression.....46
- 9. Screening Program Expression..... 47
- 10. Condition Element.....48
- 11. Nested Condition.....49
- 12. Using 'isNotNull' Custom Function..... 49
- 13. For-each Loop..... 50
- 14. For-each Loop with Criteria..... 50
- 15. Simplified For Loop..... 50
- 16. For Loop with Entity and Criteria..... 51
- 17. While Loop..... 51
- 18. Nested Loop..... 52
- 19. Relationship Page XML.....54
- 20. Relationship Page XML with Caretaker Indicator..... 54
- 21. Relationship Page XML with Relationship Attributes..... 54
- 22. Relationship Summary List XML..... 55
- 23. XML for Editable Cluster..... 55
- 24. Cluster with No Layout..... 57
- 25. Layout with Label Width.....57
- 26. Layout with Compact-flow and 3 Columns..... 58
- 27. Layout with Width for Cluster..... 58
- 28. Layout with Input Alignment Set to Right..... 59
- 29. Cluster Container XML..... 60
- 30. List Container XML..... 60
- 31. List Container with Width XML..... 61
- 32. Superseded IEG Script Definition Before Migration..... 84
- 33. IEG Script Definition After Migration..... 85
- 34. Superseded IEG Script Definition..... 85
- 35. Superseded IEG Question Group Definition..... 86
- 36. IEG script definition..... 86

List of Tables

1. Add-link Attributes.....	4
2. Add-Link Child Elements.....	5
3. Alias attributes.....	5
4. Argument attributes.....	5
5. Cluster attributes.....	6
6. Cluster child elements.....	6
7. Column attributes.....	7
8. Column child elements.....	7
9. container attributes.....	8
10. Container child elements.....	8
11. custom-output element attributes.....	9
12. Description Attributes.....	10
13. Description Child Elements.....	10
14. display-text attributes.....	10
15. display-text child elements.....	10
16. Divider Attributes.....	11
17. Edit-link attributes.....	11
18. footer-field attributes.....	12
19. Help-Text attributes.....	13
20. Help-Text Child elements.....	13
21. Icon attributes.....	14
22. informational-message element attributes.....	15
23. informational-message child elements	15
24. Item-label child elements.....	16
25. label attributes.....	16
26. label child elements.....	16
27. Label-element attributes.....	17
28. Legislation attributes.....	17
29. List attributes.....	18
30. List child elements.....	18
31. list-question attributes.....	19
32. list-question child elements.....	20
33. Message attributes.....	21
34. Message Child Elements.....	21
35. next-button-label attributes.....	21
36. next-button-label child elements.....	22
37. Policy attributes.....	22
38. Question attributes.....	23
39. Question child elements.....	24

40. Question-page attributes.....	24
41. Question-page child elements.....	26
42. relationship-page attributes.....	27
43. relationship-page child elements.....	28
44. row-help attributes.....	28
45. row-help child elements.....	28
46. relationship-detail-header attributes.....	29
47. relationship-detail-header child elements.....	29
48. relationship-summary-list attributes.....	29
49. relationship-summary-list child elements.....	30
50. Subtitle attributes.....	30
51. Subtitle child elements.....	31
52. summary-page attributes.....	31
53. summary-page child elements.....	32
54. Title attributes.....	33
55. Title child elements.....	33
56. codetable-hierarchy-layout attributes.....	33
57. Layout child elements.....	35
58. Condition Attributes.....	37
59. Condition Child Elements.....	37
60. Loop Attributes.....	38
61. Loop Child Elements.....	38
62. Validation Attributes.....	39
63. Validation Child Elements.....	39
64. Callout Attributes.....	39
65. Identifier Attributes.....	40
66. ieg-script Attributes.....	40
67. Ieg-Script Child Elements.....	41
68. ieg-sub-script attributes.....	42
69. ieg-script Child Elements.....	42
70. Section Attributes.....	42
71. Section Child Elements.....	42
72. Set-attribute Attributes.....	43
73. Operator Precedence.....	44
74. Data Types and Supported Operations.....	44
75. Cluster Layout Options.....	58
76. Page Banner Configuration Properties.....	62
77. Progress Panel Configuration Properties.....	63
78. Person Tabs Panel Configuration Properties.....	64
79. Action Links Configuration Properties.....	65
80. Relationship Page Configuration Properties.....	65
81. Help Panel Configuration Properties.....	66
82. Page Title Panel Configuration.....	68

83. Navigation Panel Configuration.....	68
84. List Configuration.....	69
85. Other Page Layout Configuration Properties.....	70
86. Configuration Properties for Cluster Help Panel.....	73
87. Configuration Properties for Section Panel.....	74
88. Configuration Properties for Filed Level Help.....	74
89. Configuration Properties for Page Buttons.....	75
90. Configuration Properties for Relationship Page.....	75
91. Configuration Properties for Question Pages.....	75
92. Configuration Properties for Clusters.....	75
93. Other Configuration Properties.....	76
94. Sections Panel Configuration Properties (Universal).....	76
95. Sections Panel Configuration Properties (Horizontal).....	77
96. Configuration Properties for IEG Player in a Modal Dialog.....	78
97. Configuration Properties for IEG Player in a High Contrast Mode in the Universal Access interface.....	79
98. IEG Widget Configuration Properties.....	80

Chapter 1. Authoring Intelligent Evidence Gathering scripts

Use this information to develop Intelligent Evidence Gathering scripts. Intelligent Evidence Gathering scripts have a defined structure and a set of supported operations for IEG expressions. The presentation of pages in a script is configurable. Scripts can be migrated between systems.

Introduction

Purpose

The purpose of this guide is to provide instructions on how to use the features available in Intelligent Evidence Gathering (IEG). IEG is a technology provided as part of the IBM Cúram Application Suite which allows customers to create dynamic scripts for collecting data. Technically, this can be any data you like and can be used for whatever purpose you have in mind, but typically the data in question is required as part of an application for a program or to determine potential eligibility. All such information comes under the general heading of evidence. The dynamic nature of the scripts manifests itself in two ways:

- The IEG Engine interprets the scripts and creates the appropriate screens and screen flows at runtime, which means that new scripts can be created and existing ones modified through an administration interface as part of a live application.
- IEG scripts contain conditional logic (the intelligent bit) which allows you as the script writer to decide whether or not certain pages will be displayed and how many times, and also whether certain questions will be asked, based on answers which the user has already given as part of this script. The intelligent nature of the scripts means that users are only asked the minimum information required based on the answers they have given so far (for example, you may want to find out whether a claimant is pregnant or not, but if you already know that they are male, then there's no need to ask that question).

By combining these features, IEG allows you to create attractive, easy to use screens for capturing evidence in an easily configurable way.

Audience

This guide is targeted at script authors whose goal is to design scripts which capture information intelligently. Given its instructional style, the guide refers to you, the script author, directly.

Prerequisites

You, the reader, must have a basic understanding of XML. It is also helpful to have an understanding of:

- Using databases to store data, e.g., knowledge of database entities
- Customizing the look of applications using stylesheets, e.g., cascading stylesheets
- Adding simple functions to web-based applications, e.g., adding links

Defining IEG Script Structures

In its simplest form, an IEG script consists of pages which include questions to be posed to users of IEG. The structure of the IEG script is a logical grouping of these pages so that answers to the questions can be captured effectively. Sequences of pages can be grouped into logical sections. The purpose of these sections is to give users a higher level view of the kind of information captured by the IEG script.

Overview

In addition to including a variable number of pages, each section should contain one summary page. This page is used to give feedback to the user on the information entered on the pages in a section. Summary pages typically contain clusters and lists displaying read-only versions of the answers to questions asked. The summary page is always be the last page displayed within a section and is also displayed whenever a user clicks on the link for that section in the sidebar of the IEG Player.

To summarize, IEG scripts consist of a hierarchy of elements structured something like this:

- Script
 - Section
 - Page
 - Cluster
 - Question
 - Summary Page

IEG scripts are defined using XML files which match this hierarchical structure. The following section demonstrates how to organize script pages into sections in an XML file.

Organizing Script Pages into Sections

The easiest way to understand how to organize script pages into sections is through an example.

Suppose you were given the following list of required information that an IEG script needs to capture:

- Name and Contact Details
- Race and Ethnicity
- Household Members
- Household Relationships
- Income from Wages and Salaries
- Income from Tips and Commissions
- Utility Payments
- Travel Expenses
- Medical Expenses

To give the user a sense of what information will be requested of him or her, you could organize your pages into sections as follows:

- About You
 - Name and Contact Details
 - Race and Ethnicity
- Household
 - Household Members
 - Household Relationships
- Income
 - Income from Wages and Salaries
 - Income from Tips and Commissions
- Expenses
 - Utility Payments
 - Travel Expenses
 - Medical Expenses

The bigger the script (i.e., the more pages you have in it), the more important it is to group them into logical sections. The following is an example of how the XML should be structured for this IEG script:

```
<ieg-script>
  <section>
    <title id="AboutYouSection.Title">
      <![CDATA[About You]]>
    </title>
    <question-page id="AboutYouPage">
      <cluster>
        <question id="firstName">
          ...
        </question>
        ...
      </cluster>
    </question-page>
    <summary-page id="AboutYouSummary">
      ...
    </summary-page>
  </section>
  <section>
    <title id="Household.Title">
      <![CDATA[Household]]>
    </title>
    ...
  </section>
  ...
</ieg-script>
```

Figure 1: Script XML Overview

Each page within a script can contain a number of questions, which in turn are visibly grouped together into 'clusters' with the aim of making the screens more intuitive for a user. Each question consists of the text used to ask the question and an input control used to capture the answer. The type of input control used is determined by the data-type defined to store the answer, which will be discussed in further detail later. Each cluster on a page has various properties associated with it which allow you to control the layout and position of the questions within it.

Subscripts

The subscripts are standalone scripts that can be included in another script. A subscript can be included at script-level or in a section.

When a subscript is included in a section of another script then it should not contain any sections. Subscripts are reusable scripts they can be included in multiple scripts. Subscripts may themselves contain subscripts. A subscript can only be included in a particular script hierarchy once.

A subscript can be included in a script by using 'ieg-sub-script' element and specifying the subscript ID, version number and type.

```
<ieg-script>
  <section>
    <ieg-sub-script internal-id="2" start-progress="0"
      end-progress="20">
      <identifier id="SampleSubscript"
        scriptversionnumber="V1" type="Intake" />
    </ieg-sub-script>
  </section>
  ...
</ieg-script>
```

Figure 2: Subscript element

Behavior of progress bar in ieg subscripts

Progress in a sub script is calculated using the start-progress and end-progress attributes specified in the parent script.

The start-progress attribute should refer to the progress value of where the sub script begins in the parent script and the end-progress where the subscript ends.

The actual progress of each page in the subscript is calculated using the formula below:

$$\text{start-progress} + (\text{end-progress} - \text{start-progress}) * \text{page-progress} / 100$$

Where start-progress and end-progress refer to parent script and page-progress refers to the sub-script.

IEG Script Element Reference

This section provides a high level description of all IEG script elements described throughout this guide. These elements are presented in alphabetical order.

The following is provided for each element: a description of the element's attributes, and information on child elements, if relevant. Where it is helpful, an image which demonstrates how the element is used in IEG scripts is included.

Display elements

The following sections detail the elements that are used to create the layout, content and action options that are visible on an IEG page.

add-link

Add the add-link element to a list element when you want users to be able to add extra records through a link at the top of the list.

An icon can optionally be displayed alongside the link.

For more information about configuring the text on the link by using IEG properties, see [“Configuring IEG” on page 57](#).

Attributes

Name	Description
start-page	The page that the user goes to when they click the add link. This attribute is mandatory and would normally be the first page in the loop that is used to populate this list.
end-page	An optional page to end the process of adding a record to this list. Use an end page when there are either more than one page in the loop that is used to populate the list, or when pages after the loop need to be revisited after users add a record. For more information, see “Adding Records to Lists” on page 56 .
criteria	The criteria to use when retrieving the records from the entity to be displayed in the option list when adding an entity. If no criteria is present, then all records (for this instance of the root entity in the data store) from the entity are retrieved.
skip-to-summary	When set to true, the skip-to-summary attribute goes back to the summary page when Next is selected after using the add-link to add a new entity. If skip-to-summary is false, and values are added that are used in any expressions in the subsequent pages, the Engine displays those pages before returning to the summary page. If not specified, skip-to-summary is false by default.

Child Elements

An optional title element can be added as a child.

<i>Table 2: Add-Link Child Elements</i>		
Name	minOccurs	maxOccurs
Title	0	1

alias

Use question aliases to specify different question labels and help text depending on the viewer of the question script. For example, aliases might be 'Customer', 'Third Party', or 'Third Party Medical'.

Attributes

<i>Table 3: Alias attributes</i>	
Name	Description
type	The alias type indicates which category of script viewer can see the current alias.

Child elements

None.

argument

You can add the `argument` element to a message element when you want to substitute an attribute value in the message text.

For more information, see [“validation” on page 39](#).

Attributes

<i>Table 4: Argument attributes</i>	
Name	Description
id	The id of an argument refers to the entity name and attribute name that is used to retrieve the value for the attribute in the data store. The id should be in the form <code>entityName.attributeName</code> .

Child elements

None.

cluster

Use the `cluster` element to visually group questions and answers on an IEG page.

All questions in a cluster are displayed under the same heading and display according to the layout of the cluster. For more information about modifying the display of clusters, see [“Using the Layout Element to Change the Appearance of Clusters”](#) on page 57.

Attributes

Name	Description
entity	Use the <code>entity</code> attribute to specify which entity in the associated data store schema is related to the attributes in this cluster. To specify an entity for a cluster, you must also specify an entity for the page that contains the cluster. The IEG Engine assumes that the entity that is specified for the cluster is a direct child of the entity that is specified for the page. The IEG Engine assumes only one occurrence of the entity that is specified for the cluster (when combined with the criteria) in the entity that is specified for the page. All attributes that are referenced in this cluster are assumed to belong to the entity that is specified for the cluster. If no entity is specified for the cluster, all attributes that are referenced in the cluster are assumed to belong to the entity that is specified for the page instead.
criteria	Where there can be more than one instance of the entity that is specified for a cluster in the entity that is specified for the page, use the <code>criteria</code> attribute to identify which instance to use for this cluster.
read-only-expression	A Boolean expression that sets all content of the cluster to be read-only if its value is <code>true</code> at run time.
collapsed-expression	An expression that is evaluated to define whether the cluster is collapsed or expanded in its initial state. If the expression evaluates to <code>true</code> , the cluster is initially collapsed. If the expression evaluates to <code>false</code> or if the <code>collapsed-expression</code> attribute is not specified, the cluster is initially expanded.
grouping-id	This attribute is applicable to IBM Cúram Universal Access (New) only. Use this attribute to merge several clusters into one. For more information, see Merging clusters with the cluster element grouping-id attribute .

Child elements

Name	MinOccurs	MaxOccurs
title	0	1
description	0	1
help-text	0	1
edit-link	0	1
layout	0	1

Table 6: Cluster child elements (continued)

Name	MinOccurs	MaxOccurs
container	0	unbounded
question	0	unbounded
list-question	0	1
display-text	0	unbounded
skip-field	0	unbounded
informational-message	0	1
class-names IBM Cúram Universal Access (New) only.	0	1

Related concepts

[Merging clusters with the cluster element grouping-id attribute](#)

column

Use the `column` element to specify a title to display as the heading for an individual column in a list, and what information to display within it. The information usually comes from an attribute in the entity that is specified for the list. However it is also possible to display data from a related entity in the column, using a `column` entity and `link-entity`.

Attributes

Table 7: Column attributes

Name	Description
id	The id of a column refers to the attribute from which to read the values displayed within this column. This attribute must be a valid attribute of the entity that is specified for this list.
entity	The entity in the data store from which to read the elements to be displayed within the column.
link-entity	You can use <code>link-entity</code> instead of the <code>id</code> attribute to specify the name of the entity that stores a relationship between the entity specified for this column and another data store entity, For example, <code>InsuranceRelationship</code> in the IEG Sample script.
criteria	The criteria to use when retrieving the records from the entity.

Child elements

Table 8: Column child elements

Name	MinOccurs	MaxOccurs
title	0	1

<i>Table 8: Column child elements (continued)</i>		
Name	MinOccurs	MaxOccurs
layout	0	1

container

Use the `container` element to group questions in a cluster and also to group multiple columns in a list into a single column.

For more information, see [“Using the Container Element to Control Layout of Questions and Columns”](#) on page 59.

Attributes

<i>Table 9: container attributes</i>	
Name	Description
show-container-help	The <code>show-container-help</code> container attribute is set as a Boolean value. Setting <code>show-container-help</code> to <code>true</code> displays a help icon at the right side of the container. Selecting the help icon displays a dialog box with the help text for that container. The help text can be set for a container by using the <code>help-text</code> child element.

Child elements

<i>Table 10: Container child elements</i>		
Name	MinOccurs	MaxOccurs
question	0	unbounded
column	0	unbounded
layout	0	1
divider	0	unbounded
title	1	1
help-text	0	1

custom-output

Use the `custom-output` display element in IEG scripts to render custom HTML on a summary page, in both the IBM Cúram Social Program Management user interface, and in the Universal Access user interface. The `custom-output` display element enables entity data to be retrieved from a data store and accessed within a custom renderer so that the data can be rendered by using custom HTML.

Note: You can use `custom-output` elements only on summary pages.

For more information about using the `custom-output` element in scripts, see the related link.

Attributes

Table 11: custom-output element attributes

Name	Description
data-accessor	The data-accessor attribute represents the name of the data accessor class that is used to retrieve entity data that is rendered as required in the custom renderer class that is specified in the class-name attribute. The fully qualified name of the data accessor class must be specified in the attribute field. The data accessor class must implement the <code>curam.ieg.external.impl.IEGCustomOutputEntityData</code> interface and its <code>getRequiredEntitiesForCustomOutput(IEGScriptExecution)</code> method.
class-name	The class-name attribute represents the name of the custom renderer class that is used to output custom HTML on a summary page. The class-name attribute value must specify the fully qualified name of the custom renderer class. The custom renderer class must inherit from the <code>curam.ieg.player.IEGCustomOutputRenderer</code> class.

Child elements

None.

Related concepts

[Rendering Custom HTML on a Summary Page](#)

delete-link

Add the `delete-link` element to a list when you want users to be able to delete records from that list.

A delete link is provided with each record in the list. When clicked, a confirmation dialog asks the user to confirm whether they want to delete the record. If the user clicks **Yes**, this record and all its child entities are removed from the data store. For information about configuring the text for both the link and the confirmation dialog, see [“Deleting Records from Lists”](#) on page 56.

Attributes

None.

Child elements

None.

description

Use the `description` element to add a description to any titled element within a script.

The description element has an `id` attribute, which is used to reference a text property in the appropriate locale-specific properties file. For simplicity, script writers can add the text to use for the default locale directly into the script definition by adding a CDATA section as a child of the description element. The IEG Engine removes this text from the script for you when you import the script, and store it in the appropriate properties file instead, using the `id` of the description element as the key.

Attributes

Table 12: Description Attributes

Name	Description
id	An identifier for this description text, which is used as the key with which to reference the text within the associated properties file. The id must be unique within the context in which it is used. For example, if the description is added to a cluster, then the id must be unique only within the page in which the cluster is contained. Another page can contain an id with the same value.

Child elements

The description element can contain a CDATA section to store the text for the default locale.

Table 13: Description Child Elements

Name	MinOccurs	MaxOccurs
argument	0	unbounded

display-text

Use the `display-text` element in a cluster to display text with question elements.

The `display-text` element can also be used to add a non-repeating piece of text to a relationship page. For simplicity, script writers can add text for the default locale directly into the script definition by adding a CDATA section as a child of the `display-text` element. The IEG Engine removes this text from the script for you when you import the script, and store it in the appropriate properties file instead, using the id of the `display-text` element as the key.

Attributes

Table 14: display-text attributes

Name	Description
id	An identifier for this display text, which is used as the key with which to reference the text within the associated properties file. The id must be unique within the page in which it is used.

Child elements

Table 15: display-text child elements

Name	MinOccurs	MaxOccurs
argument	0	unbounded

divider

Use the `divider` element in a container to separate question elements.

This can be placed before, after or between questions or columns. For simplicity, script writers can add the text to use for the default locale directly into the script definition by adding a CDATA section as a child

of the divider element. The IEG Engine removes this text from the script for you when you import the script, and store it in the appropriate properties file instead, using the id of the divider element as the key.

Attributes

<i>Table 16: Divider Attributes</i>	
Name	Description
id	An identifier for this divider, which will be used as the key with which to reference the text within the associated properties file. The id must be unique within the page in which it is used.

edit-link

Add the `edit-link` element to a list when you want users to be able to edit records in the list.

A link is provided with each record in the list which, when clicked, will take the user to the page containing the full details of the record so that they can make the required changes. The edit link can be configured to only show specific clusters on the linked page. An icon can optionally be displayed alongside the link. This icon configured through the IEG properties. Depending on what the user does on that page (and whether an end-page is specified) the user will either be brought straight back to the summary page containing the list, or they will be forced to continue through the remaining pages in the section so as to verify that their previous answers are still valid. For more information, see [“Controlling the Flow of Your IEG Script”](#) on page 47.

Attributes

<i>Table 17: Edit-link attributes</i>	
Name	Description
start-page	The page that the user goes to when they click the edit link. This attribute is mandatory and would normally be the first page in the loop that is used to populate this list.
end-page	If more than one page was used to create this record, then the end page can be set to indicate the set of pages which the user must visit to edit a record within the list.
skip-to-summary	When set to <code>true</code> , the <code>skip-to-summary</code> attribute goes back to the summary page when Next is selected after using the add-link to add a new entity. If <code>skip-to-summary</code> is <code>false</code> , and values that have been modified are used in any expressions on subsequent pages, the Engine will display those pages before going back to the summary page. If not specified, <code>skip-to-summary</code> is <code>false</code> by default.
show-page-elements	A list of comma separated cluster identifiers that should be displayed on the specified start-page. If this optional attribute is specified only the listed clusters will be displayed. If this attribute is not specified all of the clusters on the start-page will be displayed. Conditional clusters included in the show-page-element list may not be displayed depending on the expression used to control the condition.

Child elements

None.

footer-field

Use the `footer-field` element to display calculated values within the `footer-row` element of a list. The value displayed in a footer field is provided using an expression.

Attributes

Name	Description
id	The id is used to uniquely identify a footer field in cases where there are multiple footer fields.
expression	This attribute is used to specify an expression that will be evaluated at runtime to determine the value that will be presented to the user. The constraints on other expressions hold true for these expressions and any attributes referred to in the expressions should have a real value by the time the expression is evaluated.
type	The data type of the value. This is used by the IEG Player to display the value correctly. The value used for this attribute must be a valid domain definition.

Child elements

None.

footer-row

Add the `footer-row` element to lists to display total or summary information. A footer row is displayed as an extra row at the bottom of a list. Footer rows can be used to display text or values that are provided by expressions.

Attributes

None.

Child elements

None.

help-text

The `help-text` element can be used to specify the help text for a number of IEG Script elements.

- Cluster
- Question
- Container
- Row-level help for lists

The `help-text` element has an `id`, which is used to reference a text property in the appropriate locale-specific properties file. For simplicity, script writers can add the text to use for the default locale directly into the script definition by adding a `CDATA` section as a child of the `help-text` element. The IEG Engine removes this text from the script for you when you import the script, and stores it in the appropriate properties file instead, using the `id` of the `help-text` element as the key.

There are three mechanisms for displaying help in a cluster, at the cluster-level itself, for questions that are contained within a cluster, and for containers in contained within a cluster.

Cluster-level help

When an IEG script is run, the help text for each question and container within a cluster is combined with the overall help text for the cluster itself to create a help panel within the cluster. This panel is initially hidden but can be revealed by clicking a link at the right side of the cluster's title. The panel can be hidden by either clicking the same link again or by using a close link within the panel itself.

Question-level help

Help can optionally be displayed for each question. When help text has been specified for a question, a help icon is displayed alongside that question. Selecting this icon will display the help text for that question.

Container-level help

Help text can also be contained within a container. Similar to when you add help to a question, this will result in a help icon being displayed at the right side of the container. In order for the help icon to appear at the right side of the container 'show-container-help' must be set to true on the container element. Selecting the help icon displays the help text for that container.

Within a container it is possible to display the help text at either the container or question-level. Setting the help text for questions within the container will result in the combined help text of all questions within the container displaying in the help dialog. You can also set the help at the container-level which will result in a single entry for the container in the container help dialog. It is possible to use a combination of field-level help and container-level help within a container. All help text added to the container will appear at the cluster-level help if the IEG configuration property `compile.cluster.help=true` is set to true.

Row-level help for lists

Row-level help for lists is displayed in a similar fashion to question-level help. A help icon is displayed on each row. Selecting the icon displays the relevant help for that row.

Attributes

<i>Table 19: Help-Text attributes</i>	
Name	Description
id	An identifier for the help text, which is used as the key with which to reference the text within the associated properties file. The id must be unique within the page in which it is used.

The help-text element can contain a CDATA section to store the text for the default locale.

<i>Table 20: Help-Text Child elements</i>		
Name	MinOccurs	MaxOccurs
argument	0	unbounded
alias	0	unbounded

icon

Use the `icon` element to add images to either the title area of a page or the sections panel.

When added to a section, the icon element must contain three attributes to specify the images to use for when the section link is enabled, disabled, or the current section. When used in a page, the icon element should only contain the image attribute as the image for the page title can only have one state.

Attributes

Name	Description
image	a reference to the image to use for the title area of a page. If not specified, then a default image will be used instead.
enabled-image	a reference to the image to use for a section when it is enabled. If not specified, then a default image will be used instead.
disabled-image	a reference to the image to use for a section when it is disabled. If not specified, then a default image will be used instead.
current-image	a reference to the image to use for a section when it is the current highlighted section. If not specified, then a default image will be used instead.
hover-image	a reference to the image to use for a section when the mouse hovers over the section. If not specified, then a default image will be used instead.

Child elements

None.

informational-message

Use an `informational-message` display element to display an informational message within the heading of either a cluster, a list, or a relationship summary list.

The purpose of an informational message is to describe, at a glance, the contents of either the cluster, the list, or the relationship summary list where it occurs. An informational message consists of text and an image. For example, if a cluster, a list, or a relationship summary list is contained in a section that can be expanded and collapsed, an informational message can display information about the content when the content is in a collapsed state. The cluster, the list, or the relationship summary list must have a title for an informational message to be displayed.

The following list describes the child element and the attributes that the `informational-message` display element supports:

message child element

The `informational-message` display element supports an optional message child element. A script designer can use the message child element to define custom text to display on a particular informational message. For simplicity, script writers can add the text to use for the default locale directly into the script definition by adding a CDATA section as a child of the message element. When you import the script, the Intelligent Evidence Gathering (IEG) Engine removes the text from the script. Instead, the IEG engine stores the text in the appropriate properties file by using the ID of the message element as the key.

Custom text that is too long to fit within the heading of the cluster, list, or relationship summary list is truncated and represented by three ellipses. You can display more truncated text by hovering your cursor over the informational message. If no custom text is defined through a message child element for a particular informational message, the default text `Information Message` is displayed. You can configure the value of the default text through the `informational.message.text` property.

image attribute

The `informational-message` display element supports an optional `image` attribute that script designers can use to define a custom image to display on a particular informational message. The value of the `image` attribute is used to reference an image in the resource store that is displayed on the informational message. If no `image` attribute is defined on a particular `informational-message` display element, a default image is used. You can configure the value of the default image

through either the `informational.message.external.image` property, or through the `informational.message.internal.image` property.

expression attribute

The `informational-message` display element supports an optional `expression` attribute. If the value of an `expression` attribute on a particular `informational-message` element is evaluated as `true`, the informational message is displayed. If the value of an `expression` attribute on a particular `informational-message` element is evaluated as `false`, the informational message is not displayed. If an `expression` attribute is not defined on a particular `informational-message` element, an informational-message is always displayed.

For information about configuring the default text and image properties, see the related link.

Script example

The following sample shows an example implementation of the `informational-message` display element in a `cluster` element. Instead of a `cluster` element, you can also insert an `informational-message` display element in either a `list` element, or a `relationship-summary-list` element. Substitute the tags accordingly in the example.

```
<cluster>
<title id="Example.Title"><![CDATA[Title]]></title>
<informational-message image="SomeCustomImage.png"
  expression="Entity.attribute=="&quot;SomeValue&quot;">
  <message id="SomeID">
    <![CDATA[Some custom text.]]>
  </message>
</informational-message>
</cluster>
```

Attributes

Table 22: informational-message element attributes	
Name	Description
image	A reference to an image to display on the informational message. If a value is not specified, a default image is displayed instead.
expression	A Boolean expression that is used to determine whether to display the informational message. If a value is not specified, the informational message is always displayed.

Child elements

Table 23: informational-message child elements		
Name	MinOccurs	MaxOccurs
message	0	1

Related concepts

[Other Page Layout Configurations](#)

item-label

Use the `item-label` element in a list question to define the text to display for each element in the list question. The text can be built up from one or more attributes of the entity that represents each element in the list question by having one or more label elements.

Attributes

None.

Child elements

<i>Table 24: Item-label child elements</i>		
Name	MinOccurs	MaxOccurs
label-element	1	unbounded

label

Use the `label` element to define the label text for any question or answer in a script.

Each label element has an `id`, which is used to reference a text property in the appropriate locale-specific properties file. For simplicity, script writers can add the text to use for the default locale directly into the script definition by adding a CDATA section as a child of the label element. The IEG Engine removes this text from the script for you when you import the script, and store it in the appropriate properties file instead, using the `id` of the label element as the key.

Attributes

<i>Table 25: label attributes</i>	
Name	Description
<code>id</code>	An identifier for this label text, which will be used as the key with which to reference the text within the associated properties file. The <code>id</code> must be unique within the page in which it is used.

Child elements

The label element can contain a CDATA section to store the text for the default locale.

<i>Table 26: label child elements</i>		
Name	MinOccurs	MaxOccurs
<code>argument</code>	0	unbounded
<code>alias</code>	0	unbounded

label-element

Use the `label-element` element in the `item-label` element of a list question to represent part of the text to be used as the label for each element within the list.

For example, if the elements within the list are people and you just want to show their first name, then there would just be one label element, but if you wanted to show their first and last names, there would be two label elements.

Attributes

<i>Table 27: Label-element attributes</i>	
Name	Description
attribute-id	the name of the attribute within the entity specified for list question from which to read the value for this label element.

Child elements

None.

legislation

You can use legislation links at page and question level to point the user to legislative information that relates to the questions.

It is possible, but not mandatory, to specify one legislation link for each page and one legislation link for each question. A link specified at page level will appear in the page title banner at runtime, in the top-right corner. Clicking on the link will open the target in a new window. A link specified at question level will appear in the Help section for the parent page, and as such will only be visible after the user expands this section. Like the page level links, clicking will open the target in a new window. Links should be inputted as full addresses, including protocol (e.g. <http://google.com>).

Attributes

<i>Table 28: Legislation attributes</i>	
Name	Description
id	An identifier for this legislation link, which is used as the key with which to reference the link within the associated properties file. The id must be unique within the page in which it is used.

Child elements

None.

list

Use the `list` element to display the details of multiple records from an entity in the data store. A separate column should be added to the list for each attribute of the entity you wish to display. A standard list contains records from a single entity.

A list can optionally include Footer Rows. These are typically used to display total information for lists. Help can be added to the rows on a list to provide context for the listed data.

You can also nest one list inside another if what you are trying to do is to show a list of records grouped together by their parent record. An example of this would be the scenario described earlier in this document where your script allows the user to enter multiple incomes per person in their household.

To achieve this, you need an outer list for the parent entity (Person in this case), which contains a single column (the First Name in this case) and then another list which contains the details of the child records (the incomes in this case).

Lists can be added as children of question pages and summary pages. When a list is added as a child of a question page, it cannot contain Add, Edit or Delete links.

Attributes

<i>Table 29: List attributes</i>	
Name	Description
entity	The entity in the Datastore from which the records displayed in this list will be retrieved.
criteria	The criteria to use when retrieving the records from the entity. If a criteria was specified for the loop used to enter these records, then the same criteria should be used on the list which displays them to ensure that the system functions properly. If no criteria is present, then all records (for this instance of the root entity in the Datastore) from the entity will be retrieved.
show-icons	If the show-icons attribute is set to true, then person icons will be displayed in the first column of the list.
collapsed-expression	The expression that must be evaluated to define whether the list is collapsed or expanded in its initial state. If the expression evaluates to <code>true</code> , the list is initially collapsed. If the expression evaluates to <code>false</code> or if the <code>collapsed-expression</code> attribute is not specified, the list is initially expanded.

Child Elements

<i>Table 30: List child elements</i>		
Name	MinOccurs	MaxOccurs
title	0	1
description	0	1
edit-link	0	1
delete-link	0	1
add-link	0	1
column	0	unbounded
list	0	1
container	0	unbounded
row-help	0	1
footer-row	0	1
informational-message	0	1

list-question

Use the `list-question` element to display a panel that contains a list of elements, each with a label, image, and a check box to allow the user to select an item. Typically, the items displayed within the list are people. However, other kinds of entity can also be used, for example, employers.

To create a list question, specify the label for the overall question, the entity from which to read the items in the list, the attributes to use as the label for each item, and the attribute on the entity to set based on whether the check box is checked or not for each item. Example XML for a list question is shown.

```
<list-question entity="Person" id="isBlind">
  <label id="IsBlind.Label">
    <![CDATA[Please choose the people who are blind:]]>
  </label>
  <item-label>
    <label-element attribute-id="firstName"/>
  </item-label>
</list-question>
```

List questions can also be used on summary pages to display the choices the user has made. In this case, only those items selected will be displayed.

Cluster flow/compact-flow layout is only supported for list questions that use a display of type 'drop down'. When using a flow layout on the cluster (question label to the left, drop down to the right), it is possible to set the label-width to a value greater than 0. The default behavior with no label-width set, will default to 50% for the label and 50% for the drop down. For all other types of list questions, they must be placed on a cluster with label-width set to 0 in order to ensure that the full width of the page is available

The list question element can also be used to display a question matrix, which is a list question with a code table data type. In this case, there will be a table displayed with a column for each entity, containing radio buttons to indicate the multiple-choice options defined in the code table.

Validating list questions

The `IEG2Context` object is available in custom functions and contains the following methods for validating:

- `List<Long> getListQuestionSelectedEntityIDs(final String idOrLinkEntity, final String entity).`
- `List<Map<Long, String> getListQuestionValues(final String idOrLinkEntity, final String entity).`

Given a list-question identifier (the `id` and `entity` attributes as defined on the list-question in the script definition, or the `link-entity` and `entity` in case of list-question relationships), the first method will return a list of entity IDs that were selected, whereas the second one will return a list of all the entity IDs that were displayed associated with the selected value for each ID. This value will be "true" or "false" for list questions that are "selectable" (for Boolean list-questions or list-question relationships), and a code table value for code table list-questions.

It should be noted that the methods can return null if the list-questions are not present on the page, and they can also return empty lists if nothing was selected or displayed. The script designers should not assume that a value is always present even if the mandatory flag has been set, as the mandatory validation takes place at the same time as this validation. Therefore if a value is expected the validation should not fail but succeed and let the mandatory validation be displayed, otherwise two validation messages will appear.

Attributes

<i>Table 31: list-question attributes</i>	
Name	Description
entity	The entity in the Datastore from which to read the elements to be displayed within the list.

Table 31: list-question attributes (continued)

Name	Description
criteria	The criteria to use when retrieving the records from the entity.
id	The name of the attribute on the entity in which to store the answers (i.e. true for each item in the list whose check-box is checked and false for the others).
mandatory	If the mandatory attribute is set to true, at least one selection must be made for the list-question.
link-entity	A list-question can be used to create relationships between entities instead of setting an attribute on an entity. For example, if you were entering an insurance record and wanted to record which people in the household were covered by it, you could use a list question to display the people in the household and create the relationships from the insurance entity to the person entity. In this case, the link-entity should be used instead of the id attribute to specify the name of the entity which stores the relationship between the other two entities (e.g. InsuranceRelationship). For more informatin, see Creating Datastore Schemas .
single-select	A Boolean indicator to specify that the list-question is single-select, whereby only one option in the list can be selected. The default is false, i.e. the list-question can have multiple answers.
display	The format in which to display the list question. By default, list questions are displayed in 'horizontal' format, with the options appearing across the screen in a row. This attribute can also have the values 'vertical' (to display the options in a vertical column) or 'dropdown' (to display the options in a dropdown box).
input-alignment	Indicates if the input element should appear to the left or right of the associated image.
read-only-expression	A Boolean expression which, if evaluated to true, causes the list-question to become read-only on the page.

Child elements

Table 32: list-question child elements

Name	MinOccurs	MaxOccurs
label	1	1
item-label	1	unbounded
help-text	0	1

message

Use the message element in validations for the the message to be displayed to the user if the validation fails, or within an informational-message element to describe the contents of a cluster, a list, or a relationship summary list..

Each message element has an id, which is used to reference a text property in the appropriate locale-specific properties file. For simplicity, script writers can add the text to use for the default locale directly into the script definition by adding a CDATA section as a child of the message element. The IEG Engine will remove this text from the script for you when you import the script, and store it in the appropriate properties file instead, using the id of the message element as the key.

Attributes

<i>Table 33: Message attributes</i>	
Name	Description
id	An identifier for this message text, which will be used as the key with which to reference the text within the associated properties file. The id must be unique within the page in which it is used.

Child elements

The message element has no child elements but can contain a CDATA section to store the text for the default locale.

<i>Table 34: Message Child Elements</i>		
Name	MinOccurs	MaxOccurs
argument	0	unbounded

next-button-label

Use the `next-button-label` element to set a label for the next button on the current page. This element is applicable to IBM Cúram Universal Access (New) only.

The `next-button-label` element is a child of the `question-page`, `relationship-page` `summary-page` elements. For example:

```
<summary-page id="SummaryPage" show-back-button="true" show-exit-button="false" show-save-exit-button="true" entity="Person" criteria="isPrimaryParticipant==true">
<next-button-label id="NextButton.Label">Submit</next-button-label>
...
</summary-page>
```

Attributes

<i>Table 35: next-button-label attributes</i>	
Name	Description
id	An identifier for this text, which is used as the key with which to reference the text within the associated properties file. The id must be unique within the page in which it is used.

Child elements

Table 36: next-button-label child elements

Name	MinOccurs	MaxOccurs
argument	0	unbounded

policy

You can specify policy links at page and question level to point the user to policy information that relates to the questions.

It is possible, but not mandatory, to specify one policy link for each page and one policy link for each question. A link specified at page level will appear in the page title banner at runtime, in the top-right corner. Clicking on the link will open the target in a new window. A link specified at question level will appear in the Help section for the parent page, and as such will only be visible after the user expands this section. Like the page level links, clicking will open the target in a new window. Links should be inputted as full addresses, including protocol (e.g. <http://google.com>).

Attributes

Table 37: Policy attributes

Name	Description
id	An identifier for this policy link, which is used as the key with which to reference the link within the associated properties file. The id must be unique within the page in which it is used.

Child elements

None.

question

Use the `question` element to ask a question on a page and capture the answer. It consists of an optional label, the question text, and an input control for the user to enter or select their answers.

If a script has duplicate questions, put each question in mutually exclusive conditional clusters, to ensure that only one can be displayed at a time. The use of duplicate questions is only supported in static conditions.

Attributes

Table 38: Question attributes

Name	Description
id	<p>The id of a question refers to the name of the attribute used to store the answer to the question within the Datastore. The entity to which this attribute belongs is taken from the cluster which contains this question (if one has been specified) or the page containing the cluster.</p> <p>For example, if the question is contained within a cluster on a page which has the entity Address specified for the cluster and Person specified for the page, and the id of the question is 'firstName', then the answer to this question will be stored in Address.firstName. If the cluster had no entity specified, the answer to the question will be stored in Person.firstName.</p> <p>When a page is loaded, if an answer exists in the Datastore for the entity/attribute specified, that value will be displayed as the initial contents of the input field for the question, otherwise the field will be empty, or the configurable text 'please select' will appear for combo boxes.</p>
mandatory	<p>Specifies whether or not an answer to this question is required or not. If set to true, then an asterisk is placed beside the question to indicate that it is mandatory and validations are performed by the Engine when the user clicks the Next button to ensure that an answer has been provided for this question.</p>
control-question	<p>This attribute can be used to indicate that the answer to this question is used purely to control the flow of the script and is therefore not to be found in the Datastore. Instead, the IEG Engine will maintain its own copy of this value. The id of a control question must be unique within the script.</p>
control-question-type	<p>If the control-question attribute is set to true, then you must also set this attribute so that the IEG Player knows what type of input control to use and the Engine knows how to handle it when used in expressions. The value used for this attribute must be a valid domain definition.</p>
multi-select	<p>This attribute can be used to indicate if the question represents a multi select list question or not. If the specified question data type is a CODETABLE_CODE, then this question is displayed to a user as a drop-down list containing the possible answers (i.e. the descriptions of the codetable entries). Setting this attribute to true (the default is false) ensures that a user may select none/one/all of the possible answers for the specified question. It is the codetable code corresponding to the description of the answer(s) selected that is stored in the Datastore in this instance.</p>
default-value-expression	<p>This attribute can be used to specify an expression that will be evaluated at runtime to determine an initial answer value that will be presented to the user. The user can then choose to accept the initial value or overwrite it with some other value. The default value expression for a question is only evaluated once per script execution and is evaluated just before the page on which the question is declared is displayed. Default value expressions can be complex expressions and can reference answers already supplied during script execution. The constraints on other expressions hold true for default value expressions and any attributes referred to in the expressions should have a real value by the time the expression is evaluated. Default value expressions can also be defined for control questions.</p>
read-only-expression	<p>A Boolean expression which, if evaluated to true at runtime, causes the question to become read-only.</p>

Table 38: Question attributes (continued)

Name	Description
show-field-help	A Boolean attribute which, if true, results in a help icon being displayed alongside the question input field. In the case of list questions and codetable-type questions without dropdown inputs, the help icon is displayed alongside the question label. Clicking the help icon opens a modal dialog that displays the help text for the question.

Child elements

Table 39: Question child elements

Name	MinOccurs	MaxOccurs
label	0	1
help-text	0	1
layout	0	1
legislation	0	1
policy	0	1
codetable-hierarchy-layout	0	1

question-page

Use the `question-page` element to represent a standard page for capturing the answers to questions in a script.

Attributes

Table 40: Question-page attributes

Name	Description
id	a unique identifier for this page. This id can be used to reference this page when linking to it from another part of the script, or when indicating what page to start at when re-entering a script.
entity	<p>the name of an entity from the associated Datastore schema. If used directly within a section, the entity referenced on a page must be a child of the root element within the schema. Any attribute within a cluster on this page which does not have its own entity specified is assumed to be an attribute of this entity.</p> <p>When a page is used within a loop, there is no need to set the entity on the page if it is the same as the entity used for the loop itself</p>

Table 40: Question-page attributes (continued)

Name	Description
criteria	<p>used in combination with the entity attribute to identify the exact record(s) to display/modify on this page. The criteria attribute can be thought of in the same way as the 'where' clause in an SQL statement. If the question-page is not contained within a loop, the IEG Engine will assume that there is only one record which matches the criteria and therefore only select the first record returned, so to ensure that everything works as intended, the criteria should only match a single record in this situation. If no record matches the specified criteria, then no values are displayed in the input fields on this page, and when the data entered by the user is saved, a new record will be created. If no criteria is specified, then there it is assumed that only one instance of the entity exists per root element in the Datastore.</p> <p>If the page is contained within a loop, no criteria should be specified as the criteria for the loop is what will be used.</p>
progress	the percentage to use for the progress bar when the user gets to this page.
show-person-tabs	when set to true, this attribute will indicate to the IEG Engine and Player that person tabs should be displayed at the top of the page, more information on which can be found in the Flow Control chapter. This should only be used on pages within a loop.
read-only	when set to true, this attribute will indicate to the IEG Player that no input controls should be available to the user on this page and instead the values presented should be read-only, as on summary pages.
show-back-button	when set to true, this attribute will indicate to the IEG Player that the Back button should be displayed. If not specified, this attribute will be set to true by default.
show-exit-button	when set to true, this attribute will indicate to the IEG Player that the Exit button should be displayed. If not specified, this attribute will be set to false by default.
show-next-button	when set to true, this attribute will indicate to the IEG Player that the Next button should be displayed. If not specified, this attribute will be set to true by default.
show-save-exit-button	when set to true, this attribute will indicate to the IEG Player that the 'Save & Exit' button should be displayed. If not specified, this attribute will be set to true by default.
read-only-expression	a boolean expression which, if evaluated to true at runtime, causes the question page to become read-only.
set-focus	when set to false, this attribute will indicate to the IEG Player that no initial focus should be set on the form. If not specified, this attribute will be set to true by default.

Child elements

Table 41: Question-page child elements

Name	MinOccurs	MaxOccurs
title	1	1
subtitle	0	1
description	0	1
icon	0	1
help-text	0	1
legislation	0	1
policy	0	1
cluster	0	unbounded
list	0	unbounded
condition	0	unbounded
validation	0	unbounded
set-attribute	0	unbounded
next-button-label	0	1

relationship-page

Use the relationship-page element only to capture relationships between people in a household.

You must have the following entity structure in your data store schema for this element to work correctly.

```

<xsd:element name="Person">
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element ref="Relationship"
        minOccurs="0"
        maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="personID" type="D:SVR_KEY" />
    ...
  </xsd:complexType>
  <xsd:key name="PersonKey">
    <xsd:selector xpath="./Person" />
    <xsd:field xpath="@personID" />
  </xsd:key>
  <xsd:keyref name="RelationshipRef" refer="PersonKey">
    <xsd:selector xpath="./Person/Relationship" />
    <xsd:field xpath="@personID" />
  </xsd:keyref>
</xsd:element>

<xsd:element name="Relationship">
  <xsd:complexType>
    <xsd:attribute name="relationshipType" type="IEG_STRING" />
    <xsd:attribute name="personID" type="D:SVR_KEY" />
  </xsd:complexType>
</xsd:element>

```

Figure 3: Datastore Schema Required for Relationship Page

Please note that the Relationship entity may have other attributes defined in order to capture more information in regarding the relationship itself. For example you may wish to avail of the option of using an indicator to record whether a relationship is a non-parent caretaker relationship or you may wish to add other custom relationship attributes. For example:

```

<xsd:element name="Relationship">
  <xsd:complexType>
    <xsd:attribute name="relationshipType" type="IEG_STRING" />
    <xsd:attribute name="personID" type="D:SVR_KEY" />
    <xsd:attribute name="isNonParentPrimaryCaretaker"
      type="IEG_BOOLEAN" />
    <xsd:attribute name="startDate" type="IEG_DATE" />
  </xsd:complexType>
</xsd:element>

```

Figure 4: Datastore schema for relationship Attributes

You need to have captured the people in your household before you visit the relationship pages, otherwise there will be no relationships to enter. Assuming there are people in the household, a page will be displayed for all members of the household bar the last one. More information about how to create relationships pages and how they look can be found in [“Controlling the Flow of Your IEG Script”](#) on page 47.

Attributes

Table 42: relationship-page attributes	
Name	Description
id	A unique identifier for this page. This id can be used to reference this page when linking to it from another part of the script, or when indicating what page to start at when re-entering a script.
progress	The percentage to use for the progress bar when the user gets to this page.
show-person-tabs	When set to true, this attribute will indicate to the IEG Engine and Player that person tabs should be displayed at the top of the page. For more information, see “Controlling the Flow of Your IEG Script” on page 47.
show-back-button	When set to true, this attribute will indicate to the IEG Player that the back button should be displayed. If not specified, this attribute will be set to true by default.
show-exit-button	When set to true, this attribute will indicate to the IEG Player that the exit button should be displayed. If not specified, this attribute will be set to false by default.
show-next-button	When set to true, this attribute will indicate to the IEG Player that the next button should be displayed. If not specified, this attribute will be set to true by default.
show-save-exit-button	When set to true, this attribute will indicate to the IEG Player that the 'Save & Exit' button should be displayed. If not specified, this attribute will be set to true by default.
read-only-expression	A Boolean expression which, if evaluated to true at runtime, causes the relationship page to become read-only.
set-focus	When set to false, this attribute will indicate to the IEG Player that no initial focus should be set on the form. If not specified, this attribute will be set to true by default.
mandatory	When set to true, this optional attribute will indicate to the IEG Player that the relationship type field on the relationships page is mandatory. If not specified, this attribute will be set to false by default.

Child elements

Name	MinOccurs	MaxOccurs
title	0	1
subtitle	0	1
description	0	1
icon	0	1
question	0	1
cluster	0	unbounded
display-text	0	1
next-button-label	0	1

row-help

Use the `row-help` element to specify help for rows in a list.

When an IEG script is run, the help text is not visible. A help icon is displayed on the right side of the row. The help text can be viewed by selecting help icon. To add row help to a list the row help is associated with an attribute of the entity displayed in the list. The help text displayed for each row will be based on the value of this attribute.

Attributes

Name	Description
id	Specifies the data store attribute that stores the property key for each row in the list. The value of this attributes specified the id of the help-text that is displayed for each row.

Child elements

Name	MinOccurs	MaxOccurs
help-text	1	unbounded

relationship-detail-header

Use the `relationship-detail-header` element to customize the header of the relationship list. The `relationship-detail-header` accepts an ICU message template. This element is applicable to IBM Cúram Universal Access (New) only.

For more information about the ICU messaging format, see <http://icu-project.org/apiref/icu4j/com/ibm/icu/text/MessageFormat.html>.

You can define the title as follows:

```
{index, select, 0 {Your relationships} other {{personName}'s relationships}}
```

For example:

```
<relationship-summary-list>
  <title id="RelationshipSummaryList.Title"><![CDATA[Relationships]]></title>
  <relationship-detail-header id="RelationshipDetail.Header"><![CDATA[{index, select, 0 {Your
relationships} other {{personName}'s relationships}}
relationships]]></relationship-detail-header>
</relationship-summary-list>
```

Attributes

Table 46: relationship-detail-header attributes	
Name	Description
id	An identifier for this text, which is used as the key with which to reference the text within the associated properties file. The id must be unique within the context in which it is used. For example, if the title is added to a cluster, then the id need be unique only within the page in which the cluster is contained. Another page could contain an id with the same value.

Child elements

The title element can contain a CDATA section to store the text used.

Table 47: relationship-detail-header child elements		
Name	MinOccurs	MaxOccurs
argument	0	unbounded

relationship-summary-list

Use the `relationship-summary-list` element on a summary page to display a list of all the household relationships that were captured by using the `relationship-page` element.

For more information about how to use the `relationship-summary-list` element, see [“Controlling the Flow of Your IEG Script”](#) on page 47.

Attributes

Table 48: relationship-summary-list attributes	
Name	Description
collapsed-expression	The expression that must be evaluated to define whether the relationship summary list is collapsed or expanded in its initial state. If the expression evaluates to <code>true</code> , the relationship summary list is initially collapsed. If the expression evaluates to <code>false</code> or if the <code>collapsed-expression</code> attribute is not specified, the relationship summary list is initially expanded.

Child Elements

Table 49: relationship-summary-list child elements

Name	MinOccurs	MaxOccurs
title	0	1
description	0	1
edit-link	0	1
column	0	unbounded
informational-message	0	1
relationship-detail-header	0	1

skip-field

Use the `skip-field` element for a more flexible layout of elements within clusters or footer rows in lists. You can use the `skip-field` element in clusters and footer rows where no visible display element is needed.

When the IEG Player displays a page containing a cluster or footer row that contains skip fields, those fields are rendered as spaces. This behavior allows rows in a cluster to have different numbers of visible display elements.

Attributes

None.

Child elements

None.

subtitle

Use the optional `subtitle` element in a script to define localized subtitle text that is displayed within a page subtitle directly below the title of a page. You can define a `subtitle` element within a `question-page` element, a `summary-page` element, or a `relationship-page` element.

The `subtitle` element has a mandatory ID attribute, `id`, that is used as a key to reference a text property in the appropriate locale-specific properties file. The value of the `id` attribute must be unique within the context in which it is used; for example, it must be unique within the page in which it is contained. For simplicity, you can also add the text to use for the default locale directly into the script definition by adding a CDATA section as a child of the `subtitle` element. The IEG engine removes the localized subtitle text from the script when you import the script. Instead, the IEG engine stores the subtitle text in the appropriate properties file, by using the `id` attribute of the `subtitle` element as the key.

Attributes

Table 50: Subtitle attributes

Name	Description
id	An identifier for the subtitle text that is used as the key to reference the text within the associated properties file. The value of the identifier attribute must be unique within the context in which it is used; for example, it must be unique within the page in which it is contained. Another page can contain a subtitle element identifier that has the same value.

Child elements

The subtitle element can contain a CDATA section to store the text that is used.

Name	MinOccurs	MaxOccurs
argument	0	unbounded

summary-page

Use the summary - page element to define the last page in a section.

The summary - page element displays read-only values of questions that were previously answered in the section. The summary page is displayed when you navigate to previous sections in the script. For more information, see [“Controlling the Flow using Sections” on page 47](#).

Some summary pages might contain much material that is displayed on multiple pages. Therefore, you might want to implement custom layout requirements so that you can use user interface design concepts. You can use the custom - output display element to render custom HTML on a summary page. The custom - output display element enables data from a data store instance to be retrieved and accessed from a custom renderer so that the data can be rendered by using custom HTML. For more information, see the related link.

Attributes

Name	Description
id	a unique identifier for this summary page. This id can be used to reference this page when linking to it from another part of the script.
entity	the name of an entity from the associated Datastore schema. The entity referenced must be a child of the root element within the schema. Any attribute within a cluster on this page which doesn't have its own entity specified, is assumed to be an attribute of this entity.
criteria	used in combination with the entity attribute to identify the exact record(s) to display/modify on this page. The criteria attribute can be thought of in the same way as the 'where' clause in an SQL statement. If the question-page is not contained within a loop, the IEG Engine will assume that there is only one record which matches the criteria and therefore only select the first record returned, so to ensure that everything works as intended, the criteria should only match a single record in this situation. If no record matches the specified criteria, then no values are displayed in the input fields on this page, and when the data entered by the user is saved, a new record will be created. If no criteria is specified, then it is assumed that only one instance of the entity exists per root element in the Datastore. If the page is contained within a loop, no criteria should be specified as the criteria for the loop will be used.
progress	the percentage to use for the progress bar when the user gets to this page.
show-back-button	when set to true, this attribute will indicate to the IEG Player that the Back button should be displayed. If not specified, this attribute will be set to true by default.

Table 52: summary-page attributes (continued)

Name	Description
show-exit-button	when set to true, this attribute will indicate to the IEG Player that the Exit button should be displayed. If not specified, this attribute will be set to false by default.
show-next-button	when set to true, this attribute will indicate to the IEG Player that the Next button should be displayed. If not specified, this attribute will be set to true by default.
show-save-exit-button	when set to true, this attribute will indicate to the IEG Player that the Save & Exit button should be displayed. If not specified, this attribute will be set to true by default.
read-only-expression	a boolean expression which, if evaluated to true at runtime, causes the summary page to become read-only. In this instance, Edit, Delete and Add links are not displayed.
set-focus	when set to false, this attribute will indicate to the IEG Player that no initial focus should be set on the form. If not specified, this attribute will be set to true by default.

Child Elements

Table 53: summary-page child elements

Name	MinOccurs	MaxOccurs
title	1	1
subtitle	0	1
description	0	1
icon	0	1
help-text	0	1
cluster	0	unbounded
condition	0	unbounded
list	0	unbounded
validation	0	unbounded
set-attribute	0	unbounded
relationship-summary-list	0	unbounded
next-button-label	0	1
custom-output	0	unbounded

Related concepts

[Rendering Custom HTML on a Summary Page](#)

title

Use the `title` element to define the localizable text for any titled element within a script, such as sections, pages, or clusters.

Each `title` element has an ID that references a text property in the appropriate locale-specific properties file. For simplicity, you can add the text to use for the default locale directly into the script definition by adding a CDATA section as a child of the title element. The IEG engine removes the text from the script when you import the script, and instead stores it in the appropriate properties file by using the ID of the title element as the key.

Also, you can use the optional `subtitle` element to define localized subtitle text that is displayed within a page subtitle directly below the title of a page.

Attributes

Name	Description
id	An identifier for this title text, which is used as the key with which to reference the text within the associated properties file. The id must be unique within the context in which it is used. For example, if the title is added to a cluster, then the id need be unique only within the page in which the cluster is contained. Another page could contain an id with the same value.

Child elements

The title element can contain a CDATA section to store the text used.

Name	MinOccurs	MaxOccurs
argument	0	unbounded

Meta-display elements

Meta-display elements, while not shown on the screen themselves, provide information for how display elements are displayed.

codetable-hierarchy-layout

Use the `codetable-hierarchy-layout` element in a question with a code-table hierarchy type to control the layout of that element. These layout options are not supported within compact layout clusters or in Universal Access (New).

Attributes

Name	Description
vertical	Set this boolean attribute to true for the IEG Player to display the code table hierarchy in a vertical layout. If not set, it defaults to false.

Table 56: codetable-hierarchy-layout attributes (continued)

Name	Description
hide-description	Set this boolean attribute to true for the IEG Player to not display description text for the code table hierarchy. If not set, it defaults to false. If show-path is true, and the question is read-only, this attribute does not apply. Only one of the hide-description or hide-label attributes can be true.
hide-label	Set this boolean attribute to true for the IEG Player to not display labels for the code table hierarchy path. If not set, the value of this attribute defaults to false. If show-path is false, and the question is read-only, this attribute does not apply. Only one of the attributes hide-description or hide-label can be true.
show-path	Set this boolean attribute to true for the IEG Player to display the path taken through the code table hierarchy to arrive at the final answer chosen. It only applies when the question is read-only. If not set, it defaults to false.

Child elements

None.

label-alignment

The label-alignment element can be used within a layout element for a cluster to control the alignment of the text within all the labels for that cluster.

See [“Summary of Cluster Layout Options” on page 58](#) for more details.

Attributes

None.

Child elements

The label-alignment element does not have any child elements, but must contain one of the following values:

- LEFT - to align the text to the left in the space that is allocated to the label.
- CENTER - to align the text to the center in the space that is allocated to the label.
- RIGHT - to align the text to the right in the space that is allocated to the label (the default).

label-width

The label-width element can be used within a layout element for a cluster to control the width of the labels within that cluster.

See [“Summary of Cluster Layout Options” on page 58](#) for more details.

Attributes

None.

Child elements

The label-width element does not have any child elements, but instead should contain an integer value between 0 and 100 to indicate the percentage of the width available to each question within the cluster which should be allocated to the label. The remaining width will then be allocated to the input control or value for the question.

layout

Use the layout element to control the layout of cluster and question elements, and, for IBM Cúram Universal Access (New), to format and constrain input fields.

For questions, you can control the width of the input control that is used to enter the answer to the question.

For more information about controlling the layout of clusters, see [“Using the Layout Element to Change the Appearance of Clusters”](#) on page 57.

The class-names child element

If you are developing for IBM Cúram Universal Access (New), you can use the class-names child element to improve input field readability by formatting or constraining the typed data. For more information, see [Configuring formats and constraints for input fields](#).

Attributes

None.

Child elements

Name	MinOccurs	MaxOccurs
type	0	1
num-cols	0	1
num-rows	0	1
width	0	1
label-width	0	1
label-alignment	0	1
class-names (Universal Access (New) only)	0	1

num-cols

The num-cols element can be used within a layout element for a cluster to control the number of columns within that cluster.

See [“Using the Layout Element to Change the Appearance of Clusters”](#) on page 57 for more details. Each column consists of both labels and input controls or values for each question.

Attributes

None.

Child elements

The num-cols element does not have any child elements, but instead should contain an integer value greater than 1 (the default) to indicate the number of columns in the cluster.

num-rows

The num-rows element can be used within a layout element for a cluster to display the text fields as a text area which spans the specified number of rows.

Attributes

None.

Child elements

The num-rows element does not have any child elements, but instead should contain an integer value greater than 1 (the default) to indicate the number of rows to display in the text area.

type

The type element can be used within a layout element for a cluster to control the layout of labels in relation to input controls.

See [“Using the Layout Element to Change the Appearance of Clusters” on page 57](#) for more details.

Attributes

None.

Child elements

The type element does not have any child elements, but instead should contain a string value. Furthermore, the applicable string values depend on the context of this type element: either the child of a cluster or the child of a question.

In the context of a cluster, use "flow" as the layout type if the labels should be displayed to the left of input controls or values, or "compact-flow" if the labels should be displayed above the input controls or values.

In the context of a question, use "radio" as the layout type if the question has a code table as its data type and the possible answers should be displayed as a group of radio buttons to the side of the question label. Use "radio-indent" as the layout type if the question has a code table as its data type and the possible answers should be displayed as a group of radio buttons underneath the question label and tabbed in from the side. The "radio-indent" layout type should only be used in clusters that have one column.

width

The width element can be used within a layout element for a cluster to control the width of the cluster on the page.

See [“Using the Layout Element to Change the Appearance of Clusters” on page 57](#) for more details.

Attributes

None.

Child elements

The width element does not have any child elements, but instead should contain an integer value between 0 and 100 to indicate the percentage of the width available on the page which should be allocated to the cluster.

Flow-control elements

Flow-control elements are logical constructs to indicate to the IEG Engine which display elements should be shown.

They are invisible to a user. The category includes loops and conditions.

condition

The condition element can be used to instruct the IEG Engine as to whether or not to display a given element (whatever is contained within the condition), based on the answers to some previously asked questions.

The most common usages of the condition element are within a section (where they typically contain question pages - see “Controlling the Flow of Your IEG Script” on page 47 for more details for more details), or within a page, where it can be used to conditionally display a cluster on the page. On a summary page, the condition element can also be used to conditionally display a list.

When used with clusters, this can lead to two different types of behavior. If the value of the expression for the condition can be determined by the IEG Engine before it displays the page, then it either includes or excludes the cluster from the page and the cluster will remain either hidden or displayed for the duration of this visit to the page. However, if the expression cannot be evaluated before displaying the page (i.e., if one of the questions on this page is used in the expression), then the Engine hands responsibility over to the IEG Player, which will dynamically hide/display the cluster based on the answer the user gives to the relevant question. Dynamically conditional clusters are highlighted differently from other clusters on a page, they have a border and are shaded.

Please note when defining a condition for a dynamically conditional cluster the following restrictions apply:

- the expression cannot refer to custom functions
- the result of evaluating the expression must be a boolean

Attributes

Name	Description
expression	a Boolean expression used to determine whether or not to display the elements contained within this condition.
fast-path	when set to true, this attribute will indicate to the IEG Engine that Fast Path navigation should be switched on for this condition and the enclosed elements on which it applies. If not specified, this attribute will be set to false by default. See Fast Path Navigation .

Child elements

Name	MinOccurs	MaxOccurs
cluster	0	1
condition	0	unbounded
list	0	unbounded

Table 59: Condition Child Elements (continued)

Name	MinOccurs	MaxOccurs
loop	0	unbounded
question-page	0	unbounded
summary-page	0	1
callout	0	unbounded

loop

The loop element can be used to instruct the IEG Engine to repeat the page(s) contained within the loop multiple times. How many times the pages are repeated is dependent on the type of loop and the criteria/expressions used.

Full details of all the available loop types and how to use them can be found in [“Controlling the Flow of Your IEG Script”](#) on page 47.

Attributes

Table 60: Loop Attributes

Name	Description
loop-type	the type of loop you wish to use. The loop-type can be either for, for-each or while.
entity	the entity to which the data on the pages within the loop will be saved, and from which it will be loaded when re-iterating through the loop. This is a required attribute for all loop types.
criteria	the criteria to use when retrieving the records from the entity over which you'll iterate in this loop. This is a required attribute for all loop types.
loop-expression	an integer expression used to determine the number of iterations in a for loop the first time you pass through it (thereafter it will be based on the number of records, as in a for-each loop).
fast-path	when set to true, this attribute will indicate to the IEG Engine that Fast Path navigation should be switched on for this loop and the enclosed elements on which it applies. If not specified, this attribute will be set to false by default. See Fast Path Navigation for more details on Fast Path.

Child Elements

Table 61: Loop Child Elements

Name	MinOccurs	MaxOccurs
condition	0	unbounded
loop	0	unbounded
question-page	0	unbounded
callout	0	unbounded

validation

The validation element is used to validate answers on a question page. Each validation element has an expression, which is used to evaluate if an answer is a valid answer.

The expression must evaluate to true for the script execution to proceed beyond the page containing the validation element. For example, the following validation element displays a warning message if the value for the wage amount is less than or equal to zero:

```
<validation expression="Income.wageAmount > 0 ">
  <message id="Page2.noWageValidationMessage">
    <![CDATA[You entered %1d as your wage amount.
    Please enter a value greater than zero.]]>
    <argument id="Income.wageAmount" />
  </message>
</validation>
```

Figure 5: Validation XML

Attributes

Name	Description
expression	a Boolean expression used to determine whether or not answer(s) on the page are valid.

Child elements

Name	MinOccurs	MaxOccurs
message	1	1

Structural, administrative and other elements

A list of IEG elements that are neither displayed or used explicitly for flow control.

callout

The callout element signifies the invocation of code that is not part of IEG to perform some validation or other operation for the script execution, with access to the Datastore provided to the function through the appropriate identifiers.

Callouts can be placed anywhere question pages can exist in the script, except before the first question page. They will invoke a custom function specified in the expression before moving to the next page. Note that if a page is accessed from a summary page link and is followed by callouts, the callouts will get invoked unless the 'skip-to-summary' attribute has been set to true on the summary link.

Attributes

Name	Description
id	an identifier for this callout, which must be unique on a page.

Table 64: Callout Attributes (continued)

Name	Description
expression	an expression that is used to invoke a callout custom function at this point in the script. Datastore information (root entity ID, execution ID and current entity ID) is provided to the function automatically.

Child elements

None.

identifier

The identifier element is a mandatory element of a script and contains the information required to identify a script in the database.

All three attributes described below form part of the key for the script record, which allows for multiple versions of a script with the same id.

Attributes

Table 65: Identifier Attributes

Name	Description
id	the identifier, or name, of the script
scriptversionnumber	the version number of the script.
type	a type for the script. For example, this can be used to create a logical grouping of scripts which are used for different purposes so that they can be filtered when displaying lists of scripts.

Child elements

None.

ieg-script

The ieg-script element is the root element of the XML file that contains an IEG script and is used to define the script element itself.

Attributes

Table 66: ieg-script Attributes

Name	Description
config-properties	this attribute allows you to specify a properties file to use for modifying the text and style used for various components in the IEG Player.
fast-path	when set to true, this attribute will indicate to the IEG Engine that Fast Path navigation should be switched on for all script elements on which it applies. If not specified, this attribute will be set to false by default. See the Fast Path section of the Working with IEG guide for more details on Fast Path.

Table 66: *ieg-script* Attributes (continued)

Name	Description
finish-page	the id of the UIM page to which the user will be taken when there are no more pages left to display in the script (i.e. when the user clicks on the Next button on the last page in the script).
hide-for-control-question	when set to true, the label and value of questions that control for loops will be hidden once the loop it controls has been entered. If not specified, it will default to false and the question value will be read-only.
highlight-validation	when set to true, this attribute will indicate to the IEG Player that the mandatory or domain validation errors displayed at the top will also be repeated next to the failing question. If not specified, the value will be taken from the script configuration properties (using the key <code>validation.highlight</code>), and if not present, it will default to false. If two questions on the page have the same label (not recommended) and one of them has failed validation, both questions will be highlighted.
quit-page	the id of the UIM page to which the user will be taken when they click on the Save and Exit button at any stage within this script.
show-progress-bar	when set to true, this attribute will indicate to the IEG Player that a progress bar should be displayed at the top of the page. If not specified, this attribute will be set to true by default.
show-sections	when set to true, this attribute will indicate to the IEG Player that the sections panel should be displayed on the left side of the page. If not specified, this attribute will be set to true by default.
validate-save-and-exit	when set to false, this attribute will indicate to the IEG Player that page validations and mandatory validations should not be performed when the Save and Exit button is selected. If not specified, this attribute will be set to true by default.

Child elements

Table 67: *Ieg-Script* Child Elements

Name	MinOccurs	MaxOccurs
identifier	1	1
section	0	unbounded

ieg-sub-script

The `ieg-sub-script` element can be used to include the subscript in the script. The subscripts are standalone scripts that can be included in another script. A subscript can be included at script-level or in a section. When a subscript is included in a section of another script then it should not contain any sections.

Attributes

<i>Table 68: ieg-sub-script attributes</i>	
Name	Description
start-progress	the starting percentage value to use for calculating the progress for the progress bar for each page in the subscript.
end-progress	the ending percentage value to use for calculating the progress for the progress bar for each page in the subscript.

Child elements

<i>Table 69: ieg-script Child Elements</i>		
Name	MinOccurs	MaxOccurs
identifier	1	1

section

The section element represents a section of pages within an IEG script.

Attributes

<i>Table 70: Section Attributes</i>	
Name	Description
read-only-expression	a boolean expression which, if evaluated to true at runtime, causes the section to become read-only.
visible	the expression specified in this attribute will be evaluated at the start of a script execution and if it evaluates to false, the section will be removed from the execution. The expression won't be re-evaluated during script execution.

Child elements

<i>Table 71: Section Child Elements</i>		
Name	MinOccurs	MaxOccurs
title	1	1
icon	0	1
question-page	0	unbounded
loop	0	unbounded
condition	0	unbounded
relationship-page	0	unbounded
summary-page	1	1
callout	0	unbounded

set-attribute

The set-attribute element can be used within a question page to set the value of an attribute within the entity specified for the page, without asking the user a question.

This can be used in scenarios where you know what value to set because of the page you are on. An example of this would be if you had a page at the start of your script which is used to capture the primary person (usually the person filling in the details for this script), and want to be able to distinguish that person from the rest of the people you capture. To do this, you could have an attribute in the Person entity called 'isPrimary' and set it to true on the primary person page as follows:

```
<question-page id="AboutYouPage" entity="Person"
  criteria="isPrimary==true">
  <set-attribute id="isPrimary" expression="true" />
```

Figure 6: set-attribute XML

You can either have a set-attribute which sets isPrimary to false on the pages capturing other people, otherwise, you can set the default value for isPrimary to false in the schema definition.

Attributes

Name	Description
id	the identifier of a set-attribute refers to the name of the attribute within the entity specified for the page for which to set a value for in the Datastore.
expression	the value for which to set the attribute to in the Datastore.

Child elements

None.

Operations Supported for IEG Expressions

This section provides information on the operations and data types and that are supported for IEG expressions. It also provides information on the bracketing of terms, operator precedence and the use of custom functions in expressions.

Expressions are used in various parts of an IEG script to control the flow of the script and the content of pages, e.g. loops, conditions, criteria. They are also used in callouts to invoke external functionality.

Bracketing of Terms

The bracketing of terms can have a significant impact on the result of a calculation. The behavior is as normal for mathematical operations, but the effects of brackets can be combined with operator precedence and may add complexity to an expression. Any operation that should be carried out in advance of another operation should be bracketed, e.g., $5 * (3/4) = 3.75$.

Operator Precedence

The precedence of operators is as defined for the Java™ programming language.

The operators in the following table are listed in order of precedence:

Table 73: Operator Precedence

Operator	Associatively	Type
()	left to right	parentheses
* /	left to right	multiplicative
+ -	left to right	additive
< <= > >=	left to right	relational
== !=	left to right	equities

Data Types and Supported Operations

The operations that are explicitly supported between the data types are detailed in the following table.

It is possible to perform operations between the data types not listed in the table if the underlying data type of an attribute can be converted into one of the types for which an operation is supported.

For example, the addition of IEG_INT8 and IEG_MONEY is possible, because IEG_INT8 is converted into IEG_DOUBLE and the addition of IEG_DOUBLE and IEG_MONEY is supported.

It is possible to add or subtract integers from dates. Integers represent the number of days to be added or subtracted.

Table 74: Data Types and Supported Operations

The first parameter type	The second parameter type	Operations supported	Result type
IEG_STRING	IEG_STRING	==, !=	IEG_BOOLEAN
IEG_CHAR	IEG_CHAR	==, !=	IEG_BOOLEAN
IEG_MONEY	IEG_MONEY	==, !=, <, >, <=, >=	IEG_BOOLEAN
IEG_MONEY	IEG_DOUBLE	==, !=, <, >, <=, >=	IEG_BOOLEAN
IEG_DOUBLE	IEG_MONEY	==, !=, <, >, <=, >=	IEG_BOOLEAN
IEG_DOUBLE	IEG_DOUBLE	==, !=, <, >, <=, >=	IEG_BOOLEAN
IEG_DATE	IEG_DATE	==, !=, <, >, <=, >=	IEG_BOOLEAN
IEG_MONEY	IEG_MONEY	+, -, /, *	IEG_DOUBLE
IEG_MONEY	IEG_DOUBLE	+, -, /, *	IEG_DOUBLE
IEG_DOUBLE	IEG_MONEY	+, -, /, *	IEG_DOUBLE
IEG_DOUBLE	IEG_DOUBLE	+, -, /, *	IEG_DOUBLE
IEG_FLOAT	IEG_FLOAT	+, -, /, *	IEG_DOUBLE

Table 74: Data Types and Supported Operations (continued)

The first parameter type	The second parameter type	Operations supported	Result type
IEG_INT8	IEG_INT8	+, -, /, *	IEG_INT32
IEG_INT16	IEG_INT16	+, -, /, *	IEG_INT32
IEG_INT32	IEG_INT32	+, -, /, *	IEG_INT32
IEG_INT64	IEG_INT64	+, -, /, *	IEG_INT64
IEG_DATE	IEG_INT32	+, -	IEG_DATE

Custom functions in expressions

Custom functions can also be referenced by expressions defined in an IEG script. By default, the `setFullNameAge` and `isNotNull` functions are included in the application.

As custom functions are on the server side, they cannot be referenced by expressions that are evaluated on the client side. This means that custom functions cannot be referenced by expressions of dynamically conditional clusters. Custom functions cannot accept a variable number of parameters.

setFullNameAge

The `setFullNameAge` custom function sets a Person's `fullNameAge` attribute, it concatenates the person's `firstName` and `lastName` attributes and the calculation of the age based on the `dateOfBirth` attribute.

```
CustomFunctionMetaData.xml
<CustomFunctions>
  <CustomFuncor name="CustomFunctionsetFullNameAge">
    <parameters />
    <returns>curam.util.rules.functor.Adaptor$BooleanAdaptor</returns>
  </CustomFuncor>
</CustomFunctions>
```

You must update the Person schema definition and call the function in the script definition.

Example of an updated Person schema definition:

```
<xsd:element name="Person">
  <xsd:complexType>
    <xsd:attribute name="firstName" type="IEG_STRING" default=""/>
    <xsd:attribute name="lastName" type="IEG_STRING" default=""/>
    <xsd:attribute name="dateOfBirth" type="IEG_DATE"/>
    <xsd:attribute name="fullNameAge" type="IEG_STRING" />
  ...
  </xsd:complexType>
</xsd:element>
```

Example of calling the function in the script definition:

```
<callout id="setFullNameAgeCallout" expression="setFullNameAge()"/>
```

The `fullNameAge` attribute is very useful when it is set for the first column of the list element, as it enables singling out persons very easily. As it is the first column of the list, it is used to generate the `add-link` options. The following code snippet shows you how to set the attribute.

```
<list entity="Person" show-icons="false">
  <title id="EmploymentIncome.Title">Employment Income</title>
  <edit-link start-page="EmploymentIncomePage"/>
```

```

        <delete-link/>
        <add-link start-page="EmploymentIncomePage" skip-to-summary="true">
            <title id="AddIncome.Title">Add income</title>
        </add-link>
    <column id="fullNameAge">
        <title id="Person.Title">PERSON</title>
    </column>
    ...
</list>

```

isNotNull

Refer to the following example for the `isNotNull` function.

```

<loop loop-type="for-each" entity="Person"
    criteria="isNotNull(Person.hasIncome) and hasIncome==true">

```

IntakeProgramType and ScreeningProgramType in Expressions

IEG allows customers to create dynamic scripts for collecting data which is typically used as part of an application for a program or to determine potential eligibility.

As such IEG provides some exceptional processing in determining the programs for which a claimant is applying.

When defining expressions in an IEG script you can refer to what appears to be two entity types named `IntakeProgramType` and `ScreeningProgramType`. However, these entity types are not actually defined in the schema used to execute the script. IEG performs a transformation on these expressions and the entity types that should be defined in the schema are `IntakeProgram` and `ScreeningProgram`. These entity types should be defined with an attribute called `programTypeReference` with the type `IEG_STRING`. The root entity is then defined to contain collections of `IntakeProgram` and `ScreeningProgram` entities as follows:

```

<xsd:element name="Application">
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element ref="IntakeProgram"
        minOccurs="0" maxOccurs="unbounded" />
      <xsd:element ref="ScreeningProgram"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="IntakeProgram">
  <xsd:complexType>
    <xsd:attribute name="programTypeReference" type="IEG_STRING"/>
  </xsd:complexType>
</xsd:element>
<xsd:element name="ScreeningProgram">
  <xsd:complexType>
    <xsd:attribute name="programTypeReference" type="IEG_STRING"/>
  </xsd:complexType>
</xsd:element>

```

Figure 7: Intake and Screening Program Schema

This allows a single IEG script to be used to gather information required in the processing of applications for multiple programs.

IEG does not create the `IntakeProgram` and `ScreeningProgram` entities but merely checks their existence and the value of their attributes. Therefore to use this feature, the Datastore should be pre-populated with the required entities. When pre-populating the entities the value of the `programTypeReference` attribute should correspond to what appears to be the attribute name referred to in the expression in the script definition. For example, an expression can be defined as follows:

```

<condition expression="IntakeProgramType.FoodStamps==true">
  ...
</condition>

```

Figure 8: Intake Program Expression

When this expression is evaluated, IEG checks to see if the root entity contains a child entity of type IntakeProgram where the attribute programTypeReference, contains the string "FoodStamps". ScreeningProgram can be referred to similarly:

```
<condition expression=
  "ScreeningProgramType.CashAssistanceProgram ==true">
  ...
</condition>
```

Figure 9: Screening Program Expression

When this expression is evaluated, IEG checks to see if the root entity contains a child entity of type ScreeningProgram where the attribute programTypeReference, contains the string "CashAssistanceProgram".

Only the entity types IntakeProgram and ScreeningProgram are supported in this way but there is no restriction on the value of the programTypeReference attribute in either entity type.

Controlling the Flow of Your IEG Script

This section outlines how a script developer can control the presentation of pages in an IEG script through the use of page order; sections; conditions; loops; control questions; and action links.

Judicious use of flow control constructs can help make a script intuitive and efficient for a user by ensuring that questions are asked in a context that makes sense. It also helps the developer to avoid the presentation of unsuitable or irrelevant questions.

Natural Flow of an IEG Script

The natural flow of an IEG script is governed by the order in which the pages are defined. The first page in the first section of the script will be the first page displayed when the script is executed, and each time the user hits the Next button, the next page in that section will be displayed.

When there are no more pages in that section to be displayed, the summary page for the section will be displayed. Clicking on the Next button on a summary page takes the user to the first page in the next section. Clicking Next on the summary page for the last section in the script takes the user to a configured finish page, which exists outside of the script.

The Back button allows the user to work their way back through a script in a similar way.

If the user clicks on the Back button having already entered, but not saved, some information on this page, the information they have entered *is discarded*.

In IBM Cúram Universal Access (New), information is not lost when navigating through the application.

In addition to the Back and Next buttons, a Save and Exit button will be available on each page. On clicking the 'save and exit' button, the system will attempt to save the information entered on the page, and then (assuming there are no validation errors) take the user to a configured quit page, which exists outside of the script.

The Exit and Save and Exit buttons allow navigation to cease before a script execution has concluded. The Exit button can also be enabled on a question page element by specifying show-exit-button="true". This will allow you to exit the script without saving the information entered. The Back, Next and Save and Exit buttons are also optional but are enabled by default. For more information, see ["question-page"](#) on page 24.

Controlling the Flow using Sections

As the user navigates through the different sections in a script using the 'next' button, a link for each completed section will be enabled in the sections panel to the left of the screen (in the default configuration). Clicking on this link will bring the user to the summary page for that section. All previously completed sections remain enabled as the user jumps back and forward through the script.

For example, if the first four sections are completed in the script, a user can jump to the summary page for Section 3, then to the summary page for Section 1, and back to the summary page for Section 3. Sections are only enabled up to the furthest point in the script which the user has visited. If the furthest section has not yet been completed (i.e., the user has not accessed the summary page for that section), instead of the link for that section bringing the user to its summary page, it will go to the furthest page visited in the section. If no summary page was defined for the section, or if the summary page is conditional and wasn't displayed, the navigation will still bring the user to the furthest visited page in the section.

The list of enabled sections can change depending on the actions a user takes when editing and deleting previous answers or adding new ones. For example, a user progresses through a script which contains 5 sections and has made it to the summary page for the fourth section. The user then returns to the second section and edits some answers. If the new answers are used to determine the flow of later parts of the script, then it is no longer safe to allow the user to jump over all the pages in the script as the original route taken through the script may no longer be valid. All sections after the section containing the page on which the new answers were given will be automatically disabled. The user is forced to use the next button to progress through the script so that the correct route can be determined based on the new answers. No information previously entered is discarded when sections are disabled unless the pages containing the information no longer form part of the script execution.

Some sections might contain questions that are only applicable based on prior answers. It is possible to make some sections conditional: a 'visible' attribute can be specified on sections. It will contain an expression that will be evaluated at the start of the execution. If it evaluates to false, the section will be removed from the execution and won't be displayed in the sections panel. Another possibility is to wrap all the elements contained in a section (including the summary page) into a single condition. This conditional navigation will follow the same logic as existing conditions, so the expression will be evaluated when it is encountered during script execution. In this case, the sections will be displayed in the sections panel even if they are not encountered by the user (but they will be disabled).

Controlling the Flow using Conditions

Conditions can be used to control a user's flow through an IEG script. A user's progression through an IEG script will follow a linear path from page to page within the script, unless you as the script author decide to change that flow. The main reason for changing this flow is to ensure users are not faced with unnecessary or irrelevant questions.

A user's answers should be used to determine which questions do or do not relate to that user. For example, you may have a page which captures detailed information about a person's third-level education. You would not want to display this page to the user if the user had already indicated that he or she never attended a third-level college. In order to achieve this, you can surround one or more pages in a condition element to indicate under what conditions that page (or pages) is to be displayed; something like this:

```
<condition expression="attendedThirdLevel==true">
  <question-page id="ThirdLevelDetailsPage">
    ...
  </question-page>
</condition>
```

Figure 10: Condition Element

When the user clicks on the Next button on the page prior to this condition, the system will evaluate the expression 'attendedThirdLevel==true'. If it evaluates to true, the page(s) within the condition will be displayed to the user; if it evaluates false, then the system will skip the page(s) and display the page after the condition element.

In this example, attendedThirdLevel is the id of a control question asked earlier in this script. The answer given will be used in the evaluation of the expression. If you want to use the value of an attribute within the Datastore instead, then you just need to prefix it with the name of the entity in which it is contained (e.g., Person.attendedThirdLevel).

When using expressions on conditions (or anywhere else for that matter), you must ensure that any attributes used within the expression will have a real value by the time they are evaluated. By default,

attributes within the Datastore are null until a value has been set for them, and will stay that way if the user does not enter or choose a value for the appropriate answer.

Generally speaking, there are two ways to ensure that attributes have values before they are used in expressions: make the questions that populate them mandatory or give them default values in your Datastore schema (you could also use the set-attribute element in some cases, see [“set-attribute” on page 43](#)).

Conditions can contain any combination of pages (including summary pages), loops and other conditions. For example:

```
<condition expression="attendedThirdLevel==true">
  <question-page id="ThirdLevelDetailsPage">
    ...
  </question-page>
  <condition expression="hasMasters==true">
    <question-page id="MastersDetailsPage">
      ...
    </question-page>
  </condition>
</condition>
```

Figure 11: Nested Condition

This means that ThirdLevelDetailsPage would only be displayed if the attendedThirdLevel answer is true, and MastersDetailsPage would only be displayed if attendedThirdLevel is true and hasMasters is also true.

It is also possible to invoke a custom function from a condition or other expression. Information is automatically provided to the function to access the Datastore, i.e. root entity ID, script execution ID and currently entity ID (if the condition is within a loop). It should be noted that custom functions that have a side-effect (for example to populate some answers in the Datastore) shouldn't be used in such expressions as they won't necessarily be evaluated before the page content is loaded.

The custom function `isNotNull` is provided out-of-the-box in IEG to allow expressions to handle null values as parameters. For example, to validate a person's date of birth, it might first be necessary to ensure that a value exists:

```
<validation expression="
  isNotNull(Person.dateOfBirth)
  and isNotNull(Person.today)
  and (subDates(Person.dateOfBirth, Person.today) < 0)">
  <message id="DateOfBirthValidation">
    Your date of birth must be before today
  </message>
</validation>
```

Figure 12: Using 'isNotNull' Custom Function

Controlling Page Content using Conditional Clusters

Conditions can also be used within a page to introduce two types of conditional cluster:

- clusters which are dynamically hidden/displayed based on answers to questions on the current page, or a combination of questions on the current page and previous pages. The expression used to control dynamic clusters cannot refer to custom functions and must refer to at least one question from the current page. Any control questions that the condition expression refers to must occur at the start of the expression.
- clusters which are hidden/displayed based on the answers to questions on previous pages. In this case, the visibility of the cluster is determined before the current page loads and cannot be changed by answers given the page.

See [“condition” on page 37](#) for more details.

A conditional cluster may contain one or more mandatory questions, making these questions *conditionally mandatory*. For dynamically conditional clusters, the mandatory questions contained will only be validated as such if the cluster is visible when the page is submitted.

Controlling the Flow using Loops

Another way to control the flow of your script is to repeatedly display the same page or set of pages. This is achieved by adding a *loop* element to a script. There are three types of loops available within IEG.

The For-each loop

This is the simplest form of loop available as it always behaves the same way. When adding a for-each loop to your script, you must specify an entity in the Datastore to use for the loop. When the system encounters the loop, it will retrieve all the instances of that entity within its parent entity, and perform one iteration of the loop (presenting each page within the loop) for every instance returned.

For example, the following loop could be used to display the `ExtraPersonDetailsPage` for each person in this application (the root entity in our earlier example):

```
<loop loop-type="for-each" entity="Person">
  <question-page id="ExtraPersonDetailsPage">
    ...
  </question-page>
</loop>
```

Figure 13: For-each Loop

More commonly, you might not want to loop through all instances of an entity, in which case you can add a criteria which the system will use to select only those instances of the entity which match the criteria. For example, to loop through all the people for whom the 'hasIncome' attribute has been set to true, use a loop like this:

```
<loop loop-type="for-each" entity="Person"
criteria="hasIncome==true">
  <question-page id="IncomeDetailsPage" entity="Income">
    ...
  </question-page>
</loop>
```

Figure 14: For-each Loop with Criteria

If a criteria is used in a for-each loop it is recommended the criteria should contain a simple expression referring just to a single attribute of type Boolean (for example "hasIncome==true"). If a single attribute is referred to in the criteria the attribute can be automatically updated by IEG when using summary links (for example when all the nested entities are removed the attribute can be set to false or when the first nested entity is added the attribute can be set to true). This functionality is not available if the criteria contains a complex expression. It is also recommended a default value should be defined in the Datastore schema for this attribute.

The For loop

The for loop is used to perform a given number of iterations of the loop.

The number of iterations is determined by the value of the loop-expression attribute, which looks something like this:

```
<loop loop-type="for" loop-expression="numPeople">
  <question-page id="PersonDetailsPage" entity="Person">
    ...
  </question-page>
</loop>
```

Figure 15: Simplified For Loop

In other words, the number of times the `PersonDetailsPage` is displayed will be determined by the value of the answer to the `numPeople` control question. While this might work fine the first time through the loop, it is important to consider what happens when going through the loop a second or third time when reviewing or changing answers. For example, during the previous iterations, one or more persons may have been captured so it might make sense to loop through them rather than continuing to add new people.

In effect, the for loop becomes a for-each loop once some data has been entered in the entity its recording entries against. As such, it is necessary to give the for loop the same information given to a for-each loop: an entity to iterate over and an optional criteria. Once the entity is specified on the loop, there

is no need to specify it for the pages within the loop so long as they are the same. The loop might look something like this:

```
<loop loop-type="for" loop-expression="numPeople"
      entity="Person" criteria="isPrimary==false">
  <question-page id="PersonDetailsPage">
    ...
  </question-page>
</loop>
```

Figure 16: For Loop with Entity and Criteria

It is recommended when using for loops that the loop expression should be a simple expression referring just to the id of a question that is asked prior to the loop to determine the number of records to create. This question should be a control-question of type Integer.

This control-question will not be updated automatically, so it will be out of sync with the actual number of entities if entities are added or deleted through a summary page. Therefore its value shouldn't be used for anything other than the loop expression.

Once the loop has been started, it will be impossible to modify the value of this control-question, it will be read-only by default, unless the "hide-for-control-question" attribute has been set to true on the ieg-script element, in which case the label and value of the control-question will be hidden. The script designer should then ensure that the control-question is not the only question on the page where it is defined as this would lead to an empty page being displayed.

In practice, for loops have restricted application and therefore while loops are usually recommended for capturing information as their use can be more intuitive to the user.

The While Loop

The while loop is used in situations where the number of required loop iterations is unknown. The number of loop iterations is decided by a user's answer to a question within each iteration of the loop.

For example, you might want to ask the user to enter some income details and at the same time, ask whether the user has any more income to enter. This could be achieved with a loop like this:

```
<loop loop-type="while" loop-expression="hasMoreIncome"
      entity="Income">
  <question-page id="IncomePage">
    <cluster>
      <question id="type">
        <label id="Type.Label">
          <![CDATA[Type:]]>
        </label>
      </question>
      <question id="amount">
        <label id="Amount.Label">
          <![CDATA[Amount:]]>
        </label>
      </question>
      <question id="hasMoreIncome"
        control-question="true"
        control-question-type="IEG_BOOLEAN">
        <label id="ContinueQuestion.Label">
          <![CDATA[More income?]]>
        </label>
      </question>
    </cluster>
  </question-page>
</loop>
```

Figure 17: While Loop

The while loop will always perform at least one iteration (which makes it more of a do-while loop in programming parlance). If you have a situation where you want to check whether to go into the loop at all, then it should be wrapped in a condition.

The while loop suffers from the same complication as the for loop when it comes to going back though the loop when information has already been entered. It too effectively becomes a for-each loop up to the point at which the user has iterated through all the previously entered records. The while loop also

requires the entity attribute to be set (as in the above example) and gives you the option of specifying a criteria.

Loops can be nested inside other loops and one of the most common usages of the while loop is to nest it inside a for-each loop. To extend the above example, multiple people might already have been captured by the time the income loop is reached in the IEG script. To capture multiple incomes per person, assuming the user has already been asked which persons have any income, the nested loop would look something like this:

```
<loop loop-type="for-each" entity="Person"
criteria="hasIncome==true">
  <loop loop-type="while" loop-expression="hasMoreIncome"
entity="Income">
  <question-page id="IncomePage">
    ...
    <cluster>
      ...
      <question id="hasMoreIncome"
control-question="true"
control-question-type="IEG_BOOLEAN">
        <label id="ContinueQuestion.Label">
          <![CDATA[More income?]]>
        </label>
      </question>
    </cluster>
  </question-page>
</loop>
</loop>
```

Figure 18: Nested Loop

It is recommended when using while loops that the loop expression should be a simple expression referring just to the id of a question that is asked inside the loop to determine if more records should be added. This question should be a control-question of type Boolean.

The control-question will be updated automatically when adding or deleting a record through the summary page. When reviewing the answers by going through the loop after the initial pass, the question will be read-only, except on the last iteration, to provide the opportunity to add more entities.

Controlling the Flow Using Nested Loops

IEG provides the ability to create entities in the Datastore that are nested inside other entities. This section provides guidance on defining scripts to gather and display information in this area.

Scenario: It is required that several entities of the same type be registered in the Datastore. These entities may also contain entities themselves. For example, a number of Person entities need to be created to represent the members of a household. Each Person entity may also contain a number of Income entities representing the sources of income each household member has.

This information can be gathered in an IEG script using nested loops (a loop element containing another loop element). The information gathered can be displayed on a summary page using nested lists (a list element containing another list element).

The following should be considered when choosing the loop types to capture the required information:

- If the outer entity has already been created (because it has been created in another loop or the Datastore has been pre-populated), the loop type should be *for-each*.
- If the number of entities to create can be predetermined (for example the user has been asked "How many children do you have?"), the loop type should be *for*.
- If the number of entities to be created cannot be predetermined, the loop type should be *while*.

If using nested loops it is then recommended to use one of the following 6 combinations:

- For-Each/While
- For-Each/For
- While/While
- For/While

- While/For
- For/For

Note that the use of While/For-each or For/For-each nested loops is not recommended. As a for-each loop relies on existing entities to iterate over, this would involve the outer loop in creating entities containing entities that already exist. This situation is not possible as it does not make sense.

Control-Questions

In IEG, a question may be specified as being a *control question*. Control questions are defined by setting the control-question attribute to true and specifying a control-question-type. Control questions can be used to control the flow of the script or to control the display of clusters on a page.

The answers supplied to control questions are not persisted in the Datastore, hence having to specify a type in the script definition.

Control questions may be referred to in:

- loop expressions: If a control question is referenced in the loop-expression of a for loop, the control question type should be defined as integer. If a control question is referenced in the loop-expression of a while loop, the control question type should be defined as boolean. See [“Controlling the Flow Using Nested Loops”](#) on page 52 for more details.
- condition expressions: If a control question is referenced in the expression of a condition, the control question type should be defined as boolean.

The scope of control questions is global within a script execution. Defining multiple control questions with the same ID will result in unexpected behavior and should be avoided. For example, two separate while loops should not be controlled by the same "hasMore" control question.

When control questions are referenced by for loops, once an answer is supplied and the loop execution begins the answer to the control question may not be changed. When control questions are referenced by while loops, once an answer is supplied and the loop execution begins the answer to the control question may not be changed except for the last record in the loop.

Looping through People

By far, the most common type of entity to loop over in IEG scripts is the *Person* entity. IEG comes with some handy features to help you in this regard. The first feature is *person tabs*.

When using person tabs, the user will be presented with a panel between the page title and the main contents of the page which shows all the people in the household and highlights the person for whom the user is currently entering information.

Each person is represented by his or her first name and an icon to depict whether the person is a man, woman, boy or girl. A generic person icon is also provided for persons whose gender and date-of-birth has not yet been provided. Configuring a page to use person tabs is as simple as setting the show-person-tabs attribute to true for that page. Note that the page must be within a loop whose entity attribute is set to 'Person' for this to work.

When used on a page within a for loop, the first time the user enters the loop, the only information known is the number of people to be captured. The system then builds up information about the people as the user goes through the loop. The only indication the person tab can give is how many people are left to enter and not their age or gender.

Note that the show-person-tabs attribute can also be set on pages within a nested loop, so long as the entity for the outer loop is set to 'Person'. In that way, the user can still see the person for whom they are collecting the information in the inner loop.

Another feature of IEG which can be used to help capture information about people in a household is the relationship-page element. This element provides a simple way of instructing the system to capture the relationships between the household members. Including a relationship-page element in a script looks something like this:

```

<relationship-page id="RelationshipPage" show-person-tabs="true">
  <title id="RelationshipPage.Title">
    <![CDATA[Household Relationships]]>
  </title>
</relationship-page>

```

Figure 19: Relationship Page XML

The system will automatically take the user through a loop of the people entered so far, and allow the user to enter details of the user's relationships with each of the other members of the household. The system will only ask the user to enter relationships which have not yet been entered, so for each person in the household, there will be one less relationship to enter. This means that no relationships will be captured for the last person as his or her reciprocal relationships would have already been entered.

By default, the relationships page will only ask for the type of each relationship. You also have the option of using an indicator to record whether a relationship is a non-parent caretaker relationship. This can be done using the following syntax:

```

<relationship-page id="RelationshipPage" show-person-tabs="true">
  <title id="RelationshipPage.Title">
    <![CDATA[Household Relationships]]>
  </title>
  <question id="caretakerInd">
    <label id="CaretakerInd.Label">
      <![CDATA[Is this a non-parent caretaker relationship]]>
    </label>
  </question>
</relationship-page>

```

Figure 20: Relationship Page XML with Caretaker Indicator

The caretaker indicator is the only question that can be added directly to the relationship page. Questions regarding other attributes of a Relationship entity must be added to clusters that have been added to the relationship page. For example:

```

<relationship-page id="RelationshipPage" show-person-tabs="true">
  <title id="RelationshipPage.Title">
    <![CDATA[Household Relationships]]>
  </title>
  <question id="caretakerInd">
    <label id="CaretakerInd.Label">
      <![CDATA[Is this a non-parent caretaker relationship?]]>
    </label>
  </question>
  <cluster>
    <question id="startDate" mandatory="true">
      <label id="StartDate.Label">
        <![CDATA[Relationship Start Date:]]>
      </label>
    </question>
  </cluster>
</relationship-page>

```

Figure 21: Relationship Page XML with Relationship Attributes

The clusters added to a relationship page will be repeated for each relationship to be captured.

Display Text can be added directly to a relationship page. This text will be displayed once on the page regardless of the number of relationship captured. The display text will be displayed at the top of page, above the relationships.

A summary of the relationships captured for the household can easily be included on a summary page by adding a relationship-summary-list element.

The relationships list will always contain at least three columns to display the two people involved in the relationship and the relationship type. If you have captured the caretaker indicator on your relationships page, or you have captured other information about relationships, columns may be added to the relation summary list to display this information. For example:

```

<relationship-summary-list>
  <title id="RelationshipSummaryPage.Title">
    <![CDATA[Person Relationships Summary]]>
  </title>
  <description id="PersonRelationshipSummaryPage.Description">
    <![CDATA[Person Relationship Summary Details]]>
  </description>
  <column id="caretakerInd">
    <title id="CaretakerInd.Title">
      <![CDATA[Caretaker?]]>
    </title>
  </column>
  <column id="startDate">
    <title id="StartDate.Title">
      <![CDATA[Start Date]]>
    </title>
  </column>
  <edit-link start-page="RelationshipPage" />
</relationship-summary-list>

```

Figure 22: Relationship Summary List XML

The edit-link element can be used in a relationship-summary-list to edit relationships in the same way the edit-link element works in a list ([“Editing Records in Lists” on page 56](#))

Customizing Links on Summary Pages

Summary pages are designed to give users feedback on the answers they have given to the questions asked so far in a section. They can also be used to provide a means for users to change the information they have entered so far. This section describes how to customize links on summary pages enabling users to add, edit, and remove summary data.

Editing Information in Clusters

Any cluster of answers on a summary page can have an associated Edit link which appears to the right-hand side of the cluster title.

This link is created by adding an edit-link element to the cluster as in the example below:

```

<cluster>
  <title id="DetailsCluster.Title">
    <![CDATA[Person Details]]>
  </title>
  <edit-link start-page="AboutYouPage" />
  <layout>
    <num-cols>2</num-cols>
  </layout>
  <question id="firstName">
    <label id="FirstName.Label">
      <![CDATA[First Name:]]>
    </label>
  </question>

```

Figure 23: XML for Editable Cluster

The start-page attribute can be used to specify which page to link to (typically the same page on which they entered the information in the first place), using the id of the appropriate question-page element. This page should be in the same section as the summary page, otherwise a validation error will be thrown.

Once the user clicks on the Edit link from a summary page, the user is taken to the specified start page so that he or she can edit the data on it. Where the user goes from there depends on whether the user actually does anything on the page (i.e., change any answers) and what the implications of these changes might be. The options for what happens when a user clicks the next button are as follows:

- If the user made no changes to the answers previously captured on the page, then he or she will be taken straight back to the summary page for the section (where the user came from)
- If the user does make changes, then the system will check to see whether any of the answers on this page are used as part of a condition or loop expression anywhere in this script:
 - If not, then the user is returned to the summary page as above.
 - If so, then the Next button behaves as it normally would on a page and takes the user through the remaining pages in the section, evaluating conditions and loop expressions as it goes. As described

earlier, all enabled sections beyond the section containing the first page in which the changed answers are referenced will be disabled at this point.

The optional `show-page-elements` attribute can be used to specify a list of clusters that should be displayed on the specified start page. If the attribute is not specified, the clusters on the page are rendered normally. Conditional clusters which are listed in the `show-page-elements` attribute behave as follows.

- Conditional clusters which are controlled by answers to questions on previous pages are shown if the expression controlling the condition evaluates to true.
- Dynamically conditional clusters where the elements contained in the expression are in potentially visible clusters on the start page are displayed if the expression evaluates to true. These clusters may be dynamically hidden or displayed as questions on the page are answered.
- Dynamically conditional clusters where the elements contained in the expression are not in potentially visible clusters on the start page are displayed if the expression evaluates to true. These clusters will not be dynamically hidden or displayed based on the users input.

Editing Records in Lists

The edit-link element can also be added to a list in much the same way as for a cluster, except this time it will result in an edit link per row in the list.

Typically the start and end pages specified for an edit link in a list will relate to pages within the loop used to capture the information displayed in the list. If so, then the user will be taken to the iteration of that loop used to capture this particular record and with all the information previously captured filled in. The loop will progress through to the end page, if specified.

As with the Edit link on clusters, what happens next depends on what the user changes and whether it has an impact on the flow of the script thereafter. If the user makes a change to an answer which is used in a condition or expression further on in the script, then the Next button will behave as if it were the first time through the loop: the user will be taken through all the subsequent pages. Otherwise they will be taken straight back to the summary page.

Deleting Records from Lists

Lists can also have Delete links which allow records to be removed from lists. When a user clicks on the Delete link, a dialog will pop up which asks the user to confirm that he or she wishes to delete this record or not.

If the user chooses 'OK' in this dialog, then the entity related to this record will be deleted from the Datastore, as will any of its child entities. When an entity is deleted, any other entities that were created on the same page are also deleted. If entities should have independent existence, they should be created on separate pages.

Also upon clicking Delete, the page that created the entity and pages that created its child entities will be removed from the list of visited pages, thereby not appearing as the user navigates through the script. In addition, pages that reference the entity or any of its child entities will be removed from the list of visited pages provided that other entities are not created or referenced on the page.

If the user chooses Cancel, then the dialog will close and nothing will be deleted.

Adding Records to Lists

Lists can also have links to add new records to them.

Not surprisingly, this is created by adding an add-link element to the list and specifying the start and end pages to take the user to when creating the new record. While the start page for an add link will almost certainly be part of the loop used to populate the list in the first place, the end page may not necessarily be. This is because you may want to force the user to go through some extra pages after creating the record to ensure that all the other information entered so far is up-to-date. An example of this would be when adding people to a list in situations where the user has already captured relationships for the existing people. Once the new person has been added, the user should be taken through the relationships pages (which typically come after the loop), so the end page for the add link can be set accordingly.

Configuring IEG

You can configure the layout and presentation of IEG pages by using the `layout` element and using configuration properties. You can also configure the integration of the IEG Player into an application in both tab and modal contexts. For the IBM Cúram Universal Access (New) application, forms that are implemented in IEG forms are now rendered in the browser by IEG React components from the design system, which replace the IEG player, and in some cases, the IEG behavior has changed.

For information about IEG that is specific to the IBM Cúram Universal Access (New) application, see [Configuring IEG for Universal Access \(New\)](#).

Using the layout element to customize IEG pages

Use the layout element to customize the appearance of clusters and questions.

Using the Layout Element to Change the Appearance of Clusters

The default appearance of a cluster is to display all the questions it contains in one column, with the questions displayed in the order in which they are defined in the script and with the label and input field or value each taking 50% percent of the available width.

To change this default appearance, a layout element can be added to the cluster. For example, the following cluster has no layout element and therefore adheres to the default behavior:

```
<cluster>
  <title id="DetailsCluster.Title">
    <![CDATA[Personal Details]]>
  </title>
  <description id="DetailsCluster.Description">
    <![CDATA[Enter your details here]]>
  </description>
  <question id="firstName" mandatory="true">
    <label id="FirstName.Label">
      <![CDATA[First Name:]]>
    </label>
  </question>
  <question id="middleName">
    <label id="MiddleName.Label">
      <![CDATA[Middle Name:]]>
    </label>
  </question>
  <question id="lastName">
    <label id="lastName.Label">
      <![CDATA[Last Name:]]>
    </label>
  </question>
  <question id="gender" mandatory="true">
    <label id="Gender.Label">
      <![CDATA[Gender:]]>
    </label>
  </question>
  <question id="dateOfBirth" mandatory="true">
    <label id="DateOfBirth.Label">
      <![CDATA[Date Of Birth:]]>
    </label>
  </question>
</cluster>
```

Figure 24: Cluster with No Layout

A layout element can be added which changes the label width to be 75% as follows:

```
<cluster>
  <layout>
    <label-width>75</label-width>
  </layout>
```

Figure 25: Layout with Label Width

A layout element can also be used to change the layout type and the number of columns, as follows:

```
<cluster>
  <layout>
    <type>compact-flow</type>
    <num-cols>3</num-cols>
  </layout>
</cluster>
```

Figure 26: Layout with Compact-flow and 3 Columns

The default width for clusters is 100% of the available space. It is possible to alter the width of clusters using the layout element, as shown below:

```
<cluster>
  <layout>
    <width>80</width>
  </layout>
</cluster>
```

Figure 27: Layout with Width for Cluster

Summary of Cluster Layout Options

The following list describes all the possible layout options that can be applied to clusters:

Name	Description
<i>type</i>	The type can be set to either 'flow' (the default) or 'compact-flow'. A cluster using 'flow' will lay its questions out from left to right, top to bottom, with the label always appearing to the left of the input control or value. The compact-flow behaves in much the same way but with the label displayed above the input control or value. This can allow you to fit more columns into a cluster (than if the labels and input controls were side-by-side).
<i>num-cols</i>	The number of columns, can be used to specify the number of question elements to layout across the cluster. The default number of elements to display in a column is 1. This attribute can also be used to specify the number of columns in which to display the options for a multiple-select question.
<i>width</i>	The width of the cluster can be used to alter the percentage of the available width of the page which this cluster uses from its default of 100%.
<i>label-width</i>	The label width can be used to alter the percentage of a column width given to the label of a question (and by implication, the percentage given to the input control or value) from its default of 50%.
<i>label-alignment</i>	The label alignment can be used to alter the alignment of the text within the question labels in this cluster. The default is to align text to the right (beside the input control or value) and the other options available are left and center.

By combining these options and varying the number of clusters on your page, you can exercise a high degree of control over what the user ultimately sees, with the goal of presenting a friendly, intuitive user interface.

Using the Layout Element to Change the Appearance of Multiple-Choice Questions

The default appearance of a multiple-choice question (for example, a question with a codetable data type) is to display the options in a dropdown box. By using the layout element as a child of the question, it is possible to display the question in several formats.

If the layout element contains the *num-cols* element, the options will be displayed in the specified number of columns:

Layout with Number of Columns

```
<layout>
  <num-cols>4</num-cols>
</layout>
```

If the layout element contains the *num-rows* element, the options will be displayed in a scrollable list box with the specified number of rows visible at one time:

Layout with Number of Rows

```
<layout>
  <num-rows>6</num-rows>
</layout>
```

Note that when *num-rows* and *num-cols* are both used, *num-cols* takes precedence.

If the layout element contains the *autosize* element, and that contains a text value of 'true', the options will be arranged in the number of columns specified by the *multiselect.layout.optimum.columns* configuration property.

If the layout element contains the *input-alignment* element, and that contains a text value of 'left' the checkboxes or radio buttons will be displayed to the left of the text for each option. Conversely, if the text value is 'right', the checkboxes or radio buttons will be displayed to the right of the text for each option:

```
<layout>
  <input-alignment>right</input-alignment>
  <num-rows>6</num-rows>
</layout>
```

Figure 28: Layout with Input Alignment Set to Right

Summary of Multiple-Choice Question Layout Options

The following list describes the layout options that pertain specifically to codetable-type questions.

- The number of columns, *num-cols*, specifies the number of columns in which to display the options available for a multiple-choice question. The number of rows is implicit once the number of codetable items is known.
- The number of rows, *num-rows*, specifies the number of rows in which to display the options available for a multiple-choice question. When this attribute is used, a scrollable list box with the specified number of rows is displayed. The *num-rows* attribute is ignored if *num-cols* is also present.
- *autosize* can be used to obtain the configured default for number of columns to display. This default number is defined in the *multiselect.layout.optimum.columns* property of *ieg-config.properties*.
- The input alignment, *input-alignment*, can be used to align the input field to the left or right of its associated label. This defaults to left in left-to-right configuration.

Using the Container Element to Control Layout of Questions and Columns

The container element can be used in two cases:

- to group questions within a cluster: Display multiple questions alongside each other and use one label.
- to group multiple columns in a list: Display the answer to multiple questions in one column by grouping multiple columns in a container element.

The questions or columns within a container can be visually separated using dividers. Each successive questions or columns can be separated by one divider element, but the divider can also be placed before the first question and after the last one.

Using the Container Element in a Cluster

When the container element is used in a cluster to group questions, the questions will be displayed alongside each other and the title for the container will be displayed alongside the grouped questions instead of the individual question labels.

A container can be used in a cluster as follows:

```

<container>
  <title id="ContactNumber.Title">Contact number:</title>

  <question id="countryCode">
    <layout>
      <width>15</width>
    </layout>

    <label id="CountryCode.Label">Country code</label>
  </question>

  <question id="areaCode">
    <layout>
      <width>20</width>
    </layout>

    <label id="AreaCode.Label">Area Code</label>
  </question>

  <question id="phoneNumber">
    <layout>
      <width>40</width>
    </layout>

    <label id="PhoneNumber.Label">Phone number</label>
  </question>
</container>

```

Figure 29: Cluster Container XML

Note that when questions are wrapped in a container, although the container title is displayed instead of the individual question labels, a label should still be included for each question. The question label will be displayed as a tool tip, in the example above, if the mouse is hovering over the second text field in the phone number the label 'area code' will be displayed as the tool tip.

Using the Container Element in a List

When a container is used within a list to group columns, the grouped columns will be displayed in one column with the container title displayed as the heading for that column.

A container can be used in a list as follows:

```

<list entity="Person" criteria="isPrimary==false"
  show-icons="true">
  ...
  <container>
    <title id="FullName.Title">Full Name</title>
    <column id="firstName">
      <title id="FirstName.Title">First Name</title>
    </column>
    <column id="lastName">
      <title id="LastName.Title">Last Name</title>
    </column>
  </container>
  ...
</list>

```

Figure 30: List Container XML

It is possible to apply a width value to a container, as shown below where the container takes up 60% of the width available to the list:

```

<list entity="Person" criteria="isPrimary==false"
  show-icons="true">
  ...
  <container>
    <layout>
      <width>60</width>
    </layout>
    <title id="FullName.Title">Full Name:</title>
    <question id="firstName" mandatory="false"
      control-question="false" multi-select="false">
      <label id="FirstName.Label">First Name:</label>
    </question>
    <question id="lastName" mandatory="false"
      control-question="false" multi-select="false">
      <label id="lastName.Label">Last Name:</label>
    </question>
  </container>
  ...
</list>

```

Figure 31: List Container with Width XML

Using Dividers

When the container element is used in a cluster to group questions, the questions will be displayed alongside each other and the title for the container will be displayed alongside the grouped questions instead of the individual question labels.

A container can be used in a cluster as follows:

Figure 1. Container and Dividers XML

```

<container>
  <title id="ContactNumber.Title">Contact number:</title>
  <divider id="CountryCode.Divider">+</divider>
  <question id="countryCode">
    <layout>
      <width>15</width>
    </layout>
    <label id="CountryCode.Label">Country code</label>
  </question>
  <divider id="AreaCode.Divider">-</divider>
  <question id="areaCode">
    <layout>
      <width>20</width>
    </layout>
    <label id="AreaCode.Label">Area Code</label>
  </question>
  <divider id="PhoneNumber.Divider">-</divider>
  <question id="phoneNumber">
    <layout>
      <width>40</width>
    </layout>
    <label id="PhoneNumber.Label">Phone number</label>
  </question>
</container>

```

If a white space needs to be added at the start or at the end of a divider string, this will need to be placed within tags (e.g. " - ") as properties strip leading and trailing spaces. In the absence of divider elements, this will continue to behave as before i.e. a white space will be added between questions when they are not editable. The width set on each question should be adjusted depending on the length of the dividers.

Using configuration properties to customize IEG pages

You can customize the style, content, and layout of elements on IEG pages by using configuration properties.

Beyond the options available in the `Layout` element in scripts, IEG provides extensive customization options for the look-and-feel of screens. This customization is done by setting application properties rather than modifying CSS, so reduces the need for web design knowledge.

Default values of the properties that are used for configuring the layout of IEG Pages are set in `ieg-config.properties`. They can be customized by specifying a new configuration file for a script on the `config-properties` attribute of the `ieg-script` element, see [“ieg-script” on page 40](#). This file contains the properties and values that differ from `ieg-config.properties`.

Changing the Look-and-Feel of the Pages

There are configuration properties that allow the look-and-feel (the size of elements, the color scheme, the images used, and so forth.) of each part of the page to be modified. These are outlined in the tables below, grouping together properties that affect particular items on the screen.

Configuring the Page Banner

The following configuration properties can be used to modify the look and feel of the page banner:

Property	Description
<code>banner.show</code>	Boolean value that hides the page banner if set to 'false'. By default, the banner is shown.
<code>banner.systitle</code>	The 'system title' text to be displayed next to the logo.
<code>banner.apptitle</code>	The 'application title' text to be displayed next to the logo.
<code>banner.background.color</code>	The background color for the banner panel.
<code>banner.background.image</code>	The background image for the banner panel.
<code>banner.border.color</code>	The color to use for the panel border.
<code>banner.text.color</code>	The text color in the banner.
<code>banner.text.weight</code>	The text weight in the banner.
<code>banner.link.print</code>	The text to be displayed for the 'print' link.
<code>banner.link.print.desc</code>	The description/alt text for the 'print' link.
<code>logo</code>	The logo image.
<code>logo.alt</code>	The alt text for the logo image.
<code>print.logo</code>	The image to use for displaying the print option.
<code>print.logo.hover</code>	The image to use when hovering over the print option.

Table 76: Page Banner Configuration Properties (continued)

Property	Description
print.logo.click	The image to use when clicking on the print option.
banner.link.button	Optional property that controls the display of a button/link in the page banner. This property contains the label of the link. By default the button is not displayed. If the button is present it will be displayed to the right of the print button.
banner.link.button.url	The URL to be associated with the banner button. It can be either absolute or relative.
banner.link.button.desc	The description/alt text for the banner button link.
banner.button.logo	The image to use for displaying the banner button.
banner.button.logo.hover	The image to use when hovering over the banner button.
banner.button.logo.click	The image to use when clicking on the banner button.
menu.item.color	The color of the text for the print link.
notes.button.text	The text to display for the Notes button.
notes.button.hide.image	The image to use for the 'close notes panel' option.
notes.button.hide.selected.image	The image to use when the 'close notes panel' option is clicked.
notes.button.show.image	The image to use for the 'show notes panel' option.
notes.button.show.selected.image	The image to use when the 'show notes panel' option is clicked.
notes.panel.title.text	The title text for the Notes panel.

Configuring the Progress Panel

The configuration properties that can be used to modify the look and feel of the progress panel are listed in the following table:

Table 77: Progress Panel Configuration Properties

Property	Description
progress.panel.border.color	The color for the border of the progress panel.
progress.panel.background.color	The background color of the progress panel.

Property	Description
progress.panel.background.image	The background image of the progress panel.
progress.bar.border.color	The color for the border of the progress bar.
progress.bar.background.color	The background color of the progress bar.
progress.bar.text	The text to use in the progress bar, following the percentage value.
progress.bar.text.color	The color of the progress bar text.
progress.total.bar.background.color	The background color of the 'total' section of the progress bar.
progress.total.bar.background.image	The background image of the 'total' section of the progress bar.
progress.total.bar.border.color	The color of the border of the 'total' section of the progress bar.
progress.completed.bar.background.color	The background color of the 'completed' section of the progress bar.
progress.completed.bar.background.image	The background image of the 'completed' section of the progress bar.
progress.completed.bar.border.color	The color of the border of the 'completed' section of the progress bar.
progress.pagetext.color	The color of the text specifying the current page title in progress panel.

Configuring the Persons Tab Panel

The configuration properties that can be used to modify the look and feel of the progress panel are listed in the following table:

Property	Description
persontabs.background.color	The background color for the person tabs panel.
persontabs.background.image	The background image for the person tabs panel.
persontabs.border.color	The color of the person tabs border.
persontabs.max.word.size	The maximum number of characters in the name displayed in the person tabs before the string is truncated.
persontabs.tab.width	The width of each person tab, in pixels.

Table 78: Person Tabs Panel Configuration Properties (continued)

Property	Description
persontabs.hide.panel.if.one.person	Indicates if the persons tab panel should be hidden if there is only one person to be displayed.

Configuring the Action Links

The Action Links are:

- The Edit link on a cluster
- The Add link on a list
- The Edit and Delete links on a row in a list

The configuration properties that can be used to modify the look and feel of the action links are listed in the following table:

Table 79: Action Links Configuration Properties

Property	Description
action.edit	The text to display for edit links.
action.desc.edit	The description/alt text for edit links.
action.desc.cxt.edit	The description/alt text for edit links, with a parameter to identify the entity to edit.
list.action.label	The text to display at the head of the action links column in a list.
list.action.add	The text to display for Add links.
list.action.desc.add	The description/alt text for Add links.
list.action.desc.add.select	The title text of the dropdown list used for Add links.
list.action.edit	The text to display for Edit links.
list.action.desc.edit	The description/alt text for Edit links.
list.action.delete	The text to display for Delete links.
list.action.desc.delete	The description/alt text for Delete links.
list.action.delete.confirm	The text to display in the confirmation dialog box for deleting items.

Configuring Relationship Pages

You can configure all of the standard text on relationship pages by using the properties.

Table 80: Relationship Page Configuration Properties

Property	Description
relationship.capture.item.1	The first item to appear in the relationship. Set to the 'from' or subject person.

Table 80: Relationship Page Configuration Properties (continued)

Property	Description
relationship.capture.item.2	The second item to appear in the relationship. Set to be the input control.
relationship.capture.item.3	The third item to appear in the relationship. Set to the 'to' or object person.
relationship.type.domain.name	The domain to use to populate the input control for a relationship.
relationship.from.label	The text to display for the heading of the subject column in a relationship summary list.
relationship.type.label	The text to display for the heading of the type column in a relationship summary list.
relationship.to.label	The text to display for the heading of the object column in a relationship summary list.
relationship.action.label	The text to display for the heading of the action column in a relationship summary list.
relationship.dropdown.message	The title attribute of the drop-down list for relationship questions.
relationship.question.label	This property is applicable to the IBM® Cúram Universal Access (New) application only. The title attribute of the question list for relationship.
relationship.type.date.label	This property is applicable to the IBM® Cúram Universal Access (New) application only. The text for the attribute type and relationship start date at relationship-summary-page.

Configuring the Help Panel

The configuration options that can be used to modify the look and feel of the help panel are listed in the following table:

Table 81: Help Panel Configuration Properties

Property	Description
help.link.moreinfo	Specifies the text to be displayed for the 'Help' link.
help.link.moreinfo.hide	Indicates if there should be descriptive text displayed beside the cluster 'Help' link. By default, this is true.
help.link.desc.moreinfo	Used as the title text for the 'Help' link.
help.link.desc.cxt.moreinfo	Used as the title text for the 'Help' link.

Table 81: Help Panel Configuration Properties (continued)

Property	Description
help.panel.background.color	The background color for the expanded help panel.
help.panel.color	The color of texts in the expanded help panel.
help.panel.heading.color	The color of the text for the help panel headings.
help.bottompanel.background.color	The background color for the bottom of the help panel, which contains the 'Close' link.
help.bottompanel.background.image	The background image for the bottom of the help panel, which contains the 'Close' link.
help.link.image.open	The image used as 'open' icon for expanding the help panel.
help.link.image.selected	The image used to indicate that the 'open' icon for expanding the help panel has been selected.
help.link.policy	The text to display for policy links in the help panel.
help.link.desc.policy	The description text for policy links in the help panel.
help.link.legislation	The text to display for legislation links in the help panel.
help.link.desc.legislation	The description text for legislation links in the help panel.
help.panel.close	The text to display for the 'close' link that hides the help panel.
help.close.title	The title text for the 'close' link that hides the help panel.
help.link.image.close	The image to use for the 'close' link that hides the help panel.
help.link.image.close.roll	The roll-over image to use for the 'close' link that hides the help panel.
dialog.help.link.image.close	The image to use for the link that closes help modal dialogs.
dialog.help.link.image.close.roll	The roll-over image to use for the link that closes help modal dialogs.
compile.cluster.help	<p>A boolean property that indicates if the cluster help panel should compile the help texts of the questions in the cluster. By default, this is true.</p> <p>This property is not supported for Universal Access (New).</p>

Table 81: Help Panel Configuration Properties (continued)

Property	Description
show.cluster.help.dialog	A boolean property that indicates if cluster help should be displayed in a modal dialog (similar to field level help). By default, this is false.

Configuring the Page Title Panel

There are several configuration properties that can be used to modify the look and feel of the page title panel. These are listed in the following table:

Table 82: Page Title Panel Configuration

Property	Description
pagetitle.border.color	The border color for the page title panel.
pagetitle.background.color	The background color for the page title panel.
pagetitle.background.image	The background image for the page title panel.
pagetitle.color	The text color of the page title.
pagetitle.description.color	The text color of the page description.
pagetitle.imagecell.width	The width of the cell that contains the page title image.

Configuring the Navigation Panel

The configuration properties that can be used to modify the look and feel of the navigation panel are listed in the following table:

Table 83: Navigation Panel Configuration

Property	Description
navpanel.button.background.image	Background image for the navigation buttons.
navpanel.button.background.image.hover	The roll-over image used for the navigation buttons.
navpanel.button.background.image.corner	The image used for the corners of the buttons.
navpanel.button.background.image.corner.hover	The roll-over image used for the corners of the buttons.
navpanel.button.background.color	The background color for the navigation buttons.
navpanel.button.color	The text color for the navigation buttons.
navpanel.button.selected.background.image	The background image for selected navigation buttons.

Table 83: Navigation Panel Configuration (continued)

Property	Description
navpanel.button.selected.background.color	The background color for selected navigation buttons.
navpanel.button.active.background.image	The background image for active navigation buttons.
navpanel.button.active.background.color	The background color for active navigation buttons.
navpanel.button.back.text	The text to display for the Back button.
navpanel.button.exit.text	The text to display for the Exit button.
navpanel.button.quit.text	The text to display for the Save and Exit button.
navpanel.button.next.text	The text to display for the Next button.
navpanel.button.back.desc	The description/alt text for the Back button.
navpanel.button.exit.desc	The description/alt text for the Exit button.
navpanel.button.quit.desc	The description/alt text for the Save and Exit button.
navpanel.button.next.desc	The description/alt text for the Next button.

Configuring Lists

The configuration properties that can be used to modify the look and feel of Lists are listed in the following table:

Table 84: List Configuration

Property	Description
style.list.as.cluster	A boolean property that indicates if lists should be styled similar to clusters. By default, this is false. If set to true, this will have an effect on the list header and the list body. This property affects the following list types... <ul style="list-style-type: none"> • Lists specified on summary pages • Lists specified on non-summary pages • Nested Lists • Relationship Summary Lists
list.title.color	The color of list titles when lists are styled similar to clusters.
list.title.border.color	The color of list title borders when lists are styled similar to clusters.

<i>Table 84: List Configuration (continued)</i>	
Property	Description
list.link.add.show	A boolean property that controls the display of an add icon alongside the add link for a list.
list.link.add.image.open	The icon used for the add link.
list.link.add.image.roll	The icon used for the add link on mouse rollover.

Other Page Layout Configurations

There are several other configurable items on an IEG page. Properties relating to these are listed in the following table:

<i>Table 85: Other Page Layout Configuration Properties</i>	
Property	Description
font.family	The font family to use on pages.
icon.mandatory	The icon used to indicate a mandatory field.
icon.mandatory.alt	The alt text for the mandatory icon.
cluster.title.color	The text color of cluster titles.
cluster.title.border.color	The border color of cluster titles.
messages.panel.color	The text color for the messages panel.
messages.panel.border.color	The border color for the messages panel.
messages.panel.background.color	The background color for the messages panel.
messages.panel.description	The description text for the messages panel.
messages.panel.reset.script.message	The text to be displayed in the messages panel when the script has been reset.
list.no.data.text	The message displayed when a list does not have any data (i.e. zero rows)
dropdown.list.blank.entry.description	The text displayed for the blank entry in a dropdown list.
dropdown.list.description	The title text of dropdown lists used for codetable-type questions.
messages.highlight.color	The color used to highlight fields with related validation messages.
messages.label.color	The text color for validation message labels.
messages.label.weight	The text weight for validation message labels.
true	Translation for boolean value.
false	Translation for boolean value.

Table 85: Other Page Layout Configuration Properties (continued)

Property	Description
calendar.today	The text representing today in the calendar widget.
calendar.icon.alt	The alt text for the calendar icon.
multiselect.layout.optimum.columns	The optimum number of columns to use for multi-select questions.
multiselect.mandatory.message	The message to display for mandatory validations of multi-select questions.
listquestion.mandatory.message	The message to display for mandatory validations of list questions.
checkbox.mandatory.message	The message to display for mandatory validations of multi-select questions using checkboxes.
radioButton.mandatory.message	The message to display for mandatory validations of single-select questions using radio buttons.
navigation.error.message	The message to display when unsupported use of the browser back button is detected.
navigation.link.message	The text to display for the link to resume script execution after the 'browser back button' message is displayed.
navigation.update.message.display	Indicates if a message should be displayed to users when some section navigation has been disabled.
navigation.update.message	Message to display when some navigation options have been disabled.
session.timeout.error.message	Session timeout message.
session.timeout.link.message	Text to display for the 'continue' link after a session timeout.
session.timeout.link.url	Text to display for the resume link after a session timeout.
item.itemLabel.maxLength	The maximum length for a list question item label. If the label length exceeds this value, it will be truncated to this length, including ellipsis to indicate the truncation.
matrix.image.selected	Question matrix summary image alt-text for a selected item.
matrix.image.notSelected	Question matrix summary image alt-text for an unselected item.
matrix.selected	Question matrix summary image for a selected item.

Table 85: Other Page Layout Configuration Properties (continued)

Property	Description
matrix.unselected	Question matrix summary image for an unselected item.
policy.logo	The image to display for policy links.
policy.logo.hover	The image to display for policy links when hovered over.
policy.logo.click	The image to display for policy links when clicked.
legislation.logo	The image to display for legislation links.
legislation.logo.hover	The image to display for legislation links when hovered over.
legislation.logo.click	The image to display for legislation links when clicked.
person.adultAge	The age at which the adult image should be displayed for a person in the persons tab, list questions, relationship questions, etc.
date.field.width	The width, as a percentage of available space, to be used for date input fields. This can be overridden for specific questions by setting a width on the question layout. By default, this is 60%.
confirm.delete.title	The title of the delete confirmation dialog.
confirm.delete.message	The confirmation text in the delete confirmation dialog.
confirm.delete.ok.button	The confirmation button text in the delete confirmation dialog.
confirm.delete.cancel.button	The cancel button text in the delete confirmation dialog.
label.align	The standard alignment of label texts. By default, this is left.
cluster.link.edit.show	A boolean property that controls the display of an edit icon alongside the edit link for a cluster.
cluster.link.edit.image.open	The icon used for the edit link.
cluster.link.edit.image.roll	The icon used for the edit link on mouse rollover.
link.skip	A skip link is a hidden link which allows a user to skip to the main content area of an IEG page. This property allows the text associated with this link to be configured.

Table 85: Other Page Layout Configuration Properties (continued)

Property	Description
transitions.perform	A boolean property that controls the animation of transitions. When the main content area changes (by clicking a navigation button, hitting a link, ...) the new content and the other panels (sections, page title, progress bar...) will be updated using animations. By default, this is true. This is not applicable when running in a modal.
informational.message.text	The default text to display on each informational message that does not have custom text defined through the message child element.
informational.message.external.image	The default image to display on each informational message that does not have a custom image defined through the image attribute. The default external image is displayed when an IEG script displays an informational message in the Universal Access user interface.
informational.message.internal.image	The default image to display on each informational message that does not have a custom image defined through the image attribute. The default internal image is displayed when an IEG script displays an informational message in the IBM Cúram Social Program Management user interface.

Related reference

[informational-message](#)

Use an `informational-message` display element to display an informational message within the heading of either a cluster, a list, or a relationship summary list.

Changing the Look-and-Feel of Universal Access Scripts

IBM Cúram Universal Access (UA) is a web-based self service application that allows citizens to interact with the agency. As the scripts provided through UA are designed for use by citizens as opposed to agency workers, they have a distinct look and feel.

For information about IEG that is specific to the IBM Cúram Universal Access (New) application, see [Configuring IEG for Universal Access \(New\)](#).

Table 86: Configuration Properties for Cluster Help Panel

Property	Description
help.panel.external.background.color	The background color for the expanded help panel.
help.panel.external.heading.color	The color of the text for the help panel headings.
help.panel.external.heading.weight	The weight of the text for the help panel headings.

Table 86: Configuration Properties for Cluster Help Panel (continued)

Property	Description
help.panel.external.color	The color of the text in the expanded help panel.
help.panel.external.border.bottom	The style and color of the bottom border of the help panel
help.external.links.color	The text color of the policy and legislation links.
help.external.links.weight	The text weight of the policy and legislation links.
help.link.close	The link text displayed to close the help panel.
help.panel.external.description.color	The text color of the help panel description.
help.panel.external.description.link.color	The link color of the help panel description.

Table 87: Configuration Properties for Section Panel

Property	Description
sectionspanel.width	The width of the section panel.
sectionspanel.external.selected.color	The text color for the currently selected section.
sectionspanel.external.disabled.color	The color of the disabled sections.
sectionspanel.external.enabled.color	The color of enabled sections.
sectionspanel.external.enabled.background.color	The background color for the enabled sections.
sectionspanel.external.selected.background.color	The background color for the currently selected section.
sectionspanel.external.current	Image for the current section.
sectionspanel.external.disabled	Image for a disabled section.
sectionspanel.external.enabled	Image for an enabled section.
sectionspanel.external.enabled.roll	The image for rollover events on enabled sections.
sectionspanel.external.background.color	The background color for the sections panel.

Table 88: Configuration Properties for Filed Level Help

Property	Description
help.external.link.image.open	The image used as 'open' icon for expanding the help panel.
help.external.link.image.roll	The help image displayed on a mouse over event.

Table 88: Configuration Properties for Filed Level Help (continued)

Property	Description
help.external.panel.heading.color	The color of the text for the help panel headings.
help.external.links.color	The text color of the policy and legislation links.
help.external.links.weight	The text weight of the policy and legislation links.

Table 89: Configuration Properties for Page Buttons

Property	Description
navpanel.external.button.font.weight	The weight of the text used in navigation buttons.
navpanel.external.button.font.family	The font family of the text used in navigation buttons.
navpanel.external.button.font.size	The size of the text used in navigation buttons.

Table 90: Configuration Properties for Relationship Page

Property	Description
relationship.external.caretaker.font.color	The caretaker question text color.
relationship.external.caretaker.font.weight	The caretaker question text weight.

Table 91: Configuration Properties for Question Pages

Property	Description
pagedescription.external.font.color	The text color of the page description.
pagedescription.external.font.weight	The text weight of the page description.
pagetitle.external.font.color	The text color of the page title.
pagetitle.external.font.weight	The text weight of the page title.
pagetitle.external.background.color	The background color for the page title panel.
pagetitle.external.border.style	The border style of the page heading.
pagetitle.external.border.bottom.color	The border color of the page heading.
pagetitle.external.border.width	The border width of the page heading.

Table 92: Configuration Properties for Clusters

Property	Description
cluster.external.title.color	The text color of cluster titles
cluster.external.title.weight	The text weight of cluster titles
cluster.external.title.border.color	The border color of cluster titles
cluster.external.description.color	The text color of cluster descriptions

Table 92: Configuration Properties for Clusters (continued)

Property	Description
cluster.external.description.weight	The text weight of cluster descriptions
cluster.external.background.color	The background color of clusters
cluster.external.border.color	The border color of clusters
cluster.external.border.style	The border style of clusters

Table 93: Other Configuration Properties

Property	Description
external.field.label.text.color	The font color for question labels
external.field.label.text.weigh	The font weight for question labels
radiobutton.icon.external	Image for a disabled and selected radio button

Configuring the Layout of the Sections Panel

By default, the sections panel in IEG is displayed in a vertical alignment, from the first section at the top to the last section at the bottom.

By setting the configuration property `sectionspanel.style=horizontal`, it is possible to display the panel horizontally along the top of the page instead.

In addition, configuration properties can be set to control the display of the sections panel in both formats. The following properties are applicable to both horizontal and vertical layouts:

Table 94: Sections Panel Configuration Properties (Universal)

Property	Description
sectionspanel.border.color	The border color of the sections panel.
sectionspanel.background.color	The background color for the sections panel.
sectionspanel.background.image	The background image for the sections panel.
sectionspanel.selected.color	The text color for the currently selected section.
sectionspanel.selected.border.color	The border color of the currently selected section.
sectionspanel.selected.background.color	The background color for the currently selected section.
sectionspanel.selected.background.image	The background image for the currently selected section.
sectionspanel.enabled.border.color	The border color for enabled sections.
sectionspanel.enabled.background.color	The background color for the enabled sections.
sectionspanel.enabled.background.image	The background image for the enabled sections.

Table 94: Sections Panel Configuration Properties (Universal) (continued)

Property	Description
sectionspanel.desc.prev	The description/alt text for a completed section. The parameter is the section title.
sectionspanel.desc.current	The description/alt text for the current section. The parameter is the section title.

The configuration properties that apply only to the horizontal sections panel are outlined below:

Table 95: Sections Panel Configuration Properties (Horizontal)

Property	Description
sectionspanel.horizontal.max.display	The maximum number of sections to display in the panel at any one time.
sectionspanel.horizontal.arrow.move.amount	The number of sections to move on the panel when navigating using the left and right arrows.
sectionspanel.horizontal.truncate.limit	The number of characters displayed for the section title in the panel before the string is truncated.
sectionspanel.horizontal.text.top.margin	Used to change the position of the text in the horizontal panel.
sectionspanel.horizontal.text.size	The size of the text displayed in the horizontal panel.
sectionspanel.horizontal.background.color	The background color of the horizontal sections panel.
sectionspanel.horizontal.border.color	The border color of the horizontal sections panel.
sectionspanel.horizontal.enabled.color	The background color for the enabled sections in the horizontal panel.
sectionspanel.horizontal.enabled.text.color	The text color for the enabled sections in the horizontal panel.
sectionspanel.horizontal.text.disabled.color	The text color for the disabled horizontal sections.
sectionspanel.horizontal.bar.text.color	The color of text in the bar that displays the current section title.
sectionspanel.horizontal.bar.text.color	The color of text in the bar that displays the current section title.
sectionspanel.horizontal.bar.background.color	The background color of the bar that displays the current section title.
sectionspanel.horizontal.bar.background.color	The background color of the bar that displays the current section title.
sectionspanel.horizontal.enabled.background.image	The background image for the enabled horizontal sections.

<i>Table 95: Sections Panel Configuration Properties (Horizontal) (continued)</i>	
Property	Description
sectionspanel.horizontal.box.height	The height of the individual section 'boxes'.
sectionspanel.horizontal.box.width	The width of the individual section 'boxes'.
sectionspanel.horizontal.back.arrow.image	An image used as the icon for the link to navigate through the sections panel.
sectionspanel.horizontal.next.arrow.image	An image used as the icon for the link to navigate through the sections panel.
sectionspanel.horizontal.back.arrow.title	The title/alt text for the icon link to navigate through the sections panel.
sectionspanel.horizontal.next.arrow.title	The title/alt text for the icon link to navigate through the sections panel.
sectionspanel.horizontal.arrow.width	The width of the divs holding the section navigation links/icons.

Configuring the Layout of the IEG Player in Modal Dialogs

IEG is most commonly used in conjunction with other screens which form a logical and seamless flow for an application. This flow may be launched from a given page and open a modal dialog instead of a new tab or link to a new page.

This is where the agency wants to indicate to the client that an application for a service or benefit has started.

There is a slightly different look and feel when screens and the IEG Player are opened in a modal. To accommodate for changes to the display of the IEG Player when running in a modal rather than in a tab, there are configuration properties set. Some examples of these are:

- Set the width and height of the modal dialog in which the player is displayed
- Hide the page title to create more real estate for questions in a modal dialog
- Anchor the navigation buttons in a panel at the bottom of the modal dialog

These properties are set in the configuration file associated with the script, and by default in `ieg-config.properties`. A full list is provided below:

<i>Table 96: Configuration Properties for IEG Player in a Modal Dialog</i>	
Property	Description
modal.anchor.nav.panel	Indicates to player whether the navigation panel should be anchored.
modal.hide.title.panel	Indicates to player whether the page title panel should be hidden.
modal.width	Sets the width of the modal dialog.
modal.height	Sets the height of the modal dialog.
modal.close.dialog.on.exit	Indicates to the player whether the modal dialog should be automatically dismissed when the Exit button is selected.

Table 96: Configuration Properties for IEG Player in a Modal Dialog (continued)

Property	Description
navpanel.modal.button.background.image	The background image for navigation buttons in a modal dialog.
navpanel.modal.button.right.corner.image	The background image for the corners of navigation buttons in a modal dialog.
navpanel.modal.button.wrapper.image	The wrapper image for navigation buttons in a modal dialog.

Configuring the IEG Player for High Contrast Mode in the Universal Access UI

The IBM Cúram Social Program Management user interface is a high contrast design by default. However, in the Universal Access user interface, the IEG Player supports a high contrast mode. To avail of the high contrast mode functionality, the high contrast user preference must be switched on.

For information about IEG that is specific to the IBM Cúram Universal Access (New) application, see [Configuring IEG for Universal Access \(New\)](#).

To support the high contrast mode feature, several configuration properties allow various aspects of the high contrast look and feel to be configured. You must add the properties into any custom IEG script configuration property files that are currently being used if a high contrast mode is required in the IEG Player in the Universal Access user interface. The Universal Access interface is also known as external mode.

The properties are set in the configuration file that is associated with the script, and by default in `ieg-config.properties`. The following table lists all the properties:

Table 97: Configuration Properties for IEG Player in a High Contrast Mode in the Universal Access interface

Property	Description
sectionspanel.external.hc.current	The high contrast icon for the currently selected section in the sections panel.
sectionspanel.external.hc.disabled	The high contrast icon for the disabled sections in the sections panel.
sectionspanel.external.hc.enabled	The high contrast icon for the enabled sections in the sections panel.
sectionspanel.external.hc.enabled.roll	The high contrast roll-over icon for the enabled sections in the sections panel.
sectionspanel.external.hc.disabled.color	The high contrast color of the text on disabled sections in the sections panel.
sectionspanel.external.hc.enabled.background.color	The high contrast color of the background on enabled sections in the sections panel.
pagetitle.external.hc.font.color	The high contrast font color for the page title panel.
cluster.external.title.hc.color	The high contrast text color of cluster titles.
messages.external.panel.error.icon.hc	The high contrast error icon used in the validations panel.
help.panel.color.external.hc	The high contrast color of text in the expanded help panel.

Table 97: Configuration Properties for IEG Player in a High Contrast Mode in the Universal Access interface (continued)

Property	Description
help.panel.heading.color.external.hc	The high contrast color of text used for the heading in help modal dialogs.
dialog.help.close.external.hc	The high contrast image used for the link that closes help modal dialogs.
list.external.link.add.image.open.hc	The high contrast icon used for the add link.
list.external.link.add.image.roll.hc	The high contrast roll-over icon used for the add link.
matrix.selected.external.hc	The high contrast question matrix summary icon used for a selected item.
matrix.unselected.external.hc	The high contrast question matrix summary icon used for an unselected item.

Configuration properties to customize IEG widgets

You can use the following configuration properties to modify the default behavior of IEG widgets.

Table 98: IEG Widget Configuration Properties

Property	Description
dropdown.list.blank.entry.description	Code table drop-down list blank entry description.
dropdown.list.description	Code table drop-down list description.
dropdown.expand	Cause all drop-down lists to expand on the initial mouse click.

IEG Administration

In IEG, a number of simple administration screens have been provided which allow you to access, download, edit, run and remove existing scripts and to upload new scripts into the system.

Listing all Scripts

To gain access to the IEG administration screens, you will need to log in as an admin user. Once logged in, you will see a section in your navigation panel called IEG and when you click on it you will see a menu for 'IEG' which contains a link called 'IEG Scripts'.

If you click on this, you will see a screen containing a list of all the IEG scripts currently in the system and various links to allow you to perform the following activities on these scripts.

Downloading an Existing Script

Each script listed on the page above has a link to let you download the xml for that script as it is currently stored on the system.

Clicking on this link will take you to another screen from which a link to the document itself will be available for you to either save or open. If you choose to save this document to your file system, you will then be able to modify the script and upload the updated script via the 'Import IEG Script' link.

Removing an Existing Script

Each script listed on the page above also has a 'Delete' link to let you remove this script from the system so that it is no longer available to run.

When you click on this link, a confirmation dialog will be displayed asking you to confirm that you want to remove this script from the system. Note that deleted scripts cannot be accessed again, so only use this in exceptional circumstances and consider downloading the latest version of the script first.

Running a Script

A 'Run' link is provided for each script listed on the page above. Using the 'Run' link, you can conduct test runs of your scripts. When you click on the 'Run' link, you will be presented with another screen where a schema from the Datastore may be selected against which execute the script.

This allows the IEG Engine know the structure and types of the data you intend to use when running the script. Any data elements you wish to save/retrieve from the Datastore as part of your script must be defined in this schema. If an appropriate schema is not available in the list, you should use the Datastore administration screens to upload one. Once you choose the appropriate schema, click on the 'run script' link to invoke the IEG Player to run your script.

Validating a Script

A 'Validate' link is provided for each script on the 'IEG Scripts' page. When this link is clicked, the Validate Script page will be displayed on which you must specify the data store schema to associate with the script.

Once the 'Validate' link on this page is selected, validation errors for the IEG script will be listed. If no validation errors are found, a message specifying 'The specified IEG script definition is valid.' will be shown at the top of the page.

The 'Identifier' references either the page id or the script id for which the validation failed. The 'problem' describes the validation error that occurred.

The following are some of the reasons a script may fail validation:

- List question ids must match attributes contained in the Datastore entity.
- All parent-child entity relationships defined in the script must have a similar relationship defined in the Datastore. A child entity must have a parent entity defined and must also be a child of the root entity or the specified parent entity.
- Any entity that is specified in the script is present in the Datastore.
- Duplicate page ids
- If the script contains a relationship summary page:
 - The Person entity must exist in the Datastore.
 - The Relationship entity must exist in the Datastore.
 - The relationshipType attribute must be specified on the Relationship entity.

Uploading a New Script

At the top of the 'IEG Scripts' page is a 'Import IEG Script' link which lets you upload, or import, a new IEG script.

When you click on this link, you will be presented with this screen which asks you to choose the file on your file system which contains the script definition and to provide a logical name for the script (the id, version number and type will be read from the script definition itself).

If the script you are uploading is intended to replace an existing script with the same id, then you should select the 'Overwrite' check-box, otherwise an error will be thrown.

Read-only Mode

When this flag is set to true, all pages are immutable. A user, most likely a caseworker, can review the script by using the navigation buttons and the sections panel, without the ability to edit any information.

A read-only flag can be set on a script execution using the public API ([“Public API” on page 90](#)).

When the read-only flag is reset to false and a script execution is resumed in editable mode, the current page will be the page the editing user was on previously, not the page the reviewing user last visited.

Command Line Development Options

This section describes command line options for developing IEG scripts.

There is an Import IEG Script command line option available which as an alternative to uploading scripts using the administration screens. This command line option allows a script developer to import script definition files into the database and options which are not available when uploading a script using the administration screens. Details on this Import Script command line option can be found below:

```
appbuild ieg.importscript -Dscript.file=<path to the script file> -Doverwrite=<true or false>-Dgenerate.properties=<true or false>
```

The last two options, `overwrite` and `generate.properties` are optional. These default to false and true respectively. If `overwrite` is false, the import will fail if the script already exists on the database.

If `generate.properties` is false, only the script definition will be inserted to the database. The text elements won't be inserted into the Resource Store properties.

If `generate.properties` is true, all the text elements contained in the script definition will be extracted and inserted into the Resource Store. If a text element has not been defined, its value will be taken from the existing property in the Resource Store if it exists.

Migrating Superseded IEG Scripts

This section provides information on how to run the migration tool and also provides an overview of the process that occurs when the tool is run.

Question scripts developed in the superseded version of IEG cannot be executed in IEG. They have to be migrated, i.e. reconstituted in a format that can be interpreted by the IEG Engine and IEG Player. A migration tool has been provided to help carry out this task.

The intention of this tool is not to produce a polished script, but rather to perform the bulk of the conversion. The result should be reviewed, and it should be assumed that some modifications will need to be performed. For example, changes to page layout will probably be required.

Migration process

Running the Migration Tool

The migration tool can be run from the command line.

From the EJBServer directory, run the following:

```
build migrateiegscrip -Dscriptfilename=myscript.sx  
-Dinputdir=C:\mydir -Doutputdir C:\mydir\output
```

All three of the following parameters are mandatory:

- `scriptfilename` : the name of the superseded IEG script definition file to be migrated, without specifying its location.
- `inputdir` : the full path to the directory where the superseded IEG artifacts to be migrated are located. This will contain the script definition specified in `scriptfilename`, and all the question groups and subscripsts that it references.

- `outputdir` : the full path to the directory where the new definition artifacts will be created. If the tool is run multiple times, the content should be deleted to avoid surprises.

The generated artifacts will be as follows:

- The new script definition will be located in the `clob` directory, under `outputdir`. Using the original script ID, it will be named `scriptID.xml`. Subscripts will be in the same location.
- The Datastore schema will be located in the `clob` directory, under `outputdir`. It will be named `scriptID.xsd`.
- All the texts from the original scripts will be inserted into new properties files located in the `blob` directory, under `outputdir`. One file per script plus one file per page will be generated. One file per subscript page will be added as well. The file names will be `scriptID_version_type.properties` or `scriptID_version_type_pageID.properties`.
- Different DMX files will be automatically created so that the generated artifacts can be inserted easily in a database. The following files are created:

APPRESOURCE.dmx

This DMX file will insert one row per properties file contained in the `blob` directory into the `AppResource` table. Each row has a `ResourceID`. This ID starts at 0 but it should be unique as it is a key to the table. If your table already contains records, you will possibly have to change those IDs manually before running the data manager so that the IDs remain unique. The properties will be for the default (empty) locale.

IEGSCRIPTINFO.dmx

This DMX file will insert the script and subscripts definitions contained in the `clob` directory to the `IEGScriptInfo` table. The script name will be the same as the script ID.

IEGSCRIPTRELS.dmx

This DMX file will insert the relationships between the script and its subscripts in the `IEGScriptRels` table. If no subscript has been produced, this file won't contain any row.

DATASTORESHEMA.dmx

This DMX file will insert the Datastore schema contained in the `clob` directory to the `DatastoreSchema` table.

- A system-generated log file named `migration-log.txt` contains any information, warnings and errors that have been encountered in the migration process. This file should be reviewed after running the tool. Here are the possible contents of this file:
 - **INFO**: gives information about the steps performed by the tool: which scripts and subscripts are migrated, which files are created.
 - **WARNING**: some manual steps may be required to complete the script migration, or some of the old script content cannot be introduced in the new script.
 - **SEVERE**: if an unforeseen event occurs during the migration, some information about the issue could be found here.

Script flow migration

During script flow migration, a superseded IEG script is migrated to an IEG script. This new script is identified using the original script ID, a version number that will default to 1 unless the scriptID ends with the letter V followed by a number.

In this case, the version will be set to that number. The type will be copied from the original script, but if it was empty it will become "DefaultType". The Quit and Finish pages are extracted from the application's `IEGPlayerConfig.xml`. If the Quit and Finish pages need to be modified manually, they have to be changed in the Script Definition file as well as in the script-level properties file.

The new script will contain one section that will not be displayed by default (as the concept of sections does not exist in superseded IEG scripts). The section title will be set to the original script name, and the entire content of the migrated script will reside in this section.

For each page from the original script (whether it's a top-level page or a child page), a new page will be added to the section. The new page ID and description are set to the question group ID and description,

whereas the page title comes from the old page name. The presence of loopsize or precondition attributes will enclose the new pages into loop or condition elements.

Loop types used to be implicit, depending on the expression. They are now explicitly set on loop elements.

The progress bar is enabled by default and the progress value for each page is calculated by allocating weights equally.

Postconditions will automatically be replaced by validation elements.

The following example illustrates how the script flow gets migrated, firstly by showing a superseded IEG script definition before the migration process:

```
<?xml version="1.0" encoding="UTF-8"?>
<IEGScript id="SampleScript">
  <QuestionPageDefinition questionpageid="1"
    questiongroupid="QuestionGroup1" loopsize="" precondition="">
    ...
  </QuestionPageDefinition>
  <QuestionPageDefinition questionpageid="2"
    questiongroupid="QuestionGroup2"
    precondition="QuestionGroup1.Q1>0"
    loopsize="">
  <postcondition id="ValueUnder100"
    expression="QuestionGroup2.Q1<100">
    <message>
      <Translation id="en"
        value="Value must be under 100" />
    </message>
  </postcondition>
  ...
  <QuestionPageDefinition questionpageid="3"
    questiongroupid="QuestionGroup3" precondition=""
    loopsize="QuestionGroup2.Q1">
    ...
  </QuestionPageDefinition>
</QuestionPageDefinition>
<ScriptName>
  <Translation id="en" value="Sample Script" />
</ScriptName>
<ScriptDescription>
  <Translation id="en" value="" />
</ScriptDescription>
</IEGScript>
```

Figure 32: Superseded IEG Script Definition Before Migration

The following XML sample represents the IEG Script Definition created by the migration tool, having operated on the above superseded Script Definition:

```

<?xml version="1.0" encoding="UTF-8"?>
<ieg-script finish-page="IEGPlayer_summary"
quit-page="IEGPlayer_summary" show-sections="false"
show-progress-bar="true">
  <identifier id="SampleScript" scriptversionnumber="1"
type="DefaultType" />
  <section>
  <title id="Section.Title" />
  <question-page id="QuestionGroup1" entity="QuestionGroup1"
progress="0" ... >
  ...
</question-page>
  <condition expression="QuestionGroup1.Q1>0">
  <question-page id="QuestionGroup2"
entity="QuestionGroup2" progress="33" ... >
  ...
  <validation expression="QuestionGroup1.Q1 < 100">
  <message
id="QuestionGroup2.ValueUnder100.Message" />
  </validation>
  ...
</question-page>
  <loop loop-type="for" expression="QuestionGroup2.Q1"
entity="QuestionGroup3">
  <question-page id="QuestionGroup3"
entity="QuestionGroup3" progress="67" ... >
  ...
  </question-page>
  </loop>
</condition>
  <summary-page id="SummaryPage" progress="100" ... >
  ...
</summary-page>
</section>
</ieg-script>

```

Figure 33: IEG Script Definition After Migration

Page Content Migration

The elements contained in a question group (questions and hyperlink labels) are added to the new page under the same cluster. The cluster title is set to the original page name.

Questions and Hyperlink labels will be added to this cluster as new Question or DisplayText elements.

If the HyperlinkLabel is a URL, the display-text created will contain the necessary anchor markup.

Question text and help and their aliases are extracted to create new Question elements. String meta data indicating a multi-line input are converted to a layout element being added to the question with num-rows set to the correct number of lines. If List meta data is supplied, then the question becomes a multi-select question that has to be added in a separate cluster.

If default value expressions are specified for a particular question, they will be set on the new question.

Legislation and Policy links are migrated without the need for modification to new Legislation and Policy links.

The following examples illustrate how the page content gets migrated. The first two examples show the superseded IEG Script Definition and the superseded Question Group Definition:

```

<?xml version="1.0" encoding="UTF-8"?>
<IEGScript id="SampleScript">
  <QuestionPageDefinition questionpageid="1"
questiongroupid="QuestionGroup1" loopsize="" precondition="">
  ...
  </QuestionPageDefinition>
  <ScriptName>
  <Translation id="en" value="Sample Script" />
  </ScriptName>
  <ScriptDescription>
  <Translation id="en" value="" />
  </ScriptDescription>
</IEGScript>

```

Figure 34: Superseded IEG Script Definition

```

<?xml version="1.0" encoding="UTF-8"?>
<QuestionGroup id="QuestionGroup1">
  <Question id="Q1" recordunanswered="false"
    answertype="SVR_BOOLEAN" mandatory="false">
    <Questions>
      <Translation id="en" value="Disabled"/>
    </Questions>
    <ScriptTexts>
      <Translation id="en"
        value="Is <OtherGroup.Q1> disabled"/>
    </ScriptTexts>
    <HelpTexts>
      <Translation id="en"
        value="Does <OtherGroup.Q1>
        have any disability"/>
    </HelpTexts>
    <LegislationLinks>
      <Translation id="en" value=""/>
    </LegislationLinks>
    <PolicyLinks>
      <Translation id="en" value=""/>
    </PolicyLinks>
  </Question>
  <GroupName>
    <Translation id="en" value="Additional Questions"/>
  </GroupName>
  <GroupDescription>
    <Translation id="en"
      value="Please give us general information about
      the household members"/>
  </GroupDescription>
</QuestionGroup>

```

Figure 35: Superseded IEG Question Group Definition

The following XML sample represents the IEG Script Definition created by the migration tool, having operated on the above superseded Script Definition and Question Group Definition:

```

<?xml version="1.0" encoding="UTF-8"?>
<ieg-script finish-page="IEGPlayer_summary"
  quit-page="IEGPlayer_summary" show-sections="false"
  show-progress-bar="true">
  <identifier id="SampleScript" scriptversionnumber="1"
    type="DefaultType" />
  <section>
    <title id="Section.Title" />
    <question-page id="QuestionGroup1" entity="QuestionGroup1"
      progress="0" ... >
      <title id="QuestionGroup1.Title" />
      <description id="QuestionGroup1.Description" />
      <legislation id="QuestionGroup1.LegislationLink" />
      <policy id="QuestionGroup1.PolicyLink" />
      <cluster>
        <title id="QuestionGroup1.Cluster.Title" />
        <question id="Q1" mandatory="false"
          control-question="false" multi-select="false">
          <label id="QuestionGroup1.Q1.Label">
            <argument id="OtherGroup.Q1" />
          </label>
          <help-text id="QuestionGroup1.Q1.HelpText">
            <argument id="OtherGroup.Q1" />
          </help-text>
          <legislation
            id="QuestionGroup1.Q1.LegislationLink" />
          <policy id="QuestionGroup1.Q1.PolicyLink" />
        </question>
      </cluster>
    </question-page>
    <summary-page id="SummaryPage" progress="100" ... >
      ...
    </summary-page>
  </section>
</ieg-script>

```

Figure 36: IEG script definition

Summary Page After Migration

A summary page will be added to the single section the migration tool creates.

It will contain one cluster per page but the questions that are not straightforward (questions from loop pages, multi-select lists,...) will be omitted. Edit links and summary lists will have to be added manually, or the summary page can be removed altogether as it is not mandatory.

Expressions Migration

All the expressions defined in a superseded IEG script will be migrated without modification except when question group IDs contain a dollar sign ('\$').

This is not a valid entity identifier in IEG, so it will be replaced with an underscore ('_'). If list RDOs were used in the original scripts and referenced using the 'current' keyword, the tool will not fail but the generated script will not run.

Subscripts Migration

In a superseded IEG script, subscripts could be added at the top-level (just under the script) or under a page. In an IEG script, a subscript can be added at the top-level or just under a section.

This is why there is no one-to-one mapping between old and new subscripts. Top-level subscripts will be migrated as independent scripts (but they won't have any section) and referenced in the scripts, whereas other subscripts will be merged in the new scripts: their pages will be directly added to the section and not referenced as subscripts.

When inserting a subscript reference, the start-progress and end-progress attributes will be set based on the number of pages in the subscript and the total number of pages in the script, each page having the same weight.

Datastore Schema Generation in the Migration Process

Each question group will be used to add an entity in the Datastore schema, the question group ID becoming the entity name (except if the group ID contains a '\$', in which case it will be replaced by a '_'). As a consequence, each page in the new script is associated with its own entity.

If the original group references RDOs, one entity per RDO is added to the schema, using the RDO name as the entity name and the data items as the attributes.

All the questions in a particular group become the attributes of the page entity. The data type will be copied from the original question definition. If a root domain was used (for example SVR_BOOLEAN), the corresponding IEG domain will be used (IEG_BOOLEAN). If the domain is not a root domain (for example MyApp_BOOLEAN), the corresponding IEG domain will be used, and a warning will be logged as the original domain may have to be manually redefined in the domains schema. In the case of a code table, a warning will be logged as well as it needs to be added to the domains schema.

Properties Generation

All the texts from the translation elements in the original script will be extracted and put in properties to be added in the Resource Store.

Compliance for IEG

Follow these guidelines to develop in a compliant manner with IEG. By following these considerations, customers will also find it easier to upgrade to future versions of Cúram.

Customizing IEG Scripts

IEG scripts may be shipped as part of a Cúram Solution or Module. These scripts can be customized according to the rules documented in the relevant Cúram Solution or Module Customization/Developers guide.

When customizing an IEG script it is important to note that resources referenced by the script are stored in the Resource Store of the Curam application. Examples of IEG resources are textual elements, images and icons displayed during the execution of an IEG script. After making changes to a script that affect

these resources, they should be extracted from the Resource Store. The script and resources should be placed under source control.

Creating a Custom Copy of an IEG Script

About this task

Unless otherwise stated in the relevant customization/developer guides, Cúram recommends that customers create a new copy of the relevant script and update the script identifier (script ID, type and version). Please note that when customizing IEG sub-scripts, the appropriate parent scripts will need to be customized to reference the new custom sub-script. A script can be copied as follows:

Procedure

1. Download the IEG script using the download option available from the IEG section of the Cúram administration application
2. Open the script using an XML or text editor and change the scriptID, version and/or type
3. Use the import IEG script option in the administration application to upload the custom script
4. Make any changes to the script using the IEG Editor in the Cúram administration application
5. Save the script

Script Upgrades

When editing an IEG script using the IEG Editor, the script is edited in-place. This means that changes can be made to scripts that may have been executed or are currently being executed.

In a development environment, this is required to allow changes to be verified in a development/test cycle. In a production environment, in-place changes, other than changes to text elements, should not be made to scripts that are currently being executed or may be re-executed. If non-text changes are required where script executions may be resumed, it is recommended that the script should be copied and the changes made to the new script. In this scenario any existing executions of the original script will not be affected and can be completed. Any subsequent executions should invoke the new script definition.

Database Representation

IEG scripts and resources are stored in the database and may be discarded or overwritten by files from the file system during a database build. This build process uses a data manager configuration to tell what files should be included in the build.

Customer specific data such as IEG scripts and resources should be stored on the file system as DMX files, CLOBs (large character objects) and BLOBs (large binary objects).

An IEG script definition is stored in the IEGSCRIPTINFO database table as a CLOB. Application resources are stored in the APPRESOURCE database table as BLOBs.

It is important therefore that any custom IEG scripts, custom application resources, the IEGSCRIPTINFO.dmx file and the APPRESOURCE.dmx file are on the file system and placed under source control. Cúram recommend the following steps to ensure the custom artifacts are picked up by the build database process:

IEG Database Representation

About this task

The custom IEG script and IEGSCRIPTINFO.dmx file:

Procedure

1. Download the custom script into a custom component directory. For example "custom\data\demo"
2. Use the build target "build extractdata -Dtablename=IEGSCRIPTINFO" to extract an up-to-date copy of the IEGSCRIPTINFO.dmx file which is generated into the "EJBServer\build\dataextractor" directory.

Note that it will also extract the IEG script definitions into a clob folder in the same directory. The naming convention of these resources is IEGSCRIPTINFO<number> e.g. "IEGSCRIPTINFO3"

3. Copy this DMX file into a custom component directory. From the example above this file should be located in the "custom\data\demo" directory
4. Open the IEGSCRIPTINFO.dmx file and delete all the row elements except for the row element that references the new custom script
5. Ensure the script identifier matches the script identifier in the custom script. For readability the CLOB files should be renamed to correspond to the script identifier and the Script Definition attribute in the row should be modified accordingly. For example the original value of the script definition attribute is ".\clob\IEGSCRIPTINFO3". The custom script file should be renamed to something like "IEGSample_custom_v1_Intake". Here the new value of the script definition attribute should also be changed to ".\clob\IEGSample_custom_v1_Intake"
6. Save the IEGSCRIPTINFO.dmx file

APPRESOURCE Database Representation

About this task

The Application Resources and APPRESOURCE.dmx file:

Procedure

1. Use the build target "build extractdata -Dtablename=APPRESOURCE" to create an up-to-date copy of the APPRESOURCE.dmx file which is generated into the build\dataextractor directory. Note that it will also extract all the resources into a blob folder in the same directory. The naming convention of these resources is APPRESOURCE<number> e.g. "APPRESOURCE3"
2. Open the APPRESOURCE.dmx file using an XML or Text editor
3. Search for the custom script resources using the script identifier e.g. "IEGSample_custom_v1_Intake"
4. Each row in the dmx file will have an attribute called "content" that references the resource file in the "build\dataextractor/blob directory"
5. Find that resource file and rename that to the value specified in the "name" attribute for that row in the DMX file e.g. from resource3 to "IEGSample_custom_v1_Intake_AboutYouPage"
6. Ensure the content attribute for that row also uses the same resource name
7. Copy that resource file to a custom component location e.g. custom\data\demo/blob
8. Perform steps 3 through 7 until all the resources are found, renamed and copied to the custom component location
9. Finally save the APPRESOURCE.dmx file and copy it to the custom component directory e.g. custom\data\demo

Results

To ensure the location of all these artifacts is picked up by the build database process, ensure the datamanager_config.xml file reference that custom component directory e.g. <entry name="components/custom/data/demo/" type="dmx" base="basedir"/>

Internal IDs and Script Executions

When an IEG script is executed, the IEG Engine checks the script definition to ensure that all scripts elements that require an "internal-id" have one assigned.

As previously mentioned, IEG scripts are stored in the database. If there are internal IDs missing, the script definition is modified to populate the missing IDs and is updated in the database. So the first time the script is executed, the internal IDs are set and should not be changed subsequently. The internal IDs are used by the IEG Engine to support script execution, for example they are used to determine what the current page in the script execution should be. In order for script executions to continue or for script executions to be resumed, the internal IDs in the script definition must be consistent with the internal IDs

when the script execution was created. For this reason, it is important to ensure that the upgrade environment is in sync with the production environment.

IEG Functional Identifiers (FIDS)

The following IEG classes contain the functional identifiers to run IEG Scripts:

1. IEGRuntime, and
2. IEGPageNavigator.

IEGRuntime is used to create IEG Script Executions. The IEGRuntime class sets and retrieves various elements of an IEG Script Execution at runtime. Not all of the FIDS listed below may be required to run the IEG Script. If there is no manipulation of the script during execution, the FIDS for creating the script executions will suffice. IEGRuntime contains the following FIDS:

1. IEGRuntime.createScriptExecution,
2. IEGRuntime.createScriptExecutionExistingRootEntity,
3. IEGRuntime.resetExecution,
4. IEGRuntime.getScriptExecutionRootEntityID,
5. IEGRuntime.setReadOnlyFlag,
6. IEGRuntime.getReadOnlyFlag, and
7. IEGRuntime.setDisplayBannerFlag.

IEGPageNavigator provides the functions for page navigation in IEG. The FIDS listed below are all required to run an IEG script. IEGPageNavigator contains the following FIDS:

1. IEGPageNavigator.getCurrentPage,
2. IEGPageNavigator.getPersonTabDetails,
3. IEGPageNavigator.getRelationships,
4. IEGPageNavigator.getRelationshipsSummary,
5. IEGPageNavigator.getScript, and
6. IEGPageNavigator.move

Public API

IEG has a public API which you may use in your application code. This public API will not have any components changed or removed without following Cúram standards for handling customer impact.

Identifying the API

The JavaDoc shipped is the sole means of identifying which public classes, interfaces and methods form the public API.

Outside the API

IEG also contains some public classes, interfaces and methods, which do not form part of the API.

Important: To be compliant, dependencies on any class or interface should not be made. No methods should be called other than those described in the JavaDoc.

Classes, interfaces and methods outside of the public API are subject to change or removal without notice. Unless otherwise stated in the JavaDoc, you must not place any of your own classes or interfaces in the same package as that of IEG.

Model Customization

Model files delivered as part of IEG should not be customized, as such customization is not supported. These are files with .emx and .efx extensions.

- IntelligentEvidenceGathering.emx

- IEGScriptAdmin_cat.efx
- ResourceAdmin_cat.efx

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.

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

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

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

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



Part Number:

(1P) P/N: