



IBM Cúram Social Program Management

# Authoring Scripts using Intelligent Evidence Gathering (IEG)

Version 6.0.4

**Note**

Before using this information and the product it supports, read the information in Notices at the back of this guide.

This edition applies to version 6.0.4 of IBM Cúram Social Program Management and all subsequent releases and modifications unless otherwise indicated in new editions.

Licensed Materials - Property of IBM

Copyright IBM Corporation 2012. All rights reserved.

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

© Copyright 2011 Cúram Software Limited

# Table of Contents

Chapter 1 Introduction .....	1
1.1 Purpose .....	1
1.2 Audience .....	1
1.3 Prerequisites .....	2
Chapter 2 Defining IEG Script Structures .....	3
2.1 Overview .....	3
2.2 Organizing Script Pages into Sections .....	4
2.3 Subscripts .....	5
Chapter 3 IEG Script Element Reference .....	7
3.1 Overview .....	7
3.2 Display elements .....	7
3.2.1 add-link .....	7
3.2.2 alias .....	8
3.2.3 argument .....	8
3.2.4 cluster .....	9
3.2.5 column .....	10
3.2.6 container .....	11
3.2.7 delete-link .....	11
3.2.8 description .....	12
3.2.9 display-text .....	12
3.2.10 edit-link .....	13
3.2.11 help-text .....	14
3.2.12 icon .....	15
3.2.13 item-label .....	16
3.2.14 label .....	16
3.2.15 label-element .....	17
3.2.16 legislation .....	17
3.2.17 list .....	18
3.2.18 list-question .....	19
3.2.19 message .....	21
3.2.20 policy .....	21
3.2.21 question .....	22
3.2.22 question-page .....	24
3.2.23 relationship-page .....	26
3.2.24 relationship-summary-list .....	28

3.2.25	summary-page .....	29
3.2.26	title .....	31
3.3	Meta-display elements .....	31
3.3.1	codetable-hierarchy-layout .....	32
3.3.2	label-alignment .....	32
3.3.3	label-width .....	33
3.3.4	layout .....	33
3.3.5	num-cols .....	34
3.3.6	num-rows .....	34
3.3.7	type .....	34
3.3.8	width .....	35
3.4	Flow-control elements .....	35
3.4.1	condition .....	35
3.4.2	loop .....	37
3.4.3	validation .....	38
3.5	Structural, Administrative and Other elements .....	38
3.5.1	callout .....	39
3.5.2	identifier .....	39
3.5.3	ieg-script .....	40
3.5.4	section .....	41
3.5.5	set-attribute .....	42
3.5.6	ieg-sub-script .....	43
Chapter 4	Operations Supported for IEG Expressions .....	44
4.1	Introduction .....	44
4.2	Bracketing of Terms .....	44
4.3	Operator Precedence .....	44
4.4	Data Types and Supported Operations .....	45
4.5	Custom Functions in Expressions .....	46
4.6	IntakeProgramType and ScreeningProgramType in Expressions .....	46
Chapter 5	Controlling the Flow of Your IEG Script .....	48
5.1	Introduction .....	48
5.2	Natural Flow of an IEG Script .....	48
5.3	Controlling the Flow using Sections .....	49
5.4	Controlling the Flow using Conditions .....	50
5.4.1	Controlling Page Content using Conditional Clusters .....	51
5.5	Controlling the Flow using Loops .....	52
5.5.1	The For-each loop .....	52
5.5.2	The For loop .....	53
5.5.3	The While Loop .....	54
5.5.4	Controlling the Flow Using Nested Loops .....	55
5.6	Control-Questions .....	56
5.7	Looping through People .....	57
5.8	Customizing Links on Summary Pages .....	59
5.8.1	Editing Information in Clusters .....	59
5.8.2	Editing Records in Lists .....	60
5.8.3	Deleting Records from Lists .....	61
5.8.4	Adding Records to Lists .....	61

Chapter 6 Configuring IEG .....	62
6.1 Introduction .....	62
6.2 Using the Layout Element to Customize IEG Pages .....	62
6.2.1 Using the Layout Element to Change the Appearance of Clusters .....	62
6.2.2 Summary of Cluster Layout Options .....	63
6.2.3 Using the Layout Element to Change the Appearance of Multiple-Choice Questions .....	64
6.2.4 Summary of Multiple-Choice Question Layout Options .....	65
6.2.5 Using the Container Element to Control Layout of Questions and Columns .....	66
6.3 Using Configuration Properties to Customize IEG Pages .....	67
6.3.1 Changing the Look-and-Feel of the Pages .....	68
6.3.2 Configuring the Layout of the Sections Panel .....	80
6.3.3 Configuring the Layout of the IEG Player in Modal Dialogs .....	83
Chapter 7 IEG Administration .....	85
7.1 Introduction .....	85
7.2 Listing all Scripts .....	85
7.3 Downloading an Existing Script .....	85
7.4 Removing an Existing Script .....	85
7.5 Running a Script .....	86
7.6 Validating a Script .....	86
7.7 Uploading a New Script .....	87
7.8 Read-only Mode .....	87
Chapter 8 Migrating Superseded IEG Scripts .....	88
8.1 Migration process .....	88
8.1.1 Introduction .....	88
8.1.2 Running the Migration Tool .....	88
8.1.3 Script flow migration .....	90
8.1.4 Page Content Migration .....	91
8.1.5 Summary Page After Migration .....	93
8.1.6 Expressions Migration .....	93
8.1.7 Subscripts Migration .....	94
8.1.8 Datastore Schema Generation in the Migration Process .....	94
8.1.9 Properties Generation .....	94
Appendix A Compliancy .....	95
A.1 Introduction .....	95
A.2 Customizing IEG Scripts .....	95
A.2.1 Creating a Custom Copy of an IEG Script .....	95
A.2.2 Script Upgrades .....	96
A.2.3 Database Representation .....	96
A.2.4 Internal IDs and Script Executions .....	98
A.3 Public API .....	98
A.4 Identifying the API .....	98
A.5 Outside the API .....	98
A.6 Model Customization .....	99
Notices .....	100



# Chapter 1

## Introduction

### 1.1 Purpose

The purpose of this guide is to provide instructions on how to use the features available in *Intelligent Evidence Gathering*<sup>™</sup> (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.

### 1.2 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.

### 1.3 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



# Chapter 2

## Defining IEG Script Structures

### 2.1 Overview

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.

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 will always be the last page displayed within a section and will also be 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 struc-

ture. The following section demonstrates how to organize script pages into sections in an XML file.

### 2.2 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>
```

### Example 2.1 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.

## 2.3 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>
```

## Authoring Scripts using Intelligent Evidence Gathering (IEG)

```
</ieg-sub-script>  
</section>  
...  
</ieg-script>
```

Example 2.2 Subscript element

# Chapter 3

## IEG Script Element Reference

### 3.1 Overview

This chapter 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.

### 3.2 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.

#### 3.2.1 add-link

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

The text used on this link is configured through the IEG properties, as described in Chapter 6, *Configuring IEG* chapter.

None

Attributes

Name	Description
start-page	the page the user should be taken to when they click on the add link. This attribute is mandatory and would normally be the first page within the loop used to populate this list in the first place.

Name	Description
end-page	an optional page on which to end the process of adding a new record to this list. This is usually used when there are either more than one page within the loop used to populate the list, or there are pages after the loop which need to be revisited after adding a new record. See Section 5.8.4, <i>Adding Records to Lists</i> for more details.
criteria	the criteria to use when retrieving the records from the entity to be displayed in the option list when adding a new entity. If no criteria is present, then all records (for this instance of the root entity in the Datastore) from the entity will be retrieved.
skip-to-summary	when set to true, the skip-to-summary attribute will navigate back to the summary page once Next is selected after using the add-link to add a new entity. If skip-to-summary is false, and values have been added that are used in any expressions in the subsequent pages, the Engine will display those pages before going back to the summary page. If not specified, skip-to-summary will be false by default.

Table 3.1 Add-link Attributes

#### Child Elements

None.

### 3.2.2 alias

Question aliases are used to specify different question labels and help texts depending on the viewer of the question script. For example, aliases might be 'Customer', 'Third Party', 'Third Party Medical', etc.

#### Attributes

Name	Description
type	the alias type indicates which category of script viewer should see the current alias.

Table 3.2 Alias Attributes

#### Child Elements

None.

### 3.2.3 argument

The argument element can be added to a message element when you want to substitute an attribute value in the message text See Section 3.4.3, *validation* for more details.

### Attributes

Name	Description
id	the id of an argument refers to the entity name and attribute name used to retrieve the value for the attribute in the Datastore. The id should be in the form entity-Name.attributeName

Table 3.3 Argument Attributes

### Child Elements

None.

### 3.2.4 cluster

The cluster element is used to visually group questions and answers on a page. All questions within the cluster will be displayed under the same heading and will be subject to whatever layout management is in place for the cluster. See Section 6.2.1, *Using the Layout Element to Change the Appearance of Clusters* for more details on how this works for clusters.

### Attributes

Name	Description
entity	the entity attribute can be used to specify which entity within the associated Datastore schema the attributes within this cluster relate to. In order to validly specify an entity for a cluster, there must also be an entity specified for the page containing the cluster. The IEG Engine will then assume that the entity specified for the cluster is a direct child of the entity specified for the page and that there is only one occurrence of the entity specified for the cluster (when combined with the criteria) within the entity specified for the page. All attributes referenced within this cluster are assumed to belong to the entity specified for the cluster. If no entity is specified for the cluster, all attributes referenced within it are assumed to belong to the entity specified for the page instead.
criteria	in situations where there could potentially be more than one instance of the entity specified for a cluster within the entity specified for the page, the criteria attribute can

Name	Description
	be used to identify which instance to use for this cluster
read-only-expression	a boolean expression which, if evaluated to true at runtime, causes all content of the cluster to become read-only.

Table 3.4 Cluster Attributes

### Child Elements

Name	MinOccurs	MaxOccurs
title	0	1
description	0	1
help-text	0	1
edit-link	0	1
layout	0	1
container	0	unbounded
question	0	unbounded
list-question	0	1
display-text	0	unbounded

Table 3.5 Cluster Child Elements

### 3.2.5 column

The column element represents an individual column within a list, and can be used to instruct the Engine as to what title to display as the heading for the column, and what information to display within it. This information will usually come from an attribute within the entity 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

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 specified for this list.
entity	the entity in the Datastore from which to read the elements to be displayed within the column.
link-entity	link-entity can be used instead of the id attribute to spe-



Name	Description
	Specify the name of the entity which stores a relationship between the entity specified for this column and another Datastore entity (e.g. InsuranceRelationship in the IEG Sample script).
criteria	the criteria to use when retrieving the records from the entity.

Table 3.6 Column Attributes

#### Child Elements

Name	MinOccurs	MaxOccurs
title	0	1
layout	0	1

Table 3.7 Column Child Elements

### 3.2.6 container

The container element can be used to group questions within a cluster and also to group multiple columns in a list into a single column. See Section 6.2.5, *Using the Container Element to Control Layout of Questions and Columns* for more details.

#### Attributes

None.

#### Child Elements

Name	MinOccurs	MaxOccurs
question	0	unbounded
column	0	unbounded
layout	0	1

Table 3.8 Container Child Elements

### 3.2.7 delete-link

The delete-link element can be added to a list when you want users to be able to delete records from that list. A link will be provided with each record in the list which, when clicked, will pop up a confirmation dialog asking the user to confirm whether they wish to delete the record. If the user clicks

Yes, this record and all its child entities will be removed from the Datastore. The text for both the link and the confirmation dialog are configurable ( see the Configuration chapter for more details ).

Attributes

None.

Child Elements

None.

### 3.2.8 description

The description element can be used 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 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 description element as the key.

Attributes

Name	Description
id	an identifier for this description 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 context in which it is used. For example, if the description is added to a cluster, then the id need only be unique within the page in which the cluster is contained. Another page could contain an id with the same value.

Table 3.9 Description Attributes

Child Elements

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

Name	MinOccurs	MaxOccurs
argument	0	unbounded

Table 3.10 Description Child Elements

### 3.2.9 display-text

The display-text element is used within a cluster to display text alongside question elements. 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 display-text 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 display-text element as the key.

### Attributes

Name	Description
id	an identifier for this display 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.

Table 3.11 Display-text Attributes

### Child Elements

Name	MinOccurs	MaxOccurs
argument	0	unbounded

Table 3.12 Display-text Child Elements

### 3.2.10 edit-link

The edit-link element can be added to a list when you want users to be able to edit records within that list. A link will be 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. 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. See Chapter 5, *Controlling the Flow of Your IEG Script* for more details.

### Attributes

Name	Description
start-page	the page to which the user will be taken when they click on the edit link. This attribute is mandatory and would normally be the first page within the loop used to populate this list in the first place.
end-page	if more than one page was used to create this record,

Name	Description
	then the end page can be set to indicate the set of pages which the user must visit in order to edit a record within the list.
skip-to-summary	when set to true, the skip-to-summary attribute will navigate back to the summary page once next is selected after using the edit-link to modify an existing entity. If skip-to-summary is false, 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, skip-to-summary will be false by default.

Table 3.13 Edit-link Attributes

#### Child Elements

None.

### 3.2.11 help-text

The help-text element can be used to specify the help text for a cluster or question within a script. When an IEG script is run, the help text for each question 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 on a link at the right-hand side of the cluster's title, and can be hidden by either clicking the same link again or using a close link within the panel itself.

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 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 help-text element as the key.

#### Attributes

Name	Description
id	an identifier for this help 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.

Table 3.14 Help-Text Attributes

## Child Elements

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

Name	MinOccurs	MaxOccurs
argument	0	unbounded
alias	0	unbounded

Table 3.15 Help-Text Child Elements

### 3.2.12 icon

The icon element is used to add images to either the title area of a page or the sections panel. When added to a section, the icon element should contain three attributes to specify the images to use when the section link is enabled, disabled and the current section. When used within 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.

Table 3.16 Icon Attributes

## Child Elements

None.

### 3.2.13 item-label

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

#### Attributes

None.

#### Child Elements

Name	MinOccurs	MaxOccurs
label-element	1	unbounded

Table 3.17 Item-label Child Elements

### 3.2.14 label

The label element is used to define the label text for any question or answer within 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 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 label element as the key.

#### Attributes

Name	Description
id	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 id must be unique within the page in which it is used.

Table 3.18 Label Attributes

#### Child Elements

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

Name	MinOccurs	MaxOccurs
argument	0	unbounded
alias	0	unbounded

Table 3.19 Label Child Elements

### 3.2.15 label-element

The label-element element is used within 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

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.

Table 3.20 Label-element Attributes

#### Child Elements

None.

### 3.2.16 legislation

Legislation links can be specified at page and question level to point the user to pertinent legislative information relating to the questions presented. 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

Name	Description
id	an identifier for this legislation link, which will be used

Name	Description
	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.

Table 3.21 Legislation Attributes

### Child Elements

None.

### 3.2.17 list

The list element can be used to display the details of multiple records from an entity in the Datastore. 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.

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).

Please note, lists may 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

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.



Table 3.22 List Attributes

Child Elements

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

Table 3.23 List Child Elements

### 3.2.18 list-question

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

To create a list question, you need to specify the label for the overall question, the entity from which to read the items in the list, the attribute(s) 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. The xml used to create the above list question would therefore look something like this:

```
<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>
```

#### Example 3.1 List question XML

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.

One thing to be aware of when using list questions is that in order to ensure that the full width of the page is available to them, they should always be added to a cluster whose label-width has been set to 0.

The list question element can also be used to display a question matrix,

which is a list question with a codetable 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 codetable.

### Attributes

Name	Description
entity	the entity in the Datastore from which to read the elements to be displayed within the list.
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). See the <a href="#">Creating Datastore Schemas</a> guide for more details.
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.

Table 3.24 List-question Attributes

### Child Elements

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

Table 3.25 List-question Child Elements

### 3.2.19 message

The message element is used within a validation to represent the message to be displayed to the user if the validation fails. 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

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.

Table 3.26 Message Attributes

### Child Elements

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

Name	MinOccurs	MaxOccurs
argument	0	unbounded

Table 3.27 Message Child Elements

### 3.2.20 policy

Policy links can be specified at page and question level to point the user to

pertinent policy information relating to the questions presented. 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

Name	Description
id	an identifier for this policy link, which will be 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.

Table 3.28 Policy Attributes

### Child Elements

None.

### 3.2.21 question

The question element is used to ask the user a question on a page and capture their answer. It consists of an optional label (the question text) and an input control (for the user to enter/select their answer).

### 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 Data-</p>

Name	Description
	store 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.
mandatory	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.
control-question	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.
control-question-type	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.
multi-select	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.
default-value-expression	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

Name	Description
	should have a real value by the time the expression is evaluated. Default value expressions can also be defined for control questions.
read-only-expression	a boolean expression which, if evaluated to true at runtime, causes the question to become read-only.
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.

Table 3.29 Question Attributes

### 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

Table 3.30 Question Child Elements

### 3.2.22 question-page

The question-page element represents a standard page used for capturing the answers to questions in a script.

#### 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	the name of an entity from the associated Datastore schema. If used directly within a section, the entity ref-

Name	Description
	<p>erenced 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>
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

Name	Description
	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.

Table 3.31 Question-page Attributes

### Child Elements

Name	MinOccurs	MaxOccurs
title	1	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

Table 3.32 Question-page Child Elements

### 3.2.23 relationship-page

The relationship-page element should only be used when you want to capture household relationships within your script (i.e. relationships between the people in your household). In order for this element to work properly, it relies on the following entity structure in your Datastore schema:

```
<xsd:element name="Person">
  <xsd:complexType>
    <xsd:sequence minOccurs="0">
      <xsd:element ref="Relationship"
        minOccurs="0"
        maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```



```

    <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>

```

### Example 3.2 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>

```

### Example 3.3 Datastore schema for relationship Attributes

You will obviously also need to have captured the people in your household prior to visiting 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 on how to create relationships pages and how they look can be found in Chapter 5, *Controlling the Flow of Your IEG Script*.

#### 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

Name	Description
	Engine and Player that person tabs should be displayed at the top of the page. More information can be found in Chapter 5, <i>Controlling the Flow of Your IEG Script</i> .
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.

Table 3.33 Relationship-Page Attributes

### Child Elements

Name	MinOccurs	MaxOccurs
title	1	1
description	0	1
icon	0	1
question	0	1
cluster	0	unbounded

Table 3.34 Relationship-Page Child Elements

### 3.2.24 relationship-summary-list

The relationship-summary-list element can be used on a summary page to display a list of all the household relationships captured using the relationship-page element. More details on how to use the relationship-summary-list element and how it behaves can be found in Chapter 5, *Con-*

*trolling the Flow of Your IEG Script .*

Attributes

None.

Child Elements

Name	MinOccurs	MaxOccurs
title	0	1
description	0	1
edit-link	0	1
column	0	unbounded

Table 3.35 Relationship-Summary-List Child Elements

### 3.2.25 summary-page

The summary page element is used to define the last page in a section. It displays read-only values of questions previously answered in the section. The summary page will be displayed when navigating to previous sections in the script. See Section 5.3, *Controlling the Flow using Sections* for more details.

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

Name	Description
	<p>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.</p> <p>If the page is contained within a loop, no criteria should be specified as the criteria for the loop will be used.</p>
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.
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.

Table 3.36 Summary-page Attributes

### Child Elements

Name	MinOccurs	MaxOccurs
title	1	1
description	0	1
icon	0	1
help-text	0	1
cluster	0	unbounded
condition	0	unbounded
list	0	unbounded

Name	MinOccurs	MaxOccurs
validation	0	unbounded
set-attribute	0	unbounded
relationship-summary-list	0	unbounded

Table 3.37 Summary-page Child Elements

### 3.2.26 title

The title element is used to define the localizable text for any titled element within a script (sections, pages, clusters, etc.). Each title 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 title 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 title element as the key.

#### Attributes

Name	Description
id	an identifier for this title 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 context in which it is used. For example, if the title is added to a cluster, then the id need only be unique within the page in which the cluster is contained. Another page could contain an id with the same value.

Table 3.38 Title Attributes

#### Child Elements

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

Name	MinOccurs	MaxOccurs
argument	0	unbounded

Table 3.39 Title Child Elements

## 3.3 Meta-display elements

Meta=display elements, while not shown on the screen themselves, provide

information on how display elements should be displayed.

### 3.3.1 codetable-hierarchy-layout

The `codetable-hierarchy-layout` element can be used within a question with a `codetable hierarchy` type to control different aspects of the layout of that element. Note that these layout options are not supported within compact layout clusters.

Attributes

Name	Description
<code>vertical</code>	this boolean attribute, when set to true, instructs the IEG Player to display the codetable hierarchy in a vertical layout. If not set, it will default to false.
<code>hide-description</code>	this boolean attribute, when set to true, instructs the IEG Player not to display description text for the codetable hierarchy. If not set, it will default to false. If <code>show-path</code> is true, and the question is read-only, this attribute does not apply. Only one of the attributes <code>hide-description</code> or <code>hide-label</code> should be true.
<code>hide-label</code>	this boolean attribute, when set to true, instructs the IEG Player not to display labels for the codetable hierarchy path. If not set, the value of this attribute will default to false. If <code>show-path</code> is false, and the question is read-only, this attribute does not apply. Only one of the attributes <code>hide-description</code> or <code>hide-label</code> should not be true.
<code>show-path</code>	this boolean attribute, when set to true, instructs the IEG Player to display the path taken through the codetable hierarchy to arrive at the final answer chosen. It only applies when the question is read-only. If not set, it will default to false.

Table 3.40 Codetable Hierarchy Layout Attributes

### 3.3.2 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 Section 6.2.2, *Summary of Cluster Layout Options* for more details.

Attributes

None.

### Child Elements

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

- LEFT - to align the text to the left within the space allocated to the label
- CENTER - to align the text to the center within the space allocated to the label
- RIGHT - to align the text to the right within the space allocated to the label (the default)

### 3.3.3 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 Section 6.2.2, *Summary of Cluster Layout Options* 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.

### 3.3.4 layout

The layout element can be used within either a cluster or a question to control different aspects of the layout of that element. Controlling the layout of a cluster is described in more detail in Section 6.2.1, *Using the Layout Element to Change the Appearance of Clusters*. For a question, controlling the layout is much simpler as the only aspect of its layout you can control is the width of the input control used to enter the answer to the question.

#### Attributes

None.

### Child Elements

Name	MinOccurs	MaxOccurs
type	0	1

Name	MinOccurs	MaxOccurs
num-cols	0	1
num-rows	0	1
width	0	1
label-width	0	1
label-alignment	0	1

Table 3.41 Layout Child Elements

### 3.3.5 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 Section 6.2.1, *Using the Layout Element to Change the Appearance of Clusters* 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.

### 3.3.6 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.

### 3.3.7 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 Section 6.2.1, *Using the Layout Element to Change the Appearance of Clusters* 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 codetable 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 codetable 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.

### 3.3.8 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 Section 6.2.1, *Using the Layout Element to Change the Appearance of Clusters* 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.

## 3.4 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.

### 3.4.1 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 Chapter 5, *Controlling the Flow of Your IEG Script* 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 the Fast Path section of the <i>Working with IEG</i> guide for more details on Fast Path.

Table 3.42 Condition Attributes

### Child Elements

Name	MinOccurs	MaxOccurs
cluster	0	1
condition	0	unbounded

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

Table 3.43 Condition Child Elements

### 3.4.2 loop

As the name suggests, 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 Chapter 5, *Controlling the Flow of Your IEG Script*.

#### 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 the Fast Path section of the <i>Working with IEG</i> guide for more details on Fast Path.

Table 3.44 Loop Attributes

#### Child Elements

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

Table 3.45 Loop Child Elements

### 3.4.3 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 %ld as your wage amount.
    Please enter a value greater than zero.]]>
    <argument id="Income.wageAmount" />
  </message>
</validation>
```

Example 3.4 Validation XML

#### Attributes

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

Table 3.46 Validation Attributes

#### Child Elements

Name	MinOccurs	MaxOccurs
message	1	1

Table 3.47 Validation Child Elements

## 3.5 Structural, Administrative and Other elements

This section lists the elements in IEG which are neither displayed nor used explicitly for flow control.

### 3.5.1 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.
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.

Table 3.48 Callout Attributes

#### Child Elements

None.

### 3.5.2 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

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 dis-

Name	Description
	playing lists of scripts.

Table 3.49 Identifier Attributes

Child Elements

None.

### 3.5.3 ieg-script

The ieg-script element is the root element of the xml file containing an IEG script and as the name suggests is used to define the script element itself.

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.
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

Name	Description
	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.

Table 3.50 leg-script Attributes

#### Child Elements

Name	MinOccurs	MaxOccurs
identifier	1	1
section	0	unbounded

Table 3.51 leg-Script Child Elements

### 3.5.4 section

As the name suggests, the section element represents a section of pages within an IEG script.

#### Attributes

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.

Table 3.52 Section Attributes

### Child Elements

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

Table 3.53 Section Child Elements

### 3.5.5 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" />
```

#### Example 3.5 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.



Table 3.54 Set-attribute Attributes

Child Elements

None.

### 3.5.6 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

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.

Table 3.55 ieg-Sub-Script Attributes

Child Elements

Name	MinOccurs	MaxOccurs
identifier	1	1

Table 3.56 ieg-script Child Elements

# Chapter 4

## Operations Supported for IEG Expressions

### 4.1 Introduction

This chapter 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.

### 4.2 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 (see below) and may add complexity to an expression. Any operation that should be carried out in advance of another operation should be bracketed, e.g.,  $5 * (3/4) = 3.75$ .

### 4.3 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:

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

Operator	Associativity	Type
== !=	left to right	equities

Table 4.1 Operator Precedence

## 4.4 Data Types and Supported Operations

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

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.

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
IEG_INT8	IEG_INT8	+, -, /, *	IEG_INT32
IEG_INT16	IEG_INT16	+, -, /, *	IEG_INT32

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

Table 4.2 Data Types and Supported Operations

## 4.5 Custom Functions in Expressions

Custom functions may also be referenced by expressions defined in an IEG script. For example:

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

### Example 4.1 Custom Function in an Expression

As custom functions are a server side phenomenon they may not be referenced by expressions that are evaluated on the client side. This means that custom functions may not be referenced by expressions of dynamically conditional clusters. Please also note that custom functions cannot accept a variable number of parameters. See the *Cúram Rules Codification Guide* for more information on defining and implementing custom functions.

## 4.6 IntakeProgramType and ScreeningProgramType in Expressions

As previously mentioned in Chapter 1, *Introduction*, 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>
```

## Authoring Scripts using Intelligent Evidence Gathering (IEG)

```
<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>
```

### Example 4.2 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>
```

### Example 4.3 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>
```

### Example 4.4 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.

# Chapter 5

## Controlling the Flow of Your IEG Script

### 5.1 Introduction

This chapter 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.

### 5.2 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 *will be discarded* . 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. See Table 3.31, *Question-page Attributes* for further details.

### 5.3 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).

### 5.4 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>
```

#### Example 5.1 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 Section 3.5.5, *set-attribute*).

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>
```

### Example 5.2 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>
```

### Example 5.3 Using 'isNotNull' Custom Function

#### 5.4.1 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 Section 3.4.1, *condition* 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.

## 5.5 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.

### 5.5.1 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>
```

#### Example 5.4 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>
```

#### Example 5.5 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.

### 5.5.2 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>
```

#### Example 5.6 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>
```

#### Example 5.7 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.

### 5.5.3 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>
```

#### Example 5.8 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 through 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>
```

### Example 5.9 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.

#### 5.5.4 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.

## 5.6 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 Section 5.5.4, *Controlling the Flow Using Nested Loops* for more details.
- condition expressions: If a control question is referenced in the expres-

sion 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.

### 5.7 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>
```

### Example 5.10 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>
```

### Example 5.11 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>
```

### Example 5.12 Relationship Page XML with Relationship Attributes

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

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>
```

### Example 5.13 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 ( Section 5.8.2, *Editing Records in Lists* )

## 5.8 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.

### 5.8.1 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>
```

### Example 5.14 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.

### 5.8.2 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.

### 5.8.3 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.

### 5.8.4 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.

# Chapter 6

## Configuring IEG

### 6.1 Introduction

This chapter provides information how to configure IEG. IEG configuration includes customizing the layout of IEG pages using the layout element and using configuration properties. IEG configuration also includes integrating the IEG Player into an application in both tab and modal contexts.

### 6.2 Using the Layout Element to Customize IEG Pages

#### 6.2.1 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>

```

### Example 6.1 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>

```

### Example 6.2 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>

```

### Example 6.3 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>

```

### Example 6.4 Layout with Width for Cluster

## 6.2.2 Summary of Cluster Layout Options

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

Name	Description
type	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).
num-cols	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.
width	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%.
label-width	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%.
label-alignment	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.

Table 6.1 Cluster Layout Options

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.

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

The default appearance of a multiple-choice question (i.e. 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>
```

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

### Example 6.5 Layout with Number of Columns

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

### Example 6.6 Layout with Number of Rows

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>
```

### Example 6.7 Layout with Input Alignment Set to Right

## 6.2.4 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 `multise-`

`lect.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.

### 6.2.5 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.

#### 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>
```

#### Example 6.8 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>
```

### Example 6.9 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>
```

### Example 6.10 List Container with Width XML

## 6.3 Using Configuration Properties to Customize IEG Pages

This section details how the style, content and layout of elements on IEG pages can be customized 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 facilitated by setting application properties rather than modifying CSS, thereby reducing the need for web design knowledge.

Default values of the properties 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 Section 3.5.3, *ieg-script*). This file will contain the properties and values that differ from `ieg-config.properties`.

### 6.3.1 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, etc.) 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.

Property	Description
logo.alt	The alt text for the logo image.
print.logo	The image to use for displaying the print option.
print.logo.hover	The image to use when hovering over the print option.
print.logo.click	The image to use when clicking on the print option.
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.

Table 6.2 Page Banner Configuration Properties

### 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:

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.
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

Property	Description
	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.

Table 6.3 Progress Panel Configuration Properties

### 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

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

Table 6.4 Person Tabs Panel Configuration Properties

### 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:

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 drop-down list used for Add links.
list.action.edit	The text to display for Edit links.
list.action.desc.edit	The description/alt text for

Property	Description
	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.

Table 6.5 Action Links Configuration Properties

### Configuring Relationship Pages

It is possible to configure all the standard texts on relationship pages, using the properties listed below:

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

Property	Description
	in a relationship summary list.
relationship.dropdown.message	The title attribute of the dropdown list for relationship questions.

Table 6.6 Relationship Page Configuration Properties

### 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:

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.
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

Property	Description
	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	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.
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.

Table 6.7 Help Panel Configuration Properties

### 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:

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.

Table 6.8 Page Title Panel Configuration

### 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:

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.
navpanel.button.selected.background.color	The background color for selected navigation buttons.

Property	Description
	tons.
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.

Table 6.9 Navigation Panel Configuration

### Configuring Lists

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

Property	Description
style.list.as.cluster	<p>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...</p> <ul style="list-style-type: none"> <li>• Lists specified on summary pages</li> <li>• Lists specified on non-summary pages</li> <li>• Nested Lists</li> </ul>

Property	Description
	<ul style="list-style-type: none"> <li>Relationship Summary Lists</li> </ul>
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.

Table 6.10 List Configuration

### Other Page Layout Configurations

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

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.
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

## Authoring Scripts using Intelligent Evidence Gathering (IEG)

Property	Description
	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.
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.

Property	Description
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.
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

Property	Description
	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.

Table 6.11 Other Page Layout Configuration Properties

### 6.3.2 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:

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 cur-

Property	Description
	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.
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.

Table 6.12 Sections Panel Configuration Properties (Universal)

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

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

Property	Description
	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.
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.



Property	Description
sectionspanel.horizontal.arrow.width	The width of the divs holding the section navigation links/ icons.

Table 6.13 Sections Panel Configuration Properties (Horizontal)

### 6.3.3 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:

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

Property	Description
	automatically dismissed when the Exit button is selected.
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.

Table 6.14 Configuration Properties for IEG Player in a Modal Dialog

# Chapter 7

## IEG Administration

### 7.1 Introduction

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.

### 7.2 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.

### 7.3 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.

### 7.4 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.

### 7.5 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.

### 7.6 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.

## 7.7 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.

## 7.8 Read-only Mode

A read-only flag can be set on a script execution using the public API ( Section A.3, *Public API* ). 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.

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.

# Chapter 8

## Migrating Superseded IEG Scripts

### 8.1 Migration process

#### 8.1.1 Introduction

This chapter 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.

#### 8.1.2 Running the Migration Tool

The migration tool can be run from the command line: from the EJBServer directory, run the following:

```
build migrateiegsript -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 subscripts

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.

### 8.1.3 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=" ">
```



## Authoring Scripts using Intelligent Evidence Gathering (IEG)

```
<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>
```

### Example 8.1 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>
```

### Example 8.2 IEG Script Definition After Migration

#### 8.1.4 Page Content Migration

## Authoring Scripts using Intelligent Evidence Gathering (IEG)

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>
```

### Example 8.3 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>

```

### Example 8.4 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>

```

### Example 8.5 IEG script definition

#### 8.1.5 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.

#### 8.1.6 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.

### 8.1.7 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.

### 8.1.8 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 '\_'). 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.

### 8.1.9 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.

# Appendix A

## Compliance

### A.1 Introduction

This appendix explains how to develop in a compliant manner. By following these considerations, customers will also find it easier to upgrade to future versions of Cúram.

### A.2 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.

#### A.2.1 Creating a Custom Copy of an IEG Script

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:

1. Download the IEG script using the download option available from the IEG section of the Cúram administration application

## Authoring Scripts using Intelligent Evidence Gathering (IEG)

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

### A.2.2 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.

### A.2.3 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:

The custom IEG script and IEGSCRIPTINFO.dmx file:

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

## Authoring Scripts using Intelligent Evidence Gathering (IEG)

er\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

The Application Resources and APPRESOURCE.dmx file:

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

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

### A.2.4 Internal IDs and Script Executions

As previously mentioned, IEG scripts are stored in the database. 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. 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.

### A.3 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.

### A.4 Identifying the API

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

### A.5 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.

### A.6 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 U.S.A. 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

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (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.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: 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 states 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 typograph-

## Authoring Scripts using Intelligent Evidence Gathering (IEG)

ical 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply 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 Corporation  
Dept F6, Bldg 1  
294 Route 100  
Somers NY 10589-3216  
U.S.A.

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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

## Authoring Scripts using Intelligent Evidence Gathering (IEG)

should be addressed to the suppliers of those products.

All 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 the names and addresses used by an actual business enterprise 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. \_enter the year or years\_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Programming Interface Information

This publication documents intended programming interfaces that allow the customer to write programs to obtain the services of IBM Cúram Social Pogram Management.

## 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/us/en/copytrade.shtml> .

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

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