IBM Cúram Social Program Management

Cúram Rules Definition Guide

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.

Table of Contents

Chapter 1

Introduction

1.1 Purpose

The purpose of the document is to introduce the concept of rules and to introduce the process of extracting the rules from legislation and designing the rulesets. It also contains the best recommended practices to follow.

1.2 Audience

This document should be read by the business analysts working on the design of the business rules.

1.3 Introduction to Rules and Rules Engine

In the scope of a Cúram application, rules are the conditions for a product delivery. The conditions are drawn from the appropriate legislation, and should also reflect any assumed conditions which are normally imposed, e.g. that the client be alive.

The Cúram Rules Engine executes the rules and provides a decision. A decision consists of an indicator specifying if the conditions succeeded and a list of financial and non-financial objectives achieved in the rules. Financial objectives are calculated using the rate components defined in the rules. Rate components represent the amount of payments associated with certain rules.

Chapter 2

Identifying Rules

2.1 Introduction

In a Cúram application, rules are the conditions for a product delivery. Rules are drawn from relevant legislation and from the conditions that are normally imposed for a product delivery. This chapter describes the process for identifying rules by analyzing legislation and looking at the rate components for a product delivery.

2.2 Identifying Rules

The first step in the rule design process is to identify the conditions which are associated with the eligibility for a particular product delivery. Rules are drawn from relevant legislation and from the conditions that are normally imposed for a product delivery, e.g., that the client is alive. Analyzing these should lead to the creation of a list of conditions.

Note that conditions which are purely manual processes may have to be redrawn to match their computerized equivalents or assigned to the procedures which drive the computerized system. For example, if there is a requirement that a certain form must have been received, the rule may check a flag that this has occurred or the procedures may dictate that no entry for a product delivery may be filed until that form has been received.

2.3 Creating a Decision Table

Decision tables are used to tabulate all possible situations in which a decision may be required and to specify which action to take in each of these situations. The decision table is the main tool used in rule design.

The following methodology is used to create a decision table:

1. Identify conditions and values

2. Identify possible rate components (these are represented by objective tags in the Rules Editor)
3. Enter all valid condition combinations
4. Enter a rate component for each condition combination

Section 2.4, *Example of Rule Identification* demonstrates how each of these steps is carried out by the business analyst.

2.3.1 Verification

After the decision table has been created, it is important to verify that it produces a correct set of rate components for a given client. This should involve drawing up a list of clients whose information leads to various combinations of rate components and clients who fail to qualify for any rate components. Note that this process may be postponed until the evidence has been identified (see Chapter 3, *Identifying Evidence*).

2.3.2 Simplification

After verification, the decision table can be simplified. This means only keeping the condition combinations that give a successful eligibility result. Simplification can be achieved by removing any conflicting conditions or duplicate columns from the decision table.

For example, if there is a condition of “Aged over 65 years” and a condition of “Aged under 65 years” these can be amalgamated into a single condition of “Not aged 65”. Also, any rate components which depend on both of these conditions being `true` or `false` should be examined for errors.

Taking a further example, if there are two columns giving the same rate component and the only difference is on a single condition where one contains `true` and the other `false`, they can be amalgamated into a single column with a don't care “-” value in that condition.

It should be noted that while this simplification is valid and important in reducing the amount of work required, it is very important to ensure that all circumstances which make a client ineligible for a product delivery be correctly determined by the conditions described in the decision table.

2.4 Example of Rule Identification

The following example illustrates the process for designing rules using a decision table. The example uses the personal rate component for a contributions-based old-age pension product.

The following is the relevant legislation for the contributions-based old-age pension product:

Conditions

“For Old Age Contributory Pension you must:

1. be aged 66 or over
2. have started paying contributions before reaching age 56
3. not be in receipt of any conflicting product.
4. have at least 156 weeks contributions paid (260 weeks full rate contributions paid if the yearly average is between 10 and 19 - see following condition)
5. have a yearly average of at least 48 weeks contributions paid or credited from 1979 to the end of the tax year before you reach age 66 - this will entitle you to a maximum pension (this applies only if you reached age 66 on or after 6 April 1992), or

   a yearly average of at least 10 weeks contributions paid or credited from 1953 (or the time you started insurable employment, if later) to the end of the tax year before you reach age 66.

Rates

A yearly average of 10 weeks contributions will entitle you to the minimum rate of Old Age Contributory Pension. For the maximum pension, an average of 48 weeks contributions is needed.

The current rate bands are:

1. Full Rate - Average 48+ contributions
2. Reduced Rate 1 - Average 36-4
3. Reduced Rate 2 - Average 24-35 contributions
4. Reduced Rate 3 - Average 15-23 contributions
5. Reduced Rate 4 - Average 10 Contributions

From the legislation, the following conditions can be identified:

The client must

1. Be aged 66 or over;
2. Not be in receipt of a conflicting product delivery;
3. Have commenced contributions before aged 56;
4. Have made at least 156 full rate contributions;
5. Have made an average of 48+ contributions;
6. Have made an average of 36-47 contributions;
7. Have made an average of 24-35 contributions;
8. Have made an average of 15-23 contributions;

9. Have made an average of at least10 contributions.

From the legislation, the following possible rate components can also be identified:

1. Full Rate;
2. Banded Rate for 36 to 47 contributions;
3. Banded Rate for 24 to 35 contributions;
4. Banded Rate for 15 to 23 contributions;
5. Banded Rate for 10 to 14 contributions.

Entering the condition combinations and the rate components results in the following decision table:

Decision Table					
Conditions					
Aged 66 or over	Y	Y	Y	Y	Y
Not in receipt of a conflicting product	Y	Y	Y	Y	Y
Commenced contributions before aged 56	Y	Y	Y	Y	Y
At least 156 full rate contributions	Y	Y	Y	Y	-
Average 48+ contributions	Y	N	N	N	N
Average 36-47 contributions	-	Y	N	N	N
Average 24-35 contributions	-	-	Y	N	N
Average 15-23 contributions	-	-	-	Y	N
Average 10-14 Contributions	-	-	-	-	Y
At least 260 full rate contributions	-	-	-	-	Y
Rate Components					
Full Rate	X				
BandedRate36to47		X			
BandedRate24to35			X		
BandedRate15to23				X	
BandedRate10to14					X

Table 2.1 Personal Rate Component Decision Table

Chapter 3

Identifying Evidence

3.1 Introduction

Evidence is the data on which a decision is based. In the Cúram rules environment, each piece of evidence has an association with a loader. A loader is a Java class that populates the Cúram Rules Data Object (RDO) with data from the database. The loader is in turn associated with a business process. Therefore, a rule is based on a piece of evidence that is retrieved during the running of the appropriate business process.

3.2 Identifying Evidence

The required evidence can be determined by analyzing the set of conditions and the rate components listed in the decision table. After the required evidence is collected, it can be matched to the business processes that it will be gathered from. The evidence can also be grouped to provide a logical view of the categories of evidence that are required.

The following sections describe the process for matching evidence to conditions, rate components, business processes, and evidence groups, respectively.

3.2.1 Identifying Evidence from Conditions

Based on the contributions-based old-age pension product described in the example in the previous chapter, the following list can be compiled:

Condition	Evidence
Aged 66 years or over	clientAge
Not in receipt of a conflicting product	clientRecievingConflictingServiceDeliverInd
Commenced contributions	clientDateOfEntry

Condition	Evidence
before aged 56	
At least 156 full rate contributions	totalFullRateContributions
Average 48+ contributions	averageContributions
Average 36-47 contributions	averageContributions
Average 24-35 contributions	averageContributions
Average 15-23 contributions	averageContributions
Average 10 contributions	averageContributions
At least 260 full rate contributions	totalFullRateContributions

Table 3.1 Conditions to Evidence Pairing

3.2.2 Identifying Evidence from Rate Components

Based on the contributions-based old-age pension product example, the following list can be compiled:

Rate Component	Evidence
Full Rate	fullPersonalRate
BandedRate36to47	bandedPersonalRate36to47
BandedRate24to35	bandedPersonalRate24to35
BandedRate15to23	bandedPersonalRate15to23
BandedRate10to14	bandedPersonalRate10to14

Table 3.2 Rate Component to Evidence Matching

3.2.3 Matching Evidence to Business Processes

After all required evidence has been identified, the evidence can be matched to the business processes that it will be gathered from. It it important to remember that the evidence used in a rule set is as important as the rules themselves. It must be ensured, therefore, that the business processes that are called to collect the evidence do exactly what they are expected to do.

Based on the contributions-based old-age pension product example, the following list can be compiled:

Evidence	Business Process
clientAge	ClientPersonalDetails

Evidence	Business Process
clientRecievingConflictingServiceDeliverInd	ClientInReceiptOfServiceDelivery
clientDateOfEntry	ClientContributionDetails
totalFullRateContributions	ClientContributionDetails
averageContributions	ClientContributionDetails
fullPersonalRate	ReadRateComponent
bandedPersonalRate36to47	ReadRateComponent
bandedPersonalRate24to35	ReadRateComponent
bandedPersonalRate15to23	ReadRateComponent
bandedPersonalRate10to14	ReadRateComponent

Table 3.3 Evidence to Business Process Matching

After matching the evidence to business processes, it is important to make sure that the business processes retrieve the evidence exactly in line with the expectations. For example, contributions may be calculated differently for different products and it would be important to ensure that the correct calculation is made.

It is also important to ensure that evidence is accurate over a time-line. The rule set will be called with different effective dates to assess eligibility and entitlement over time. It is vital that any business processes which give time-dependant results, for example, age calculations, take account of this fact.

3.2.4 Grouping the Evidence

The evidence can also be grouped to provide a logical view of the categories of evidence that are required. Based on the contributions-based old-age pension product example, the following list can be compiled:

Group	Evidence
Demographics	clientAge
Contributions	clientDateOfEntry, totalFullRateContributions, averageContributions,
Rate Components	fullPersonalRate, bandedPersonalRate36to47, bandedPersonalRate24to35, bandedPersonalRate15to23, bandedPersonalRate10to14

Table 3.4 Evidence Grouping

Chapter 4

Constructing a Rule Set

4.1 Introduction

A rule set is the fundamental structure used by Cúram to describe rules. A rule set is essentially a decision tree that is designed based on the conditions and rate components identified from the relevant legislation. This chapter describes the basic process for designing a rule set. It also covers key design factors. The chapter ends with a very brief overview of how rule sets and rules are implemented using the rules editor.

4.2 Grouping Rules and Rates

The grouping of rules and rates is a logical operation that makes future rules maintenance simpler. The main goal of grouping is to create groups that make sense in the context of the product to which the rules apply. Successful grouping can also have a significant impact on the numbers of rules required in a rule set.

Rate components can be grouped by “All of”, “First of“, “Highest value” and “Highest priority” operations, and rules can be grouped by “AND” and “OR” operations.

The easiest way to determine these groups is to examine the decision tables and identify the logical groups of conditions and rate components. To demonstrate this, the following table shows the decision table from the contribution-based old-age pension product example. The grouping is indicated for each condition and rate component.

Decision Table					
Conditions					
Product Level Rule Group: Aged 66 or over	Y	Y	Y	Y	Y
Product Level Rule Group: Not in receipt of a conflict-	Y	Y	Y	Y	Y

Decision Table					
ing product					
Product Level Rule Group: Commenced contributions before aged 56	Y	Y	Y	Y	Y
Rules to Rate Linkage Group: At least 156 full rate contributions	Y	Y	Y	Y	-
Rules to Rate Linkage Group: Average 48+ contributions	Y	N	N	N	N
Rules to Rate Linkage Group: Average 36-47 contributions	-	Y	N	N	N
Rules to Rate Linkage Group: Average 24-35 contributions	-	-	Y	N	N
Rules to Rate Linkage Group: Average 15-23 contributions	-	-	-	Y	N
Rules to Rate Linkage Group: Average 10-14 Contributions	-	-	-	-	Y
Rules to Rate Linkage Group: At least 260 full rate contributions	-	-	-	-	Y
Rate Components					
Rate Component Group: Full Rate	X				
Rate Component Group: BandedRate36to47		X			
Rate Component Group: BandedRate24to35			X		
Rate Component Group: BandedRate15to23				X	
Rate Component Group: BandedRate10to14					X

Table 4.1 Personal Rate Components Decision Table

4.3 Ordering Rules

The performance of a rule set can be strongly influenced by the ordering of the rules.

Generally, the rules that are most likely to fail should be placed before those that are less likely to fail.

It is also important that the rule group at the product level in the rules set contain all rules which apply to all rate components. This rule group is called the Product Level Rule Group. By executing first, it ensures that if one of the rules within it fails, that execution is halted. This prevents errors occurring at lower levels of the rule set and ensures that all required evidence has been collected. The Product Level Rule Group can also be used to ensure that cases that cannot qualify for the product delivery are caught quickly so that no unnecessary processing is conducted.

It should be noted, however, that some rules that would normally be placed in the Product Level Rule Group may be intentionally placed lower in the rule set. This is useful when the reason for failure is important to the user. If a rule is placed lower in the rule set, and the rule set fails on that rule, the user will be able to view the exact failure message.

4.4 Creating the Rule Set

This section discusses the main issues associated with adding rules to a rule set.

There are two key features which should be captured in any Rule Set. These are:

1. Correct Boundary Values

   Boundary values are important to the correct operation of a rule set and must therefore be precise. For example, if a person must be over sixty-six years of age, the condition must include the day of their sixty-sixth birthday.

2. Implicit Conditions

   Implicit conditions must be accounted for. Example of implicit conditions include the fact that certain persons associated with the product delivery must be alive and that any “certification period” conditions must be met. It is necessary to add explicit rules to cover implicit conditions as it should not be assumed that the conditions are being checked outside of the rule set.

There are a few design decisions that can strongly affect the efficiency of a Rule Set. The most important of these is Rule Placement. The rule most likely to fail should be placed first within rule set or rule group.

It is also important to conduct a transaction path analysis when constructing a rule set. The overall performance of the application will be strongly dependent on the performance of the database. It is therefore vital that a detailed examination for the database tables accessed during a rule set's execution is carried out.

4.5 Rule Set Implementation

Rule Sets, once designed, are implemented in Cúram using the Rules Editor, an application which is used to add rules to the Cúram application database. Very briefly, the process of implementing of rule set requires the following steps:

- In the modeling environment (Rational Software Architect), open the appropriate model by right-clicking on the model in the project explorer
- create Rules Data Objects (RDOs) to contain the evidence as data items

(RDO attributes, or members),

- build and start the Cúram application,
- log in to the Cúram application as administrator,
- go to the Rules Editor section in the Cúram application,
- create a new Eligibility Rule Set, and
- Write the code to populate the data items of the RDO used by the eligibility rules set.

For detailed information about rules implementation please refer to the `Cúram Rules Codification Guide` and the `Cúram Rules Editor Guide`.

4.6 Changing rules XML files in the application

4.6.1 Create new rule set

Create a new rules XML file using a text editor or an XML editor. All new files must be created under the `rulesets` subdirectory of the `EJBServer\components\custom` directory. If the `rulesets` directory does not exist, create it.

4.6.2 Change existing rule set

To change an existing rules XML file, copy it from its containing directory into the application's `EJBServer\components\custom\rulesets` directory. Make all changes to the copied file. The generators will generate from component directories in the order specified by the `SERVER_COMPONENT_ORDER` environment variable. The custom component has the highest priority so all output from lower priority components will be overwritten by the output from the custom component.

Chapter 5

Sub-rule sets

5.1 Introduction

A sub-rule set is similar to a rule set except that it cannot be called directly by the rules engine. It can only be called by other sub-rule sets or the main rule set itself.

A sub-rule set can contain any number of `input`, `output` or `local` RDOs. For more information on RDO Qualifiers see *Cúram Rules Codification Guide*

5.2 Sub-rule set Scope

The scope of the sub-rule set is controlled by the RDO qualifiers. A sub-rule set can have one of the following scopes:

- Local
- Global

5.2.1 Local Scope

A sub-rule set is executed in local scope (context) if it contains at least one qualified RDO. A sub-rule set that is executed in local context can only access the RDOs that are passed to it as input and output parameters and its local RDOs. Other sub-rule sets cannot access the values of it's RDOs unless they are passed as input RDOs or received via output RDOs.

5.2.2 Global Scope

A sub-rule set is executed in global scope (context) if it does not contain any qualified RDOs. A global context sub-rule set can access its own RDOs and all the RDOs defined both in its parent rulesets/sub-rule sets and child sub-rule sets.

5.3 Calling Sub-rule sets

Sub-rule sets are called the same way whether they are to be executed locally or globally i.e. by inserting the standard `SubRuleSetLink` element into the rule set/sub-rule set e.g.

```
<RuleGroup executionmode="executeall" operation="and">
  <Label>
    <RuleName>
      <Text value="Label"/>
    </RuleName>
  </Label>
  <!-- Call the sub-rule set as normal -->
  <SubRuleSetLink id="1" subrulesetid="SUBRULESET_001"/>
  <!-- More rules go here -->
</RuleGroup>
```

The presence of an RDO qualifier will signify that the sub-rule set is to be executed locally. The following rules snippet declares three RDOs one local, one input and one output.

```
<RuleSet>
  <SubRuleSet id="SUBRULESET_001" comment="">
    <Label>
      <RuleName>
        <Text value="Sub-rule set"/>
      </RuleName>
    </Label>
    <DataAccess name="Product" qualifier="local">
      <DataItem name="name"/>
      <DataItem name="price"/>
    </DataAccess>
    <DataAccess name="Person" qualifier="input">
      <DataItem name="name"/>
      <DataItem name="age"/>
    </DataAccess>
    <DataAccess name="Platform" qualifier="output">
      <DataItem name="operatingSystem"/>
      <DataItem name="osVersion"/>
    </DataAccess>
    <RuleGroup executionmode="executeall" operation="and">
      <Label>
        <RuleName>
          <Text value="Rule Label"/>
        </RuleName>
      </Label>
      <!-- More rules go here -->
    </RuleGroup>
  </SubRuleSet>
</RuleSet>
```

The data item values of the `Product` RDO cannot be seen outside of the sub-rule set.

The data item values of the `Person` RDO can be seen by the parent rule set/sub-rule set. *N.B. The input RDO must be declared in the parent rule set/*

sub-rule set

The data item values of the `Platform` RDO can be seen in the parent rule set/sub rule set, but only after the sub-rule set has finished executing and passed the values back to the calling rule set. *N.B. The output RDO must be declared in the parent rule set/sub-rule set*

Chapter 6

Rules Testing and Maintenance

6.1 Introduction

This chapter explains best practices for rules testing and maintenance.

6.2 Rules Testing

Testing can commence after rules have been entered and an application is generated. It is important that every possible circumstance is tested for and that all results are validated. It is also important to re-test all rules after even the smallest change has been made.

6.2.1 Testing tips

The process of testing the rules should be considered as part of the overall testing of the application environment, but the following are useful guides for the process of rules testing.

1. Use fixed test cases

   The use of fixed test cases ensures that any previously detected errors can be reproduced exactly and that if problems arise after testing, that test cases can be examined and expanded to cover the problem areas.

2. Know the “correct” result for any test case before running it.

   If the test case does not include the expected result it is very easy to miss the error by assuming that the returned result is correct.

6.2.2 Important Testing areas

The following areas are very important to cover when testing the rules:

1. Boundary cases

   The boundary cases are the most important areas to test. Boundary cases are where a client just meets or fails to meet a condition. Examples of this would include those clients just failing by being over an age condition on their birthday or missing/making a contribution condition by a single contribution unit.

2. High volume cases

   It is very important to ensure that the high volume cases or the “normal” conditions of a product delivery, are covered by the rules. It is, of course, equally important that any exceptions are dealt with correctly.

3. Multiple failures

   The circumstances where a case fails multiple conditions for a product delivery should be examined to ensure that a “sensible” failure message is returned and that the client is informed of the reason in a logical and coherent way.

6.3 Rules Maintenance

There are several issues to consider when maintaining a rule set. These include:

1. Version Control

   It is recommended that version control between development and release environments is strongly controlled and that a detailed policy for version control be drawn up in line with your business requirements. The versions of the rules in the release and development environments should be known and kept separate. It should also be possible to access historical information on the exact versions of the rules running at any point in time.

2. Clean-ups

   If changes are made to a rule set and these changes affect the original design decisions made, then it will be necessary to run a “clean-up” operation which should re-run the rules of the potentially affected cases and make the appropriate updates. There may also be legal issues associated with the changes to product deliveries that have already been made, which limits the extent to which any attempt to reclaim monies may be made.

3. Evidence Changes

   Changes to the evidence required to determine eligibility and entitlement to a product delivery also have knock-on effects on the test cases used and the operation of the system as a whole. The knock-on effects of any changes should be carefully considered to avoid problems when they are released.

4. Impact Analysis

   The impact of even the smallest change should be assessed to ensure that any knock-on effects are accounted for and that the implementation of the changes works smoothly.

After performing any maintenance on a rule set, it is important to re-test the rule set. Even the smallest change can have an unexpected affect on the performance of the rule set.

6.3.1 Maintenance "Don'ts"

The following list describes processes that should *not* be carried out during rules maintenance:

1. Do not modify a rule set in place if there are any case decisions records created using this rule set.

   The rule set is required to be present in its original form when displaying decision or evidence returned by the Rules Engine.

   This means that no modifications can be made to a rule set once it has been executed and that the results of any such execution need to be referenced (for example, by displaying a decision record using the Cúram Client application). Any modifications to such a rule set must be carried out on a cloned copy of the rule set.

2. Do not forget about evidence/business process changes.

   The impact of changes to evidence, or the business processes that retrieve them, can be as large as the impact of changes to rules themselves.

   As with rules, it is possible for several other artifacts to change over time. The evidence collected in respect of a case may expand; the business processes used to retrieve or calculate an RDO attribute may vary over time; and the data necessary to execute a rule may need to be retrieved from a different location. Similarly all of these changes can have effective dates. A new piece of evidence may only be collected from a particular point in time, but this is not to say that evidence collected before this point is wrong. All of these considerations must be taken into account when analyzing changes to be made to rule sets over time as they each will have an impact on RDO, Loader and Rules BPO design.

3. Do not overlook potentially required reassessments

   The reassessment operations resulting from changes to previously applied rule sets should never be overlooked as there are large data integrity issues arising from such processing.

   If any changes to rules artifacts have an effective date in the past it will be necessary to reassess all cases which have decisions based on that artifact. Therefore it will be necessary to analyze and evaluate the impact of such changes (to rules, RDOs, Loaders, BPOs) as appropriate.

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

Any references in this information to non-IBM 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 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