IBM Cúram Social Program Management
Version 7.0.2

Cúram Evidence Upgrade Guide

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 13

Contents

Chapter 1. Upgrading Evidence

Information is provided on how to upgrade Cúram evidence from older versions.

Introduction

Purpose

This document is intended to provide guidance to those customers upgrading their evidence solution from version 5.2 to version 6.0.

Each chapter will explain a different section of the user interface including how to develop and configure it. In addition, each chapter will detail how the Cúram Evidence Generator handles it.

Prerequisites

The reader should be familiar with the evidence capturing aspect of case management as well as its use in determining eligibility and entitlement on a case. They should also have read the Designing Cúram Evidence Solutions guide.

Audience

This document is targeted at a technical audience, both developers and architects, intending to upgrade their evidence solution.

Evidence Workspace

Introduction

The Evidence Workspace of a case, is the principal location where evidence is managed from. It comprises of several different views of the evidence, as well as functions for inserting, modifying and removing it.

With the move to a tabbed UI, the evidence workspace should normally reside in the content panel of a case tab. As UIM pages can only exist in one tab, the infrastructure provides a set of VIM files containing the functionality of the different pages in the workspace.

Each page of the workspace is typically provided as a Links VIM file as well as a View VIM file. The Links file contains the page level `ActionControl` elements while the View page contains the main implementation code. This separation allows the addition of any custom links that may be required.

Standalone Workspace Tab

A standard Evidence Workspace is provided with the system which opens in its own tab. This workspace can be used to manage evidence from any case type, by linking to one of its pages with the appropriate Case ID parameter.

The infrastructure pages that make up this Tab, use the same VIM files that are described in the following sections, so any customisations to these VIM files will affect the standalone workspace as well as any newly developed pages.

Implementation

Dashboard

The Dashboard provides a view of the evidence, grouped by category. The infrastructure provides the following files to support this:

Evidence_dashboardView.vim
    Main content

Evidence_dashboardLinks.vim
    Actions menu for the page.

Sample File

The following sample UIM file gives an example of how these VIMs could be used in a custom dashboard page

```
<PAGE
    PAGE_ID="MyProduct_dashboard"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd"
 >

    <!-- Page Title etc -->

    <ACTION_SET>

      <INCLUDE FILE_NAME="Evidence_dashboardLinks.vim"/>

      <!-- Add custom actions here -->
    </ACTION_SET>

    <INCLUDE FILE_NAME="Evidence_dashboardView.vim"/>
  </PAGE>
```

(deprecated) evidenceFlow

The evidenceFlow provides a graphical view of the evidence types, grouped by category.

The infrastructure provides the following files to support this:

Evidence_flowView.vim
    Main content

Evidence_flowLinks.vim
    Actions menu for the page.

Sample File

Note: EvidenceFlow is deprecated. For more information, see Deprecated features.

The following sample Cúram User Interface Metadata (UIM) file gives an example of how these VIMs could be used in a custom evidenceFlow page.

```
<PAGE
  PAGE_ID="MyProduct_flow"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd"
>

  <!-- Page Title etc -->

  <ACTION_SET>

    <INCLUDE FILE_NAME="Evidence_flowLinks.vim"/>

    <!-- Add custom actions here -->
```

```
      </ACTION_SET>

    <INCLUDE FILE_NAME="Evidence_flowView.vim"/>
  </PAGE>
```

Active

The Active Evidence page provides a view of the active evidence on the case. The infrastructure provides the following files to support this:

Evidence_workspaceActiveHighLevelView.vim
> Main content

Evidence_workspaceActiveHighLevelLinks.vim
> Actions menu for the page.

Sample File

The following sample UIM file gives an example of how these VIMs could be used in a custom active evidence page

```
<PAGE
  PAGE_ID="MyProduct_active"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd"
>

  <!-- Page Title etc -->

  <ACTION_SET>

    <INCLUDE
      FILE_NAME="Evidence_workspaceActiveHighLevelLinks.vim"
    />

    <!-- Add custom actions here -->
  </ACTION_SET>

  <INCLUDE
    FILE_NAME="Evidence_workspaceActiveHighLevelView.vim"
  />
</PAGE>
```

In Edit

The In Edit page provides a view of the in edit evidence on a case. The infrastructure provides the following files to support this:

Evidence_workspaceInEditHighLevelView.vim
> Main content

Evidence_workspaceInEditHighLevelLinks.vim
> Actions menu for the page.

Sample File

The following sample UIM file gives an example of how these VIMs could be used in a custom in edit page

```
<PAGE
  PAGE_ID="MyProduct_inEdit"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd"
>

  <!-- Page Title etc -->

  <ACTION_SET>
```

```
    <INCLUDE
      FILE_NAME="Evidence_workspaceInEditHighLevelLinks.vim"
    />

    <!-- Add custom actions here -->
  </ACTION_SET>

  <INCLUDE
    FILE_NAME="Evidence_workspaceInEditHighLevelView.vim"
  />
</PAGE>
```

Issues

The Issues workspace page provides a view of all the issues outstanding on the evidence on a case. The infrastructure provides a simple VIM with the functionality implemented. This can then be included in any UIM as normal.

Evidence_listIssuesForCaseView.vim
> Main content

Curam Verification

The Curam Verification pages provides views of the verifications for the case evidence. The infrastructure provides two views of the Verifications, one showing only outstanding Verifications, and the other showing all. These are provided as simple VIM files that can be included in any UIM as normal. Typically two pages would be provided and linked via in-page navigation tabs. The following files are provided to support this:

Evidence_listVerificationsForCaseView.vim
> Lists all outstanding verifications for the case.

Evidence_listAllVerificationsForCaseView.vim
> Lists all verifications on the case.

Evidence Generator

As the Evidence Workspace lives at the level of the case, no files are created by the Evidence Generator.

Evidence Type Workspace

Introduction

The Evidence Type workspace provides a complete view of all evidence objects of a given type. Functionality is also provided for managing the objects. It is linked to from the evidenceFlow control, while the standalone version provided by the infrastructure is linked from the other workspace list pages, and the dashboard page.

Typically the evidence type workspace is made up of 3 pages, the main list of evidence objects, the list of Verifications and finally the list of Issues.

Standalone Evidence Type Workspace

A standard Evidence Workspace is provided with the system which opens in its own tab. This workspace can be used to manage evidence from any case type, by linking to one of its pages with the appropriate Case ID and Evidence Type parameters.

The infrastructure pages that make up this Tab, use the same VIM files that are described in the following sections, so any customisations to these VIM files will affect the standalone workspace as well as any newly developed pages.

Implementation

To implement an evidence types own workspace, for use from the evidenceFlow widget, the 3 pages mentioned must be implemented, that is the main list page, the verifications and the issues list pages.

The following vim files are provided by the infrastructure to support evidence type workspaces

Evidence_typeWorkspace.vim
    Contains a list of evidence objects of a given type.

Evidence_listIssuesForEvidenceType.vim
    Lists all Issues for a given evidence type.

Evidence_typeVerifications.vim
    Lists all verifications for a given evidence type.

The main page should use the `Evidence_typeWorkspace.vim` and have one parameter called `caseID`. To include the other pages, they should be linked using in-page navigation.

The following sample file shows how to implement the main workspace page. The other pages follow a similar pattern.

```
<PAGE
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd"
  PAGE_ID="MyProduct_listMyEvidenceType"
>

  <PAGE_PARAMETER NAME="caseID"/>

  <CONNECT>
    <SOURCE PROPERTY="caseID" NAME="PAGE"/>
    <TARGET PROPERTY="key$caseID" NAME="DISPLAY"/>
  </CONNECT>
  <CONNECT>
    <SOURCE PROPERTY="SampleAsset.EvidenceType" NAME="CONSTANT"/>
    <TARGET PROPERTY="key$evidenceType" NAME="DISPLAY"/>
  </CONNECT>

  <!-- Page Title, Action Controls etc -->
  <MENU MODE="IN_PAGE_NAVIGATION">
    <!-- Link here to Issues and Verifications List -->
   </MENU>

  <INCLUDE FILE_NAME="Evidence_typeWorkspace.vim"/>

</PAGE>
```

Configuration

There are two ways of configuring the name of the type workspace page for a given evidence type. The older method of writing an `Evidence_resolveWorkspace.uim` or using the new database field `EVIDENCEMETADATA.WORKSPACEPAGENAME`.

When the system attempts to find the name of the type workspace page for a given evidence type it will first query the database, and if that returns blank it will then attempt to use the resolve script.

Evidence Generator

For generated evidence, an evidence type workspace will be created for each type under control of the generator. In addition, the correct resolve scripts will be generated such that there is no need to configure the system as described in this chapter.

The following files will be generated for evidence at the integrated case level:

<product.prefix>_list<Evidence Type>Evidence.uim
    Contains a list of evidence objects of a given type.

<product.prefix>_list<Evidence Type>EvidenceWSIssues.uim
  Lists all Issues for a given evidence type.

<product.prefix>_list<Evidence Type>EvidenceWSVerifications.uim
  Lists all verifications for a given evidence type.

Note: For evidence at the product delivery level, the files will be generated with the word `ProductEvidence` instead of `Evidence`

Create Pages

Introduction

There has been some major changes made to process around creating evidence, and hence some major changes to the pages in this area.

For standalone evidence types (those with no relationships) or those with no parent or pre association records, the changes are confined to having the page in a modal dialog. For more complex evidence types however, selection of related records such as parents, pre associations etc is now done via a wizard. In addition there may be multiple wizards required per evidence type depending on the number of relationships it has with others.

Note: There is no real evidence infrastructure support for these wizards (other than the standard evidence header details create VIM), as the number of related record lists differs from wizard to wizard.

Generally, each wizard requires two pages:

Selection Page
  This page allows the user to select related evidence objects (Parent, Pre Association etc)

Create Page
  The page where the actual information for the current entity is entered

Implementation

As mentioned above, when writing a wizard to handle the creation of an evidence record, you should try to keep it to two pages, the first for selecting related records, and the second for entering the evidence specific information. Both pages should be set to the same height using the WINDOW_OPTIONS attribute of the <PAGE> element.

Any evidence type that has higher level relationships that must be selected when it is being created, e.g. Parent, PreAssociation, must supply a default wizard that allows the user to select a suitable record for that relationship. This is the main create wizard and is used from all the high level workspaces such as EvidenceFlow, Dashboard, Active Evidence etc. As a result, the first page of this wizard should be considered as the evidence types main create page for purposes of configuration (see the next section).

As each evidence record can be viewed as a Business Object in its own tab, and any child evidence types can also be created from there, it follows that each evidence type must provide a suitable wizard or standalone create page that is suitable for use from these locations, where one of its relationships is already selected implicitly.

There are infrastructure methods in place to support these wizards such as the facade method `Evidence.listBusinessObjects` which will return a list of Business objects of a given type.

Example 1 - Type A has a child Type B
  Type A requires a standalone create page

  Type B requires a wizard that allows selection of a record of Type A.

  Type B also requires a standalone create page that takes as a parameter an evidenceID and evidenceType of an implicitly selected record. This way, when viewing an object of Type A, a child record can be created without the need to select a parent record.

Example 2 - Type A and Type B are both mandatory parents of Type C

> Type A requires a standalone create page
>
> Type B requires a standalone create page
>
> Type C requires a wizard that allows selection of a record of Type A and of Type B
>
> Type C requires a second wizard that only allows selection of Type A and takes an evidenceID and type of Type B as a parameter.
>
> Type C requires a third wizard that only allows selection of Type B and takes an evidenceID and type of Type A as a parameter.
>
> Type C does NOT require a standalone create page, as there is nowhere that both parents can be implicitly selected.

The following example code is of a wizard select page, using the single select widget. Note the use of an action phase server interface to read the latest Evidence ID, and Type from the chosen evidence object.

```
<PAGE
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  WINDOW_OPTIONS="height=400,width=700"
  xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd"
  PAGE_ID="MyProduct_createChildSelectParentEvidence"
>

  <!-- Page Title Etc -->

  <MENU MODE="WIZARD_PROGRESS_BAR">
    <CONNECT>
      <SOURCE
        PROPERTY="MyChildEntity.wizardMenu"
        NAME="CONSTANT"
      />
    </CONNECT>
  </MENU>

  <SERVER_INTERFACE
    CLASS="Evidence"
    OPERATION="listPossibleRelatedObjects"
    NAME="PARENTLIST"
  />

  <SERVER_INTERFACE
    CLASS="MyProductEvidenceFacade"
    NAME="PARENTPARAMS"
    OPERATION="getEvidenceAndCaseFromSuccession"
    PHASE="ACTION"
  />

  <LIST>
    <CONTAINER WIDTH="5">
      <WIDGET TYPE="SINGLESELECT">
        <WIDGET_PARAMETER NAME="SELECT_SOURCE">
          <CONNECT>
            <SOURCE
              PROPERTY="result$dtls$successionID"
              NAME="PARENTLIST"
            />
          </CONNECT>
        </WIDGET_PARAMETER>
        <WIDGET_PARAMETER NAME="SELECT_TARGET">
          <CONNECT>
            <TARGET
              PROPERTY="key$successionID"
              NAME="PARENTPARAMS"
            />
```

```
                </CONNECT>
              </WIDGET_PARAMETER>
            </WIDGET>
        </CONTAINER>

        <!-- Display Fields -->
    </LIST>

    <ACTION_SET ALIGNMENT="RIGHT">

        <ACTION_CONTROL
            ALIGNMENT="LEFT"
            LABEL="ActionControl.Label.Cancel"
            IMAGE="CancelButton"
        />

        <ACTION_CONTROL
            TYPE="SUBMIT"
            LABEL="ActionControl.Label.Next"
            IMAGE="NextButton"
        >
            <LINK
                SAVE_LINK="FALSE"
                DISMISS_MODAL="FALSE"
                PAGE_ID="MyProduct_createChildFromWizard"
            >
                <CONNECT>
                    <SOURCE PROPERTY="caseID" NAME="PAGE"/>
                    <TARGET PROPERTY="caseID" NAME="PAGE"/>
                </CONNECT>
                <CONNECT>
                    <SOURCE
                        PROPERTY="evidenceKey$evidenceID"
                        NAME="PARENTPARAMS"
                    />
                    <TARGET PROPERTY="parEvID" NAME="PAGE"/>
                </CONNECT>
                <CONNECT>
                    <SOURCE
                        PROPERTY="evidenceKey$evType"
                        NAME="PARENTPARAMS"
                    />
                    <TARGET PROPERTY="parEvType" NAME="PAGE"/>
                </CONNECT>
            </LINK>
        </ACTION_CONTROL>
    </ACTION_SET>

</PAGE>
```

Configuration

In order to ensure that the system is able to open the correct create page for an entity, it should be configured in the `EVIDENCEMETADATA.CREATEPAGENAME`, or else the Evidence_resolveCreate.uim resolve script should be implemented.

When the system attempts to find the name of the create page for a given evidence type it will first query the database, if that returns blank it will then attempt to use the resolve script

Note: As an entity can have multiple create wizards depending on the relationships involved, or even have a wizard and a standalone create page, the page that should be configured is either the standalone create page if no wizards exist, or else the select page of the most general wizard available.

Evidence Generator

For generated evidence, all wizards and create pages will be created automatically, along with all required resolve scripts, so no manual configuration is required.

View Page

Introduction

The main view page is shown as a modal dialog which is launched from the various workspace list pages.

The modal consists of two pages, the Details page, showing the evidence information and the Correction History page, which shows a list view of the updates to this individual evidence record. An additional page for viewing the history record is launched from this list, showing details of the previously recorded information.

Implementation

The following vim files are provided by the infrastructure to support the view modals

Evidence_viewHeaderForModal.vim
Contains evidence header view information.

Evidence_listCorrectionHistory.vim
Contains a list showing the full correction history of an individual evidence record.

The Details page should use the new `Evidence_viewHeaderForModal.vim` file to display the evidence records header information while the correction history page should make use of the `Evidence_listCorrectionHistory.vim`.

The additional view history record page should basically be the same as the normal Details page, but without any links to the correction history list.

```
<PAGE
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd"
  PAGE_ID="MyProduct_viewMyEvidenceType"
  WINDOW_OPTIONS="height=400"
>

  <PAGE_PARAMETER NAME="caseID"/>
  <PAGE_PARAMETER NAME="evidenceID"/>
  <PAGE_PARAMETER NAME="evidenceType"/>

  <!-- In Page Navigation links -->

  <INCLUDE FILE_NAME="Evidence_viewHeaderForModal.vim"/>

  <!-- Evidence Fields -->

</PAGE>

<PAGE
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="file://Curam/UIMSchema.xsd"
  PAGE_ID="MyProduct_viewMyEvidenceTypeCorrectionHistory"
  WINDOW_OPTIONS="height=400"
>

  <!-- In Page Navigation links -->

  <INCLUDE FILE_NAME="Evidence_listCorrectionHistory.vim"/>
```

```
</PAGE>
```

Configuration

Configuration of the view page remains the same with just the `EVIDENCEMETADATA. VIEWPAGENAME` to be filled in.

Configuration of the View History Record page can be done via the `EVIDENCEMETADATA. HISTORYRECORDPAGENAME`, or alternatively by implementing the `Evidence_resolveHistoryRecord.uim` resolve script.

Evidence Generator

For generated evidence, the Details view page, Correction History list page and View History Record page are created for each type under control of the generator. In addition, the correct resolve scripts will be generated such that there is no need to configure the system as described in this chapter.

The following files will be generated for evidence at the integrated case level:

<product.prefix>_view<Evidence Type>Evidence.uim
Details view page

<product.prefix>_view<Evidence Type>Evidence_content.vim
Content for Details view

<product.prefix>_view<Evidence Type>EvidenceCH.uim
Correction History list view page

<product.prefix>_view<Evidence Type>EvidenceHistRec.uim
Details view for correction history version.

Note: For evidence at the product delivery level, the files will be generated with the word `ProductEvidence` instead of `Evidence`

Modify Page

Introduction

The modify page is shown as a modal dialog which is launched from the various workspace list pages.

Very little has changed with evidence modify pages, other than a new header details VIM file has been provided and the old one deprecated.

Implementation

The following vim files are provided by the infrastructure to support the modify modals

Evidence_modifyHeader1.vim
Contains evidence header details for modification.

Existing evidence modify pages should simply be updated to use this new file instead of the previous one, which is now deprecated.

Configuration

There has been no changes to the configuration, with just the `EVIDENCEMETADATA. MODIFYPAGENAME` to be filled in.

Evidence Generator

For generated evidence, the created modify pages will now automatically include the new header file.

Business Object Tab

Introduction

Every evidence type should provide a tab that is linked to from within the various high level workspaces and the evidence types own workspace. This tab provides a view of everything related to that particular evidence business object (or succession).

Typically the Business Object Tab contains the following navigation level tabs:

Home
    shows a view of the latest record in the succession

History
    lists all evidence records in the succession

Verifications
    Shows all verifications for the business object

Issues
    Shows all issues for the business object

Related Tabs
    One for each related evidence type showing all related records of that type

Implementation

The tab should be written such that each page on it takes as a parameter the business object ID (successionID). Implementation of the navigation tab level pages, with the exception of the home page, typically just involves creating a page that includes a provided infrastructure vim, this is because a page can only be associated with a single Tab, so therefore the Evidence Infrastructure can only provide these pages as content VIM files.

Note: The home pages will normally be very similar to the standard view pages, with the exception of needing to be based off the successionID, rather than the more usual evidence id. The intention would be for this page to show details of the latest record in the succession. The facade method `Evidence.getEvidenceAndCaseFromSuccession` returns the latest evidenceID and type for the supplied successionID.

The following vim files are provided by the infrastructure to support the business object tabs:

Evidence_viewHeaderForModal.vim
    Contain evidence header details for display on the Home page

Evidence_listEvdInstanceChangesView.vim
    Contains a list of all records in the succession.

Evidence_listIssuesForObject.vim
    Contains a list of Issues for the evidence object.

Evidence_listVerificationsForObject.vim
    Contains a list of verifications for the evidence object.

Evidence_typeWorkspace.vim
    Used for all Related entity tab pages.

Configuration

In order to ensure that the system is able to open the business object tab for a given evidence type, you need to configure the main page of the tab (The Home page) so that the system can link to it from the correct locations.

There are two ways of configuring the name of the home page for a given evidence type. The older method of writing an Evidence_resolveObject.uim, or using the new database field `EVIDENCEMETADATA.BUSINESSOBJECTPAGENAME`.

When the system attempts to find the name of the business object home page for a given evidence type it will first query the database, if that returns blank it will then attempt to use the resolve script

Evidence Generator

For generated evidence, the business object tab, and its associated pages will be created automatically, along with all required resolve scripts, so no manual configuration is required.

The following tab pages will be generated for each type:

<Entity Name>Object.tab
Main Tab Description page

<Entity Name>Object.nav
Describes the navigation tabs for the generated main tab.

<Entity Name>Object.properties
Localised properties for navigation tab titles etc

<Entity Name>ObjectMenu.mnu
Main tab level actions menu.

<Entity Name>ObjectMenu.properties
Localised properties for the actions menu.

Additionally, the following UIM pages will be generated for each type:

<product.prefix>_view<Entity Name>EvidenceObject.uim
Main tab home page

<product.prefix>_view<Entity Name>EvidenceObject_content.vim
Content for the main tab home page.

<product.prefix>_view<Entity Name>EvidenceObjectCH.uim
Change History of the evidence object.

<product.prefix>_list<Entity Name>EvidenceObjIssues.uim
List of issues for the evidence object.

<product.prefix>_list<Entity Name>EvidenceObjVerifications.uim
List of verifications for the evidence object.

<product.prefix>_list<Related Entity Name>EvidenceFor<>Obj.uim
List of objects of type <Related Entity Name> that are related to this object.

Notices

This information was developed for products and services offered in the United States.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Privacy Policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies or other similar technologies that collect each user's name, user name, password, and/or other personally identifiable information for purposes of session management, authentication, enhanced user usability, single sign-on configuration and/or other usage tracking and/or functional purposes. These cookies or other similar technologies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at " Copyright and trademark information " at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

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

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

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

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

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

IBM ®

Part Number:

(1P) P/N: