merative™

Cúram 8.1.3

Cúram Platform Analytics Accelerator Guide

Note

Before using this information and the product it supports, read the information in

Edition

This edition applies to Cúram 8.1, 8.1.1, 8.1.2, and 8.1.3.

Contents

1 Cúram Platform Analytics accelerator

Cúram Platform Analytics (SPMP Analytics) provides business intelligence and analytics to support performance management, and analytics. Organizations can use the SPMP Analytics functions to measure and monitor the performance of an organization to detect gaps in processes, and to analyze current issues.

1.1 Overview of the SPMP Analytics accelerator

The Cúram Platform Analytics (SPMP Analytics) accelerator provides a straightforward, worked example of building analytics using BIRT. This accelerator builds on the existing Adult Social Care accelerator and provides a number of extra insights on Cúram Platform. It is designed to familiarize you with how to use SPMP Analytics to meet your business intelligence requirements.

The SPMP Analytics business scenario

The goal of the SPMP Analytics scenario is to help a caseworker analyze issues that are related to their assigned cases and to understand the reasons behind them.

A typical SPMP Analytics scenario can address the needs of Jane, an Adult Social Care caseworker. Jane has a caseload of more than 20 cases that provide benefit payments to incapacitated individuals.

Jane can view two critical caseload analytics from her user application view.

- The number of Sickness Benefit cases versus Incapacity Benefit cases is displayed as a bar chart. From the bar chart, the first bar shows the total number of active Sickness Benefit cases while the second bar shows the total number of Incapacity Benefit cases. Only cases that are assigned to the caseworker are displayed. The chart helps the caseworker understand their current caseload and to process their cases in a timely fashion. The analytic provides a view showing the number of shorter term and less costly Sickness Benefit cases versus more costly long running Incapacity Benefit cases.

- The number of clients with multiple diagnoses is displayed as a pie chart. The chart shows the total number of clients who are diagnosed with more than one medical condition and who are currently in receipt of Sickness Benefit. The pie chart shows the total number of clients with multiple diagnoses as a percentage of the total number of clients on Sickness Benefit. The chart displays clients on cases that are assigned to the caseworker. The chart helps the caseworker understand which clients have multiple diagnoses and are at risk of moving onto long term Incapacity Benefit cases.

These reports allow Jane to understand issues that are related to her cases and resolve them, if necessary.

The Cúram Platform Analytics business process

A caseworker can use the Adult Social Care application to view and monitor information about benefit cases that are made to clients on their assigned cases.

Figure 1: The Cúram Platform Analytics overview diagram
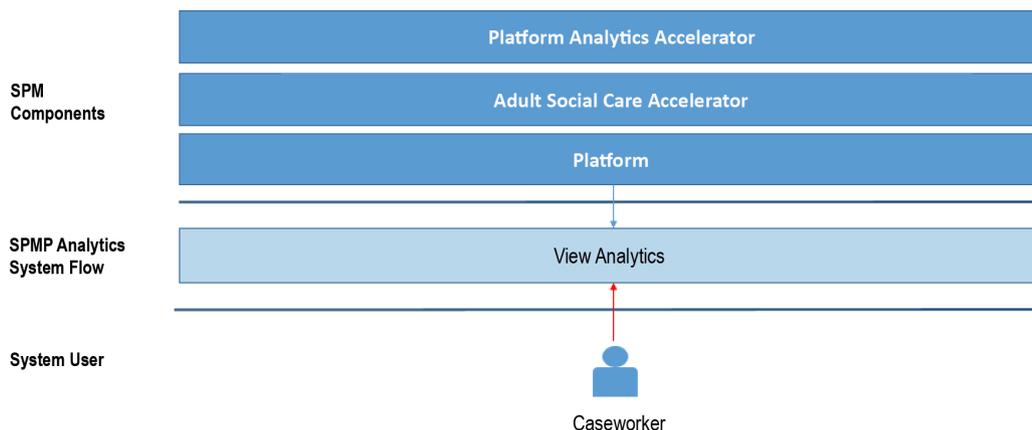
The business process uses the following components to satisfy the requirements:

- **Cúram Platform**
  This platform provides the processing and components necessary to enable the business process. The Cúram Case Management component helps to search for a client and to view their client details and case information.

- **Adult Social Care accelerator**

  Adult Social Care is a simple social program application that is built on top of the Cúram Platform.

- **Cúram Platform Analytics accelerator**
  Cúram Analytics accelerator is based on the Adult Social Care accelerator. It demonstrates how to customize an existing BIRT chart and how to create new analytics with Cúram by using the SPMP Analytics infrastructure.

Choosing the right analytic processing landscape

There are a number of factors you must consider when you decide to use SPMP Analytics or SPMP Business Intelligence and Analytics (BI&A) to meet your project's business analytical requirements. SPMP Analytics are aimed at small to midsize organizations with a small volume of data and fewer analytic workloads. SPMP BI&A is aimed at midsize and large organizations that have larger volumes of data, more system users, and more complex analytic requirements.

Identifying the right analytic processing infrastructure
A key strategic requirement for those clients who must finance, assemble, and deploy analytics is to ensure that processing infrastructures can service current and future workloads. Ultimately, you must decide on the most effective analytic processing environment based on your local project needs. The objective is to ensure that your analytic processing infrastructure aligns with your workloads.

The size and complexity of your current and projected analytic workloads must not affect the responsiveness of the Cúram application. SPMP Analytics provides a low cost easy to deploy environment to support standard analytic requirements. If you must ensure maximum application throughput, and support complex analytic requirements, then you must consider SPMP BI&A.

To determine your organization's analytics workload, you must consider the analytical requirements for each user role in your organization. This includes front line staff, for example, caseworkers, and managers. Every user wants timely, relevant, accurate, and consistent data and analysis, but each user might define those terms differently.

You must consider many characteristics when you review the complexity of your organization's analytic workload. The following points contain some of the key analytic workload markers that highlight the need for BI&A on your project.

- Systems with larger data volumes.
- Systems with complex data models.
- Systems where ad hoc queries are required.
- Systems where the computational complexity and data traversal paths cannot be anticipated or optimized.
- Systems where any queries can take many tens-of-seconds, or any processing that would otherwise affect application throughput.
- Landscapes where data must be staged, cleansed, or transformed.
- Systems where the current analytic workload will increase over time.
- Systems where the Cúram application service level agreements for case processing throughput.

- Systems where many different user groups require analytics. The following contains some examples.

  - The caseworker role typically needs to make day-to-day or real-time decisions within their area of responsibility. They require quick access to current information that can be seen from their workspace. Typically an organization can have many hundreds or thousands of caseworkers, with caseworkers viewing analytics on a frequent basis. These queries can have a significant impact on system performance. A combination of the frequency of analytic queries and the number of caseworkers can result in a large workload on the Cúram application, which can negatively affect application performance throughput. In this scenario, SPMP BI&A might be a better fit for the organization.

  - The supervisor role typically requires access to view current performance data as summary snapshots and as trends on Key Performance Indicator (KPI) dashboards. The supervisor KPI dashboards need to display performance indicators at the consolidated business unit and department levels. KPI dashboards enable the users to drill down into activity analytics to examine where activities are diverging from plans and standards of performance that are required for the enterprise to meet its set goals. As the supervisor might be responsible for managing a business unit or department with many cases, a supervisor can have hundreds of caseworkers assigned to them who would be running analytics against many cases. In this scenario, moving the workload to SPMP BI&A might be a better fit for the organization.

See also the analysis results for the analytics business scenario..

Related concepts
[Analysis results for the SPMP Analytics requirements on page 33](#)
Analysis of Fiona's requirements resulted in several key decisions. Both the business and technical decisions are discussed.

SPMP Analytics architecture

There are a number of landscape, architecture, and design decisions you must consider when you decide to use SPMP Analytics to meet your project's business analytical requirements. SPMP Analytics provides an architecture that supports performance and scalability.

The Cúram BIRT Viewer application is designed as a Java Platform, Enterprise Edition application that can be deployed in a clustered environment. Deploying SPMP Analytics in a clustered environment allows your organization to scale your analytic processing landscape by using application server clustering technologies. You can scale your landscape through application server configuration without the need to write new application code.

There are a number of deployment options for the Cúram BIRT Viewer application.

- Deploying the Cúram BIRT Viewer application on the same cluster as the Cúram application. This deployment is the default configuration.
- Deploying the BIRT viewer application on a separate cluster as the Cúram application. This configuration supports greater fault tolerance and allows both the Cúram application and the BIRT viewer application to scale independently of each other.

Summary of SPMP Analytics

Cúram Platform Analytics (SPMP Analytics) provides business intelligence and analytics to support core business intelligence, performance management, and analytics. Organizations can

use the SPMP Analytics functions to measure and monitor the performance of an organization, to detect gaps in processes, and to analyze current issues.

SPMP Analytics provides

- A runtime execution environment for running reports that are written with the Business Intelligence Reporting Tool (BIRT) system environment.
- Administration functions to help configure both the BIRT runtime environment and individual reports.
- Infrastructure to help integrate BIRT reports into application pages.
- 20 plus operational analytic charts that are embedded in application pages.

The default analytics provide decision support information to caseworkers and managers in the organization.

Default operational charts

SPMP Analytics includes a number of operational charts that can be displayed on the user workspace. The default charts cover audit, case financials, caseload summary information, investigation summary information, participant information, and workflow requirements. The purpose of these charts is to assist front line staff and managers in the decision making process.

Related concepts

Analyzing the default functionality on page 18
The following tables include descriptions for all the default SPMP Analytics charts that can be displayed on the user workspace.

Business Intelligence Reporting Tool

The Business Intelligence Reporting Tool (BIRT) is an open source technology platform. It is used to create charts and traditional reports, for example, a bank statement, that can be embedded into rich client and web applications.

BIRT is a low cost option for organizations that do not have elaborate business intelligence requirements. Since BIRT is included in SPMP, it can be used to develop analytics where the data is stored within the Cúram database or the Cúram Data-warehouse. BIRT provides interactive, summarized information in context.

BIRT allows for analytics to be integrated into the Cúram application. Analytical charts are displayed on user workspace pages, context panels, and dashboards.

Summary of SPMP Business Intelligence and Analytics

Cúram Business Intelligence and Analytics (BI&A) is a separately licensed product that provides extra features on top of SPMP Analytics. BI&A addresses the needs of organizations with complex business analytics requirements. The extra features include data warehouse capabilities, a My Reports tab on the user application views, interactive dashboards with multiple charts displayed on each dashboard, and extra operational charts. The BI&A product is not covered in this accelerator.

1.2 Installing the SPMP Analytics accelerator

The following sections describe how to install the SPMP Analytics accelerator. The accelerator is available for download only to members of Merative, such as Lab Services developers who are

assigned to work on a Cúram project on a customer site. However, you can follow the tasks in Building the SPMP Analytics accelerator to understand how the accelerator was built.

Installing the prerequisite software

You must follow the prerequisites before you install the accelerator. Make sure to follow the prerequisites and the installation steps that are specified in the release notes for each of the installers.

Procedure

1. Install an Application Development Environment (ADE).

2. Install Cúram.

3. Install the Adult Social Care accelerator and the SPMP Analytics accelerator. The accelerators are available for download only to members of Merative such as Lab Services developers who are assigned to work on a Cúram project on a customer site.

4. Extract both the accelerators at the root of the Cúram installation *%CURAM_DIR%*, which is defined by the *%CURAM_DIR%* environment variable. By default, the environment variable is set to *C:\Merative\Curam\Development.*
   The build steps are combined for building the Adult Social Care accelerator and the SPMP Analytics accelerator therefore you do not need to follow the steps for installing Adult Social Care.

Installing and building the accelerator

After you download the accelerator, complete the following steps to install and build it. The build steps are combined for building the Adult Social Care accelerator and the SPMP Analytics accelerator.

Procedure

1. Extract the SPMP Analytics accelerator at the root of the Cúram installation, which is defined by the *%CURAM_DIR%* environment variable. By default, the environment variable is set to *C:\Merative\Curam\Development.*

2. From the *%CURAM_DIR%* directory, update the *SetEnvironment.bat* or *SetEnvironment.sh* files.

3. Edit the *SERVER_COMPONENT_ORDER* variable to add the following components directly after the custom entry.

   ```
   SPMAnalytics,AdultSocialCare,ASCIntakeConfiguration
   ```

4. Edit the *CLIENT_COMPONENT_ORDER* variable to add the following components directly after the custom entry.

   ```
   SPMAnalytics,AdultSocialCare,IncapacityBenefit
   ```

5. Edit the `%CURAM_DIR%\EJBServer\project\config`
   `\deployment_packaging.xml` file to update the Cúram EAR components element,
   adding the following components directly after the custom entry.

   ```
   SPMAnalytics,AdultSocialCare,IncapacityBenefit
   ```

6. From your Cúram server development installation directory, `%CURAM_DIR%\EJBServer`,
   run the following commands.

   ```
   build clean server database
   ```

   ```
   build runExtractor createClasspaths
   ```

7. From your Cúram web client development installation directory, `%CURAM_DIR%`
   `\webclient`, run the following command.

   ```
   build clean client
   ```

8. From your Cúram BI content directory development installation directory, `%CURAM_DIR%`
   `\BIContent`, run the following command.

   ```
   build client.birt
   ```

9. Refresh the workspace within the Cúram Application Development Environment (ADE) and
   start the application as normal.

   For more information about how to install a development environment, see the *Development
   Environment Installation Guide*.

Starting the XML server

Before you start the Cúram application, you must start the XML server in your ADE.

Procedure

1. Change to the `%CURAM_DIR%\CuramSDEJ\xmlserver` directory.

   `%CURAM_DIR%` is the Cúram installation directory, which by default is ▪Windows▪ `C:`
   `\Merative\Curam\Development` or ▪Linux▪ `opt/merative/curam/`
   `development`.

2. Run the following build target to start the XML server:

   ```
   ant -f xmlserver.xml
   ```

Setting up Eclipse with the Cúram BIRT Viewer

Complete the following steps to import the Cúram BIRT View in your eclipse workspace. This
ensures that your BIRT analytic charts are displayed within application pages. Application

developers who write Java code might not need the following configuration, however, for analytics developers the following steps must be completed.

Procedure

1. From Eclipse, select **File** > **Import**.

2. From the **Import** window, select **Existing Projects into Workspace** and click **Next**.

3. Select the root directory to be `%CURAM_DIR%\BIApp\CuramBIRTViewer` and click **OK** > **Finish**.

    `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

4. From Eclipse, select the `CuramBIRTViewer` project. Right-click and select **Tomcat project** > **Update context definition**.

5. From **Window** > **Preferences** > **Tomcat** > **JVM Settings Classpath** section, click **Directory**.

6. Select the full path to `%CURAM_DIR%\EJBServer\project\properties` and click **OK**.

Testing the installation

Complete the following step to verify that the BIRT charts were built successfully and are included in the list of available reports.

Before you begin

Ensure that you ran the following build target. From the `%CURAM_DIR%\BIContent` directory, run the following command **build client.birt.**`%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

Procedure

1. From Eclipse, start the Cúram server, the RMILoginClient, and Tomcat as normal.

2. From the **RMILoginClient** window, enter the **ascewuser** user credentials.

3. From your browser, open the URL and enter `http://localhost:9080/Curam/AppController.do`. From the Eligibility Worker's home page, two new pods are displayed with BIRT analytic charts.

Table 1: PODs with BIRT chart

POD Name	Description
My Assigned Active Benefit Cases	Displays the number of cases that are receiving Sickness Benefit and Incapacity Benefit in a bar chart.
Sickness Benefit Cases By Number of Diagnosis	Displays the number of clients, which have one or more diagnoses for a Sickness Benefit case in a pie chart.

4. From your browser, enter the following URL `http://localhost:9080/CuramBIRTViewer/List`. Verify that this URL displays a list of available reports for each of the components. The first component in the list is as follows:

Component	Name
`components/core/birt/CaseAudit/`	**AuditPlanFocusAreaReport.rptdesign**

1.3 Performing the SPMP Analytics scenario

The SPMP Analytics scenario allows the caseworker to view benefit case analytics from the users workspace. The scenario provides the steps that are required to view the analytics.

Viewing the analytics on the users home page

The user workspace allows clients to view the number of Sickness Benefit cases versus the number of Incapacity Benefit cases in a bar chart. The number of clients with multiple diagnoses is displayed as a pie chart.

Procedure

From your browser, open the URL and enter `http://localhost:9080/Curam/AppController.do`. From the Eligibility Worker's home page, two new pods are displayed with BIRT charts.

Table 2: PODs with BIRT chart

POD Name	Description
My Assigned Active Benefit Cases	Displays the number of cases that are receiving Sickness Benefit and Incapacity Benefit in a bar chart.
Sickness Benefit Cases By Number of Diagnosis	Displays the number of clients, which have one or more diagnoses for a Sickness Benefit case in a pie chart.

1.4 Analyzing the SPMP Analytics functionality

Use the following practices for analyzing the default system functions against the customer's analytics requirements.

Completing a fit gap analysis

You can use the following guidance to complete a fit gap analysis between the project requirements and the default application.

Fit gap analysis scenario

Fiona, a business analyst, is asked to do a fit gap analysis of the differences between the default SPMP analytics and the customer's project requirements. Fiona is focusing on the requirements for benefit case analytic charts to be displayed on the user workspace.

The customer's project requirements are:

- The system must display a bar chart on the user workspace with the number of Sickness Benefit cases versus the number of Incapacity Benefit cases. The first bar shows the total number of active Sickness Benefit cases. The second bar shows the total number of Incapacity Benefit cases. The chart displays cases that are assigned to the caseworker.

- The system must display a pie chart on the user workspace with information about clients with multiple diagnoses. The chart must show the total number of clients who are diagnosed with more than one medical condition and who are currently in receipt of a Sickness Benefit. The pie chart shows the total number of clients with multiple diagnoses as a percentage of the total number of clients with Sickness Benefits. The chart displays clients on cases that are assigned to the caseworker.

When performing a fit gap analysis, you need to determine which features are fully available, what can be customized, and which require new development, so that project costs can be assessed. It is important to note that customizing existing BIRT analytic content requires less effort than creating new BIRT content. It is important to identify charts that can be leveraged to meet your needs, as this will reduce the project costs.

Analyzing the default functionality
The following tables include descriptions for all the default SPMP Analytics charts that can be displayed on the user workspace.

A description of each table column is provided.

- Report business name - The business name of the report.
- Report technical name - The technical name for the report that is used by BIRT. The report technical name is the physical file name with an extension of rptdesign.
- Description - A business description of the SPMP Analytics charts.
- Database table name and attribute - The Cúram database table name and attributes that are used by the report. The system uses these data items to run a query for each report object.
- Dynamically generated data items - In some cases data items for a report are generated dynamically at run time. These data items are not stored in the Cúram database but can be created from data that is available. This is possible by using SQL and/or JavaScript programming within BIRT reports. The ability to use a programming language within BIRT reports is one of the benefits of using BIRT. It allows technical users to create new data items from existing data without the need for Java development or new Curam API's.

Table 3: Case Audit Reports for the Audit Coordinator user role

Report Business Name	Report Technical Name	Description	Database table name and attribute	Dynamically Generated Data Items
Audit Plan Summary Pie Chart	AuditPlanSummaryPieChart	The pie chart shows a breakdown of the cases that are being audited. Cases are categorized into the following statuses: Not yet examined, Satisfied, and Not satisfied.	FocusAreaFinding.areaSatisfied and CaseAudit.caseAuditStatus	The table contains 1 dynamically generated data item: Overall Case Audit Status (not yet examined, satisfied or not satisfied).

Report Business Name	Report Technical Name	Description	Database table name and attribute	Dynamically Generated Data Items
Case Audit Summary Pie Chart	CaseAuditSummaryPieChart	The pie chart shows a breakdown of the cases that are assigned to an auditor, which are categorized by status. Examples: Open, Assigned, Findings complete, and Awaiting feedback, Feedback received, Feedback Complete, and Complete.	CaseAudit.caseAuditStatus	This analytic contains 1 dynamically generated data item: Overall Case Audit Status (Pending, In Progress, Complete, Delayed Processing Pending).
Case Audit Focus Areas Chart	CaseAuditFocusAreas	The chart shows a breakdown of the cases that are assigned to a specific auditor. Cases are categorized by status: Satisfied and Not satisfied.	CaseAudit.caseAuditStatus	This analytic contains 1 dynamically generated data item: Overall Case Audit Status (Satisfied and Not Satisfied).
Case Audit Summary Bar Chart	CaseAuditSummaryBarChart	The bar chart shows a breakdown of the case audits, which is categorized into the following statuses: Open, Assigned, Findings Complete & Awaiting Feedback, Feedback Received, Feedback Complete, and Complete.	CaseAudit.caseAuditStatus	This analytic contains 1 dynamically generated data item: Overall Case Audit Status (Pending, In Progress, Complete, Delayed Processing Pending).
Audit Plan Summary Bar Chart	AuditPlanSummaryBarChart	The bar chart shows a breakdown of the case audits, which is categorized into the following statuses: Pending, In Progress, Complete, Delayed Processing Pending.	AuditPlan.AuditPlanStatus	This analytic contains 1 dynamically generated data item: Overall Case Audit Status (Pending, In Progress, Complete, Delayed Processing Pending).
Audit Plan Focus Area Report	AuditPlanFocusAreaReport	The chart shows a breakdown of the case audits, which are categorized into the following statuses: Not yet examined, Satisfied, and Not satisfied	FocusAreaFinding.areaSatisfied and CaseAudit.caseAuditStatus	This analytic contains 1 dynamically generated data item: Overall Case Audit Status (Not Yet examined, Satisfied, and Not Satisfied).
Case Audit Plan Progress Summary Chart	AuditPlanProgressBarChart	The bar chart shows a breakdown of the case audits, which is categorized into the following statuses: Open, Assigned, Findings Complete & Awaiting Feedback, Feedback Received, Feedback Complete, and Complete.	CaseAudit.caseAuditStatus	This analytic contains 1 dynamically generated data item: Overall Case Audit Status (Open, Assigned, Findings Complete & Awaiting Feedback, Feedback Received, Feedback Complete, and Complete).

Table 4: Case Financial Reports for the Superuser role

Report Business Name	Report Technical Name	Description	Database table name and attribute	Dynamically Generated Data Items
All Components Graphical Benefit Statement	AllComponentsGraphicalBenefitStatement	This report compares the actual amount that was paid for a case to the reassessed amount that should have been paid.	OverUnderPaymentBreakdown.amount	None
Accounts by Payment Correction Type Report	PaymentCorrectionTypeFinancialReport	This report shows a total of the over payments for a case that is divided into the following categories: allocated payments, outstanding balance, and write off total.	InstructionLineItem.amount	The analytic contains 1 dynamically generated data item: Overall Payment Type (Overpayment amount, Payments allocated, Reversal Total, Write off Total).
Over and Under Payments Report	OverpaymentCaseFinancialReport	This report shows a breakdown of the payments for a case that is categorized into the following totals: Overpayment amount, Payments allocated, Reversal Total, Write off Total.	InstructionLineItem.amount	The analytic contains 1 dynamically generated data item: Overall Payment Type (Overpayment amount, Payments allocated, Reversal Total, Write off Total).
Single Component Graphical Benefit Statement	SingleComponentGraphicalBenefitStatement	This report compares the actual amount that was paid for a single child benefit to the reassessed amount that should have been paid.	OverUnderPaymentBreakdown.amount	None

Table 5: Intake Reports for the Super user role

Report Business Name	Report Technical Name	Description	Database table name and attribute	Dynamically Generated Data Items
Applications Awaiting Determination Report	ApplicationAwaitingDeterminationReport	This report shows a breakdown of the applications that are awaiting determination with dates of processing deadlines that are categorized in the following groups: Today, Overdue, 1-5 days, 6-14 days, and 15+ Days.	ProcessInstance.processingstatecode	The analytic contains 1 dynamically generated data item: Overall Processing Deadline (Today, Overdue, 1-5 days, 6-14 days, 15+ Days).
Assigned Applications Report	AssignedApplicationReport	This chart shows a breakdown of assigned applications with dates of processing deadlines that are categorized in the following groups: Today, Overdue, 1-5 days, 6-14 days, and 15+ Days.	Application.reference	The analytic contains 1 dynamically generated data item: Overall Processing Deadline (Today, Overdue, 1-5 days, 6-14 days, 15+ Days).

Report Business Name	Report Technical Name	Description	Database table name and attribute	Dynamically Generated Data Items
Online Application Report	OnlineApplicationReport	This chart shows a breakdown of the online applications that are awaiting determination with dates of processing deadlines that are categorized in the following groups: Today, Overdue, 1-5 days, 6-14 days, and 15+ Days.	Application.reference	The analytic contains 1 dynamically generated data item: Overall Processing Deadline (Today, Overdue, 1-5 days, 6-14 days, 15+ Days)
Applications Awaiting Disposition Chart	ApplicationAwaitingDispositionReport	This chart shows a breakdown of the applications that are awaiting disposition with dates of processing deadlines that are categorized in the following groups: Today, Overdue, 1-5 days, 6-14 days, and 15+ Days	ProcessInstance.processinstanceid	The analytic contains 1 dynamically generated data item: Overall Processing Deadline (Today, Overdue, 1-5 days, 6-14 days, 15+ Days).
Priority Complexity Risk Assessment Line Chart	PCRAssessmentLineChart	This chart shows a comparison between the priority, complexity, and risk ratings for cases.	PcrConfiguration.pcrconfigurationstatus	The analytic contains 3 dynamically generated data items: RiskClassification, ComplexityClassification, PriorityClassification (low, medium, high).

Table 6: Investigation Summary and Caseload Summary Reports for the CPM manager and the super user roles

Report Business Name	Report Technical Name	Description	Database table name and attribute	Dynamically Generated Data Items
Resolved Investigations Report	CaseWorkloadReport	This chart shows a breakdown of the number of cases that are being investigated divided into the following categories: Closed Founded, Closed Unfounded, Open Founded, and Open Unfounded. The report is for the CPM manager.	ResolutionConfiguration.resolution	The analytic contains 1 dynamically generated data item: Overall Status (Closed Founded, Closed Unfounded, Open Founded, Open Unfounded).
Employer Work Force Report	EmployerWorkforceReport	This chart shows a breakdown of the number of staff for an employer by status: Permanent Staff, and Casual Staff.	Employer.registeredname	The analytic contains 1 dynamically generated data item: Overall Staff (Permanent Staff, Casual Staff).
Prospect Employer Work Force Report	ProspectEmployerWorkforceReport	This chart shows a breakdown of the number of staff for an employer by status: Permanent Staff, and Casual Staff.	Employer.concernRoleID	The analytic contains 1 dynamically generated data item: Overall Staff (Permanent Staff, Casual Staff).

Report Business Name	Report Technical Name	Description	Database table name and attribute	Dynamically Generated Data Items
Caseload Summary Report	CaseWorkloadReport	This chart shows a breakdown of the number of cases in the following status's: Submitted, Open, Active, and Approved. The chart displays cases for all of the caseworkers who report to the supervisor.	CaseStatus.status	The analytic contains 1 dynamically generated data item: Overall Case Status (Submitted, Open, Active, Approved).

Reading product documentation

You can read the product documentation to learn about Cúram Platform.

The product documentation includes business and technical documentation for specific versions and editions of the Cúram product suite. Links to developerWorks, Redbooks, and white papers are also provided when articles about the product are available. Business analysis documentation is written to provide an overview of the business processes supported by Cúram applications. You can use this information to understand the default features, and how they can be applied to your business processes.

For the business analyst scenario, Fiona finds documentation on the platform analytics features in the Cúram Business Intelligence and Analytics Guide. The documentation provides a high-level overview of the elements of business intelligence and the business intelligence infrastructure.

Related concepts

Evaluating the default analytics scenario

Fiona logs into the Cúram application to explore the default application features to evaluate whether they meet the project requirements or if there are gaps that will need to be addressed. Fiona is analyzing the project's analytics requirements.

Examining the SPMP Analytics descriptions

Fiona is analyzing the project requirements. Fiona reads the Business Intelligence and Analytics documentation and discovers that there is a fit with the Cúram product, but needs to do further analysis to determine the alignment. Fiona compares the descriptions of the default analytics to the project requirements.

Fiona then reads the default analytics descriptions in the Completing a Fit Gap Analysis section of the accelerator. Fiona can see from the description of the Caseload Summary Report that the report might fit with the project requirements. There appears to be some level of fit with the number of Sickness Benefit Cases versus the number of Incapacity Benefit Cases bar chart. The Caseload Summary Report is a bar chart that displays the number of cases, for each caseworker that is assigned to the supervisor, by status. However, based on the description alone, Fiona cannot determine the level of fit with the project requirements and needs to examine the Caseload Summary Report further.

Based on the default analytic descriptions, there does not appear to be any default analytics that fit with the requirements for the Clients with Multiple Diagnoses pie chart.

Examining the SPMP Analytics application page captures

Fiona reads through the descriptions for the default analytics and understands that there is some level of fit to the project requirements. However, Fiona needs to do further analysis to determine

the alignment. Fiona decides to examine the screen captures of the SPMP Analytics against the project requirements.

Fiona views the screen capture for each of the default charts. Fiona finds that the Caseload Summary report displays a bar chart with the status along the x-axis and the number of cases along the y-axis. The chart is a similar format to the number of Sickness Benefit Cases versus the number of Incapacity Benefit Cases bar chart requirements. Fiona notes that there does appear to be some level of fit with the Caseload Summary report. It is often a simpler task to customize an existing Cúram feature than to create something new from scratch.

After reviewing all the screen captures, Fiona still does not find any existing default SPMP analytics that fit with the Clients with Multiple Diagnoses pie chart requirements. Fiona concludes that for this pie chart, there is no fit with SPMP Analytics and a custom chart must be created.

Examining analytics data by using BIRT and the Data Dictionary

Fiona reads through the product documentation and examines the report descriptions and screen captures. Fiona knows that there is a fit with the Cúram SPMP analytics. However, Fiona needs to do further analysis to determine the alignment. Fiona decides to explore the data items that are used from the Cúram database on a local installation of BIRT and the Data Dictionary.

Business Intelligence Reporting Tool

The Business Intelligence Reporting Tool (BIRT) is an open source technology platform. It is used to create charts and traditional reports, for example, a bank statement, that can be embedded into rich client and web applications.

BIRT is a low cost option for organizations that do not have elaborate business intelligence requirements. Since BIRT is included in SPMP, it can be used to develop analytics where the data is stored within the Cúram database or the Cúram Data-warehouse. BIRT provides interactive, summarized information in context.

BIRT allows for analytics to be integrated into the Cúram application. Analytical charts are displayed on user workspace pages, context panels, and dashboards.

Cúram Data Dictionary

The Cúram Data Dictionary is available as part of the Cúram installation process. The Data Dictionary can run on a Cúram application to generate analysis documentation that is specific to that application. Technical users can share the analysis documentation with non-technical users to facilitate technical and business analysis of a Cúram application. The analysis documentation primarily supports fit gap analysis and customization impact analysis that relates to Cúram database tables, code tables, and other components.

Use the analysis documentation to help you to explore the metadata and business functions of a Cúram application and the interactions between them. It supports you in exploring the metadata that is associated with those pages and database tables. In addition, you can see information about related artifact types like Domain Definitions, Code Tables, Message Files, Application Properties, and Workflows.

Analyzing SPMP Analytics with BIRT

You can use the Business and Intelligence Reporting Tool (BIRT) to determine which data items are used by a report, in this case the default Caseload Summary Report. You can also view and

modify SQL and view a preview of the report. Fiona further explores the Caseload Summary Report by using the tool.

Before you begin

Install the BIRT Eclipse tool and open the BIRT Eclipse batch file.

About this task

You can complete these steps of the design process, but we are also providing information on how Fiona completes them as part of the Fit Gap Analysis scenario.

Procedure

1. Open eclipse with the BIRT plugin installed. go to the **BIRT Eclipse Project Explorer** tab, select **Curam BI Content**, select **components**, select **core**, select **birt**, select **CaseloadSummary**, and select **CaseWorkloadReport.rptdesign**. The system displays the Caseload Summary Report information in the **Outline** tab at the bottom of the page.

2. From the **BIRT Eclipse Project Explorer** tab, select **Curam BI Content** > **components** > **core** > **birt** > **CaseloadSummary** > **CaseWorkloadReport.rptdesign**. The system displays the Caseload Summary Report information in the **Outline** tab at the bottom of the page.

3. From the **Outline** tab, expand **CaseWorkloadReport.rptdesign** and select **Data Sets** > **Case Workload data set**. right-click on the data set and select **Edit**. The **Edit Data Set** window is displayed.

4. Right-click and select **Edit**. The **Edit Data Set** window is displayed.

5. From the **Edit Data Set** window, select **Query** to view the SQL code that is used by the Caseload Summary Report. The system displays the query text.

   Fiona can see that the SQL code selects the CaseHeader.caseID and CaseHeader.statusCode attributes from the CaseHeader, CaseUserRole, OrgObjectLink application database tables where the CaseHeader.caseTypeCode ='CT2' AND CaseUserRole.caseID = CaseHeader.caseID AND CaseUserRole.typeCode = 'RT1'. Fiona is not familiar with these application database tables or code table values and needs to find out more information. Fiona plans to use the Cúram Data Dictionary to find more information.

6. From the **Edit Data Set** window, select **Preview Results** to view the values that are displayed on the Caseload Summary Report.

   Fiona notes the preview results as a gap. The Number of Sickness Benefit Cases versus the number of Incapacity Benefit Cases bar chart must display the case types (Sickness Benefit and Incapacity Benefit) along the x-axis. The existing Caseload Summary Report displays case statuses, not case types.

7. From the **Edit Data Set** window, select **OK** to close the window.

8. From the BIRT Eclipse home, select the globe icon at the top of the page and select **View Report as HTML** from the drop-down list. The system displays the **Parameter** window.

9. From the **Parameter** window, enter **caseworker** and select **OK** to view the report preview in the BIRT Report Viewer.

Fiona can see the Caseload Summary chart displays all the cases by status. The statuses are displayed along the x-axis and include Active, Approved, Open, and Submitted. The number of cases is displayed along the y-axis. Fiona notes that the chart displays only 1 case type and is a gap.

Analyzing requirements with the Data Dictionary

You can use the Cúram Data Dictionary to determine which application artifacts are used by a particular report, in this case the default Caseload Summary Report. Fiona further explores the Caseload Summary Report by using the tool. You can complete these steps to complete a Fit Gap Analysis scenario.

Before you begin

From the directory `%CURAM_DIR%\doc`, using your browser open the index.html. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development.`

Procedure

1. From the **Index** page, select the **Data Dictionary** link.
2. From the **Data Dictionary** page, select the **Application Database Tables** link.
3. From the Data Dictionary search text box, enter **CaseHeader** and click **e**.
4. From the search results, select **CaseHeader** from the Table Name column. You can see the CaseHeader database table details.

    Fiona can see a list of the Case Header attributes including typeCode that is used by the Caseload Summary Report.

5. From the **CaseTypeCode** page, search for the **CCT2** code table code..
6. From the CaseTypeCode code table page, search for the **CT2** code table code.

    Fiona can see that the code table value, CT2, used by the Caseload Summary Report has a description of Product Delivery . The Number of Sickness Benefit cases versus Incapacity Benefit cases in the bar chart displays case data for two types of product delivery cases. Fiona notes this as a medium fit with the project requirements as it is displaying product delivery case information. It does not display the number of cases that are needed to be displayed on the bar chart.

7. From the Data Dictionary menu, select the **Application Database Tables** link.
8. From the Data Dictionary search text box, enter **CaseUserRole** and click **Enter**.
9. From the search results, select **CaseUserRole** from the Table Name column. You can see the CaseUserRole database table details.

    Fiona can see a list of the CaseUserRole attributes including the typeCode that is used by the Caseload Summary Report.

10. From the code table column, select the **CaseUserRoleType** link.

    Fiona can see that the code table value, RT1, used by the Caseload Summary Report has a description of Case Owner. She now knows that the Caseload Summary Report is selecting and displaying cases that are assigned to the case owner. The Number of Sickness Benefit cases versus Incapacity Benefit cases bar chart must also display only the product delivery

cases that are assigned to the case owner. Fiona notes this as a high level fit with the project requirements.

Analyzing the requirements

Use the following practices for analyzing the default system functions against the customer's analytics business processes and to document project requirements.

Defining the user role requirements

You must define the reporting requirements for each role in the organization.

Users

A user is somebody who is employed by the organization to complete tasks on the system, such as creating and managing cases.

New users must be registered on the system and assigned a security role and a user application view.

Define the user role and access rights requirements

To determine your organization's analytics workload, you must consider the analytics requirements for each user role in your organization that includes front line staff and managers. Every user wants timely, relevant, accurate, and consistent data and analysis, but each user might define those terms differently.

The caseworker role typically needs to make day-to-day or real-time decisions within their area of responsibility. They require quick access to current information to review progress that can be seen from their workspace. The analytics that they view are updated on a frequent basis.

The supervisor role typically requires access to view current performance data in summary snapshots and as trends on Key Performance Indicator (KPI) dashboards. The supervisor KPI dashboards need to display performance indicators at the consolidated business unit and department levels. The KPI dashboards enable the users to drill down into activity analytics to examine where activities are diverging from plans and standards of performance that is required for the enterprise to meet its set goals. A supervisor might be responsible for managing a business unit or department with many cases.

The security authorization privileges are defined at the Cúram application page level where the chart resides. You must decide which user roles have access to view the application page.

Defining the chart type and format requirements

You must define the analytic chart types and formats to best suit the requirements of your organization.

Analytics chart types and formats

At the latest count, BIRT supports 14 different chart types. You can use this information to understand which type of chart provides a data visualization to meet your requirement.

The table provides guidance to help you understand the chart types that are available with BIRT. In addition to the types detailed, BIRT also supports tube, pyramid, cone, and difference charts. The tube, pyramid, and cone charts are functionally equivalent to the bar chart with the exception that the graphical display is slightly different.

Figure 2: Chart Types Supported by BIRT

Chart Types Supported by BIRT

Bar Chart	Used to display a comparison among items with few categories, but many items.		**Stock Diagram**	Used to examine how the value of an item fluctuates, as well as its daily, weekly, or yearly high, low, and closing price. This chart is used to track stock prices, but it can be used as well to track other variable quantities.
Line Chart	Used to display a comparison over time of many periods for non-cyclical data.		**Meter Chart**	Used to display display the current value of a certain measurement.
Pie Chart	Used to display a static composition that is a simple share of the total.		**Bubble Chart**	Used to examine the data relationships by studying the size and location of the bubbles that represent the relationships. Bubble charts are often used in financial analyses and market research.
Area Chart	Used to examine how values in different categories fluctuate over time, and see the cumulative change in values.		**Gantt Chart**	Used to display a schedule that helps to plan, coordinate, and track specific tasks in a project.
Radar Chart	Used to examine data as it relates to one central point. Data is plotted on radial points from the central point. This kind of chart is often used to make subjective performance analyses.		**Scatter Diagram**	Used to compare different numeric data point sets in space to reveal patterns and trends in data. (Similar to a bubble chart, except the data appears as points instead of bubbles.)

Each chart type has a number of settings: the subtype, dimension (2D or 3D), output format, multiple y axis, series type, and the orientation.

Each chart type has its own specific list of subtypes. A description of each subtype and when it should be used is provided in BIRT.

BIRT provides a number of output formats that include PNG, BMP, JPG, Flex, or SVG.

When the values in a chart vary widely from data series to data series, or when you have mixed types of data, you can plot one or more data series on a secondary vertical (value) axis. The scale of the secondary vertical axis reflects the values for the associated data series. BIRT supports the display of one y axis, two y axes, or multiple y axes.

The series type indicates how the data is displayed on the chart. Each chart type has its own series type values. For example, the bar chart series type values include line series, area series, cone series, and more.

The chart orientation setting can be used to change the position of the x-axis and the y-axis.

With each of these chart types the legend can be displayed above, below, left or right of the chart.

The chart title is supported; however, most charts are rendered in a pod or within a cluster. Both of these elements have a title, so in most cases BIRT chart titles are not enabled.

The label titles are supported; however, most charts are rendered in a pod or within a cluster. Both of these elements have a label title, so in some cases BIRT chart labels might not be needed.

Define the analytics chart types and formats

You must define the analytic chart types and format requirements.

You must decide which chart type best meets the project requirements.

Each chart type has relevant subtypes that are associated with it. You must decide which subtype is most suitable for your requirements.

You must also decide on the chart dimension (2D or 3D).

You can also choose to display multiple y axes on the chart and a series format for the chart.

You must decide on the position of the chart legend. It can be displayed above, below, left or right of the chart.

You must also decide on a chart title and label titles for the chart, if they are not already defined in the pod requirements.

Define the analytics time period

For each analytics chart, you must define the requirements for how often the chart needs to be generated.

For example, you might choose to use values such as month, day, and snapshot. A period value of month represents that the analytic uses a month's worth of data. A period value of day uses a day's worth of data. A period of snapshot means that the current state of the database is used with no filtering by date.

Defining the location requirements

You must define the requirements for where the analytics chart should be displayed in the application.

User workspace

The user workspace is an application page that is tailored to specific roles within an organization, such as case worker, intake worker, or supervisor. Every application has a home page. The home page can be configured to contain pods, announcements, and quick links.

User workspace components

A user workspace is composed of a number of elements that allow a user to perform functions on the system.

The figure illustrates an overview of the user workspace components in a sample Cúram application.

Figure 3: User workspace components
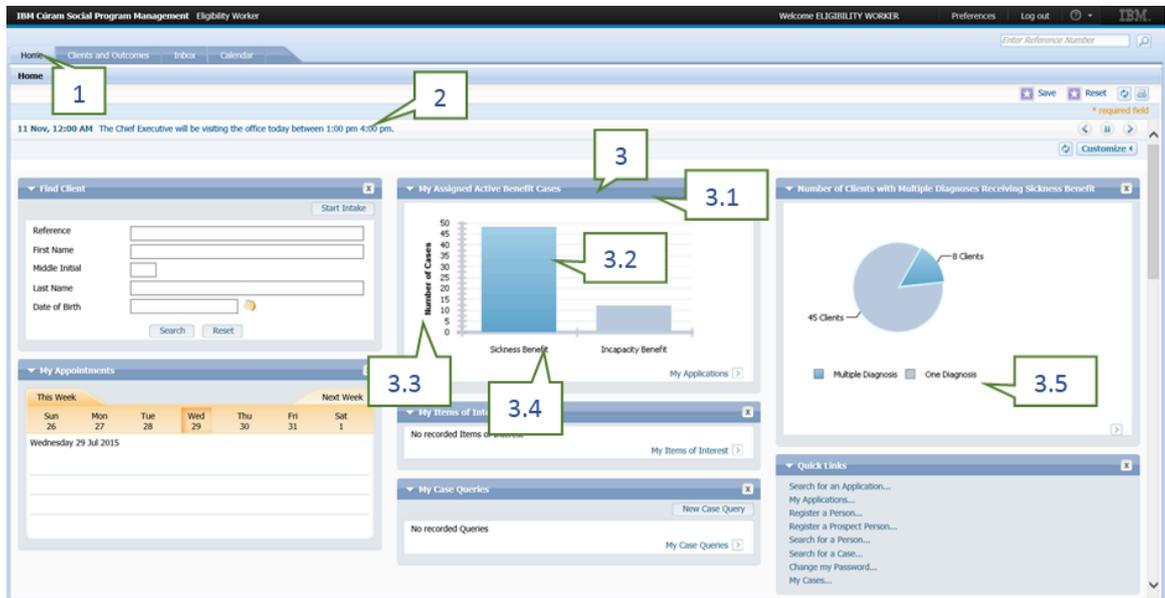
This table describes the mapping between the numbers and the user workspace elements.

Table 7:

Number	User Workspace Element Name
1	Application Home Tab
2	Announcements
3	Pod
3.1	Chart Title
3.2	Chart Type
3.3	Y axis
3.4	X axis

Number	User Workspace Element Name
3.5	Chart Legend

Interactive dashboard

A dashboard is a business specific view onto the data warehouse. Dashboards help measure business performance and quickly communicate complex information to business users in compelling visual formats, so they have a clear picture of how the business is doing. Interactive dashboards include the ability to publish graphically intuitive displays of information, including dials, gauges, and traffic lights. These displays indicate the state of the performance metric, which is compared with a goal or target value. This data is a more in-depth view into the aggregated data in the business specific data warehouse.

With SPMP Analytics, you can create dashboards for each user role.

Dashboard components

An dashboard is composed of a number of elements that allow a user to perform functions on the system.

The figure illustrates an overview of the interactive dashboard components in a sample Cúram application

Figure 4: Dashboard components
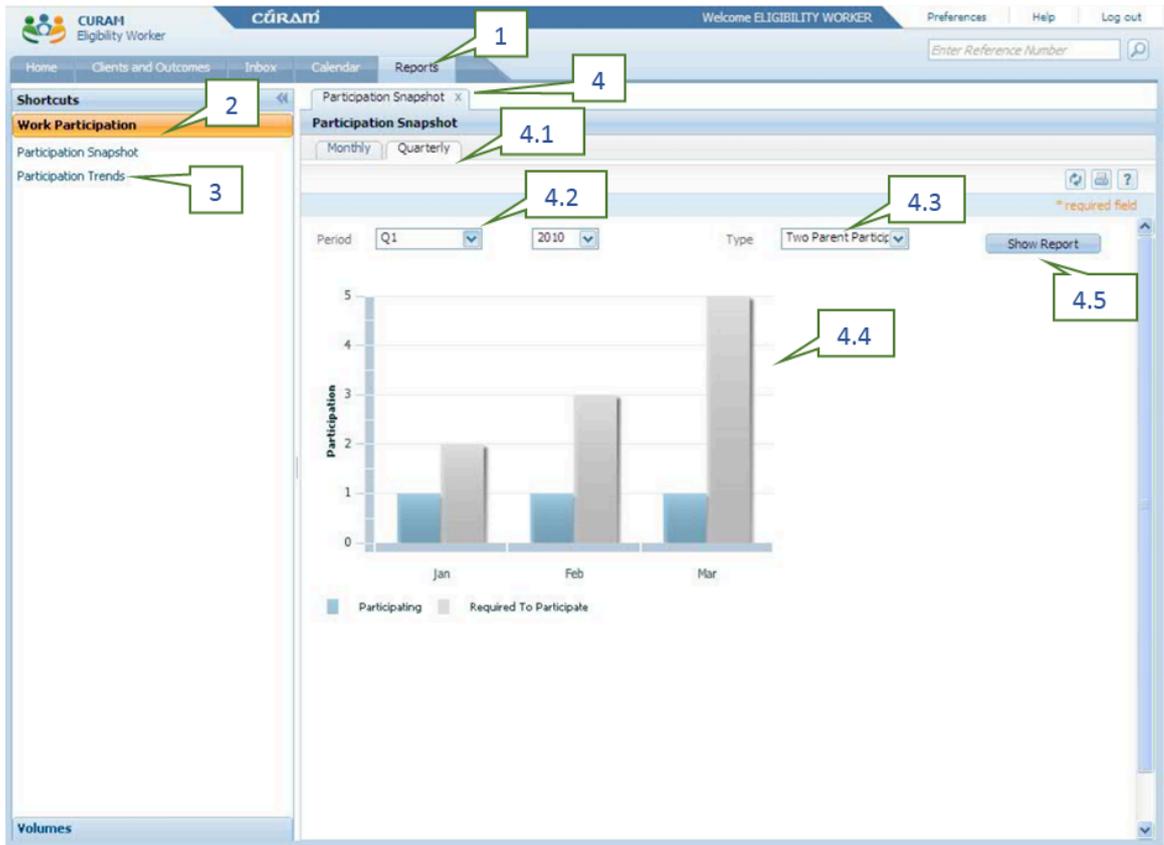
This table describes the mapping between the numbers and the dashboard elements.

Table 8: The mapping between the numbers and the dashboard elements

Number	Dashboard Element Name
1	Reports Tab
2	Dashboard Navigation Menu
3	Dashboard Navigation Menu Item
4	Dashboard Tab
4.1	Dashboard Period Tab
4.2	Period Selection
4.3	Data Type Selection
4.4	Display in Report View

Deciding on the location for the analytics

You must decide where the analytics should be displayed: context panel, user workspace, or on a dashboard.

You must decide which user role has access to the dashboard components.

You must consider who will be using the dashboard. The most effective dashboards target a single type of user role and display data specific to the user. You must select the right type of dashboard: operational, strategic, or analytical.

Define the data elements for your analytics

Identifying the data elements to be included as part of a reporting data model is important. Identifying your data items is a critical task. You are required to identify the business processes that generate your data and the physical data store. You can also find that the application database does not capture the data that you require.

You must understand which business processes generate the data elements required for your project and where the data items are physically stored. This process helps you categorize the data into different analytic groups.

You must determine where the data items are physically stored. Data items that are stored in physical database tables that are easily accessed by SQL are easy to read and process by analytical systems. Data items that are stored within large character or binary database objects are harder to process. This can require you to assess this data by using Java API's or might require you to unpack the data items from the database object before you read the data. Data items that are stored in unstructured format or in attribute-value-pairs are other examples of hard to read data items. The critical task is to identify hard to read data as hard to read data increases the cost of producing your analytics.

Analysis results for the SPMP Analytics requirements

Analysis of Fiona's requirements resulted in several key decisions. Both the business and technical decisions are discussed.

1. **User Role.** It was decided to provide the analytics to the caseworker user role. Note, this decision increases the number of users who can view the analytics thus placing a higher workload on the analytics processing landscape.
2. **Location.** The decision was to display the analytics on the Adult Social Care Eligibility Worker's user workspace. The user workspace provides quick access to current caseload information. Since both of the analytics provide information about the eligibility worker's caseload and need to be updated on a frequent basis, the user workspace was chosen as the location for the analytics.
3. **Analytic processing infrastructure.** The decision was to use SPMP Analytics as the processing infrastructure. The analysis resulted in the identification of key analytic workload markers as follows. These work load markers showed that SPMP Analytics was the correct fit as the analytic processing infrastructure.

    1. The number of caseworker's who access theses analytics was considered.
    2. The analytics are on the user workspace. It was noted that the analytics are displayed multiple times daily for each caseworker.
    3. The performance of each analytic was reviewed. After the review it was decided that placing this workload on the application database was acceptable.
    4. The current volumes of cases that are processed by the system were reviewed. The review also considered increases in the future volumes of cases to be processed.
    5. A review of future analytic requirements showed that SPMP Analytics was capable of meeting future analytic workloads.

4. **Chart Type.** The bar chart format is the best choice for displaying a comparison among items, therefore it was chosen to compare the number of Sickness Benefit and Incapacity Benefit cases. The pie chart format best displays a static composition that is a simple share of the total, therefore it was selected to display the number of clients with multiple diagnoses
5. **Customization versus new development.** The existing Caseload Summary report has a high level of fit with the requirements for the number of Sickness Benefit Cases versus the number of Incapacity Benefit cases chart. Since using an existing chart generally requires a lower level of effort to implement, the Caseload Summary report was selected to be customized. There are no existing analytics with requirements for clients with multiple diagnoses, therefore the decision was made to create a new chart to meet those requirements.
6. **Analytics Period.** The time period for both analytics were chosen to be snapshots. This time period meets the requirements of both charts.
7. **Data items.** The analysis also found that the data items required by the analytics existed within database tables and can be read by standard SQL.

1.5 Building the SPMP Analytics accelerator

Use the following information to understand each of the tasks that are involved in building this accelerator. You can follow the tasks to understand how the accelerator was built. You can also complete the tasks to build the accelerator yourself on Cúram Platform with the Adult Social Care accelerator installed. Use the following information to understand each of the tasks that are involved in creating a new BIRT chart, as well as customizing an existing one.

Configuring the component order

You must add Cúram Analytics to the server and client component orders so that the Cúram Analytics component is included in theCúram server and client builds.

Before you begin

Install a Cúram Development Environment for application development. The basic Cúram Development Environment consists of the Cúram Application Development Environment (ADE) and Java IDE.

About this task

You need to create a *CustomEnvironment.bat* file for running Cúram from Microsoft Windows or a *CustomEnvironment.sh* file for UNIX. You must not update the *SetEvironment.bat/sh* files directly to include the additional components. As the *SetEvironment.bat/sh* files are supplied with Cúram, the files are overridden if you install a new version of Cúram. *%CURAM_DIR%* is the Cúram installation directory, which by default is *C:\Merative\Curam\Development*.

> **Note:** The *CustomEnvironment* file must exist as part of the Adult Social Care accelerator installation. You need to update the file to include this new component.

Procedure

1. From the *%CURAM_DIR%* directory, create the *CustomEnvironment.bat* and *CustomEnvironment.sh* files.

2. Edit the *CustomEnvironment.bat* file to add the following details to set the component order forCúram Analytics.

    ```
    set CLIENT_COMPONENT_ORDER=SPMAnalytics,%CLIENT_COMPONENT_ORDER%
    set SERVER_COMPONENT_ORDER=SPMAnalytics,%SERVER_COMPONENT_ORDER%
    ```

3. Edit the *CustomEnvironment.sh* to add the following details to set the component order for Independent Living.

    ```
    export CLIENT_COMPONENT_ORDER=SPMAnalytics,$CLIENT_COMPONENT_ORDER
    export SERVER_COMPONENT_ORDER=SPMAnalytics,$SERVER_COMPONENT_ORDER
    ```

4. From the *%CURAM_DIR%* directory, run *SetEnvironment.bat*

Installing BIRT development prerequisites

A second Eclipse instance for BIRT development is required as the BIRT plug-in is not compatible with the RSA plug-ins that are used for Java development. For more information, see the *Deploying on IBM® WebSphere® Application Server on z/OS® Guide* guide. A BIRT development environment is not required to view default BIRT content within the application. The following steps show how to install and configure the BIRT development environment.

Downloading BIRT Eclipse-based development environment

Complete the following to install the BIRT Eclipse-based development environment. This installation is only required if you are changing or creating new BIRT reports.

Before you begin

For supported BIRT versions for product releases, review the supported prerequisites https://curam-spm-devops.github.io/wh-support-docs/spm/prerequisites.

Procedure

1. From your browser, you can download the Eclipse (that has BIRT included) from BIRT.

2. Extract the Report Designer Full Install into a separate directory, for example `C:\birteclipse`.

3. From the `%CURAM_DIR%` directory, create a batch file that is called `birteclipse.bat` to start your BIRT Eclipse environment and include the following:

   ```
   start c:\eclipsebirt\eclipse.exe -vm %JAVA_HOME%\bin\javaw.exe %CLEAN_ECLIPSE% -
   clean -data .\workspaceBIRT -vmargs -Xmx768m -Xverify:none
   ```

   Note: `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

4. Open a command prompt and change to the `%CURAM_DIR%` directory.

5. Ensure that you have the `%JAVA_HOME%` environment variable set and run `birteclipse.bat`.
   A second Eclipse instance starts, with the BIRT perspective being available to the BIRT developers.

Importing projects into BIRT Eclipse

The following steps show you how to import the required projects into the BIRT Eclipse development environment. Complete the following steps to import the projects.

Procedure

1. From Eclipse, select **File** > **Import** > **General** > **Existing Projects into Workspace** and click **Next**.

2. Select the root directory to `%CURAM_DIR%\BIApp\CuramBIRTViewer` and click **Finish**.

   `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

3. From Eclipse, select **File** > **Import** > **General** > **Existing Projects into Workspace** and click **Next**.

4. Select the root directory to `%CURAM_DIR%\BIContent` and click **Finish**.

Configuring projects within BIRT Eclipse

Complete the following steps to configure preferences within the BIRT Eclipse development environment. When the preferences are set, you can work with the default BIRT reports or create new reports.

Procedure

1. From **Window** > **Preferences** > **Report Design** > **Resource**, select **Workspace**.

2. Select the *CuramBIRTViewer/WebContent/WEB-INF/bicontent/resources* folder and click **OK**.

3. Click **Apply** > **OK**.

Results

The resource folder contains the value *${workspace_loc:CuramBIRTViewer/ WebContent/WEB-INF/bicontent/resources}*.

Adding Eclipse plug-ins

The JDBC drivers and other resources must be copied to the BIRT Eclipse installation folders directories. Complete the following steps to include JDBC driver files and other BIRT resources. With these configurations, your BIRT Eclipse environment contains the correct set of plug-in files.

Procedure

1. Open a command prompt and change to the *BIContent* directory. For example **cd %CURAM_DIR%\BIContent**.

   %CURAM_DIR% is the Cúram installation directory, which by default is *C:\Merative \Curam\Development*.

2. Run the following build target to set the location where you installed BIRT Eclipse, for example *c:\birteclipse*

   ```
   build.bat biapp.configure.eclipse -Declipseroot=C:\eclipsebirt
   ```

3. From the command prompt, you are asked to enter **y** to continue.

4. From the command prompt, review and confirm that the installation location is correct, press **ENTER** to continue.

Results

```
    [echo] ----------------------------------------------------------
    [echo] start: initializing eclipse - oda plugins
    [copy] Copying 6 files to C:\CC\DevEnv\6.1\eclipsebirt\plugins
\org.eclipse.birt.report.data.oda.jdbc_4.4.1.v201409050910\drivers
    [echo] end  : initializing eclipse - oda plugins
    [echo] start: initializing eclipse - curam plugins
    [copy] Copying 1 file to C:\CC\DevEnv\6.1\eclipsebirt\plugins
\org.eclipse.birt.report.viewer_4.4.1.v201409151154\birt\scriptlib
    [copy] Copying 1 file to C:\CC\DevEnv\6.1\eclipsebirt\plugins
    [copy] Copying 36 files to C:\CC\DevEnv\6.1\eclipsebirt\plugins
\org.eclipse.birt.report.viewer_4.4.1.v201409151154\birt\webcontent
    [copy] Copying 36 files to C:\CC\DevEnv\6.1\eclipsebirt\plugins
\org.eclipse.birt.report.viewer_4.4.1.v201409151154\birt\webcontent
    [echo] end: initializing eclipse - curam plugins
    [echo] ----------------------------------------------------------
```

Creating and customizing charts

Detailed step-by-step instructions for you to reproduce the same charts, which are included in the SPMP accelerator. The accelerator specifies how to create a new chart and how to customize an existing chart, which is delivered with Cúram. Reproducing the sample charts gives you practical hands-on experience before creating your own charts.

Before you begin

> **Attention:** Ensure that you always save your configuration files so that your configuration changes are persisted. Configuration files are overwritten each time you run a build database target.

For each generic task, child pages provide you with the specific values that you need to reproduce the sample application.

Creating a new chart and a new pod
The following sections show you how to create a new BIRT chart and save the configurations.

About this task

In this scenario, the default Adult Social Care analytic charts on the user workspace does not meet the exact needs of the Adult Social Care worker. In the following sections, a new pod is created showing Sickness Benefit cases with one or multiple diagnosis. The process involves the creation of a new BIRT chart and a new pod, which is displayed in the user workspace.

Implement a new pod and a new chart
The following sections show you how to create a new pod for your user workspace and how to create a new BIRT chart.
Creating the code table files
Complete the following steps to create a new pod type code table item and a pod binding item.

About this task

The `CT_podType.ctx` code table is used to declare a new pod.

The *CT_podLoaderBindings.ctx* declares the Java class that generates the XML fragment, which populates the pod.

Procedure

1. From *%CURAM_DIR%\EJBServer\components*, create a new directory *SPMAnalytics\codetable*.

    %CURAM_DIR% is the Cúram installation directory, which by default is *C:\Merative \Curam\Development*.

2. From *%CURAM_DIR%\EJBServer\components\SPMAnalytics\codetable*, create a code table file *CT_podType.ctx*. To create the file, templates are available, copy the *CT_podType.ctx* file from the component *core*, modify the file, and remove all existing *<code>* xml elements from the file.

3. From the *CT_podType.ctx* code table file, enter a new code table item.

    > **Note:** The java_identifier is used in the java implementation.

    For example:

    ```
    <code
        default="false"
        java_identifier="SICKNESSpod"
        status="ENABLED"
        value="PT24000">
        <locale
            language="en"
            sort_order="0">
            <description>Sickness Benefit Cases By Number of Diagnosis</description>
            <annotation/>
        </locale>
    </code>
    ```

4. From *%CURAM_DIR%\EJBServer\components\SPMAnalytics\codetable*, create a code table file *CT_podLoaderBindings.ctx*. To create the file, templates are available, copy the *CT_podLoaderBindings.ctx* file from the component *core*, modify the file, and remove all existing *<code>* xml elements from the file.

5. From the *CT_podLoaderBindings.ctx* code table file, enter the following code table item. The new code declares the Java class that generates the XML fragment that populates the pod.

    > **Important:** The pod loader bindings value must be the same as the value specified in the pod type code table item.

For example:

```
<code
    default="false"
    java_identifier="MYSICKNESSCASESpod"
    status="ENABLED"
    value="PT24000">
    <locale
        language="en"
        sort_order="0">
        <description>curam.spmanalytics.sl.pods.impl.MyBenefitCasesPodLoader</
description>
        <annotation/>
    </locale>
</code>
```

6. Open a command prompt and change to the `%CURAM_DIR%\EJBServer` directory.

7. Run **build ctgen** to generate the code tables.

8. Run **build database** to see the new code table records in the database.

9. Save your configuration files.

Code table files for SPMP Analytics

The code table configuration files specified for the SPMP Analytics accelerator.

Table 9:

File name	Location

`CT_podType.ctx`
 	
`CT_podLoaderBindings.ctx`
	`%CURAM_DIR%\EJBServer\components\SPMAnalytics\codetable`
	
	Note: `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

Implementing a pod loader

The pod loader implementation creates a pod containing a chart. Complete the following steps to create a new pod loader implementation.

About this task

To create a pod loader implementation, a new Java class is required. The pod loader extends the *curam.cefwidgets.pods.pod.impl.PodLoader* class and implements the *createPod* method.

Procedure

1. From the `%CURAM_DIR%\EJBServer\components\SPMAnalytics` directory, create the directory `source\curam\spmanalytics\impl`.

   `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

2. Create a new file called `SPMAnalyticsConst.java`. You can view the associated sample constant class implementation.

3. From the `%CURAM_DIR%\EJBServer\components\SPMAnalytics` directory, create the directory `source\curam\spmanalytics\sl\pods\impl`.

4. Create a new file called *MyBenefitCasesPodLoader.java*. You can view the associated sample pod loader implementation.

5. Open a command prompt and change to the *%CURAM_DIR%\EJBServer* directory and run the following build target to compile the implementation.

```
build compile.implemented
```

6. From Eclipse, select the *EJBServer* project, right-click and click refresh. Start the server as normal.

Pod loader implementation for SPMP Analytics

The following sample code includes the pod loader implementation and the associated constants that are defined for the SPMP Analytics accelerator.

> **Important:** You must update the java code that is specified in **BOLD** if you want to create a new pod with a different name than specified.

Table 10: MyBenefitCasesPodLoader.java

```java
/*
 * Licensed Materials - Property of IBM
 *
 * PID 5725-H26
 *
 * Copyright IBM Corporation 2015. All rights reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or disclosure
 * restricted by GSA ADP Schedule Contract with IBM Corp.
 */
package curam.spmanalytics.sl.pods.impl;

import java.util.HashMap;
import java.util.Map;

import org.w3c.dom.Document;
import org.w3c.dom.Node;

import com.google.inject.Inject;

import curam.bihelper.sl.impl.BIHelper;
import curam.cefwidgets.docbuilder.impl.BorderLayoutBuilder;
import curam.cefwidgets.docbuilder.impl.PodBuilder;
import curam.cefwidgets.pods.pod.impl.PodLoader;
import curam.cefwidgets.utilities.impl.RendererConfig;
import curam.cefwidgets.utilities.impl.RendererConfig.RendererConfigType;
import curam.cefwidgets.utilities.impl.WidgetConst;
import curam.codetable.PODTYPE;
import curam.core.sl.fact.UserAccessFactory;
import curam.core.sl.intf.UserAccess;
import curam.core.sl.pods.impl.PodsConst;
import curam.spmanalytics.impl.SPMAnalyticsConst;
import curam.util.exception.AppException;
import curam.util.exception.AppRuntimeException;
import curam.util.exception.InformationalException;
import curam.util.persistence.GuiceWrapper;
import curam.util.type.CodeTable;

public class MyBenefitCasesPodLoader extends PodLoader {

  @SuppressWarnings("restriction")
  @Inject
  protected BIHelper biHelper;

  /**
   * Constructor for the class.
   */
  public MyBenefitCasesPodLoader() {
    GuiceWrapper.getInjector().injectMembers(this);
  }

  // _____
  /**
   * Load the Benefit Case Summary pod.
   *
   * The benefit case summary pod displays a POD which contains a chart. The
   * chart provides the current user with a summary of the Benefit Cases broken
   * out by the number of diagnosis.
   *
   * @param pageDocument
   *            a DOM document describing the page to be rendered.
   *
   */
  @Override
  public Node createPod(final Document workspaceDocument,
      final Map<String, Object> contexts) {

    try {
      final PodBuilder pod = PodBuilder.newPod(workspaceDocument,
          PODTYPE.SICKNESSPOD);

      // read the pod title from code table entity
      pod.setTitle(CodeTable.getOneItem(PODTYPE.TABLENAME, PODTYPE.SICKNESSPOD));
      pod.setTextResource(SPMAnalyticsConst.kBenefitPodProperties);

      // Create a BorderLayout element with a North and Center component
      final BorderLayoutBuilder borderLayout = BorderLayoutBuilder
          .newInstance(workspaceDocument);
      final Node centerComponentElement = borderLayout
          .addComponent(BorderLayoutBuilder.Constraint.Center);
```

Table 11: SPMAnalyticsConst.java

```java
/*
 * Licensed Materials - Property of IBM
 *
 * PID 5725-H26
 *
 * Copyright IBM Corporation 2015. All rights reserved.
 *
 * US Government Users Restricted Rights - Use, duplication or disclosure
 * restricted by GSA ADP Schedule Contract with IBM Corp.
 */
package curam.spmanalytics.impl;

import curam.util.type.AccessLevel;
import curam.util.type.AccessLevelType;

/** Constants used by SPM Analytics component. */
@AccessLevel(AccessLevelType.EXTERNAL)
public abstract class SPMAnalyticsConst {

  /**
   * New pod chart name.
   */

  public static final String kNewPODChartName = "NewPodAnalyticChart";

  /**
   * New pod title for benefit cases.
   */
  public static final String kBenefitCasesPODTitle = "My Benefit Cases";

  /**
   * Contains the localizable properties for the pod.
   */
  public static final String kBenefitPodProperties =
"spmanalytics.i18n.MyBenefitCasesPod";

  /**
   * The user name constant used as a key to the report.
   */
  public static final String kuserNameParameter = "userName";

  /**
   * The title of the chart on the pod.
   */
  public static final String kSicknessCasesPODTitle = "My Sickness Cases";

}
```

Configuring a pod for a user workspace

Complete the following steps to add a pod to a user workspace container by using the default wizard provided in the Administrator application.

Procedure

1. Log in to the system as a user with administrator credentials.
2. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and select **User Interface** > **Personalized Pod Pages**.
3. From the **Personalized Pod Pages** window, select the action menu and click **Edit** for the **ASCEligibilityWorkerHome UIM Page ID**.
4. From the **Configure a Personal Page** wizard, select the **Sickness Benefit Cases By Number of Diagnosis** check box to include as an available pod to be displayed on the users workspace home page and then click **Next**.

5. From the **Defaults Pods**, ensure **Sickness Benefit Cases By Number of Diagnosis** is selected, click **Next**.

6. From the **Page Layout**, click **Save**.

Creating a BIRT chart

The following steps show you how to create a chart by using your BIRT Eclipse environment. The chart displays sickness cases with the number of diagnoses displayed.

About this task

This task describes how to create a new BIRT report. You can create a new BIRT report design file, by using the Business Intelligence infrastructure that is provided. Through the Resource Explorer, you can add pre-configured data sources, master page, and chart items to your new report. Alternatively you can copy and modify an existing chart.

Fiona previously found during the fit gap analysis that the **Case Load Summary** chart was a close fit to the requirement. In the following steps, you can copy the existing configurations that are specified for the caseload summary chart to a new name and update with your requirements.

Creating a BIRT report

Complete the following steps to create a new BIRT report. The report contains a chart that displays Sickness Benefit cases by number of diagnoses.

Before you begin

You are required to have your new BIRT Eclipse workspace setup.

Procedure

1. From Eclipse, select **Windows** > **Preferences** > **Report Design**. This step ensures that your Eclipse workspace is correctly configured for BIRT report development. Click **OK**.

2. Select **Window** > **Show View** > **Other** > **Report Design** > **Property Editor** and click **OK**. This step ensures that the BIRT property editor is available, the property editor is used in the following steps.

3. From Eclipse, select **File** > **New** > **Report**.

4. From the **New Report** window, navigate through **Curam BI Content** > **components** > **SMPAnalytics** > **BIRT**.

5. From the **Field name** text box, enter **NewPODAnalyticChart.rptdesign**.

6. Click **Finish**.

Results

The new report opens in the **Outline** tab within the **Report Design** perspective.

Configuring the data source

Complete the following steps to configure a connection to the application database.

Procedure

1. From Eclipse, select **Window** > **Show View** > **Outline**.

2. From Eclipse, select **Window** > **Show View** > **Other** > **Report Design** > **Resource Explorer** and click **OK**. The **Resource Explorer** tab is now displayed.

3. From **Resource Explorer** tab, navigate through **Shared Resources** > **Library** > **CEFLibrary.rptlibrary** > **Data Sources**, right-click on *CuramDB*, and select **Add to Report**. The new data source is displayed in the **Outline** tab under **Data Sources.** You must use the data sources that are supported. These data sources can connect to the application database when the report is deployed to Tomcat or an application server.

Table 12: Data sources supported

Data Sources	Description
CuramDB	The *CuramDB* data source is used for the Cúram database.
CuramDM	The *CuramDM* is used for the BIA data mart. This is the BI&A licensed Cúram component.

4. From the **Outline** tab, expand **DataSources** and right-click on **CuramDB** and click **Edit**.

5. From the **Driver Class**, select the drop-down list, and select the DB2 driver, if you are using DB2 on LUW select the **com.ibm.db2.jcc.DB2Driver** driver.

6. From the **BIRT JDBC Data Source** window, enter your database details and click **Test Connection**. The following example DB2 database is specified.

Table 13: Example DB2 connection details

Name	Description	Value
Driver Class:	You can select the database driver class by using your up/down arrow. In this example, the values specified	`com.ibm.db2.jcc.DB2Driver (v3.69)`
Database URL:	The database URL uses the setting, which you defined in the `%CURAM_DIR%\EJBServer\project\properties\Bootstrap.properties` during the Cúram environment setup. • SERVER_NAME: *curam.db.servername* • SERVER_PORT: *curam.db.serverport* • DATABASE_NAME: *curam.db.name* **Note:** `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.	`jdbc:db2://<SERVER_NAME>:<SERVER_PORT>/<DATABASE_NAME>` For example, `jdbc:db2://localhost:50000/curam`
User Name:	Set the user name to the value, which you specified as the *curam.db.username* in the `Bootstrap.properties` file.	For example, db2admin
Password:	Set the user name to the value, which you specified as the *curam.db.password* in the `Bootstrap.properties` file.	For example, password
JNDI URL:	You can leave this field blank.	n/a

7. When you get the message **Connection successful**, click **OK**.

8. From the **Resource Explorer** tab, select **library** > **CEFLibrary.rptlibrary** and expand the library to view data sources. This view shows the list of data sources that are supported by Cúram.

Adding a report parameter

Complete the following steps to create a report parameter. The report parameter allows the application to pass the user name to the report.

Procedure

1. From the **Outline** tab, right-click on **Report Parameters** and click **New Parameter**.
2. From the **New Parameter** window, enter *userName* in the Name text box.
3. Ensure the **Is Required** check box is selected.
4. Accept all other defaults and click **OK**.

Reading data from the application database

Complete the following steps to read data from the application database. In the following steps, you write SQL to read case data from the application tables.

Procedure

1. From the **Outline** tab, expand **NewPODAnalyticChart.rptdesign** > **Data Sets**.
2. Right-click on **Data Sets** and click **New Data Set**.
3. From the **New Data Set** window, in the **Data Set Name** text box enter **SicknessCases** and click **Next**.
4. From the **Query Text** window, enter the SQL code that meets your requirements. You can view the associated sample SQL specified for the accelerator, copy the SQL with demo data into the **Query Text** window, select **Finish**. The **Edit Data Set** window is displayed.
5. Select **Parameters** from the left hand pane, double-click the **parameter** record displayed. Create the parameter using the following example.

Table 14: Example report parameter details

Name	Description	Value
Name:	Enter a parameter name	*p_username*
Data Type	Select the parameter data type from the drop down list.	Select *String* from the drop down list.
Linked To Report Parameter	Link the SQL query parameter to the report parameter.	From the drop down list select *userName*
Other fields	Leave the defaults for any other fields.	Accept the default values.

6. Click **OK**.
7. Select **Preview Results** to view the values that are returned from this SQL query. The **Preview Results Pane** displays the following.

Results

```
NUMBEROFDIAGNOSIS     NUMBEROFCLIENTS
---------------       ---------------
Multipe Diagnosis     1
One Diagnosis         3
```

SQL specified for SPMP Analytics

The following sample SQL returns data for the Adult Social Care Sickness and Incapacity Benefit cases details. Demonstration data is provided to give you a result when you click the preview results.

Table 15: SQL

SQL
SQL to return data for Sickness and Incapacity Benefit cases

```
select
    AllData.MedicalConditions numberOfDiagnosis,
    sum( AllData.clients) numberofclients
from
(
    (select AllEvidenceWithDiagnosis.medicalConditions,
 count(AllEvidenceWithDiagnosis.caseid) clients from
      (
      select ED.caseid caseid, (CASE WHEN count(ER.parentid) > 1 THEN 'Multiple' ELSE
'One' END ) medicalConditions
      from DYNAMICEVIDENCEDATA DED join  EVIDENCEDESCRIPTOR ED on
(DED.evidenceid=ED.relatedid and ED.statusCode = 'EDS1') -- active evidence
        left outer join EVIDENCERELATIONSHIP ER  on (DED.EVIDENCEID=ER.parentid and
ER.parenttype='DET24002')--certificate evidence, 24003 is diagnosis evidence
        join (SELECT distinct CaseHeader.caseID caseid FROM CaseHeader, CaseUserRole,
OrgObjectLink, productdelivery, product
                  WHERE
                  CaseUserRole.caseID = CaseHeader.caseID AND
                  CaseUserRole.typeCode = 'RT1' AND -- case owner
                  CaseUserRole.orgObjectLinkID = OrgObjectLink.orgObjectLinkID AND
                  OrgObjectLink.userName = ? AND
                  caseheader.caseid=productdelivery.caseid and
                  productdelivery.productid=product.productid and product.TYPECODE
='IBPT24000'
                 ) CASES on (CASES.caseid=ED.caseid) -- only return active product
delivery cases.
      group by ED.caseid
    ) AllEvidenceWithDiagnosis group by AllEvidenceWithDiagnosis.medicalConditions
    )
)  AllData
    group by AllData.MedicalConditions
```

SQL

SQL including demonstration data

```
select
    AllData.MedicalConditions numberOfDiagnosis,
    sum( AllData.clients) numberofclients
from
(
(select  'One Diasnosis' medicalConditions, 3 clients from organisation union all
 select 'Multiple Diasnosis' medicalConditions, 1 clients from organisation )
    union all
    (select AllEvidenceWithDiagnosis.medicalConditions,
count(AllEvidenceWithDiagnosis.caseid) clients from
     (
     select ED.caseid caseid, (CASE WHEN count(ER.parentid) > 1 THEN 'Multiple' ELSE
'One' END ) medicalConditions
     from DYNAMICEVIDENCEDATA DED join  EVIDENCEDESCRIPTOR ED on
(DED.evidenceid=ED.relatedid and ED.statusCode = 'EDS1') -- active evidence
        left outer join EVIDENCERELATIONSHIP ER  on (DED.EVIDENCEID=ER.parentid and
ER.parenttype='DET24002')--certificate evidence, 24003 is diagnosis evidence
        join (SELECT distinct CaseHeader.caseID caseid FROM CaseHeader, CaseUserRole,
OrgObjectLink, productdelivery, product
                WHERE
                CaseUserRole.caseID = CaseHeader.caseID AND
                CaseUserRole.typeCode = 'RT1' AND -- case owner
                CaseUserRole.orgObjectLinkID = OrgObjectLink.orgObjectLinkID AND
                OrgObjectLink.userName = ? AND
                caseheader.caseid=productdelivery.caseid and
                productdelivery.productid=product.productid and product.TYPECODE
='IBPT24000'
                ) CASES on (CASES.caseid=ED.caseid) -- only return active product
delivery cases.
        group by ED.caseid
    ) AllEvidenceWithDiagnosis group by AllEvidenceWithDiagnosis.medicalConditions
    )
)  AllData
    group by AllData.MedicalConditions
```

Table 16: Expected preview results with SQL demonstration data included

NUMBEROFDIAGNOSIS	NUMBEROFCLIENTS
Multiple Diagnosis	1
One Diagnosis	3

Choosing the chart type and selecting chart properties

The following configurations include changes to your BIRT Eclipse workspace to configure the chart and report.

Adding a chart to the report

Complete the following steps to select a chart to add to the report and update the size of the chart.

Procedure

1. From the **Resource Explorer**, select **Library** > **ReportItemLibrary.rptlibrary** > **Report Items**.

2. Select **Chart-PieChartCuramStyled** and drag the chart onto the report.

3. Select the chart and right-click and click **Properties**.

4. From the **Properties Editor - Chart General** tab, set the height to 300 pixels and width to 400 pixels.

5. From the **Resource Explorer**, select **Library** > **ReportItemLibrary.rptlibrary** > **Master Pages**. Right click *CuramAnalyticMasterPage*, and select **Add to Report**.

6. From the **Outline** tab, expand **Master Pages**, right click on **Simple Master Page - Simple Master page** and select **delete**.

7. From the **Outline** tab, expand **Master Pages** and select the **Simple Master Page - CuramAnalyticMasterPage** window.

8. From the **Property Editor - Master Page** tab, set the height to 300 pixels and width to 400 pixels and set the header and foot height to zero.

9. Click **File** > **Save**.

Clearing script properties and setting the data source

Complete the following steps to updated the properties for the layout and script properties for the report.

Procedure

1. From the **Layout** tab, select the report, do not select the chart. You must select an area of white space outside the chart.

2. From the **Script** tab, go to the **Script** drop down menu and select **BeforeFactory**.

3. If there is any code listed delete all code specified in the **Script** tab.

4. From the **Layout** tab, double-click on the chart to edit the chart properties.

5. From the **Edit Chart** window, go to the **Select Data** tab and select **Use Data From** drop-down list and choose the **Sickness** data set.

6. Click **Finish**.

Setting the category definition

Complete the following steps to update the category definition properties for the chart.

Procedure

1. From the **Layout** tab, double-click the chart to edit the chart properties.

2. From the **Edit Chart** window, select the **Select Data** tab.

3. From the **Category Definition** section that is displayed below the **Chart Preview**, click **Fx**.

4. From the **Expression Builder** window, delete the content in the expression builder pane.

5. From the **Category** column, select **Available Column Bindings**.

6. From the **Sub-Category** column, select **Chart - PieChartCuramStyled**.

7. From the **Double Click to Insert** column, double-click **NUMBEROFDIAGNOSIS** to add `row["NUMBEROFDIAGNOSIS"]` to the expression builder pane.

8. Click **OK**.

Setting the slice size definition

Complete the following steps to update the slice size definition properties for the chart.

Procedure

1. From the **Select Data** tab.

2. From the **Slice Size Definition** section that is displayed to the left of the **Chart Preview**, click **Fx**.

3. From the **Expression Builder** window, delete the content in the expression builder panel.

4. From the **Category** column, select **Available Column Bindings**.

5. From the **Sub-Category** column, select **Chart - PieChartCuramStyled**.

6. From the **Double Click to Insert** column, double-click **NUMBEROFCLIENTS** to add `row["NUMBEROFCLIENTS"]` to the expression builder panel.

7. Click **OK** > **Apply** > **Finish**.

Setting the pie slice labels

Complete the following steps to set the pie slice labels for the chart.

Procedure

1. From the **Layout** tab, double-click the chart to edit the chart properties.

2. From the **Edit Chart** window, select the **Format Chart** tab.

3. Select **Series** > **Value Series**, ensure that the **Show Series Labels** check box is ticked.

4. Update the **Leader Line length** field to set the value to 5.

5. Click the **labels** button.

6. From the **Series Labels** window, below the **Values** panel, select the drop-down list and choose **Percentile value data** and click **Add**.

7. From the **Suffix** field, enter **)**.

8. From the **Separator** field, enter **clients (**. Note the space before " clients (".

9. Close the **Series Labels** window.

10. Click **Apply** > **Finish**.

11. Click **File** > **Save**.

Setting the tool tip

Complete the following steps to set the tool tip for each bar in the chart.

Procedure

1. From the **Layout** tab, double-click the chart to edit the chart properties.

2. From the **Edit Chart** window, select the **Format Chart** tab.

3. Select **Series** > **Value Series** > **Interactivity** button.

4. From the **Series Interactivity** window, select **Fx**.

5. From the **Expression Builder** window, delete the content in the expression builder pane.

6. Enter the following expression in the panel and click **OK**.

```
row["NUMBEROFCLIENTS"] + " clients have " + row["NUMBEROFDIAGNOSIS"] + "."
```

7. Close the **Series Interactivity** window.

8. Click **Apply** > **Finish**.

9. Click **File** > **Save**.

Viewing the report

Complete the following steps to view the report as HTML.

Procedure

1. From the BIRT Eclipse home, select the drop-down arrow beside the globe icon in the toolbar and click **View Report as HTML**.

2. From the **Parameter** window, enter the caseworker name as **ascewuser**, click **OK** to view the report.

Configuring a report configuration record

Every BIRT chart requires a report configuration record. A new configuration record contains the path to the new BIRT report. When the configuration record is created the report can then be displayed an application page. Complete the following steps to create a new report configuration record.

Procedure

1. Log in to the system as a user with *sysadmin* credentials.

2. Select **System Configurations** > **Shortcuts Panel** > **Business Intelligence** > **Report Configuration**.

3. From the **Report Configuration** tab, click **New**.

4. From the **New Reporting Configuration** window, enter the following details and click **Save**.

Table 17: New report configurations

Name	Description
Report Name	The name of the report.
Report File Name	The name of the report with extension and path.
Report Category	The report category type. It is configured in the *CT_BIReportCategory.ctx* file.
Report Servlet	The report server name.
Width	The report width in pixels or percent.
Height	The report height in pixels or percent.
Scrolling	The report scrolling parameter.
Frame Border	The report frame border settings.
Description	The report description.

Report configuration values for new chart

The following values are specific for the new chart configured for the SPMP Analytics accelerator.

Table 18: New report configurations

Name	Value
Report Name	*NewPODAnalyticChart*
Report File Name	*components/SPMAnalytics/birt/NewPODAnalyticChart.rptdesign*
Report Category	Default value

Name	Value
Report Servlet	blank
Width	400
Height	300
Scrolling	Default value
Frame Border	Default value
Description	This report displays Sickness Benefit cases as a pie chart by number of diagnoses.

Publishing the chart

Complete the following steps to publish the new BIRT chart and view it on the test page.

Procedure

1. From your Cúram BI content directory *%CURAM_DIR%\BIContent*, run the following command to publish BIRT reports to the run time viewer.

   ```
   build client.birt
   ```

   > **Note:** *%CURAM_DIR%* is the Cúram installation directory, which by default is *C:\Merative\Curam\Development.*

2. From your browser, enter the following URL *http://localhost:9080/CuramBIRTViewer/List*. This URL displays a list of available reports for each of the components.

3. Scroll down to the **NewPODAnalyticChart**, from the **Mode** column select **R** to display the report, the **Parameter** window is displayed.

4. From the **Parameter** window, select the **userName** text box and enter **ascewuser** and click **OK.**. Verify that the **Sickness Benefit Cases By Number of Diagnosis** chart is displayed.

Testing the user workspace page

Complete the following steps to view the new pod on the user workspace.

Procedure

1. From Eclipse, refresh the webclient and EJBServer projects. Start the Cúram server, the RMILoginClient and Tomcat as normal.

2. From the **RMILoginClient** window, enter the **ascewuser** user credentials.

3. From your browser, open the URL and enter *http://localhost:9080/Curam/AppController.do.*

4. From the user home page, select **Customize**, verify the **Sickness Benefit Cases By Number of Diagnosis** check box is selected, click **Save**.

5. From the user workspace, drag the pod to be the second column, make the pod the first widget in this column.

6. Select the **Customize** button, the **Customize** panel is displayed. Click **Save** to save your changes.

Related tasks

The user workspace allows clients to view the number of Sickness Benefit cases versus the number of Incapacity Benefit cases in a bar chart. The number of clients with multiple diagnoses is displayed as a pie chart.

Saving your chart configurations

After you configured a new chart, you must manually save all of the chart configurations to ensure that they are not overwritten when you do a build database target.

Quick reference for saving the chart configurations files

Reference table for all the files that are associated with saving a new chart.

The following descriptions detail each column in the reference table and what their purpose is:

- Task Name: The name of the current task.
- Data Contained In: The relevant file or files in the `%CURAM_DIR%\EJBServer\build\dataextractor` folder, which contains data that you need to extract for the step. `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.
- Files To Be Copied From Data Extractor: A list of files that the user must copy from the `%CURAM_DIR%\EJBServer\build\dataextractor` folder for each task.
- Files To Be Created: A list of files that need to be created by the user for each task.

Table 19: New chart creation reference table

Task Name	Data Contained In	File to be copied from Data Extractor	New Files To Be Created
Saving your pod loader implementation files			`SPMAnalyticsConst.java` `MyBenefitCasePodLoader.java`
Saving your code table files	`CODETABLEITEM.dmx`		`CT_podType.ctx` `CT_podLoaderBindings.ctx`
Saving your pod page configurations		`USERPAGECONFIG.dmx` `PAGECONFIG.dmx`	`USERPAGECONFIG.dmx` `PAGECONFIG.dmx`
Saving your report configurations		`BIREPORTCONFIGURATION.dmx`	`BIREPORTCONFIGURATION.dmx`
Saving the BIRT chart			`NewPODAnalyticChart.rptdesign`

Saving the personalized pod pages configurations

Complete the following steps to extract the specified personalized pod page DMX files from the database and put them under source control.

Procedure

1. Extract the DMX files from the database with the **build extractdata** build target.
2. From the `%CURAM_DIR%\EJBServer\build\dataextractor` directory, copy the following DMX files.

Name	Description
PAGECONFIG.dmx	Stores page templates for the personalized pages. The identifiers are referenced from the CT_PodType.ctx code table.
USERPAGECONFIG.dmx	Stores user-specific settings for the personalized pages, including default settings by user role. The identifiers are referenced from the CT_PodType.ctx code table.

> **Note:** %CURAM_DIR% is the Cúram installation directory, which by default is C: \Merative\Curam\Development.

3. Inside PAGECONFIG.dmx and USERPAGECONFIG.dmx, you must find the attribute with *name="config"* and change its *type* from *blob* to *text*.

4. The second step is to replace all < symbols with *<* and replace all > symbols with *>*

5. Inside USERPAGECONFIG.dmx, you must find the record with a *userPageConfigID* value of *4500* and change its *pageid* from *ASCEligibilityWorkerHome* to *ASCEligibilityWorkerHome2*. This change ensures that the Intake user workspace configuration record does not override the Adult Social Care configuration.

6. From %CURAM_DIR%\EJBServer, run **build database**.

7. Save your configuration files.

Personalized pod files for SPMP Analytics

The location of the personalized pod configuration files specified for the SPMP Analytics accelerator.

File name	Location
USERPAGECONFIG.dmx PAGECONFIG.dmx	%CURAM_DIR%\EJBServer\components \SPMAnalytics\data\initial > **Note:** %CURAM_DIR% is the Cúram installation directory, which by default is C:\Merative \Curam\Development.

Saving your report configurations

Complete the following steps to extract the specified personalized pod page DMX files from the database and put them under source control.

Procedure

1. Extract the DMX files from the database with the **build extractdata** build target.

2. From the %CURAM_DIR%\EJBServer\build\dataextractor directory, copy the following DMX files.

Name	Description
BIREPORTCONFIGURATION.dmx	Stores the details for the business intelligence report configurations.

> **Note:** `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

3. From `%CURAM_DIR%\EJBServer`, run **build database**.

4. Save your configuration files.

Report configuration files for SPMP Analytics

The location of the report configuration file and pod configuration file specified for the SPMP Analytics accelerator.

File name	Location
`BIREPORTCONFIGURATION.dmx`	`%CURAM_DIR%\EJBServer\components\SPMAnalytics\data\initial`
	Note: `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

Customizing a chart

The following sections show you how to reused an existing BIRT chart and modify the configurations to suit your needs.

Customizing a chart

The following sections show you how to customize a BIRT chart that is displayed on a user workspace.

In the scenario, the default Adult Social Care analytic charts on the workspace do not meet the exact needs of the Adult Social Care worker. In the following sections, an existing chart is customized without the need for any Java development. Also, a new BIRT analytic chart is created, with the chart displayed in the users workspace.

Customizing a BIRT chart

Complete these steps to customize an existing BIRT chart. The existing Caseload Summary BIRT chart is customized to display Sickness and Incapacity Benefit cases.

About this task

This task describes how to customize an existing BIRT report. You can either create a new chart from scratch or copy and modify an existing chart. In the fit gap analysis scenario, Fiona found that the **CaseLoadSummary** chart was a close fit to the requirement. In the following steps, you modify the caseload summary chart saving the chart to a new name.

Customizing a BIRT report

Complete the following steps to create a new BIRT report. The report contains a chart that displays Sickness Benefit and Incapacity Benefit cases.

Procedure

1. From Eclipse, select **Windows** > **Preferences** > **Report Design**. This step ensures that your Eclipse workspace is correctly configured for BIRT report development. Click **OK**.

2. Select **Window** > **Show View** > **Other** > **Report Design** > **Property Editor** and click **OK**. This step ensures that the BIRT property editor is available, the property editor is used in the following steps.

3. From the BIRT Eclipse workspace, select the **Project Explorer** tab.

4. From the **Curam BI Content** project, select **components** > **core** > **birt** > **CaseloadSummary**.

5. Double-click the **CaseWorkloadReport.rptdesign** file to display the case load summary report in the **Layout** tab.

6. Select **File** > **Save As** to save the report with a new name.

7. From the **Save As** window, enter the following details and click **Finish**.

Table 20: Saving new file details

Parent folder	File name
Curam BI Content/components/SPMAnalytics/birt	**CustomizedPODAnalyticChart.rptdesign**

Configuring the data source

Complete the following steps to configure a connection to the application database.

Procedure

1. From Eclipse, select **Window** > **Show View** > **Outline**.

2. From Eclipse, select **Window** > **Show View** > **Other** > **Report Design** > **Resource Explorer** and click **OK**. The **Resource Explorer** tab is now displayed.

3. From **Resource Explorer** tab, navigate through **Shared Resources** > **Library** > **CEFLibrary.rptlibrary** > **Data Sources**, right-click on *CuramDB*, and select **Add to Report**. The new data source is displayed in the **Outline** tab under **Data Sources.** You must use the data sources that are supported. These data sources can connect to the application database when the report is deployed to Tomcat or an application server.

Table 21: Data sources supported

Data Sources	Description
CuramDB	The *CuramDB* data source is used for the Cúram database.
CuramDM	The *CuramDM* is used for the BIA data mart. This is the BI&A licensed Cúram component.

4. From the **Outline** tab, expand **DataSources** and right-click on **CuramDB** and click **Edit**.

5. From the **Driver Class**, select the drop-down list, and select the DB2 driver, if you are using DB2 on LUW select the **com.ibm.db2.jcc.DB2Driver** driver.

6. From the **BIRT JDBC Data Source** window, enter your database details and click **Test Connection**. The following example DB2 database is specified.

Table 22: Example DB2 connection details

Name	Description	Value
Driver Class:	You can select the database driver class by using your up/down arrow. In this example, the values specified	`com.ibm.db2.jcc.DB2Driver (v3.69)`
Database URL:	The database URL uses the setting, which you defined in the `%CURAM_DIR% \EJBServer\project\properties \Bootstrap.properties` during the Cúram environment setup. • SERVER_NAME: *curam.db.servername* • SERVER_PORT: *curam.db.serverport* • DATABASE_NAME: *curam.db.name* **Note:** `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.	`jdbc:db2:// <SERVER_NAME>:<SERVER_PORT>/ <DATABASE_NAME>` For example, `jdbc:db2:// localhost:50000/curam`
User Name:	Set the user name to the value, which you specified as the *curam.db.username* in the `Bootstrap.properties` file.	For example, db2admin
Password:	Set the user name to the value, which you specified as the *curam.db.password* in the `Bootstrap.properties` file.	For example, password
JNDI URL:	You can leave this field blank.	n/a

7. When you get the message **Connection successful**, click **OK**.

8. From the **Resource Explorer** tab, select **library** > **CEFLibrary.rptlibrary** and expand the library to view data sources. This view shows the list of data sources that are supported by Cúram.

Reading data from the application database.

Complete the following steps to read data from the application database.

Procedure

1. From the **Outline** tab, expand **CustomizedPODAnalyticChart.rptdesign** > **Data Sets**.

2. Right-click on the **Case Load Data Sets**, and select **Edit**.

3. From the **Edit Data Set** window, select **Query** to view the SQL code in the **Query Text** window.

4. From the **Query Text** window, edit the SQL to be specific for your requirements. You can view the associated sample SQL specified for the accelerator.

5. From the **Computed Columns**, select the **STATUS_LOCALIZED** row and click **Remove**.

6. Select **Preview Results** to view the values that are returned from this SQL query.

7. From the **Edit Data Set** window, select **OK** to save and close.

SQL specified for SPMP Analytics

The following sample SQL returns data for the Adult Social Care Sickness and Incapacity Benefit cases details.

Table 23: SQL to read Sickness and Incapacity benefit cases

SQL

SQL to return data for Sickness and Incapacity Benefit cases

```
SELECT
  COUNT(ProductDeliveries.caseID)  as productcount,
  Products.description description,
  Products.sortorder as sortorder
FROM
(SELECT DISTINCT code code, description, CASE WHEN code = 'DPT24000' THEN 1 WHEN
 code = 'IBPT24000' THEN 0 end sortorder  FROM CodeTableItem where TABLENAME in
 ('ProductType','IBClaimType') AND code IN  ('DPT24000','IBPT24000')) Products
 LEFT OUTER JOIN
    (SELECT
       CaseHeader.caseID caseid,
       product.TYPECODE productcode
     FROM
       CaseHeader, CaseUserRole, OrgObjectLink, productdelivery, product
     WHERE
       CaseHeader.integratedCaseID IS NOT NULL AND
       CaseHeader.caseTypeCode ='CT2' AND -- product delivery
       CaseUserRole.caseID = CaseHeader.caseID AND
       CaseUserRole.typeCode = 'RT1' AND -- case owner
       CaseUserRole.orgObjectLinkID = OrgObjectLink.orgObjectLinkID AND
       OrgObjectLink.userName = ? and
       caseheader.caseid=productdelivery.caseid and
       productdelivery.productid=product.productid and product.TYPECODE in
 ('DPT24000','IBPT24000') -- incapacity benefit, sickness benefit
    ) ProductDeliveries ON Products.code = ProductDeliveries.productcode
GROUP BY
  Products.description, Products.sortorder
ORDER BY Products.sortorder
```

Table 24: Expected preview results

PRODUCTCOUNT	DESCRIPTION	SORTORDER
-	-	-

Table 25: SQL with demonstration data

SQL
SQL to return data for Sickness and Incapacity Benefit cases. Note, this SQL contains demonstration data, see the bold text.

```
SELECT
  COUNT(ProductDeliveries.caseID)  as productcount,
  Products.description description,
  Products.sortorder as sortorder
FROM
(SELECT DISTINCT code code, description, CASE WHEN code = 'DPT24000' THEN 1 WHEN
 code = 'IBPT24000' THEN 0 end sortorder  FROM CodeTableItem where TABLENAME in
 ('ProductType','IBClaimType') AND code IN  ('DPT24000','IBPT24000')) Products
 LEFT OUTER JOIN
    ((SELECT
       CaseHeader.caseID caseid,
       product.TYPECODE productcode
     FROM
       CaseHeader, CaseUserRole, OrgObjectLink, productdelivery, product
     WHERE
       CaseHeader.integratedCaseID IS NOT NULL AND
       CaseHeader.caseTypeCode ='CT2' AND -- product delivery
       CaseUserRole.caseID = CaseHeader.caseID AND
       CaseUserRole.typeCode = 'RT1' AND -- case owner
       CaseUserRole.orgObjectLinkID = OrgObjectLink.orgObjectLinkID AND
       OrgObjectLink.userName = ? and
       caseheader.caseid=productdelivery.caseid and
       productdelivery.productid=product.productid and product.TYPECODE in
('DPT24000','IBPT24000') -- incapacity benefit, sickness benefit
    )
  union all
     SELECT 1234 caseid , code productcode FROM CodeTableItem where TABLENAME
='ProductType'  AND code='DPT24000'
  union all
     SELECT 1234 caseid , code productcode FROM CodeTableItem where TABLENAME =
'IBClaimType' AND code = 'IBPT24000'
  union all
      SELECT 1234 caseid , code productcode FROM CodeTableItem where TABLENAME =
'IBClaimType' AND code = 'IBPT24000'
   union all
      SELECT 1234 caseid , code productcode FROM CodeTableItem where TABLENAME =
'IBClaimType' AND code = 'IBPT24000'
   union all
      SELECT 1234 caseid , code productcode FROM CodeTableItem where TABLENAME =
'IBClaimType' AND code = 'IBPT24000'
) ProductDeliveries ON Products.code = ProductDeliveries.productcode
GROUP BY
  Products.description, Products.sortorder
ORDER BY Products.sortorder
```

Table 26: Expected preview results

PRODUCTCOUNT	DESCRIPTION	SORTORDER
4	Sickness Benefit	0
1	Incapacity Benefit	1

Choosing the chart type and updating chart properties

The following configurations include changes to your BIRT Eclipse workspace to configure the chart and report.

Setting the chart height and width

Complete the following steps to set the height and width of the chart.

Procedure

1. From Eclipse, select *Curam BI Content/components/SPMAnalytics/birt/ CustomizedPODAnalyticChart.rptdesign* and open the report.
2. Select the chart and right-click and click **Properties**.
3. From the **Properties Editor - Chart General** tab, set the height to 230 pixels and width to 400 pixels.
4. Click **File** > **Save**.

Setting the axis title and refreshing the data set

Complete the following steps to set the vertical axis title and refresh the data items for the chart.

Procedure

1. From the **Layout** tab, select the report, do not select the chart. You must select an area of white space outside the chart.
2. From the **Script** tab, go to the **Script** drop down menu and select **BeforeFactory**.
3. Delete all the code that is specified in the **Script** tab.
4. From the **Layout** tab, double-click the chart to edit the chart properties.
5. From the **Edit Chart** window, go to the **Format Chart** tab and select **Chart Area** > **Axis** > **Y-Axis**.
6. Select the **Visible** check box and enter the title **Number of Cases**.
7. From the **Select Data** tab, click **Data Binding**.
8. From the **Select Data Binding** window, click **Remove** to remove all rows from the list.
9. Click **Refresh** to refresh the data from the data set.
10. Click **OK** > **Apply** > **Finish**.

Setting the axis values

Complete the following steps to update the category definition properties for the chart.

Procedure

1. From the **Layout** tab, double-click the chart to edit the chart properties.
2. From the **Edit Chart** window, select the **Select Data** tab.
3. From the **Category (X) Series** section that is displayed below the **Chart Preview**, click **Fx**.
4. From the **Expression Builder** window, delete any content in the expression builder pane.
5. From the **Category** column, select **Available Column Bindings**.
6. From the **Sub-Category** column, select **Chart - VerticalBarChartCuramStyled**.
7. From the **Double Click to Insert** column, double-click **DESCRIPTION** to add `row["DESCRIPTION"]` to the expression builder pane. Click **OK**.
8. From the **Category (X) Series** section that is displayed below the **Chart Preview**, click **Edit group and sorting**.
9. From the **Group and sorting** window, update the **Sort On** drop down list with **row["SORTORDER"]**. Click **OK**.

10. From the **Value (Y) Series Interactivity** window, select **Fx**, delete any content in the expression builder pane.

11. From the **Sub-Category** column, select **Chart - VerticalBarChartCuramStyled**.

12. From the **Double Click to Insert** column, double-click **PRODUCTCOUNT** to add `row["PRODUCTCOUNT"]` to the expression builder pane. Click **OK**.

13. Click **Apply** > **Finish**.

Setting the tool tip

Complete the following steps to set the tool tip for each bar in the chart.

Procedure

1. From the **Layout** tab, double-click the chart to edit the chart properties.

2. From the **Edit Chart** window, select the **Format Chart** tab.

3. From **Series** > **Value (Y) Series** select the **Fx** interactivity button.

4. From the **Series Interactivity** window, delete any content in the expression builder pane.

5. From the **Expression Builder** window, update the expression builder pane to show a tool tip, enter `row["DESCRIPTION"] + "\n" + row["PRODUCTCOUNT"]` into the expression builder pane.

6. Click **OK** to close the **Series Interactivity** window.

7. Click **Apply** > **Finish**.

8. Click **File** > **Save**.

Setting the bar labels

Complete the following steps to set the labels for each bar in the chart.

Procedure

1. From the **Layout** tab, double-click the chart to edit the chart properties.

2. From the **Edit Chart** window, select the **Format Chart** tab.

3. From the **Format Chart** > **Value (Y) Series**, ensure that the **Show Series Labels** check box is ticked.

4. Click the **labels** button.

5. From the **Series Labels** window, below the **Values** panel, select the drop-down list and choose **Percentile value data** and click **Add**.

6. From the **Suffix** field, enter **)**.

7. From the **Separator** field, enter **(**. Note the space before " or (".

8. Close the **Series Labels** window.

9. Click **Apply** > **Finish**.

10. Click **File** > **Save**.

Viewing the report

Complete the following steps to view the report as HTML.

Procedure

1. From the BIRT Eclipse home, select the drop-down arrow beside the globe icon in the toolbar and click **View Report as HTML**.

2. From the **Parameter** window, enter the caseworker name as **ascewuser**, click **OK** to view the report.

Configuring the report configuration record
This task describes how to customize a report configuration record.

About this task
The process of customizing a report configuration record is to cancel the original record and to create a new active record. The new configuration record contains the path to the new customized BIRT report. This report is then displayed on the user workspace. This task describes how to cancel an existing report configuration record and how to create a new active reporting configuration record.

Procedure

1. Log in to the system as a user with *sysadmin* credentials.

2. Select **System Configurations** > **Shortcuts Panel** > **Business Intelligence** > **Report Configuration**.

3. From the **Report Configuration** tab, update the **Report Category** drop-down list to select **Case** and click **Search**.

4. Select **Report Name** column to order the list.

5. For the report name **CaseWorkloadReport**, select the action menu and click **Edit**.

6. From the **Edit Reporting Configuration** window, enter the following details and click **Save**

Table 27: New report configurations

Name	Description
Report Name	The name of the report.
Report File Name	The name of the report with extension and path.
Report Category	The report category type. It is configured in the *CT_BIReportCategory.ctx* file.
Report Servlet	The report server name.
Width	The report width in pixels or percent.
Height	The report height in pixels or percent.
Scrolling	The report scrolling parameter.
Frame Border	The report frame border settings.
Description	The report description.

Report configuration values for a custom chart

The following values are specific for a custom chart configured for the SPMP Analytics accelerator.

Table 28: Custom report configurations

Name	Value
Report Name	CaseWorkloadReport
Report File Name	*components/SPMAnalytics/birt/ CustomizedPODAnalyticChart.rptdesign*
Report Category	Default value
Report Servlet	blank
Width	400
Height	250
Scrolling	Default value
Frame Border	Default value
Description	This report displays active Sickness Benefit and Incapacity Benefit cases as a bar chart.

Customizing the pod title

Complete the following steps to change the title of the CaseWorkLoad pod.

About this task

To create a new a title for the pod, a new properties file is required. This properties file changes the pod title from `Caseload Summary` to `My Assigned Active Benefit Cases.`

Procedure

1. From the `%CURAM_DIR%\webclient\components\SPMAnalytics` directory, create the directory `components\SPMAnalytics\javasource\core\i18n`.

   `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative \Curam\Development`.

2. Create a new file called `CaseWorkloadSummary.properties`. You can view the associated sample properties file.

3. Open a command prompt and change to the `%CURAM_DIR%\webclient` directory.

4. Run the **build client** build target.

5. From Eclipse, select the *webclient* project, right-click and click refresh. Start the server and Tomcat as normal.

Pod properties for SPMP Analytics

The following sample code includes the pod properties that are defined for the SPMP Analytics accelerator.

Table 29: CaseWorkloadSummary.properties

```
#
# 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.
#
######################################################################
#
#  Organization and location summary Properties
#  --------------
#
#  Property resource for all localisable
#  attributes of the case workload widget.
#
######################################################################

caseloadsummary.pod.title=My Assigned Active Benefit Cases
caseloadsummary.gotomycases=My Cases
```

Configuring the pod name
This task describes how to customize a pod type record.

About this task
This task describes how to change the name of an existing pod, the new pod name is displayed when a user is customizing their user workspace.

Procedure

1. Log in to the system as a user with *sysadmin* credentials.
2. Select **System Configurations** > **Shortcuts Panel** > **Application Data** > **Code Tables**.
3. From the **Code Tables** tab, navigate to the **Search Criteria Panel**, from the **Name** text box enter **PodType** and click **Search**.
4. From the **Search Results**, expand the **Pod Type.**
5. Select the **Technical ID(Code)** column to order the list.
6. From the row with a code value of *PT2005*, select the action menu and click **Edit**.
7. From the **Edit Item** window, go to the **Item Name** text box and enter **My Assigned Active Benefit Cases**.
8. Click **Save** to update the pod name.
9. Click **Publish**.

Configuring a pod for a user workspace
Complete the following steps to add a pod to a user workspace container by using the default wizard provided in the Administrator application.

Procedure

1. Log in to the system as a user with administrator credentials.

2. Select the **Administration Workspace** tab, expand the **Shortcuts** menu, and select **User Interface** > **Personalized Pod Pages**.

3. From the **Personalized Pod Pages** window, select the action menu and click **Edit** for the **ASCEligibilityWorkerHome UIM Page ID**.

4. From the **Configure a Personal Page** wizard, select the **My Assigned Active Benefit Cases** check box to include as an available pod to be displayed on the users workspace home page and then click **Next**.

5. From the **Defaults Pods**, ensure **My Assigned Active Benefit Cases** is selected, click **Next**.

6. From the **Page Layout**, click **Save**.

Publishing the chart
Complete the following steps to publish the new BIRT chart and view it on the test page.

Procedure

1. From your Cúram BI content directory `%CURAM_DIR%\BIContent`, run the following command to publish BIRT reports to the run time viewer.

```
build client.birt
```

> **Note:** `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

2. From your browser, enter the following URL `http://localhost:9080/CuramBIRTViewer/List`. This URL displays a list of available reports for each of the components.

3. Scroll down to the **CustomizedPODAnalyticChart.rptdesign**, from the **Mode** column select **R** to display the report, the **Parameter** window is displayed.

4. From the **Parameter** window, select the **userName** text box and enter **ascewuser** and click **OK.**. Verify that the **My Assigned Active Benefit Cases** chart is displayed.

Testing the user workspace page
Complete the following steps to view the new pod on the user workspace.

Procedure

1. From Eclipse, refresh the webclient and EJBServer projects. Start the Cúram server, the RMILoginClient and Tomcat as normal.

2. From the **RMILoginClient** window, enter the **ascewuser** user credentials.

3. From your browser, open the URL and enter `http://localhost:9080/Curam/AppController.do`.

4. From the user home page, select **Customize**, verify the **My Assigned Active Benefit Cases** check box is selected, click **Save**

5. From the user workspace, drag the pod to be the second column, make the pod the first widget in this column.

6. Select the **Customize** button, the **Customize** panel is displayed. Click **Save** to save your changes.

Saving your chart customizations

After you configured a new chart by customizing an existing chart, you must manually save all of the chart configurations to ensure that they are not overwritten when you do a build database target.

Saving your report configurations

Complete the following steps to extract the specified personalized pod page DMX files from the database and put them under source control.

Procedure

1. Extract the DMX files from the database with the **build extractdata** build target.

2. From the *%CURAM_DIR%\EJBServer\build\dataextractor* directory, copy the following DMX files.

Name	Description
BIREPORTCONFIGURATION.dmx	Stores the details for the business intelligence report configurations.

> **Note:** *%CURAM_DIR%* is the Cúram installation directory, which by default is *C:\Merative\Curam\Development*.

3. From *%CURAM_DIR%\EJBServer*, run **build database**.

4. Save your configuration files.

Report configuration files for SPMP Analytics

The location of the report configuration file and pod configuration file specified for the SPMP Analytics accelerator.

File name	Location
BIREPORTCONFIGURATION.dmx	*%CURAM_DIR%\EJBServer\components\SPMAnalytics\data\initial*

> **Note:** *%CURAM_DIR%* is the Cúram installation directory, which by default is *C:\Merative\Curam\Development*.

Saving the pod name configurations

Complete the following steps to extract the specified personalized pod page DMX files from the database and put them under source control.

Procedure

1. Extract the DMX files from the database with the **build extractdata** build target.

2. From the *%CURAM_DIR%\EJBServer\build\dataextractor* directory, copy the following DMX files.

Name	Description
CODETABLEITEM.dmx	Stores pod name for the customized pod. The identifiers are referenced from the *CT_PodType.ctx* code table. The code table code for the record *PT2005* was customized.

> **Note:** `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

3. From `%CURAM_DIR%\EJBServer`, run **build database**.
4. Save your configuration files.

Pod name files for SPMP Analytics

The location of the report configuration file and pod configuration file specified for the SPMP Analytics accelerator.

File name	Location
CT_PodType.ctx	`%CURAM_DIR%\EJBServer\components\SPMAnalytics\codetable` > **Note:** `%CURAM_DIR%` is the Cúram installation directory, which by default is `C:\Merative\Curam\Development`.

Notices

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the Merative website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of Merative

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of Merative.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

Merative reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by Merative, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

MERATIVE MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Merative or its licensors may have patents or pending patent applications covering subject matter described in this document. The furnishing of this documentation does not grant you any license to these patents.

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

Any references in this information to non-Merative 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 Merative product and use of those websites is at your own risk.

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.

The licensed program described in this document and all licensed material available for it are provided by Merative under terms of the Merative Client Agreement.

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 Merative, 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. Merative, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. Merative shall not be liable for any damages arising out of your use of the sample programs.

Privacy policy

The Merative privacy policy is available at https://www.merative.com/privacy.

Trademarks

Merative ™ and the Merative ™ logo are trademarks of Merative US L.P. in the United States and other countries.

IBM®, the IBM® logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

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.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

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

Microsoft™, Windows™, 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 company, product, and service names may be trademarks or service marks of others.