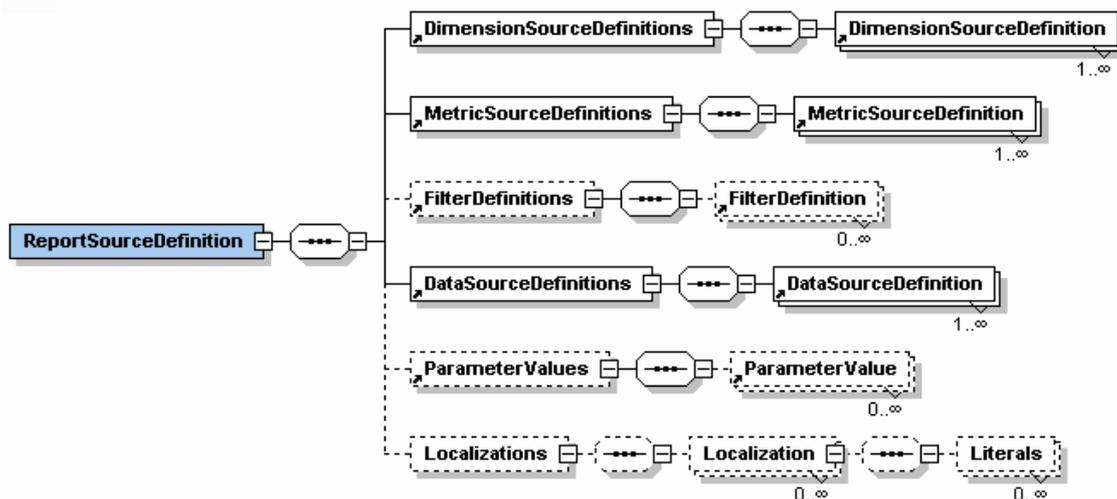


REPORTSOURCEDEFINITION.XML

Describe un origen de datos

Descripción de los elementos



ReportSourceDefinition

Atributos: definen propiedades genéricas del origen de datos

Id: identificador único. String-Requerido. Ejemplo: RSD_DISTRIBUTION_SALARY_CUBE. Es utilizado para referenciar al RSD desde el RD.

Cached: Boolean-Requerido. Este flag indica si los datos obtenidos de los orígenes descritos serán almacenados temporalmente en disco o no. Por ejemplo, si tengo un reporte con un origen de datos SQL, puedo querer lanzar la query cada 10 días y el resto del tiempo usar una “imagen” de esos datos para no utilizar el servidor de base de datos. El origen es guardado en un archivo que puede ser plano separado por “;” o una serialización nativa de java. Por lo tanto, si esta opción es true, en la primera obtención de los datos, estos serán guardados, para luego accederlos, hasta que venza el plazo de cacheo. Ejemplo: false

Expiration: indica las **horas** de vigencia de los datos almacenados temporalmente. Boolean-Opcional. Una vez cumplida esta cantidad de horas, los datos nuevamente se obtienen del origen, generando un cache actualizado. Ejemplo: 240 (diez días)

MultiLanguage: Esta opción es usada en conjunto con el elemento Localizations. Boolean-Opcional. Si es true intentará traducir cada descripción String del XML a un valor de localization según la zona configurada, siendo dicha búsqueda por códigos. De esta manera, puedo poner



como descripción de una dimensión "01", y luego hacer 2 localizations, una con "ZONA" para español y "ZONE" para ingles. Ejemplo: false

MaxRowsDimensionReference: Es el nombre de la dimensión que se usará para obtener los primeros n valores, utilizando un filtro de tipo TOP. Ejemplo: CLIENTE

MaxRowCount: indica cuantos valores se van a obtener. Utilizado para el filtro top. Más allá de ese valor, se descartarán los datos. Ejemplo :20

DimensionSourceDefinition: define las propiedades sobre los valores que serán dimensiones del reporte

Atributos:

Name: Nombre de la dimensión. Debe ser único. Ejemplo: CLIENTE.

Description: Descripción de la dimensión por defecto. Este será el valor que se muestre para el titulo de la dimensión en el reporte. Puede ser sobrescrito en el RD. Ejemplo : Cliente.

DataType: Indica el tipo de dato que contendrá la dimensión. Los valores posibles son: STRING, DATE, DATETIME, FLOAT, INTEGER, BOOLEAN. Cuando se este cargando el origen de datos, se intentará transformar los valores a su tipo correspondiente según la columna en que estén. Ejemplo: string.

Calculated: Se pueden calcular dimensiones a partir de valores de otras dimensiones. Este atributo indica si la dimensión es calculada a partir de los valores de otras dimensiones. En caso de ser true, el valor de esta dimensión no se obtiene del origen de datos, sino a partir de una dimensión obtenida anteriormente. Ejemplo: true.

Expression: Solo se usa cuando calculated es true. Es una expresión que permite obtener los valores de dicha dimensión a partir de una anterior. Ejemplo: MONTH(DIMENSION_FECHA) . Ver Anexo funciones de fecha.

ExternalData: Define el nombre "Externo" de la dimensión dentro del origen de datos. Por ejemplo en una consulta SQL "select client from ..." la ExternalData debería ser client, por más que el nombre de dimensión sea distinto. De esta manera, cuando se obtiene el valor de dimensión para el registro actual, se examina la columna que tenga el nombre igual al ExternalData. Ejemplo: CLIENT

MetricSourceDefinition: define las propiedades de los valores que se utilizarán para medir o contar (métricas del reporte).



Atributos:

Name: Nombre de la métrica. Ejemplo: PRECIO

Description: Nombre que verá el usuario asociado a esta métrica. Ejemplo: Precio

AggregateType: Operación que se utiliza para calcular el valor de una métrica. Ejemplo: SUM. Ver Anexo AggregateType y GroupFooterType

GroupFooterType: Operación que se utiliza para calcular el valor de totales de una métrica. Ejemplo : NONE. Ver Anexo AggregateType y GroupFooterType

Calculated: Indica si la métrica es calculada a partir de los valores de otras métricas. Se pueden hacer cálculos basados en operaciones matemáticas simples. Ejemplo : false.

AggregateFunction: Expresión que se utiliza para calcular el valor de una métrica calculada. Solo se utiliza en caso que calculated es true. Anexo AggregateFunction y GroupFooterFunction

GroupFooterFunction: Expresión que se utiliza para calcular el valor de totales de una métrica calculada. Solo se utiliza en caso que calculated es true. Anexo AggregateFunction y GroupFooterFunction

ExternalData: Idem a DimensionSourceDefinition

FilterDefinition: define los filtros que se aplicarán al reporte.

Los valores por defecto de los filtros de tipo fecha deben ser simple yyyyMMdd, luego se formatean según el Locale configurado de la forma apropiada para ser mostrados.

```
<FilterDefinitionName="FILTRO_FECHA" FilterParameter="FROM"
DefaultValue="19970101"/>
```

Atributos:

Name: Nombre del filtro. Ejemplo: RANGO_CLIENTE

DimensionName: Nombre de la dimensión que se va a filtrar. Ejemplo: CLIENTE

FilterType: Tipo de filtro. Este valor será uno de los siguientes: RANGE, GREATERTHAN, LESSTHAN, EQUALTO, BEGINWITH, ENDWITH, INCLUDES, RANKING, IN o EXCLUDEGROUP.

RANGE: filtra un rango de valores, tomando un parámetro inicial y uno final, excluyendo los valores que estén fuera de ese rango definido.

GREATERTHAN: toma valores que estén por encima de cierto parámetro definido. Excluye los valores que sean menores al valor dado.



LESSTHAN: toma valores que estén por debajo de cierto parámetro definido. Excluye los valores que sean mayores al valor dado.

EQUALTO: toma valores iguales a cierto parámetro dado. Excluye los valores distintos al valor dado

BEGINWITH: solo para dimensiones de tipo String. Toma valores que empiecen con el valor dado por parámetro.

ENDWITH: solo para dimensiones de tipo String. Toma valores que terminen con el valor dado por parámetro.

INCLUDES: solo para dimensiones de tipo String. Toma valores que contienen en alguna sección el valor dado por parámetro.

RANKING: Realiza un orden de la dimensión, según 2 criterios opcionales: si la dimensión tiene definida un RankMetricName (ver RD), se ordena la dimensión por sus valores correspondientes a esa métrica y solo se guarda una cantidad de valores dada por parámetro. Entonces, si quiero filtrar los 5 primeros clientes según la métrica IMPORTE, en la dimensión CLIENTE debo definir RankMetricName="IMPORTE", y en el parámetro setear 5. Si la dimensión no tiene métrica de ranking, entonces se ordena con su orden habitual (ej: alfabetico), y se filtran los primeros n según ese orden.

IN: en este caso, el parámetro define un grupo de valores separados por coma (si el valor que se quiere incluir tiene coma debe encerrarse en comillas simples ej: 'perez, juan'). El filtro toma los valores que coincidan con algún valor de ese conjunto, y excluye los que no son iguales a ningún valor dado.

EXCLUDEGROUP: este es el filtro más complejo y se explicará mediante un ejemplo. Supongamos que yo quiero mostrar todos los clientes que compraron más de 10% de mi total de ventas. Estos clientes serán supongamos 3 o 4, ya que el resto son clientes chicos que compran menos del 10%. Además, el resto de los clientes, quiero que aparezca agrupado en un nodo "OTROS", es decir, sin discriminar, y agrupando todos los valores en ese grupo. Pero además tengo un cliente especial que quiero si o si listarlo, aunque no cumpla la condición del 10%. Todo esto lo puedo realizar con el filtro EXCLUDEGROUP. Principalmente lo que hace este filtro es generar ese grupo "OTROS", que se arma respetando 2 parametros. Uno dice la operación a realizar, que puede ser GreaterThan, LesserThan, GreaterAverage, LesserAverage, Top, RangeAverage, Range. El otro dice el valor a respetar. Por ejemplo, en nuestro caso teníamos como operación GreaterAverage, y como valor 10. Entonces el filtro separa los que tienen mayor que 10% y agrupa el resto. Por otra parte con el parámetro OBLIGATORYVALUE, se puede definir una lista de valores separados por coma que estarán obligatoriamente.

DataType: Indica el tipo de dato del filtro. En caso que se opte por no asociar una dimensión para el filtro (DimensionName=null), sino que será un parámetro simplemente para el reporte, se puede optar por especificar el tipo de dato para el mismo (String, date, integer, etc). Ejemplo: string.

Visual: Indica si el filtro se debe mostrar por pantalla para ser ingresado por el usuario o no (en caso negativo se utiliza el valor default del parámetro). Ejemplo: true

DataSourceDefinition : define los orígenes de datos. Es la sección fundamental del RSD

Atributos:



Name: Nombre del origen de datos. Ejemplo: MYSQL_CLIENT_DATA

SourceType: Tipo de origen de datos. El tipo esta comprendido entre: SQL, XML, o EXCEL. El tipo SQL es para orígenes establecidos en bases de datos, orientados a consultas SQL. XML es para orígenes de características de “catálogo” dentro de un xml. El tipo Excel es para extraer datos de una planilla Excel.

Expression: una expresión que será utilizada para obtener los datos. En el caso de SQL va la sentencia: “Select client from ...”. En el caso de Excel o xml va un path (absoluto o relativo) al origen de datos: “./sourcefiles/datasources/salary.xls” (el directorio datasources dentro de sourcefiles es utilizado habitualmente para dejar los archivos de orígenes de datos).

ExternalConnectionString: definine la conexión al data source si es necesario. Es preferible utilizar los otros valores definidos para este fin, como ExternalConnectionLocalUrl y otros. Ejemplo: “DatabaseConfigurationClassName=com.mysql.jdbc.Driver;DatabaseConfigurationLocalUrl=jdbc:mysql://localhost:3306/mydb;DatabaseConfigurationDistributedUrl=java:comp/nv/jdbc/MyUrl;DatabaseConfigurationUser=user;DatabaseConfigurationPassword=password”. Se puede usar el ExternalConnectionString completo o se puede especificar por separado cada uno de los parámetros en los siguientes atributos.

ExternalConnectionClassName: define el nombre de la clase del driver jdbc a utilizar. Hay que especificar el nombre completo. Por ejemplo:

Sun ODBC: **sun.jdbc.odbc.JdbcOdbcDriver**

Oracle: **oracle.jdbc.driver.OracleDriver**

Este valor depende del driver que este utilizando para conexión a la base de datos.

ExternalConnectionLocalUrl: es la Url utilizada para que el driver se conecte a la base. También es especificada por cada fabricante la forma de escribir ese connection:

MySQL: **jdbc:mysql://localhost:3306/foodmart**

Oracle: **jdbc:oracle:thin:@host:port:base_name**

ExternalConnectionDistributedUrl: String de conexión a un origen de datos en caso de que la aplicación corra en forma remota conectando a otro server. Ejemplo: java:comp/nv/jdbc/intercarga

ExternalConnectionUser: Nombre de usuario de la base de datos.Ejemplo: root

ExternalConnectionPassword: Password del usuario de la base de datos.Ejemplo: 1



Opción para encriptar: dado que el nombre y password van directo al xml, es posible dejarlos encriptados, utilizando el método de encriptación de la clase LocalKeyEncrypter. El resultado de ese método debe ser escrito en el xml en lugar del nombre y password planos. Para seguridad, el LocalKeyEncrypter, toma datos de la máquina local para usar como clave. Esto hace que si el XML es llevado fuera del servidor, no se pueda desencriptar el valor por no contar con los datos necesarios para saber la clave.

DatePattern: Especifica el formato en que se encuentran los tipos de datos DATE para una base de datos en particular. Por ejemplo, yo puede que guarde la fecha con formato dd/MM/yyyy o mm/DD/yyyy. Especificando este valor, la aplicación sabe como convertir un dato que puede ser una representación String de una fecha en un origen de datos, en un tipo Fecha real.

DateTimePattern: Igual a DatePattern, pero para los campos DATETIME. En una base podemos guardar por ejemplo, la fecha de venta (DATE) y el instante de venta (DATETIME). Para poder aprovechar ambos datos como fecha necesitamos estos dos patterns, para convertirlos.

FloatPattern: Este pattern es para convertir un tipo de dato Float que esta representado como un String. El mismo concepto que para DatePattern.

SheetName (Solo para origen de dato Excel): Indica que nombre de hoja seleccionar dentro del documento excel.

DataInitialCell (Solo para origen de dato Excel): Es una referencia a la celda inicial donde tomar los datos. Por default es A1, pero puede que requiera cambiarla según como estén dispuestos los datos en la planilla. La sintaxis es igual a la que utiliza Excel, letra|número

DataEndingCell (Solo para origen de dato Excel): Es una referencia a la celda final donde se debe dejar de tomar los datos. Por ejemplo, puede que se necesite tomar solo las primeras líneas de la planilla, entonces, ponemos esta celda con dicho valor deseado.

ParameterValue: define los valores por defecto y comportamiento de los filtros definidos.

Atributos:

FilterDefinitionName: Nombre del filtro. Ejemplo: DESDE_CLIENTE

FilterParameter: Identifica el parámetro que estamos definiendo dentro del filtro. Esto se debe a que hay filtros con más de un parámetro asociado, por ejemplo el rango tiene parámetros FROM y TO. El valor de este atributo está comprendido entre: FROM, TO , VALUE, VALUES, OPERATION o OBLIGATORYVALUE



Para filtros de tipo BEGINWITH, ENDWITH, EQUALTO, GREATERTHAN, INCLUDES, LESSTHAN y RANKING: el único parámetro utilizado es VALUE

Para filtros de tipo RANGE: se usan FROM y TO

Para filtros de tipo IN: se usa VALUES

Para filtros de tipo EXCLUDEGROUP: se usan FROM, TO , VALUE, OPERATION o OBLIGATORYVALUE (el FROM y TO son utilizados para cuando la OPERATION de tipo rango).

DefaultValue: Valor por defecto que tendrá el parámetro.

Localizations: define el multilinguaje basado en la traducción de códigos

Atributos:

Locale: indica el lenguaje que se redefinirá a continuación. Ejemplo: EN

Literals: define cada uno de los textos

Atributos:

Code: código que identifica al literal correspondiente. Ejemplo: 01

Value: String en el lenguaje correspondiente al Locale del Localization que contiene este Literal. Ejemplo: Client

Con todos estos valores, se podrá definir por ejemplo, como nombre de una dimensión un código (01, 02, 03, etc), que luego será buscado en los Literals, y si está será traducido según el idioma que este seteado.