

Para solucionar, en gran parte, el problema de la codificación de caracteres en Pentaho, esto es cuando las xactions de análisis contienen caracteres no UTF-8, por ejemplo: ñ, á, é, etc. Hemos sistematizado algunos puntos que a continuación se detallan, cabe destacar que han sido probados en Pentaho 2 y 3 y que hay un caso en el que no funciona, que es cuando se crea un nuevo análisis y "antes de grabar" en la consulta MDX existe alguno de los caracteres no compatibles, esto es debido a que la solución trabaja con archivos físicos, esto es, luego de grabar.

Pasos:

1) Modificar el archivo de configuración de tomcat ...\\biserver-ce\\tomcat\\conf\\server.xml, asegurar la existencia del atributo **URIEncoding="ISO-8859-1"** en la definición del conector http.

```
...
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
<Connector port="8080" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true"
    URIEncoding="ISO-8859-1"/>
<!-- Note : To disable connection timeouts, set connectionTimeout value .....
```

2) Modificar el descriptor de despliegue ..\\biserver-ce\\tomcat\\webapps\\pentaho\\WEB-INF\\web.xml, asegurar que las secciones que se muestran abajo contengan como valor: **ISO-8859-1**

```
...
<context-param>
    <param-name>encoding</param-name>
    <param-value>ISO-8859-1</param-value>
</context-param>
...
...
...
<!-- This must be the first filter listed in the web.xml -->
<filter-name>Set Character Encoding Filter</filter-name>
    <filterclass>
        org.pentaho.platform.web.http.filters.PentahoAwareCharacterEncodingFilter
    </filter-class>
    <init-param>
        <param-name>ignore</param-name>
        <param-value>no</param-value>
    </init-param>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>ISO-8859-1</param-value>
    </init-param>
</filter>
...
```

3) Modificar las plantillas para nuevos análisis ..\\biserver-ce\\pentaho-solutions\\system\\analysisTemplate.tpl y ..\\biserver-ce\\pentaho-solutions\\system\\mondrian\\analysis_view_template.xaction, asegurarse que las cabeceras poseen como valor del atributo encoding **ISO-8859-1**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
    <action-sequence>
    ...
```

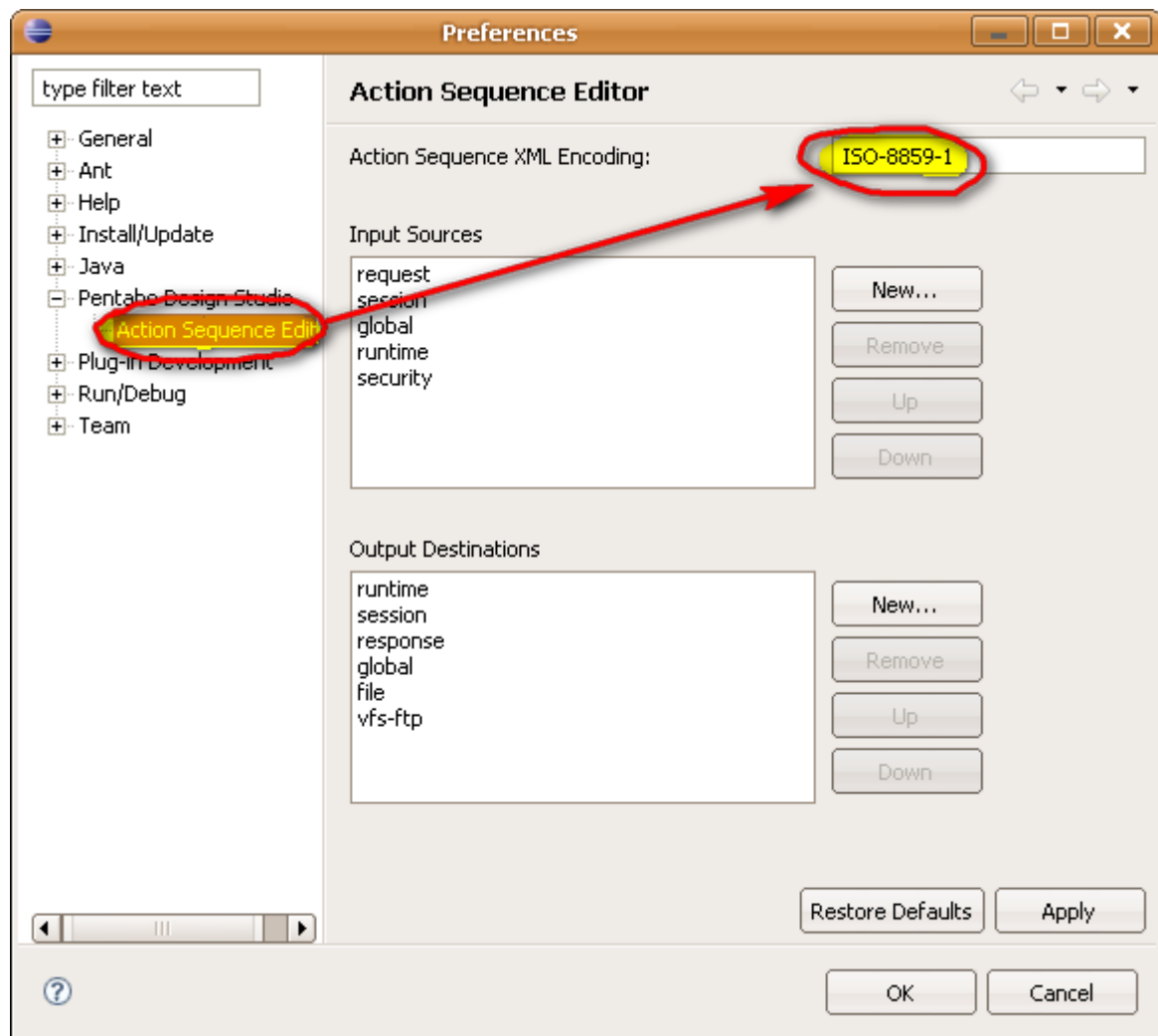
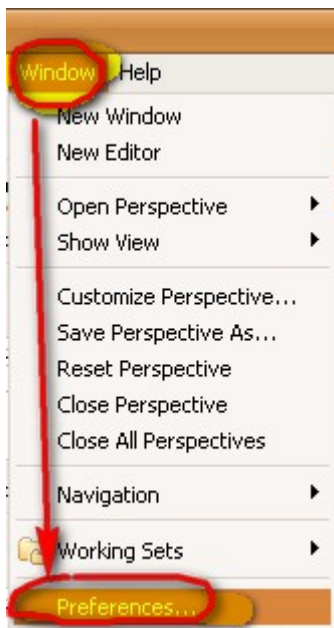
4) Modificar todas las xactions existentes y asegurar lo definido en el punto 3.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <action-sequence>
  ...
```

5) Modificar la JSP ..\biserver-ce\tomcat\webapps\pentaho\Pivot.jsp, agregando el código que se lista a continuación, este código asegura que cada xaction de análisis que se almacena (usando Guardar o Guardar como) posea en su cabecera la codificación ISO-8859-1.

```
if (( "save".equals(saveAction)) || ("saveAs".equals(saveAction))) {
    // Overwrite is true, because the saveAs dialog checks for overwrite, and we
    // never
    // would have gotten here unless the user selected to overwrite the file.
    try{
        saveResult = AnalysisSaver.saveAnalysis(userSession, props,
            request.getParameter("save-path"), request.getParameter("save-file"),
            true);
        // ===== Agregado por eGlu (Dario & Mariano)
        String solutionRootPathMagm =
            PentahoSystem.getApplicationContext().getSolutionRootPath();
        java.io.File fMagm=new java.io.File(solutionRootPathMagm,
            request.getParameter("save-path"));
        java.io.File flMagm=new java.io.File(
            fMagm,request.getParameter("save-file"));
        java.io.BufferedReader leer=new java.io.BufferedReader(
            new java.io.FileReader(flMagm));
        StringBuffer flSB=new StringBuffer();
        String lineaArch=null;
        while((lineaArch=leer.readLine())!=null){
            flSB.append(lineaArch.replaceAll("\\"UTF-8\\", "\\"ISO-8859-1\\")+"\n");
        }
        leer.close();
        java.io.BufferedWriter fSalidaMagm = new java.io.BufferedWriter(
            new java.io.FileWriter(flMagm));
        fSalidaMagm.write(flSB.toString());
        fSalidaMagm.close();
        // ===== fin agregado por eGlu (Dario & Mariano)
        switch (saveResult) {
            case ISolutionRepository.FILE_ADD_SUCCESSFUL:
                saveMessage = Messages.getString("UI.USER_SAVE_SUCCESS");
                // only set the session attribute on success, it's the only path that
                // requires it
                session.setAttribute( "save-message-01", saveMessage); //$NON-NLS-1$
                break;
            ...
```

6) Configurar Design Studio para trabajar por defecto con ISO-8859-1, el procedimiento es simple, ver las siguientes capturas de pantalla.



Enjoy
eGlu