September 2008

How to unload Hierarchies from SAP BI 7.0

Authors

Hermann Daeubler, Senior Program Manager, Microsoft Corp.,
hermannd@microsoft.com
Juergen Daiberl, Senior Technical Evangelist, D&PE, Microsoft Corp.,
jdaiberl@microsoft.com

Summary

This paper describes how to unload hierarchies from SAP NetWeaver BI 7.0. In SAP NetWeaver BI master data usually consists of attribute, text and hierarchy data. A simple example could be data about employees like name, age, title and so on in a SAP BI system. In addition one could store hierarchy information which describes the organization chart (who is a manager and who reports to whom?).

The official interface in SAP NetWeaver BI to unload data is called Open Hub Service. It allows unloading master data of type “attribute” and “text” but has no option to unload hierarchy information. This raises the question how it could be accomplished to unload hierarchical information using other interfaces or mechanisms.

This paper gives an overview of several different ways of unloading hierarchical data from SAP NetWeaver BI.

Applies to

- Microsoft .NET
- Microsoft Business Intelligence (Microsoft BI)
- Microsoft SQL Server Integration Services
- Microsoft SQL Server Reporting Services
- SAP NetWeaver 04s
- SAP NetWeaver BI
- SAP NetWeaver BI hierarchies
- SAP NetWeaver BI queryview web service
- XMLA
- C#

Keywords

SAP NetWeaver, SAP NetWeaver BI, XMLA, C#, Microsoft BI, Microsoft SQL Server, SSRS, SSIS, SAP NetWeaver BI Hierarchies, SAP NetWeaver BI queryview

Level of difficulty

IT Management, Technical Architects, Technical Consultants, Developers

Contact

This document is provided to you by Microsoft Corp. Please check the SAP interoperability area at www.microsoft.com/sap and the .NET interoperability area in the SAP Developer Network (http://sdn.sap.com) for any updates or further information.

Microsoft®

Microsoft®

Contents

Microsoft®

Introduction

In certain customer scenarios it's necessary to unload Master Data from SAP BI in order to integrate with a Microsoft BI solution. As the official interface in SAP BI for this purpose (so-called Open Hub Service) only offers the capabilities to unload attribute and text data the key question is how to unload SAP BI hierarchies?

Here is a list which shows the most common ways for unloading hierarchical data from SAP BI:

- using an ABAP report from SAP which can be downloaded from the SAP Service Marketplace (strongly recommended)
- using XMLA and C#
- using the SAP BI queryview web service and C#
- using SSRS
- using Excel 7 and SAP BI OLE DB for OLAP driver (ODBO)
- using a browser and a SAP BI query
- BizTalk Adapter Pack
- 3rd Party Tool

This paper will discuss all of these options except the usage of the BizTalk Adapter Pack; this topic will be covered in one of the next papers.

As there is no public SAP BI API to unload hierarchies the recommended way is to use the ABAP report from SAP. Other options – especially if they make use of official SAP BI APIs like XMLA – are also supported but may have some disadvantages like additional coding effort or the necessity of using an additional tool. Whereas the ABAP report from SAP will unload a hierarchy simply into a flat file.

In the end there are absolutely unsupported ways like direct database table access which shouldn't be used at all. Besides the fact that a customer might lose SAP support it's also a technical challenge as the SAP BI internal Meta data layer related to hierarchies isn't public.

IMPORTANT: Please keep in mind that 'UNLOADING' data from SAP BI requires additional SAP licensing. The customer has to be aware of this and we recommend that the customer checks this directly with SAP.

Sample

The sample uses a hierarchy out of a SAP demo cube which is part of the so-called "BI Content". The idea is to show different ways how this demo hierarchy can be unloaded.

For this paper we will not cover any security aspects. The paper doesn't describe a full solution including the creation of a hierarchy in a new target system like SQL Server Analysis Services based on the unloaded data from SAP BI, instead of this the focus was on the interface side of SAP BI and how to get the hierarchy data out.

Description of the demo hierarchy:
Out of the so-called "BI Content" (pre-defined objects and structures) the SAP Demo Cube with technical name 0D_DECU was used. The InfoObject with the technical name 0D_COUNTRY is part of this InfoCube and has a hierarchy. After activating the objects a load file with some test data will be created in the SAP work directory (0D_COUNTRY_HIER_ENGLISH.CSV) which then can be loaded via regular SAP BI mechanisms. The technical name of the hierarchy was set to HCOUNTRY01 for our sample. The hierarchy itself is pretty simple and consists of two regions with five countries:

- Europe : Germany, France, England
- America : USA, Canada

The following sections and chapters will describe how this information can be unloaded from SAP BI. The paper will just cover the basic steps.

- Using an ABAP report from SAP which can be downloaded from the SAP Service Marketplace on http://sdn.sap.com (strongly recommended).

  On the SAP Developer Network SDN you can find a big set of "How-To" papers. One of them from June 2004 with the title "How to Download a Hierarchy to a Flat File" describes an ABAP report which can be installed for this purpose. After installation you can start it through transaction SE38, the name of the report is Z_SAP_HIERARCHY_DOWNLOAD. This report will unload the hierarchy into a flat file which basically looks the same as the load file for the SAP demo hierarchy. Once the hierarchy is unloaded any tool can be used to process the data (e.g. create and fill a hierarchy in SQL Server Analysis Services).

- Using XMLA and C#

  Another option is to declare the InfoObject as an InfoProvider and create a SAP BI query using the SAP BI query designer to retrieve the hierarchy data as the query result. Using the SAP BI XMLA reporting interface it's then possible to unload the data via a C# program. A separate paper was published which describes the unloading of SAP BI master data via XMLA and C# in more detail – see reference section at the end.

- Using the SAP BI queryview web service and C#

  The three official SAP BI reporting APIs work with MDX as the query language (OLAP BAPI, OLE DB for OLAP and XMLA). For those who don't want to use MDX SAP BI offers a web service which allows you to trigger an existing SAP BI query and retrieve the result by using some proprietary structures. It's not as powerful as XMLA but might be easier to handle for users who never worked with MDX before. How to activate this queryview web service is described on http://help.sap.com. Once it's active it can be used as a web reference inside a Visual Studio program. Some sample C# code is included at the end of this document to give you an idea how the structures look like.

- Using SSRS

  SQL Server Reporting Services is certified for the SAP BI XMLA interface. By defining a SAP BI query to retrieve hierarchy data it's of course possible to use this query as a data source for SSRS and create a report. The report could then be saved e.g. as a .csv file for further processing. In case SSRS is already installed and used to access SAP BI it wouldn't be a big effort to use it for unloading hierarchy data. But it's very likely slower than the other options mentioned above because of the XML conversion as well as the overhead of using a reporting tool like SSRS due to formatting and rendering. But depending on the size of a hierarchy and how often it has to be unloaded it might not matter.

- Using Excel 7 and SAP BI OLE DB for OLAP driver (ODBO)

  SAP worked together with SIMBA to support the Excel 2007 pivot table feature through the SAP BI OLE DB for OLAP driver which is shipped by SAP. SIMBA published a paper which describes the details on http://www.simba.com/docs/Connecting-to-SAP-BW-with-Microsoft-Excel-2007-PivotTables-and-ODBO.pdf. Tests have shown that it's possible to get the hierarchy information into Excel except the country "keys" (DE, FR, GB, US, and CA). These keys can be unloaded via the official Open Hub Service API as part of the standard master data; therefore it's still possible to finally create the correct and complete hierarchy in a new target like SSAS. Due to this fact the processing is probably more complicated though than using the other options above.

- Using a browser and a SAP BI query

  SAP BI offers the possibility to run a SAP BI query via HTTP. Once the BEX service works correctly the SAP BI query can be started by entering the appropriate URL:

  //<servername>:8000/sap/bw/BEx?sap-language=EN&bsplanguage=EN&CMD=LDOC&INFOCUBE=0D_COUNTRY&QUERY=SEL_HCOUNTRY01

  The InfoObject is declared as an InfoProvider, therefore the name of the "InfoCube" is in fact the technical name of the InfoObject. The name of the SAP Bi query is "SEL_HCOUNTRY01". The advantage is obvious, a simple browser is enough to retrieve the data including the country keys. But exporting the data ends up in a format which causes quite some effort for further processing.

- BizTalk Adapter Pack

  It is possible to call the RSNDI function which is used inside the SAP ABAP report via the SAP connectivity in the BizTalk Adapter Pack. But this setup will be described in a future paper. In the meantime you can take a look to http://www.microsoft.com/biztalk/en/us/adapter-pack.aspx to find more information about the new BizTalk Adapter for SAP.

- 3rd Party Tool

Eventually there is the option to use some 3rd-Party Tool which has the capability to unload SAP BI hierarchies. As one example among many the Xtract IS package from Theobald Software was tested. It includes a component as a plug-in for SQL Server SSIS in order to unload SAP BI hierarchies. For more information on the Theobald Software see http://www.theobald-software.com.

Walk-Through

Activate the SAP Demo Cube “0D_DECU” out of the BI Content.

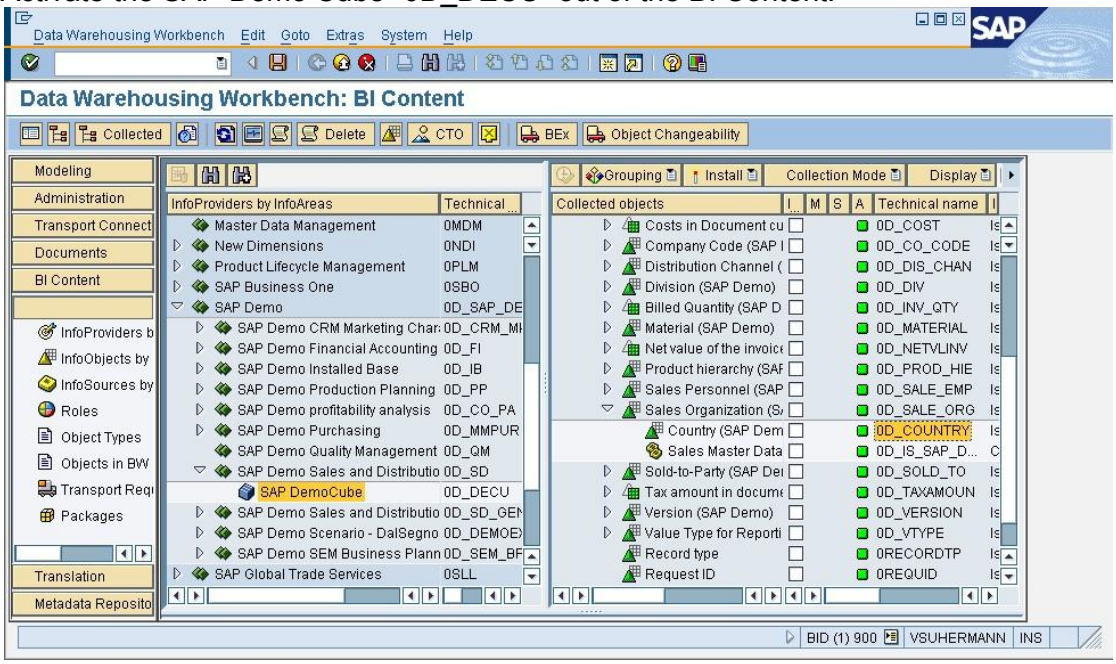


Figure 1 Data Warehousing Workbench

The InfoObject “0D_COUNTRY” includes a hierarchy.

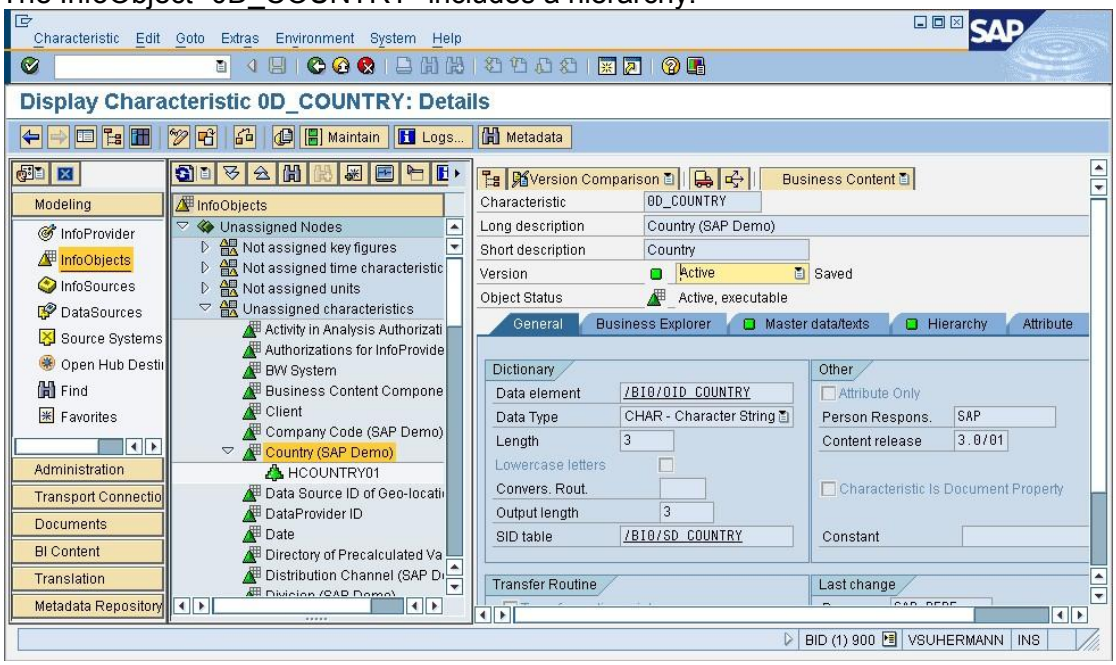


Figure 2 InfoObject “0D_Country”

Activation of the SAP Demo Cube creates the hierarchy load file with test data.

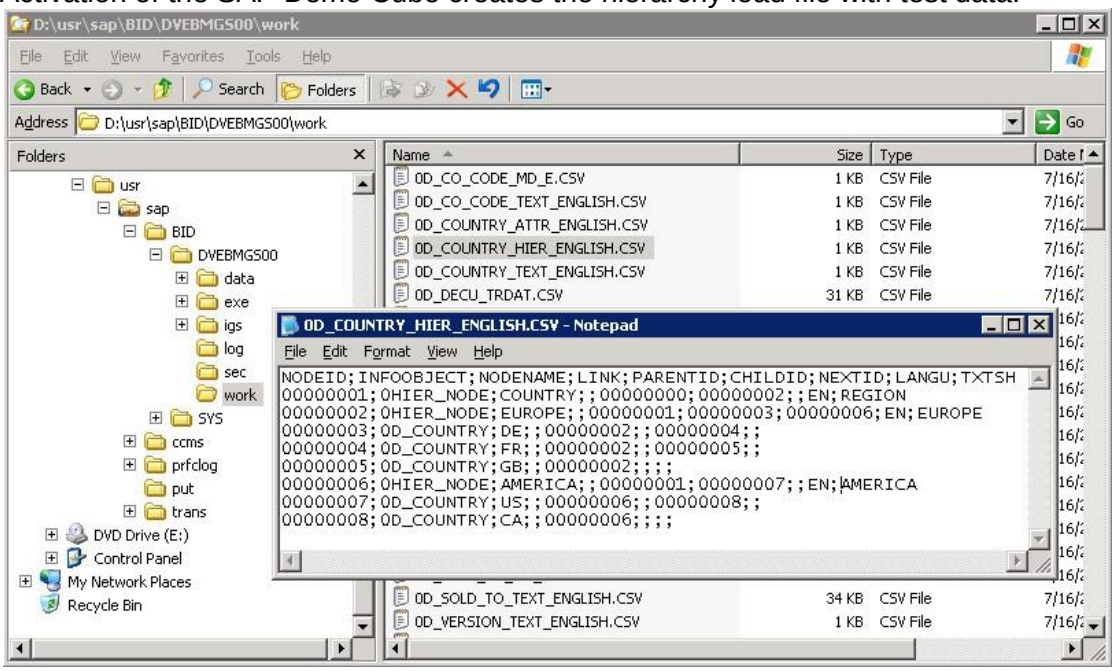


Figure 3 Activation of SAP Demo Cube

Check the hierarchy with the hierarchy maintenance in RSA1.

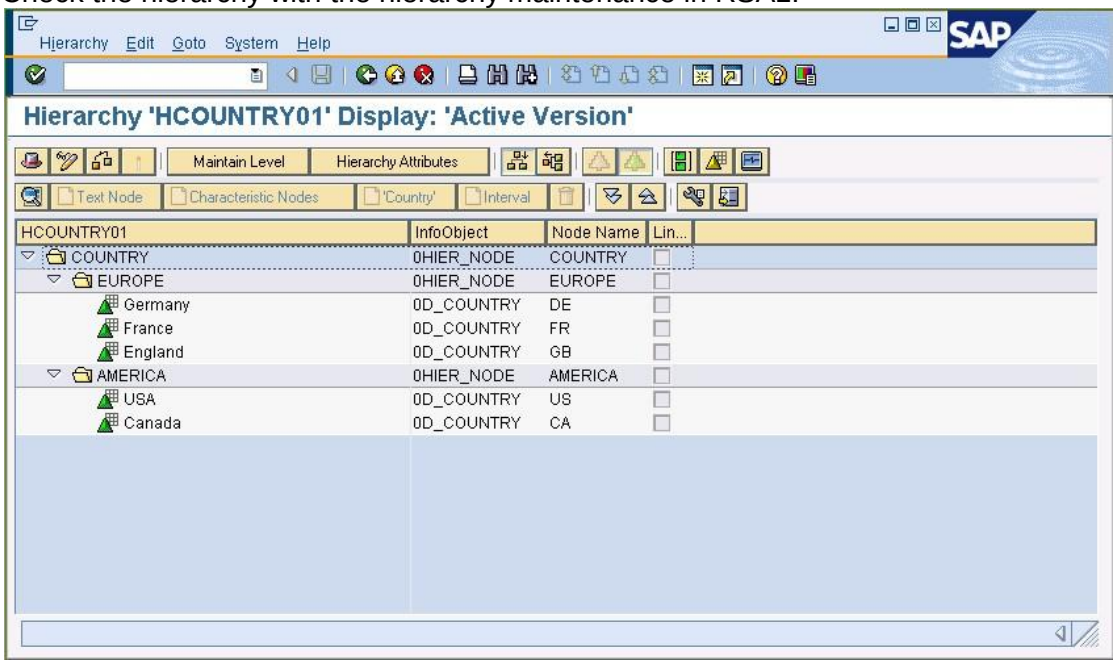


Figure 4 Hierarchy in SAPGUI

The “Hierarchy” tab shows database tables which store the data.

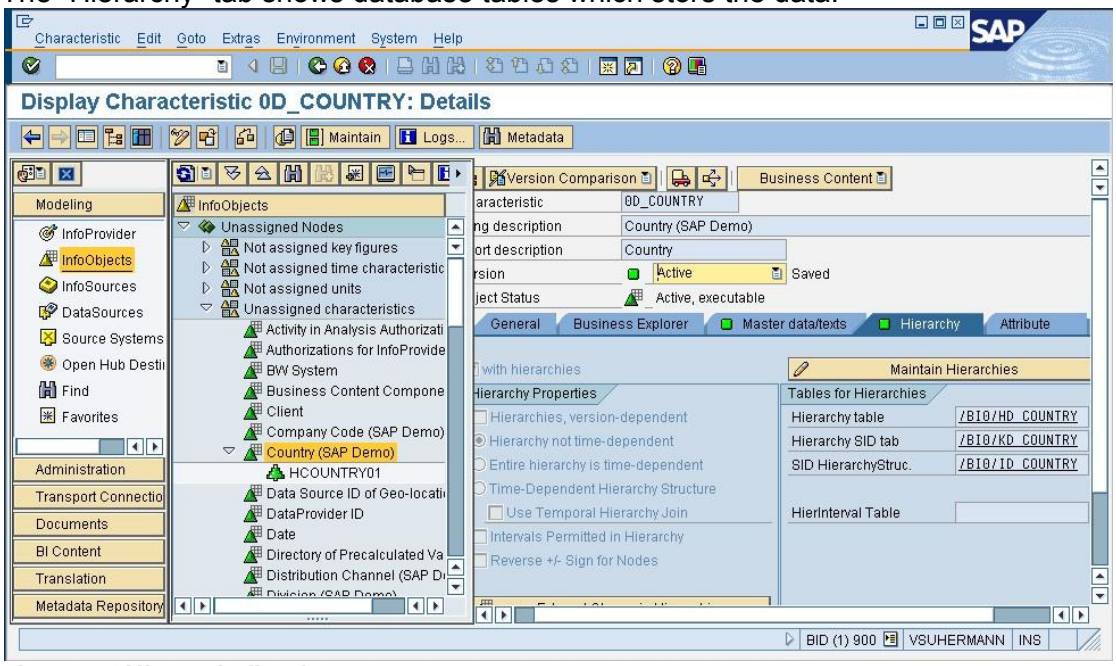
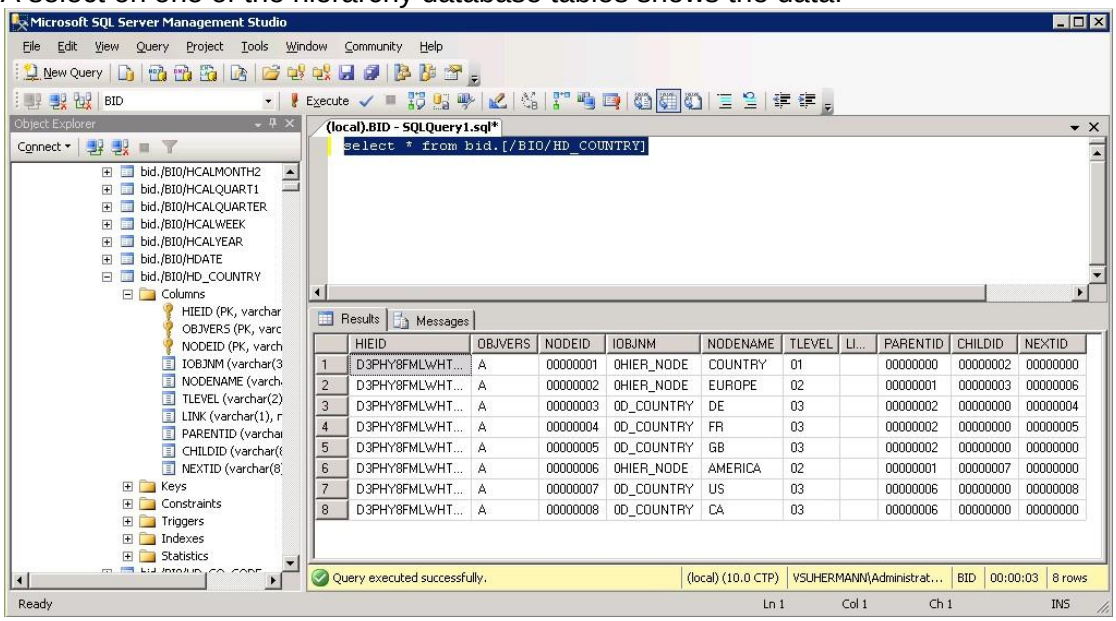


Figure 5 “Hierarchy” Tab

A select on one of the hierarchy database tables shows the data.



	HIEID	OBJVERS	NODEID	IOBJNM	NODENAME	TLEVEL	LI...	PARENTID	CHILDID	NEXTID
1	D3PHY8FMLWHT...	A	00000001	0HIER_NODE	COUNTRY	01		00000000	00000002	00000000
2	D3PHY8FMLWHT...	A	00000002	0HIER_NODE	EUROPE	02		00000001	00000003	00000006
3	D3PHY8FMLWHT...	A	00000003	0D_COUNTRY	DE	03		00000002	00000000	00000004
4	D3PHY8FMLWHT...	A	00000004	0D_COUNTRY	FR	03		00000002	00000000	00000005
5	D3PHY8FMLWHT...	A	00000005	0D_COUNTRY	GB	03		00000002	00000000	00000000
6	D3PHY8FMLWHT...	A	00000006	0HIER_NODE	AMERICA	02		00000001	00000007	00000000
7	D3PHY8FMLWHT...	A	00000007	0D_COUNTRY	US	03		00000006	00000000	00000008
8	D3PHY8FMLWHT...	A	00000008	0D_COUNTRY	CA	03		00000006	00000000	00000000

Figure 6 Tableview

IMPORTANT: It’s unsupported by SAP to try to unload this data directly as there are a few tables involved and it’s not public how they all relate to each other and what certain columns mean.

Declare the 0D_COUNTRY InfoObject as an InfoProvider to allow SAP BI queries on it.

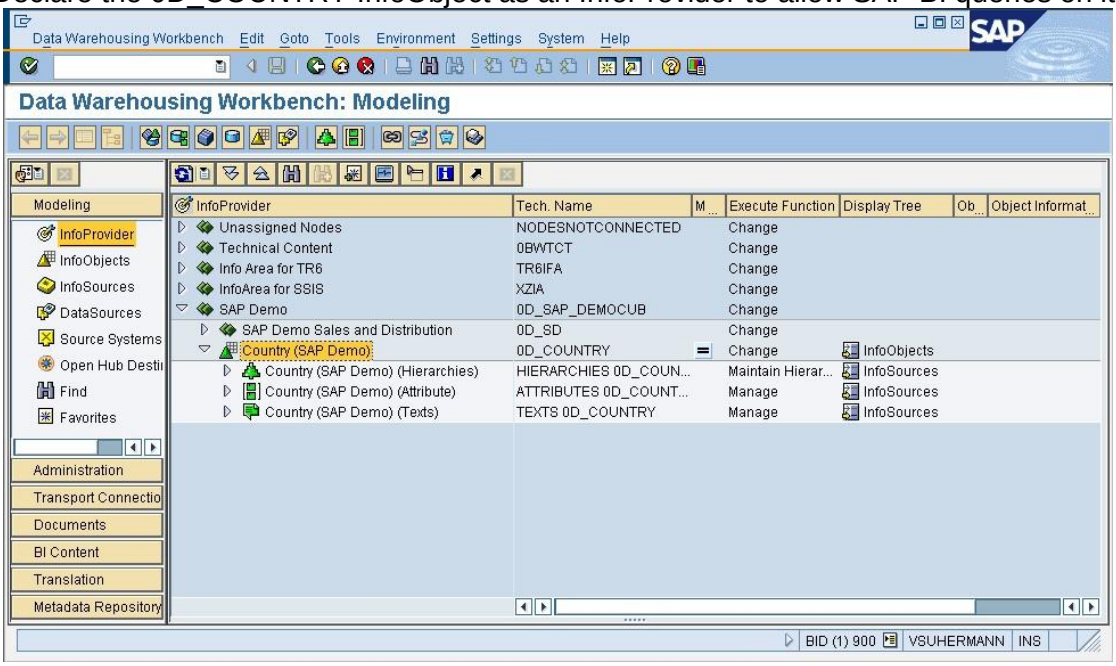


Figure 7 Declaration if 0Dcountry as InfoProvider

Define a SAP BI query to retrieve the hierarchy information of the InfoObject.

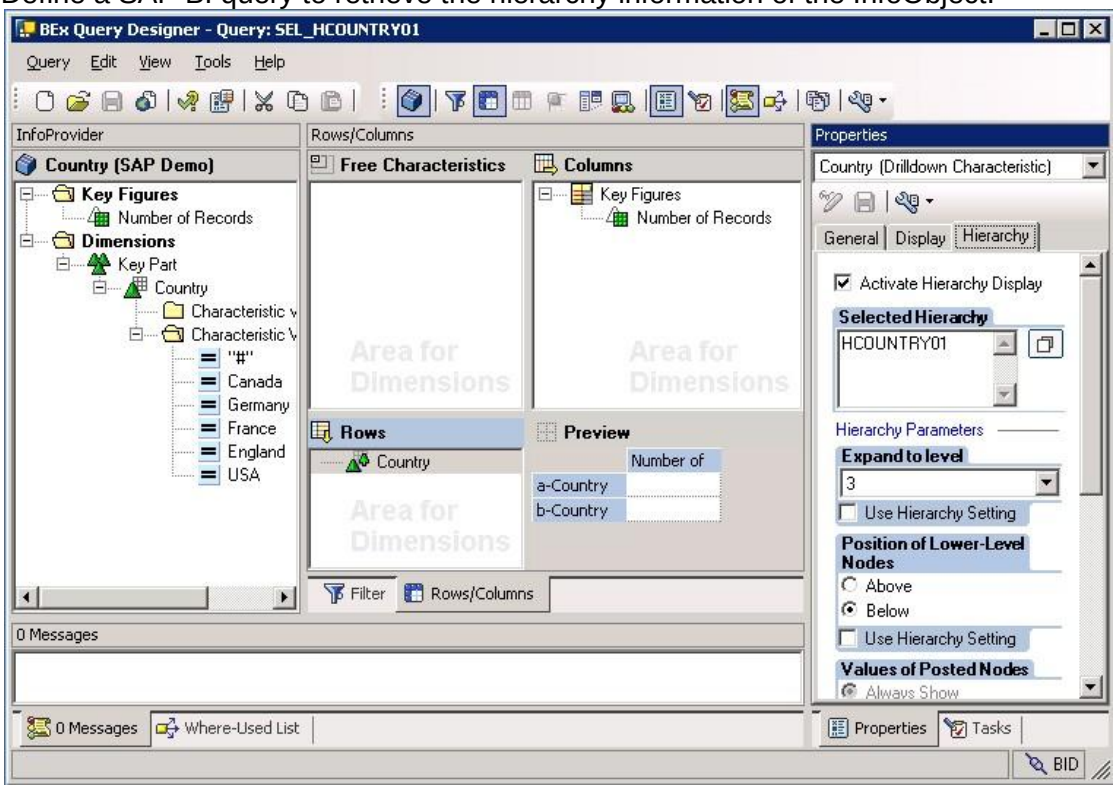


Figure 8 Definition of SAP BI Query

After the query was saved in SAP Query Designer it can be started via transaction RSRT or RSRT2 in the SAP GUI.

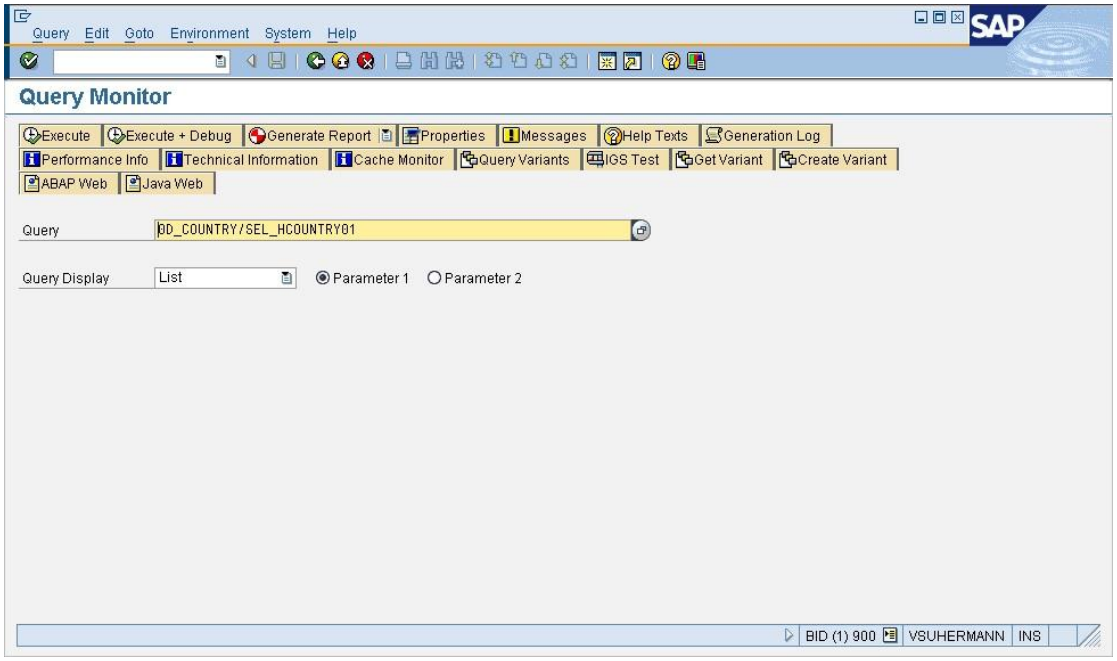


Figure 9 SAP Query Monitor

The output of the query shows all the important data we need. The following screenshots will shed some light on the different options how to call this query from extern.

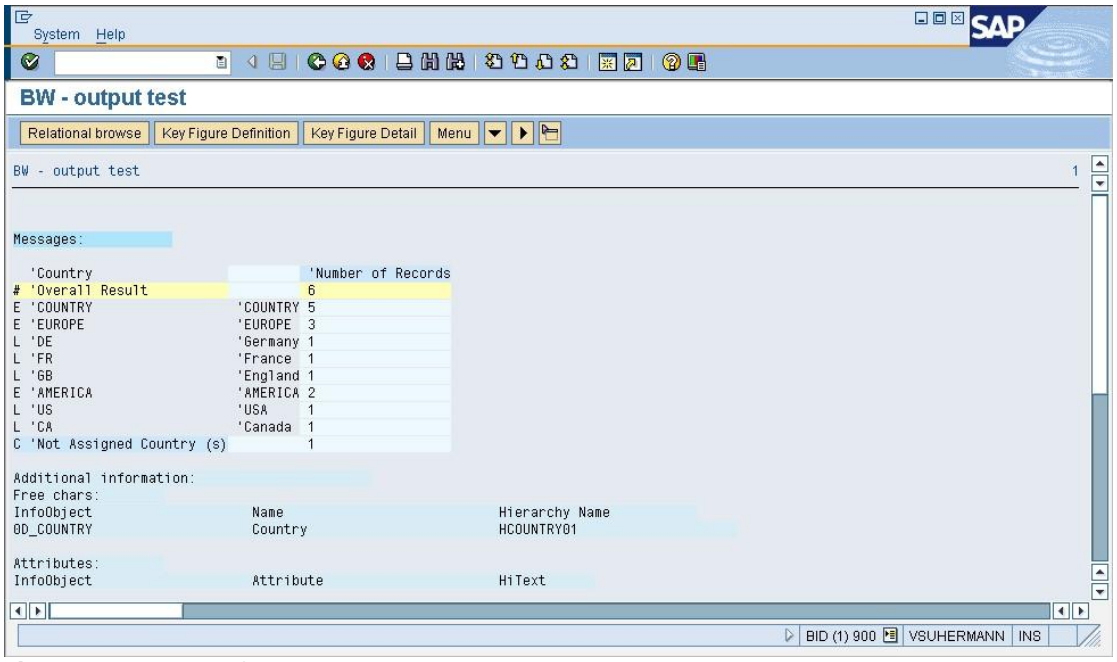


Figure 10 Output of Query

Look for the SAP ABAP report to unload hierarchies on the SDN.

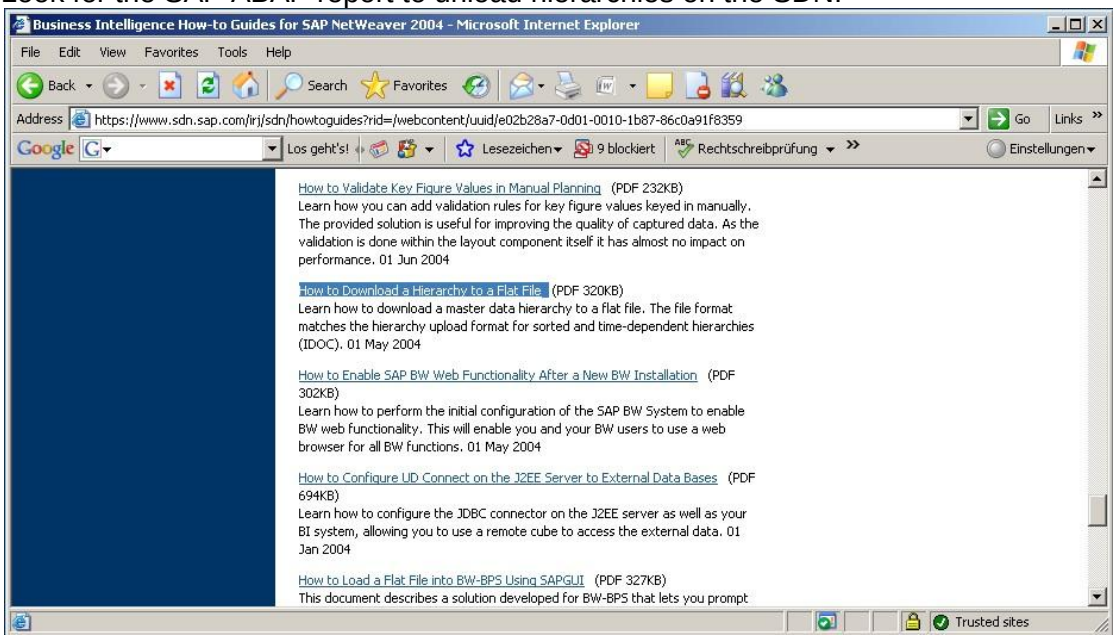


Figure 11 SDN.SAP.com Download Area

Install the SAP ABAP report and run it according to the how-to-guide.

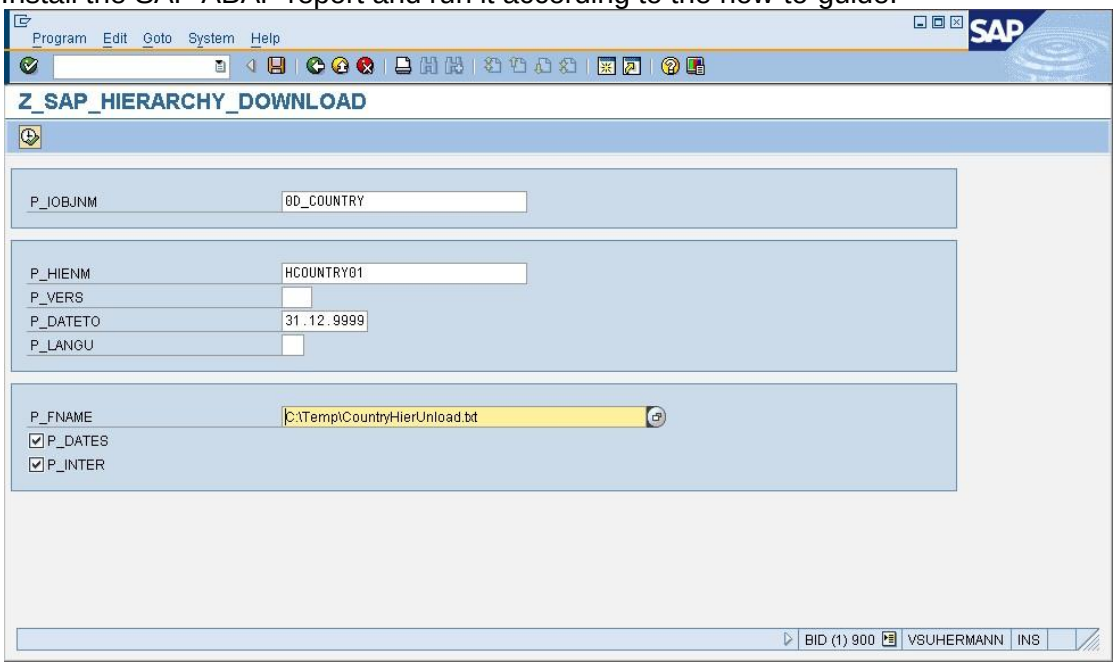


Figure 12 ABAP Report Z_SAP_HIERARCHY_DOWNLOAD

The output of the SAP ABAP report is basically the same as the load file of the SAP Demo Cube. It has fixed length though and the header row with column names is not there.

Figure 13 Output of ABAP Report

Use transaction mdxtest to create the MDX query for retrieving the hierarchy.

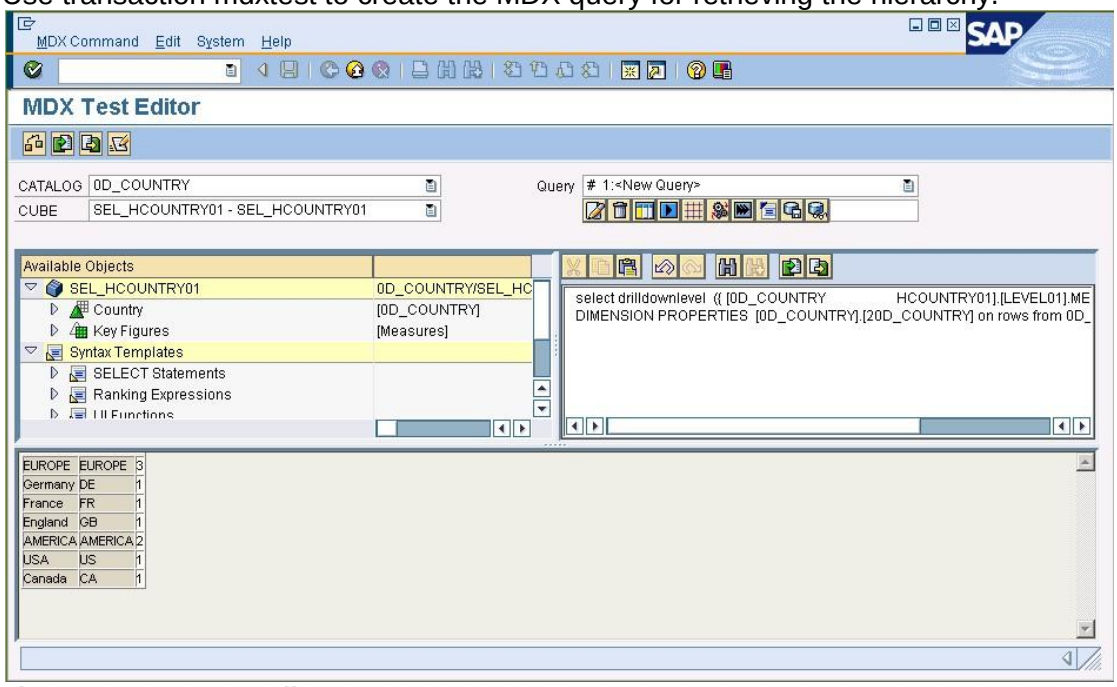


Figure 14 MDX Test Editor

Using the SAP BI XMLA Web Service as a web reference it's possible to retrieve the hierarchy information from the SAP BI query defined before via a little C# program (see sample code at the end of the paper). Having full control over the individual columns it's easy to process the data. Key part of the program is the MDX query which should be designed in transaction mdxtest as shown in the screenshot above. More details about using the XMLA interface with a C# program can be found in a separate paper which only talks about master data unload – see reference section at the end.

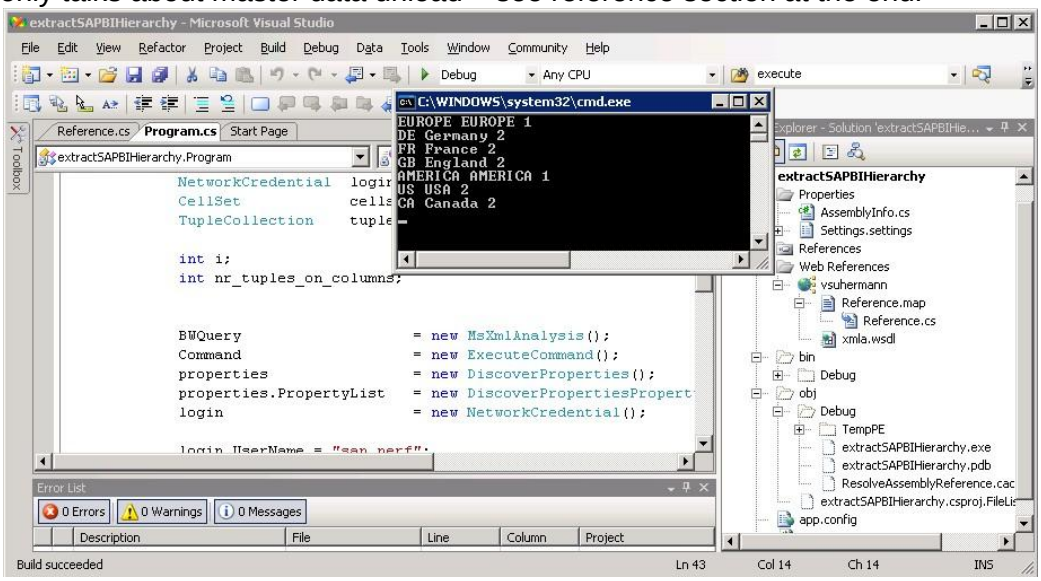


Figure 15 C# Program for SAP BI XMLA Web Service

On help.sap.com you can find a description of the SAP BI queryview web service and how to activate it.

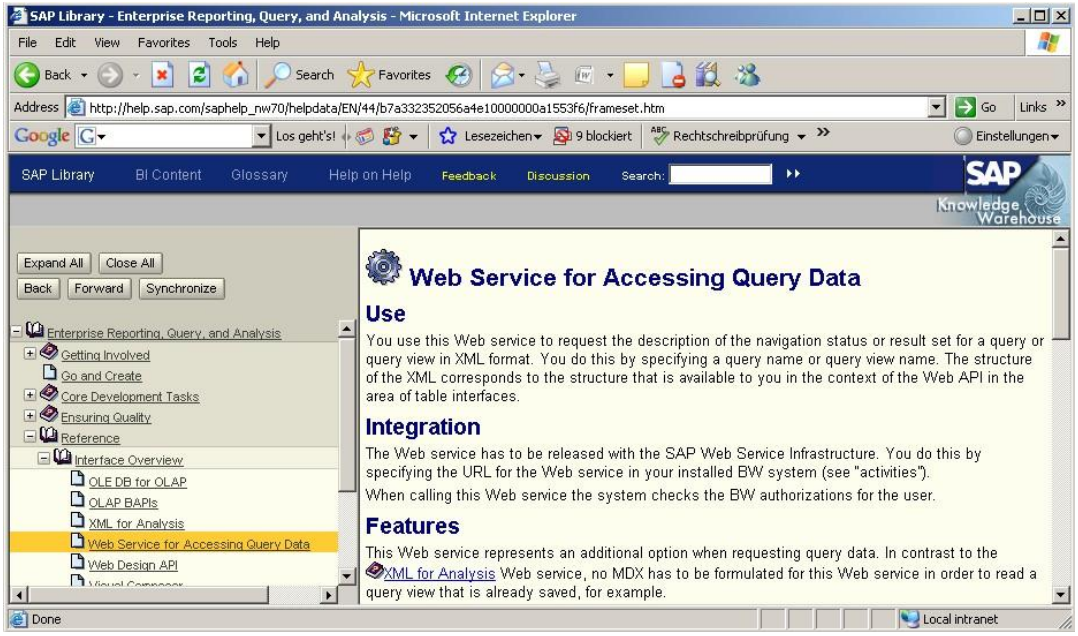


Figure 16 Help.SAP.com

Once the queryview web service is activated it should be tested using the “test service” function on the SAP BI side. This should open a browser window with the appropriate URL. This URL can be used inside Visual Studio to add a web reference in order to work with this service.

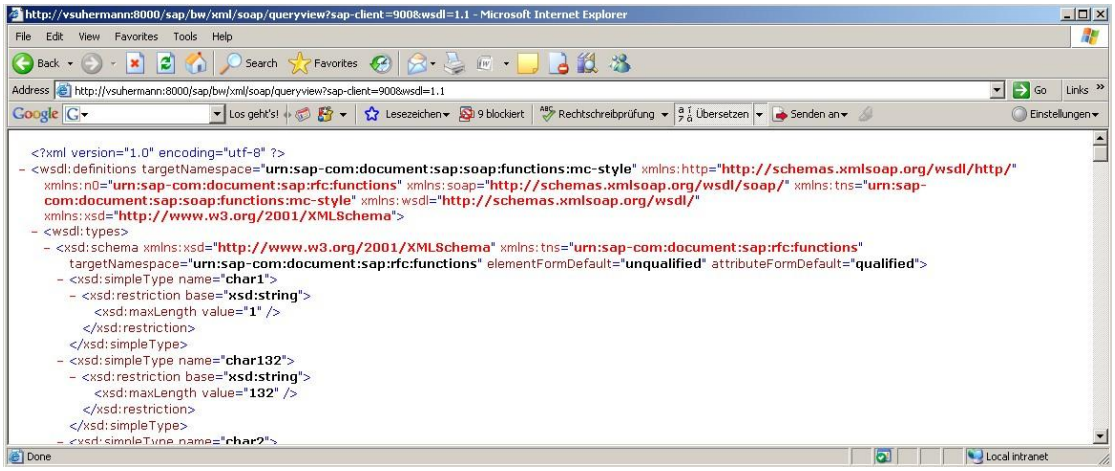


Figure 17 Result of Test Web Service

A little C# program (see sample code at the end of this paper) can then call the SAP BI query defined before and retrieve all the hierarchy data. In comparison to XMLA no MDX query language is needed.

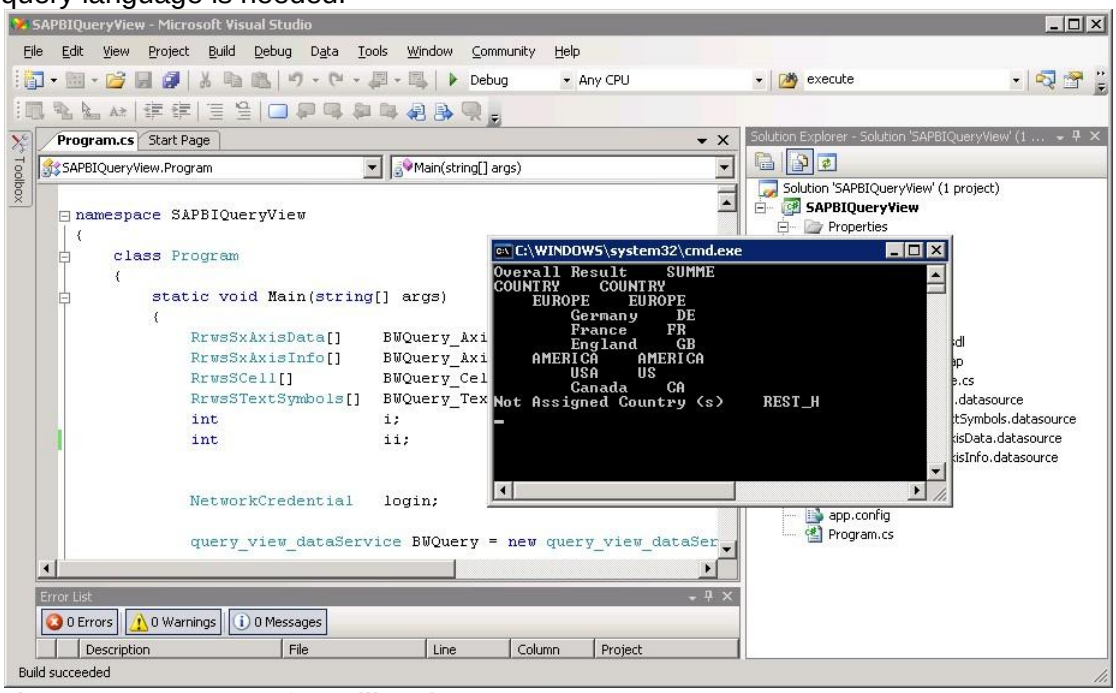


Figure 18 C# Program for calling the SAP BI Query

To get “ALL” data including the key information (DE, FR …) a little “trick” is necessary. You have to include the corresponding Member Property in the query defined in the SSRS query designer. Move the mouse on this column to see the property name (20D_COUNTRY) as it’s needed later on. For more information see the separate paper about SSRS connecting to SAP BI on www.microsoft.com/sap

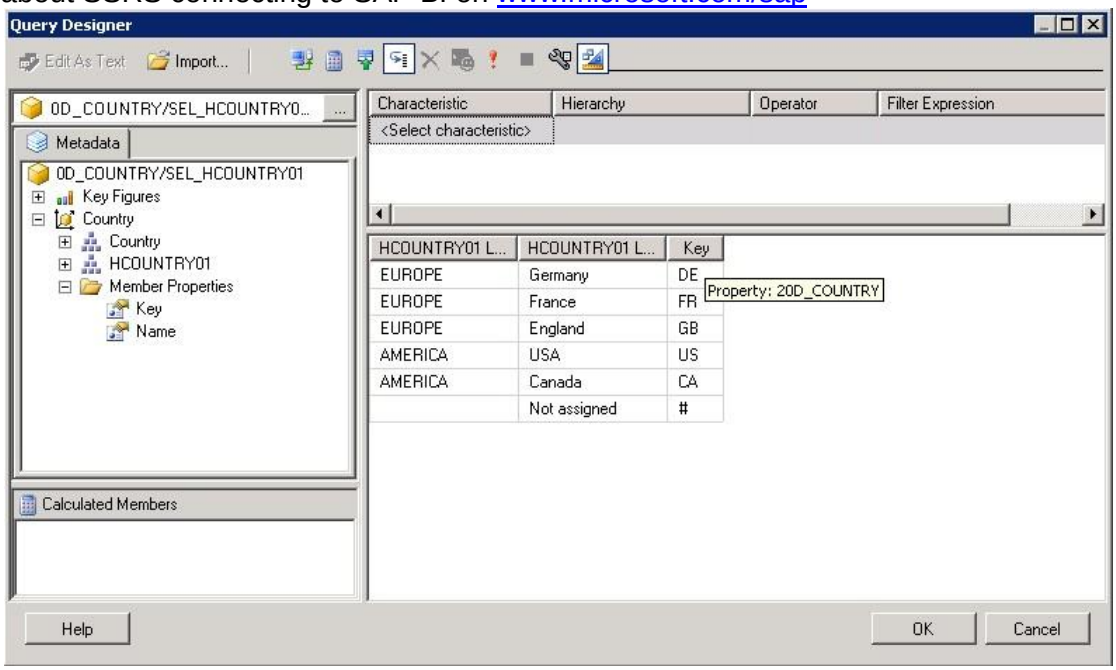


Figure 19 SSRS Query Designer

Looking at the MDX statement which SSRS generates you can see that the 20D_COUNTRY property is included.

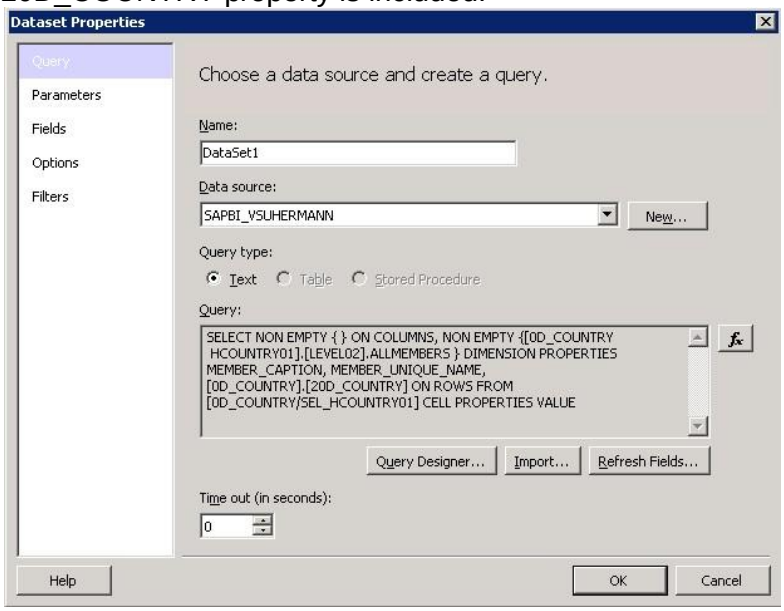


Figure 20 SSRS Query

What might be a little bit confusing the first time is the fact that the property mentioned above doesn't show up in the report layout by default even though it was selected in the SSRS query designer. An expression is necessary as shown on this screenshot to get the data for this property. That's the place where the name is needed which was shown in the SSRS query designer screenshot.

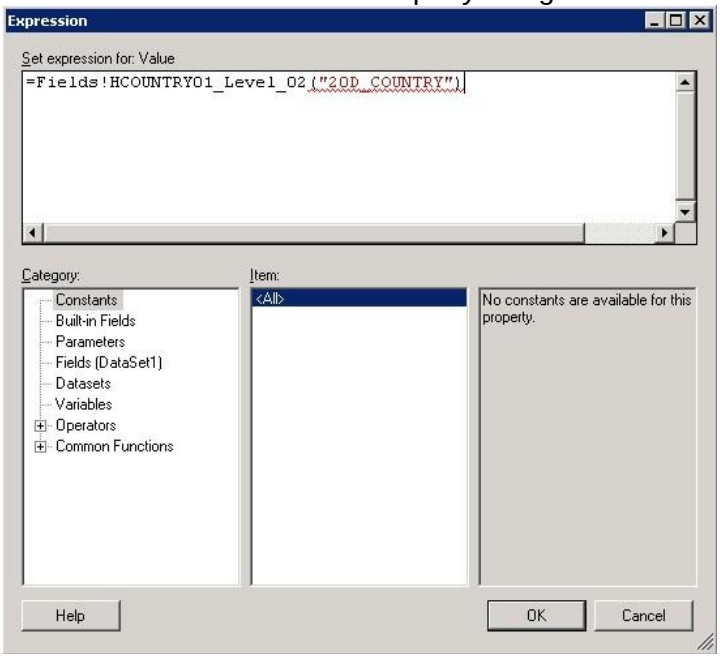


Figure 21 SSRS Expression

Finally we see all the hierarchy data in our SSRS report which we then save e.g. as a .csv file.

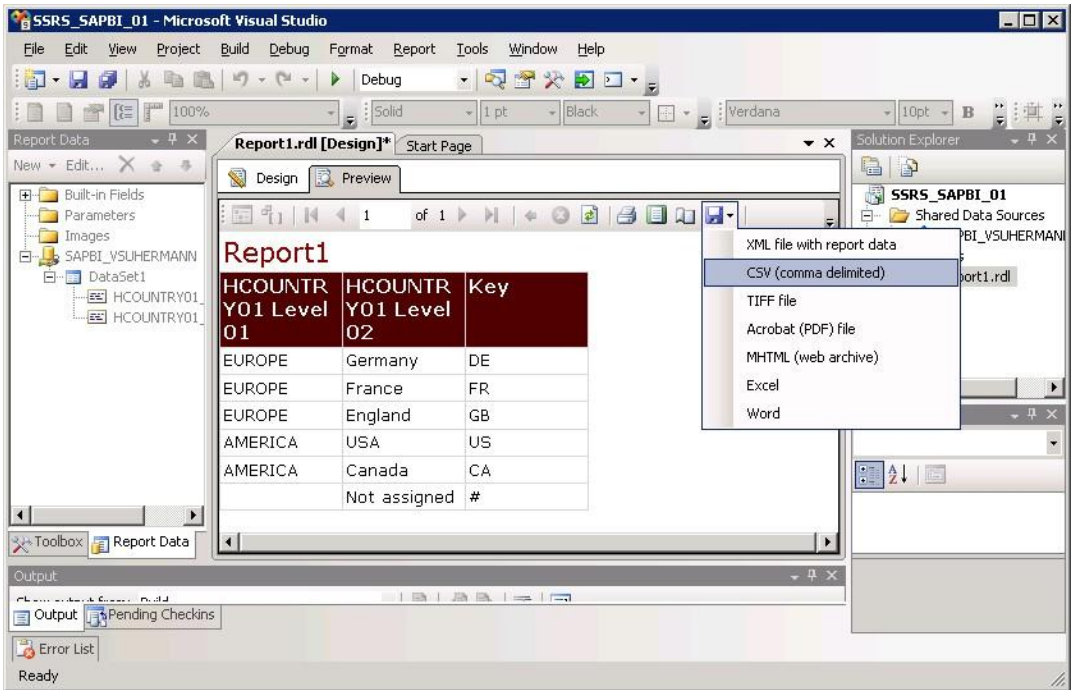


Figure 22 SSRS Report

After the data was saved any tool can be used for further processing.

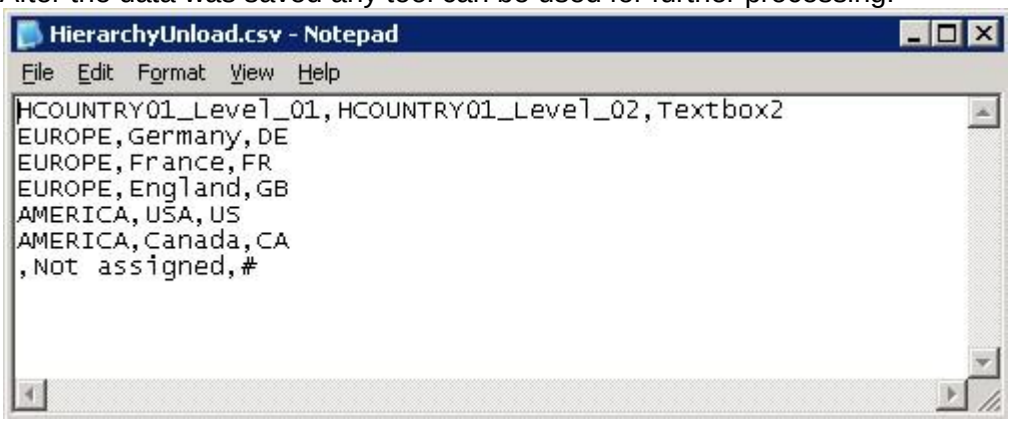


Figure 23 Data in Notepad

There is a paper from SIMBA which describes the new Excel 2007 support of SAP BI: http://www.simba.com/docs/Connecting-to-SAP-BW-with-Microsoft-Excel-2007-PivotTables-and-ODBO.pdf

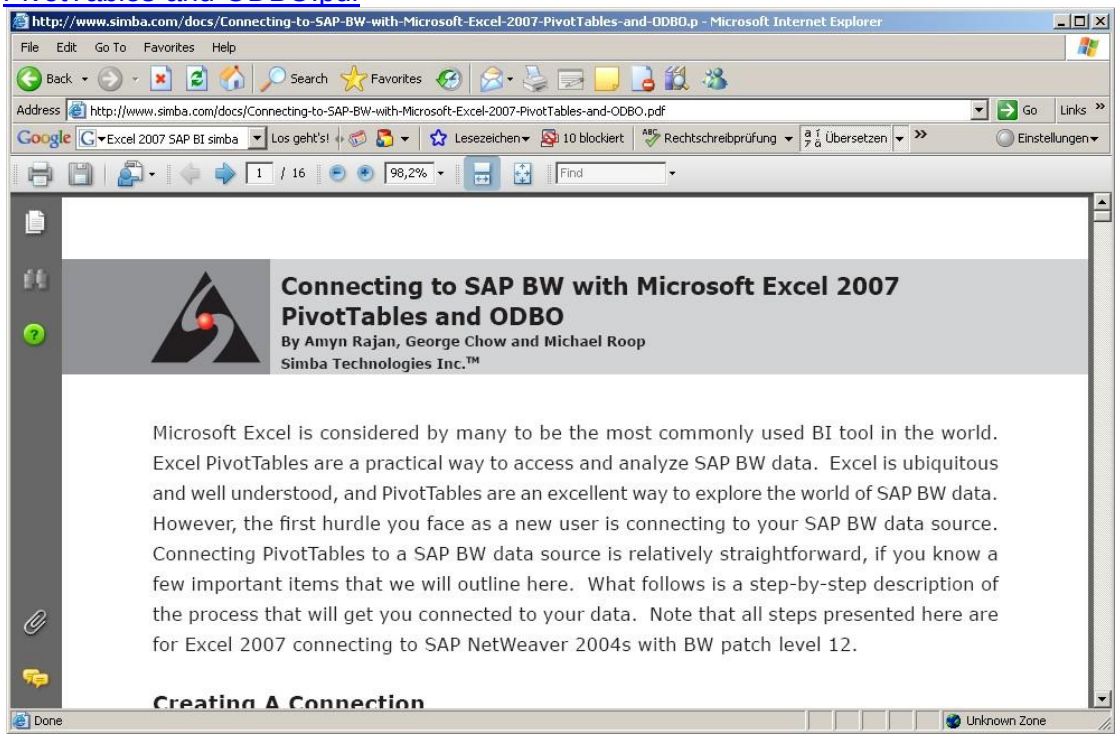


Figure 24 SIMBA Whitepaper

The Excel 2007 support also allows retrieving data from an existing SAP BI query. But it wasn't possible to get the country key data. This isn't a showstopper as it can be unloaded with the official Open Hub Service API but makes further processing of the data more difficult.

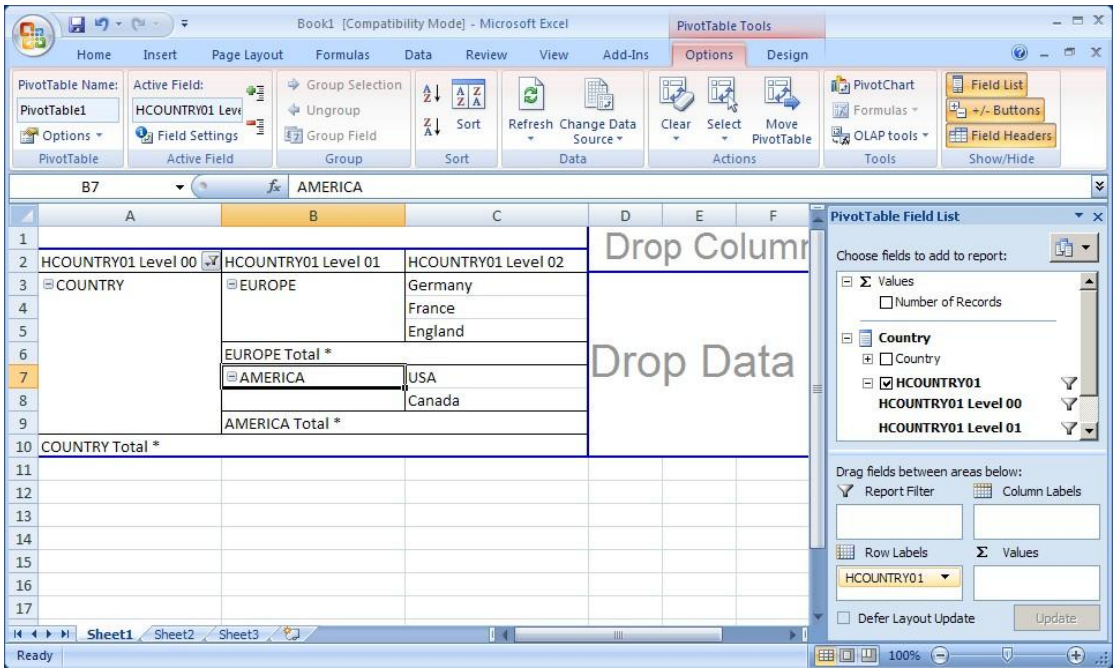


Figure 25 Excel 2007 Support

The "BEx" service inside SAP BI allows to run a SAP BI query via HTTP by entering the appropriate URL.

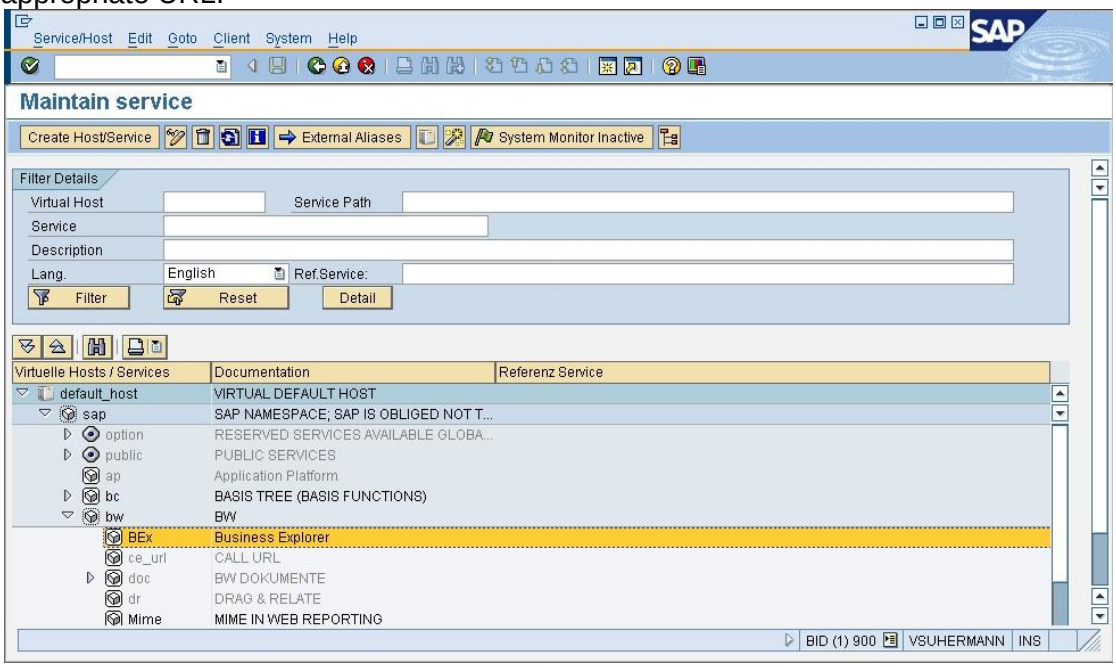


Figure 26 The "BEx" service inside SAP BI

To run the SAP query via HTTP just requires a simple browser and even returns the country keys (DE, FR …). The URL for this sample was //vsuhermann:8000/sap/bw/BEx?sap-language=EN&bsplanguage=EN&CMD=LDOC&INFOCUBE=0D_COUNTRY&QUERY=SEL_HCOUNTRY01.

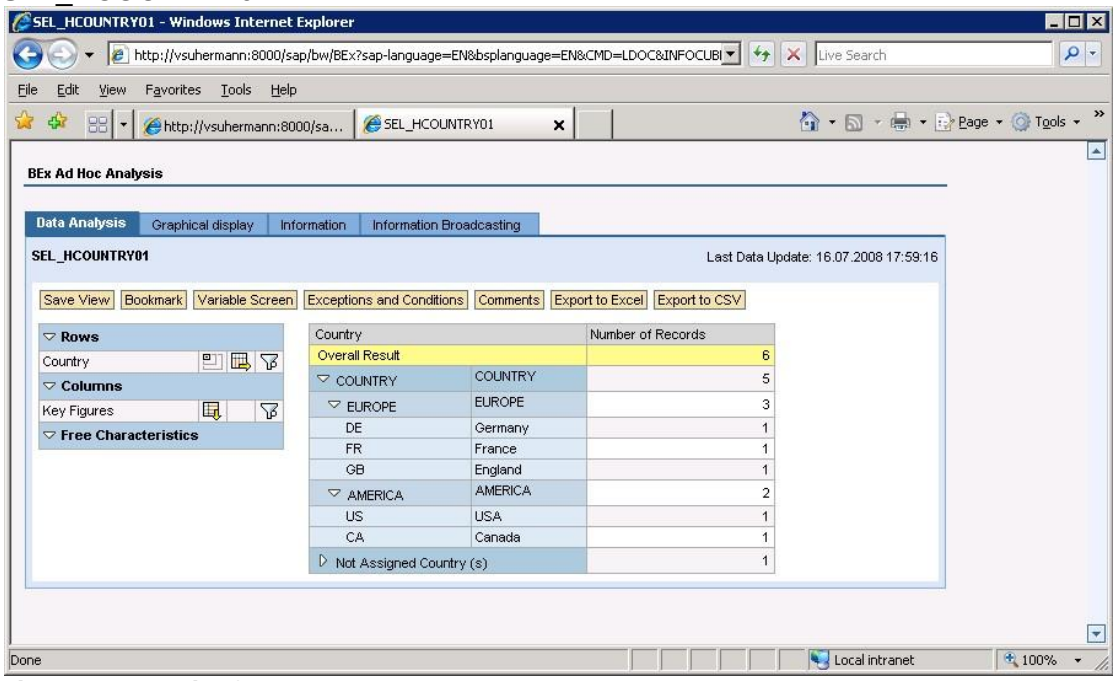


Figure 27 Result of SAP Query

Unfortunately the format after saving the HTTP output is not so optimal for further processing (e.g. '+' sign in front of a hierarchy node and so on).

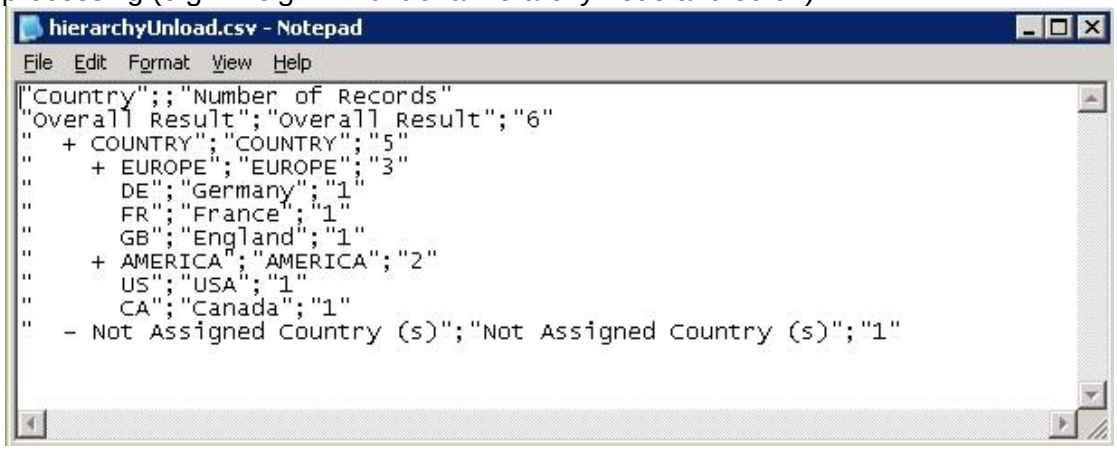


Figure 28 Output Hierarchy

The ABAP report for unloading hierarchy data makes use of it. It should be possible to call those functions from the outside world (e.g. via BizTalk Adapter Pack). But this use case is out of the scope of this paper and will be described in a separate paper.

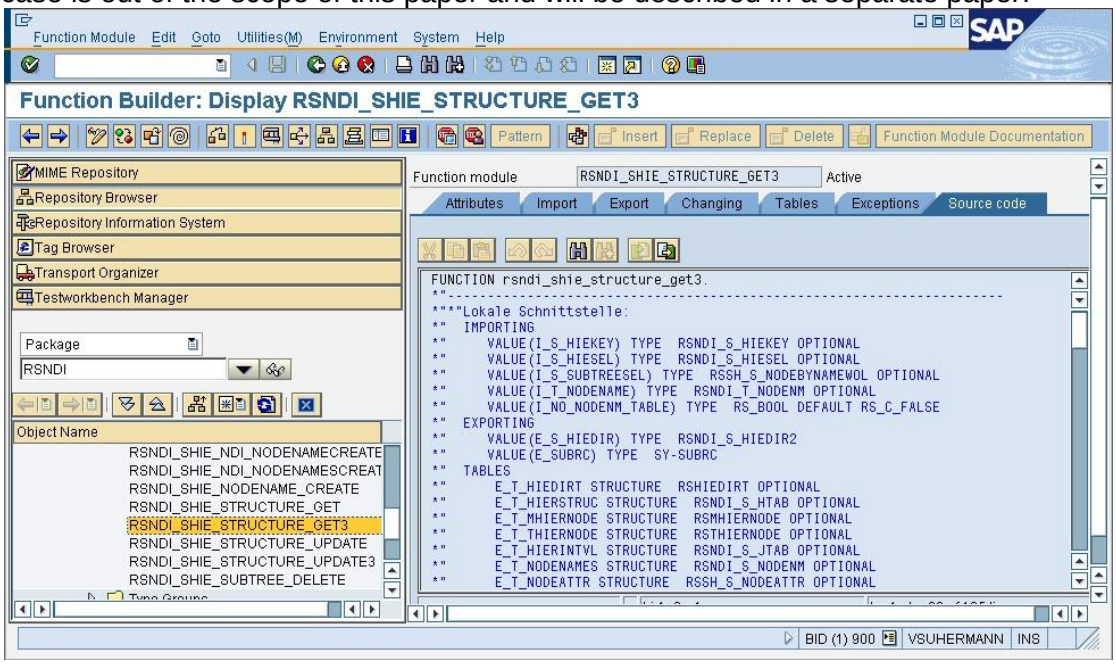


Figure 29 The RSNDI package includes a set of ABAP functions to work with hierarchies.

As one example for a potential 3rd-party tool the Xtract IS package from Theobald Software was used (www.theobald-software.com). It includes several SSIS plug-ins. One of them allows the extract of hierarchy data.

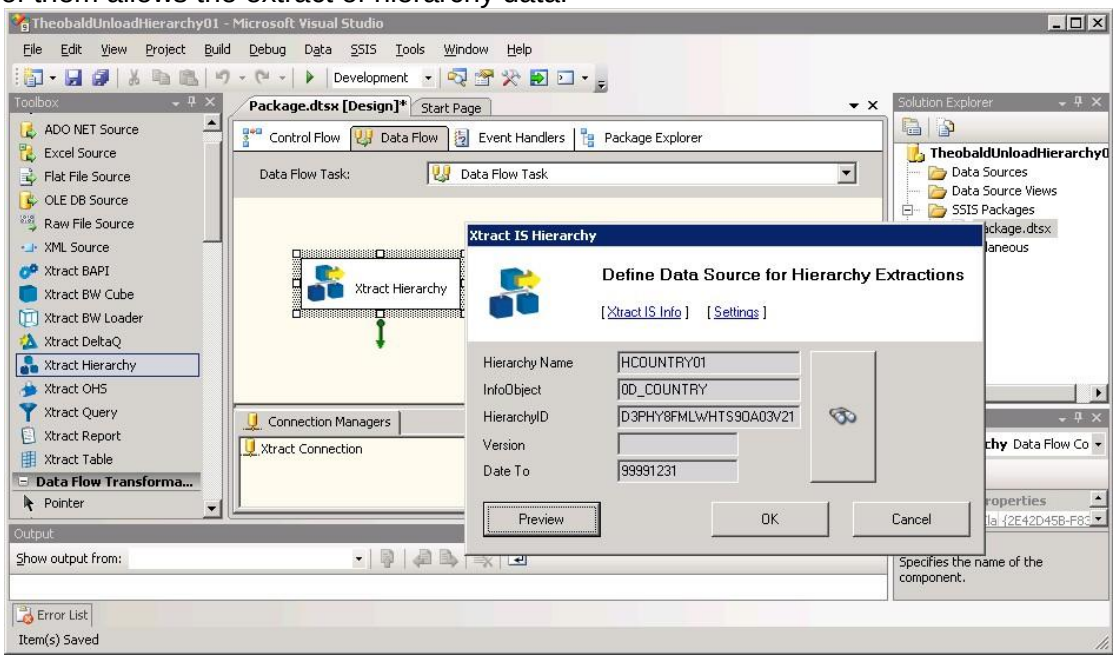
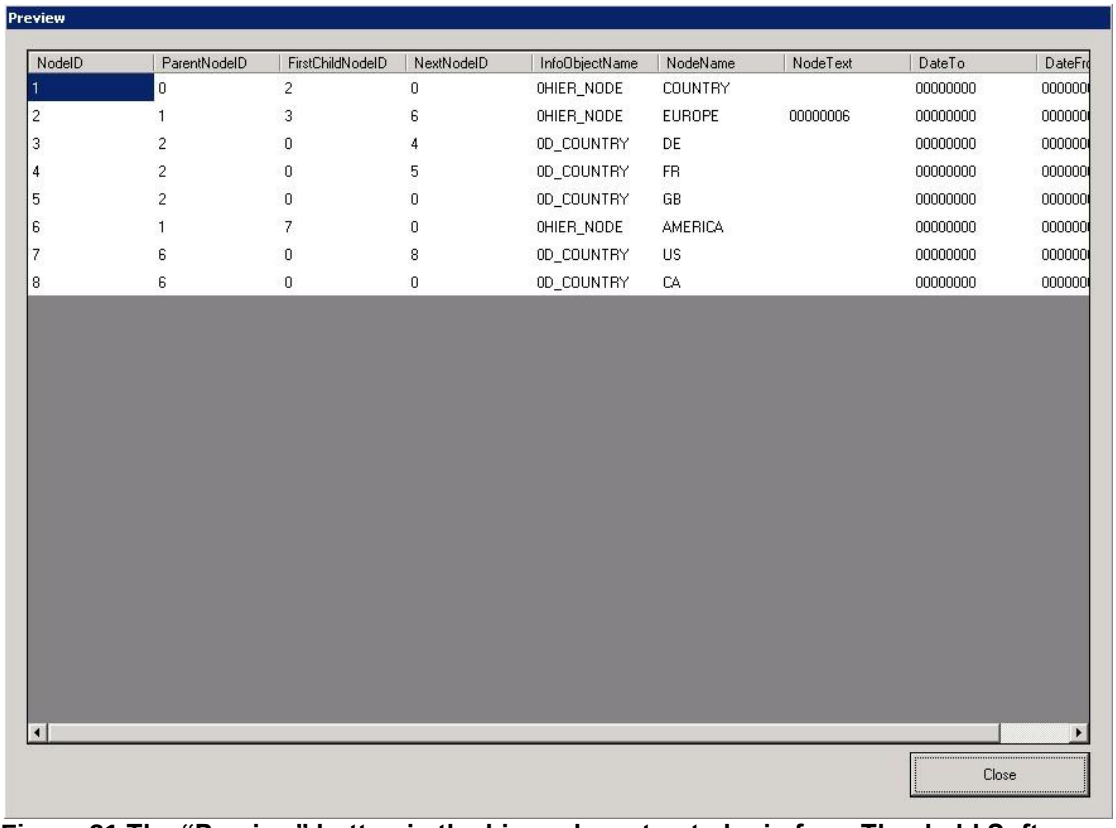


Figure 30 Xtract IS from Theobald Software

The “Preview” button shows that it retrieves all the details of the hierarchy correctly. A RFC trace on the SAP side proved that they also call a function of the RSNDI package mentioned before.



NodeID	ParentNodeID	FirstChildNodeID	NextNodeID	InfoObjectName	NodeName	NodeText	DateTo	DateFr
1	0	2	0	0HIER_NODE	COUNTRY		00000000	0000000
2	1	3	6	0HIER_NODE	EUROPE	00000006	00000000	0000000
3	2	0	4	0D_COUNTRY	DE		00000000	0000000
4	2	0	5	0D_COUNTRY	FR		00000000	0000000
5	2	0	0	0D_COUNTRY	GB		00000000	0000000
6	1	7	0	0HIER_NODE	AMERICA		00000000	0000000
7	6	0	8	0D_COUNTRY	US		00000000	0000000
8	6	0	0	0D_COUNTRY	CA		00000000	0000000

Figure 31 The “Preview” button in the hierarchy extract plugin from Theobald Software

Conclusion

As shown in this paper there are several ways to extract hierarchy data out of SAP BI. Nevertheless from a SAP support perspective the best solution would be to use the ABAP report from the SAP Service Marketplace.

Sample code C# console application

Unload hierarchy via XMLA

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Data;
using System.Xml;
using System.Net;
using Microsoft.AnalysisServices.AdomdClient;
using extractSAPBIHierarchy.vsuhermann;

namespace extractSAPBIHierarchy
{
    class Program
    {
        static void Main(string[] args)
        {
            MsXmlAnalysis          BWQuery;
            ExecuteCommand         Command;
            XmlElement             result;
            XmlNodeReader          convert_result;
            DiscoverProperties     properties;
            NetworkCredential      login;
            CellSet                cellset_result;
            TupleCollection        tuples_on_columns;

            int i;
            int nr_tuples_on_columns;

            BWQuery                    = new MsXmlAnalysis();
            Command                    = new ExecuteCommand();
            properties                 = new DiscoverProperties();
            properties.PropertyList    = new DiscoverPropertiesPropertyList();
            login                      = new NetworkCredential();
            login.UserName             = "xyz";
            login.Password             = "abc";


            BWQuery.Credentials        = login;
            BWQuery.PreAuthenticate  = true;
            properties.PropertyList.Content            = "SchemaData";
            properties.PropertyList.Format             = "Multidimensional";
            properties.PropertyList.DataSourceInfo = "default";

            Command.Statement = "select drilldownlevel({ [0D_COUNTRY " +
            “HCOUNTRY01].[LEVEL01].MEMBERS }) DIMENSION PROPERTIES " +
            “[0D_COUNTRY].[20D_COUNTRY] on rows from 0D_COUNTRY" +
            “/SEL_HCOUNTRY01";
```

```
            result = BWQuery.Execute(Command,properties);
            convert_result = new XmlNodeReader(result);
            cellset_result = CellSet.LoadXml(convert_result);
            nr_tuples_on_columns = cellset_result.Axes[0].Set.Tuples.Count;
            tuples_on_columns    = cellset_result.Axes[0].Set.Tuples;
            for (i = 0; i <nr_tuples_on_columns; i++)
            {
               if( tuples_on_columns[i].Members[0].MemberProperties.Count > 0 )
                     Console.Write
                     (tuples_on_columns[i].Members[0].MemberProperties[0].Value);
               Console.Write(" ");
               Console.Write(tuples_on_columns[i].Members[0].Caption);
               Console.Write(" ");
               Console.WriteLine(tuples_on_columns[i].Members[0].LevelDepth);
            }
            Console.ReadLine();
         }
      }
   }
```

Unload hierarchy via queryview web service

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Net;
using SAPBIQueryView.vsuhermann;

namespace SAPBIQueryView
{
   class Program
   {
      static void Main(string[] args)
      {
         RrwsSxAxisData[]      BWQuery_AxisData;
         RrwsSxAxisInfo[]      BWQuery_AxisInfo;
         RrwsSCell[]           BWQuery_Cell;
         RrwsSTextSymbols[]    BWQuery_TextSymbols;
         int                   i;
         int                   ii;

         NetworkCredential   login;

         query_view_dataService BWQuery = new query_view_dataService();

         login                 = new NetworkCredential();
         login.UserName = "xyz";
         login.Password =  "abc";

         BWQuery.Credentials = login;
         BWQuery.PreAuthenticate = true;
```

```
            BWQuery_AxisData = BWQuery.GetQueryViewData
           ("0D_COUNTRY",null,"SEL_HCOUNTRY01",null,
           out BWQuery_AxisInfo,
           out BWQuery_Cell,
           out BWQuery_TextSymbols );
           for (i = 0; i < BWQuery_AxisData.Length; i++)
           {
               if( i>0 )
                  for (ii = 0; ii < BWQuery_AxisData[i].Set.Length; ii++)
                        {
                        if (BWQuery_AxisData[i].Set[ii].Tlevel == "01")
                           { Console.Write(""); }
                        else if (BWQuery_AxisData[i].Set[ii].Tlevel == "02")
                           { Console.Write("    "); }
                        else if (BWQuery_AxisData[i].Set[ii].Tlevel == "03")
                           { Console.Write("        "); }
                        Console.Write(BWQuery_AxisData[i].Set[ii].Caption.ToString());
                        Console.WriteLine( "    " +
                            BWQuery_AxisData[i].Set[ii].Chavl.ToString());
                        }
                     }
           Console.ReadLine();
           }
       }
}
```

References

SAP Help Portal
http://help.sap.com
Microsoft SAP Customer Information Center
http://www.microsoft.com/sap
Microsoft SQL Server BI