NETConductor

PDF Guideline

Rue de Net Vesturgötu 3 101 Reykjavík
T: +354 414 5050 ruedenet@ruedenet.is www.ruedenet.is

rue de net

Index

Documentation:

20/05/2009 Alfred B. Thordarson
Read over and overhaul headings and text

19/05/2009 Richard Ottó O'Brien
Created document

NETConductor – PDF Guideline

This document explains how to use the NETConductor to retrieve a report from within the Microsoft NAV environment using web services. The NETConductor can provide reports formatted as html, xml or PDF from any report within NAV. This document explains in detail how PDF documents can be retrieved by sending parameters that are used to filter the results.

We are assuming that the NETConductor has already been installed and that the readers of this document are familiar with how the NETConductor works and are able to do basic NAV customizations. In addition, for the final part of this document we also assume you are to some degree familiar with Visual Studio 2005/8 development.

Prerequisites

To be able to retrieve PDF documents through a web service a PDF creator must be installed on the server running the NETConductor data tier. In our example here we have chosen to use PDFCreator.

NAV basically creates a PDF document before sending it. Because of this the PDFCreator needs to be setup as the default printer on the server where the data tier is hosted.

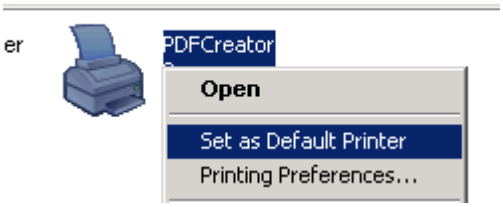

Figure 1 – Setting the PDFCreator as the default printer

The PDFCreator also needs to be setup so that it is set to auto save the documents.

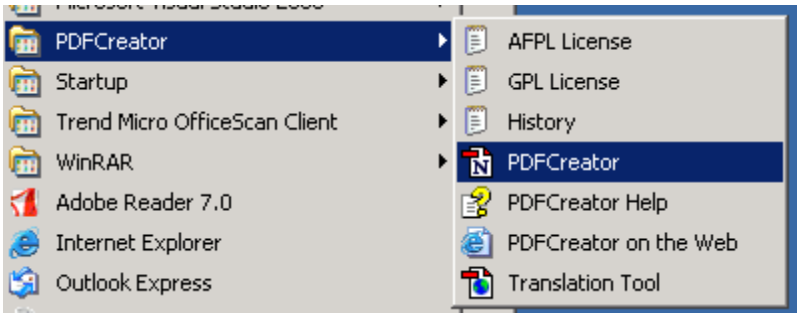

Figure 2 – Opening the PDFCreator

This is done by opening up the PDFCreator and choosing “Printer” and then “Options” (Ctrl+O).

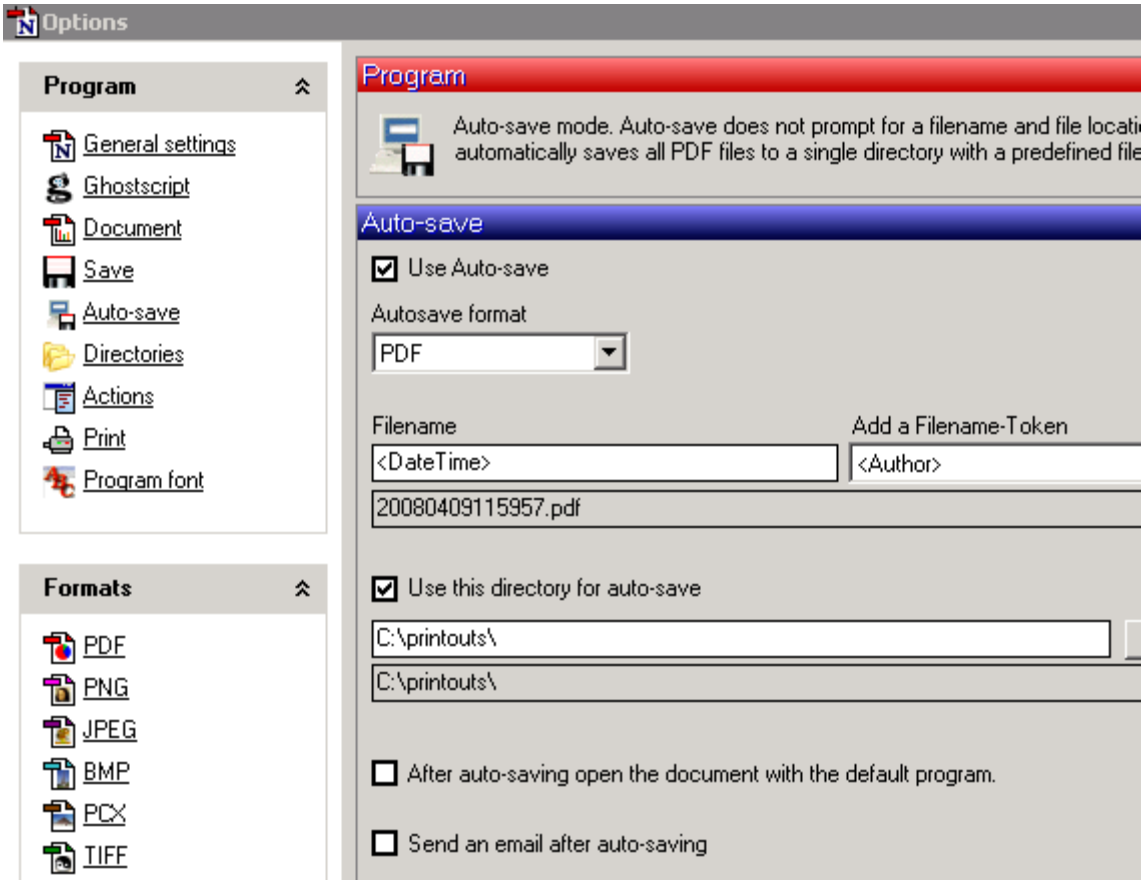

Figure 3 – Setting the PDFCreator “Auto-save” property

The “Use Auto-save” option is checked and the directory where the files should be saved is specified. When this has been done it is possible to give web access to NAV reports and to retrieve PDF documents using the NETConductor.

Invoking the service

To retrieve a PDF document we use the RunReport operation that is one of the basic NETConductor web services. In our example here we will be manually invoking this operation to demonstrate how this is done.

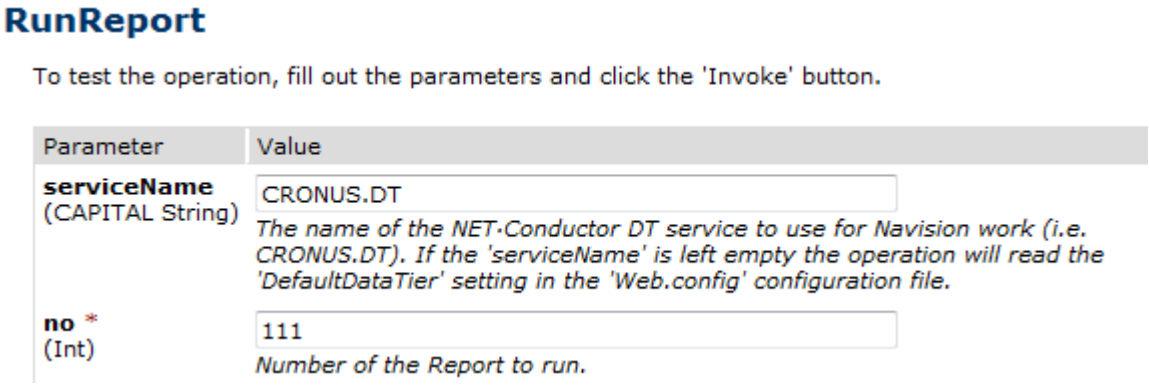

Figure 4 – Specifying the serviceName and the report to run

In figure 4 we can see that the “serviceName” and the number of the report to run need to be specified. Any report within NAV can be retrieved without doing any customization.

However if a filter is to be set for the report some minor adjustments need to be made within NAV. We can set the filter as in the example below where we only want to get entries that are within the year 2007. Then we need to adjust the report so that the correct filtering is done (see customizing NAV to enable filtering)

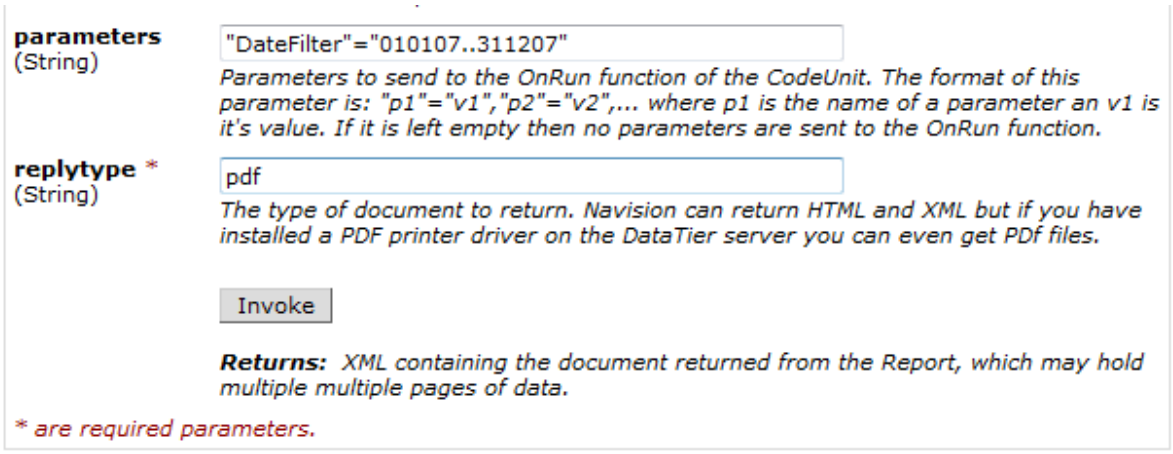

Figure 5 – Specifying the ServiceName and the report to run

The “replytype” should of course be PDF since that is the type of report we want to retrieve.

Customizing NAV to enable filtering

To enable the filtering we need to go to the Object Designer and choose the report that will be receiving the filter. In our example here we are using report “111 Customer – Top 10 List”.

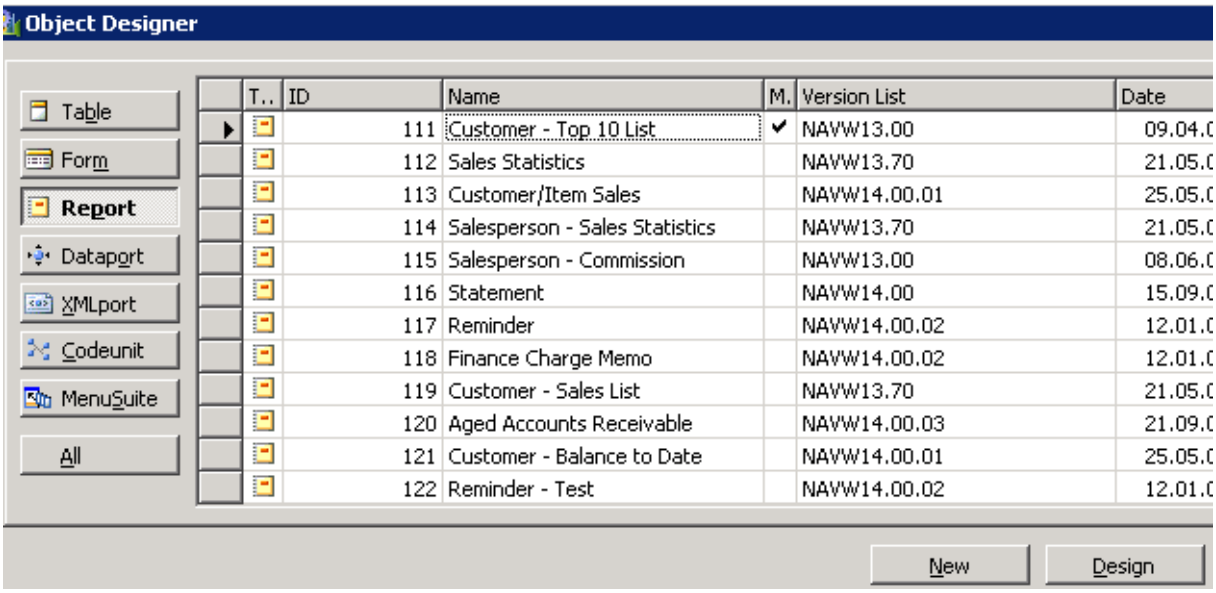

Figure 6 – Selecting to design the report that will be receiving the filters

Within the report we choose the OnPreDataItem() trigger for the record we want to filter on and within that trigger we open up the local variables form.

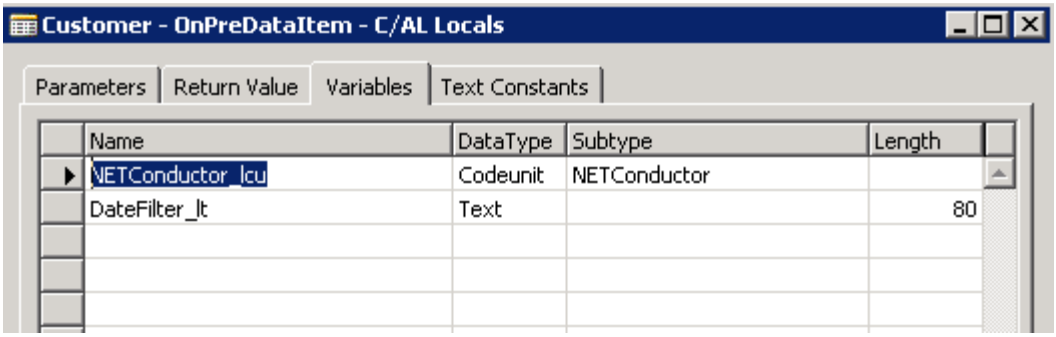

Figure 7 – Creating two new local variables

We need to create two new local variables that are needed to retrieve the parameters from the NETConductor. The variables are created as shown in figure 7.

Be careful to not create the "NETConductor" CodeUnit variable as a global variable.

Since we now have the variables all we need to do is to add the code that retrieves the parameter(s) and set the filter.

Customer - C/AL Editor

```
Customer - OnPreDataItem()
IF NETConductor_lcu.IsExecuting THEN BEGIN
  IF NETConductor_lcu.IsParameterProvided('DateFilter') THEN BEGIN
    DateFilter_lt := NETConductor_lcu.GetParameter('DateFilter');
    Customer.SETFILTER("Date Filter",DateFilter_lt);
  END;
END;
```

Figure 8 – The code responsible for retrieving a certain parameter and setting a filter

By adding the code shown in figure 8 the web service will return a PDF document with a date filter.

Obviously by using the same method as described above any kind of filtering can be applied to any report within NAV.

Reply of the service

When we have setup the PDFCreator and customized the report to enable filtering we are ready to invoke the service.

Figure 9 – The Invoke button of the RunReport operation of the NETConductor Basics web service

Using the parameters shown before, when we push the Invoke button, the NETConductor will run report number 111 with the parameter "DateFilter"="01.01.07..31.12.07". Because the replytype parameter is "PDF" the NETConductor will send the report result to the default printer, the PDFCreator, which will write a file to the "c:\printouts" directory.

```
<?xml version="1.0" encoding="utf-8" ?>
- <ReplySet name="6334337242874270872">
  - <Reply type="Report" name="111" value="Customer - Top 10 List">
    - <Reply type="File" name="Customer - Top 10 List" mimetype="application/pdf">
        <FileContent>JVBERi0xLjQKJcfsj6IKNSAwIG9iago8PC9MZW5ndGggNiAwIFIvRmlsdGVyIC9Gb
      </Reply>
    </Reply>
  </ReplySet>
```

Figure 10 – PDF reply from the NETConductor, wrapped in an XML envelope

The NETConductor will pick this file up when running the report is finished. The content of the file will be sent back to the caller wrapped in an XML envelope as shown in figure 10. In the reply the "FileContent" are the Base64 encoded bytes of the PDF document.

Getting the reply as a PDF document through a web application

Often, we would like to be able to provide the PDF results of NAV reports, through a web application, straight back to the browser of the caller, as a PDF document. For example, we would like to be able to open up a browser, punch in a URL in the form of:

http://localhost/MyWebApplication/CustomerTop10List.aspx?year=2007

and get back a PDF document straight into our browser. In the rest of this document we will show you how to do this. However, we assume that you are familiar with Visual Studio 2005/8 and know how to create, develop, configure and deploy a web application.

Making the call to NAV, using C#

The first thing we have to do is consume the NETConductor Basics web service and then we can call the RunReport operation using the shown code:

```
NC.Basics basics = new NC.Basics();

string dateTimeFilter = "\"DateFilter\"=\"";
dateTimeFilter += "01.01." + Request["year"] + "..31.12." + Request["year"];
dateTimeFilter += "\"";

XmlNode reply = basics.RunReport("CRONUS.DT", 111, dateTimeFilter, "pdf");
```

Figure 11 – The code responsible for calling the RunReport operation with the proper parameters

Converting NETConductor reply back to PDF

The code snippet above will return an XML envelope, as shown in figure 10 into the local variable “reply”. The next thing is to convert the “FileContent” into a byte array, which can be done like this:

```
string pdfText = reply.SelectSingleNode("//FileContent").InnerText;
byte[] pdfBytes = Convert.FromBase64String(pdfText);
```

Figure 12 – The code responsible for converting the Base64 string from the NETConductor into a byte array

Finally, all we have to do is write the PDF bytes to the response stream of the page request:

```
Response.ContentType = "application/pdf";
Response.AddHeader("Content-disposition", "attachment; filename=CustomerTop10List" + Reques
Response.OutputStream.Write(pdfBytes, 0, pdfBytes.Length);
```

Figure 13 – The code responsible for content type and writing out the PDF bytes

Testing out CustomerTop10List.aspx page

When we have written the CustomerTop10List.aspx page we can access it using any browser. We provide the “year” parameter at the end of the URL as expected.

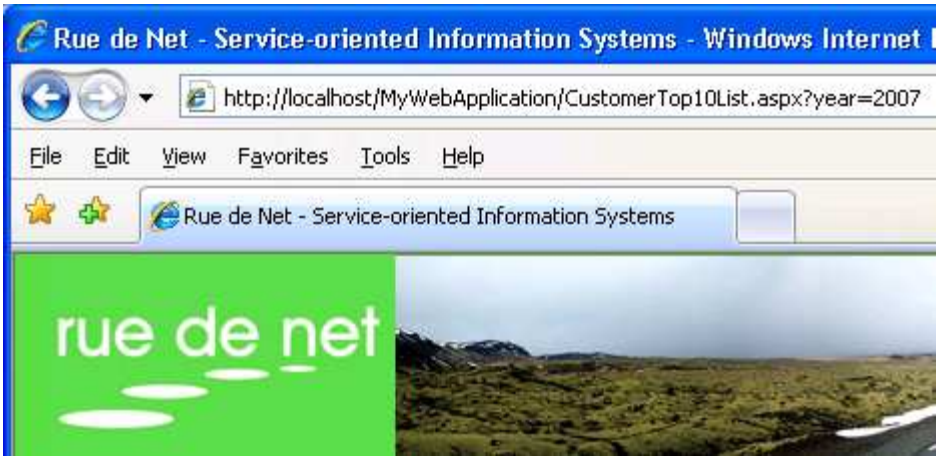

Figure 14 – Punching in our URL

This URL will result in the web server running our code and it returning back a byte array with the content type of “application/PDF”.

In my browser, this content type results in the open/save dialog popping up.

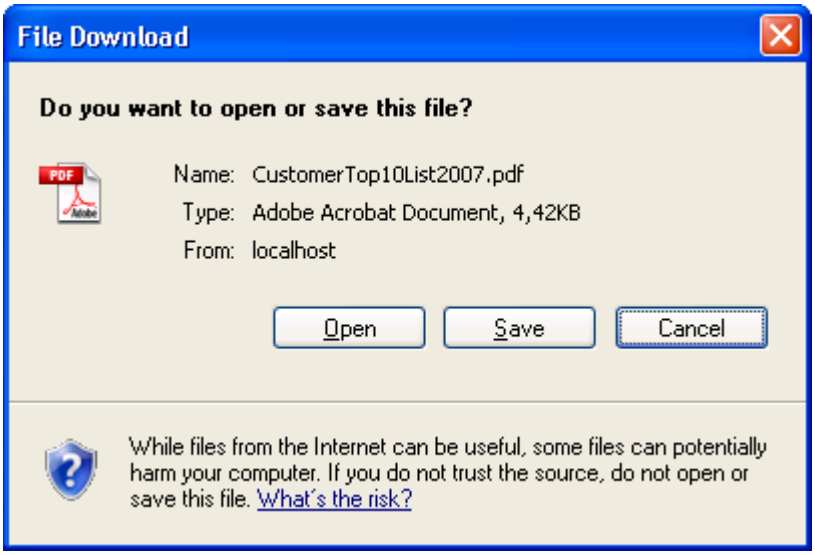

Figure 15 – The "application/PDF" content type resulting the open/save file showing up

When I have accepted to open up the PDF document it will open up in my Adobe PDF Reader 8.0:

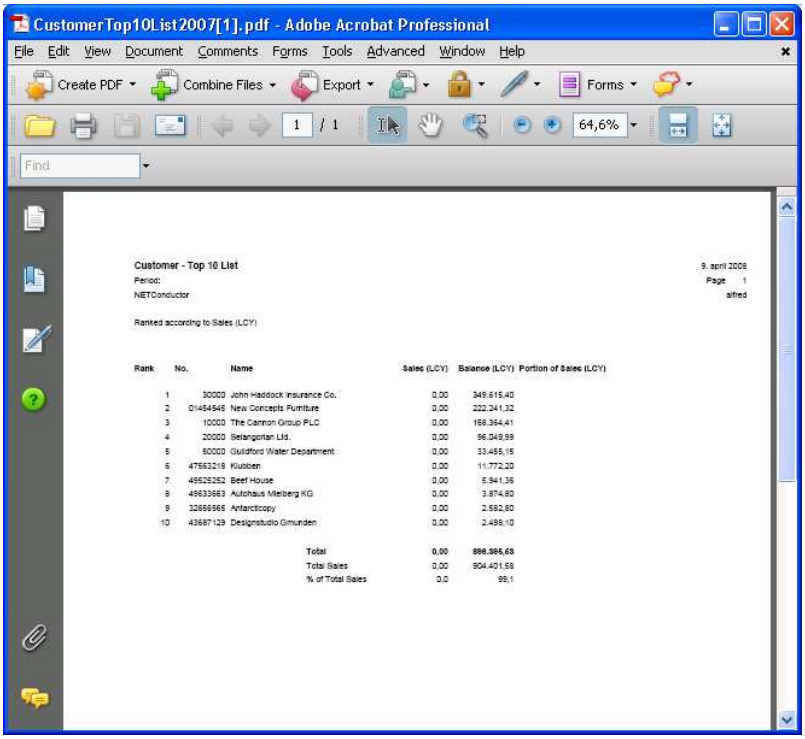

Figure 16 – The Top 10 List of 2007 in the CRONUS database

However, in the CRONUS database there are no sales in the year of 2007, let's try 2001.

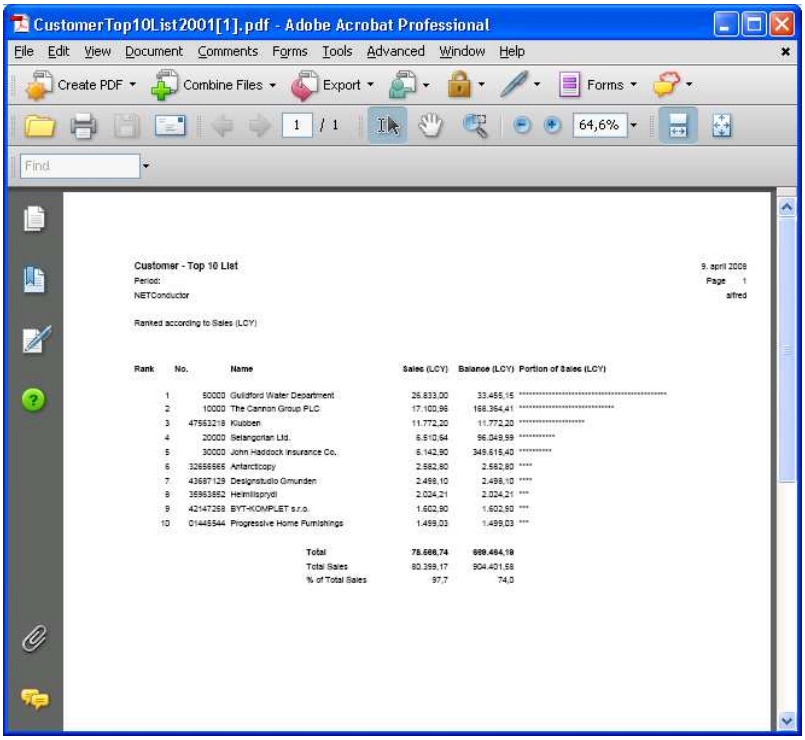

Figure 17 – The Top 10 List of 2001 in the CRONUS database

There you go!

rue de net
Vesturgata 3, 101 Reykjavik, Iceland
ruedenet@ruedenet.com
Tel: (+354) 414 5050
Fax: (+354) 414 5051