SAP NetWeaver
How-To Guide

How To Analyze Portal Activity with the Activity Data Collector

Applicable Releases:

SAP NetWeaver 7.0 >= SPS14
SAP NetWeaver '04 >= SPS21

IT Practice

User Productivity Enablement

IT Scenario

Running an Enterprise Portal

Version 1.0

December 2008

Document History

Document Version	Description
1.00	First official release of this guide

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
`Example text`	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1. Scenario

There are a couple of questions which are quite often raised whenever a SAP Enterprise Portal Environment is in place:"How many users are using the portal?", "Which is the most viewed page in the Portal?" and so on. This How-To Guide will give you an basic idea of how you can answer such questions with the help of the Activity Data Collector (ADC), and it will give you an example how you can determine the number of unique users per day.

2. Background Information

The Activity Data Collector is a tool for tracing activity in the portal, such as which users logged in and what iViews were viewed. Every server process of the SAP Web Application Server Java writes it's on trace file which contains portal requests and maintains details of each request, such as the logged-in user, processing time, and requested iView. For each request, a row is added to one or more data files.

This files can be used for further processing and analyzing for example by custom coding, Business Intelligence, Microsoft Excel or other 3rd party tools.

The Activity Data Collector is available starting with SAP NetWeaver '04 Enterprise Portal 6.0 SPS21 and SAP NetWeaver 7.0 Enterprise Portal 7.0 SPS14.

> **Tip**
>
> It is planned to offer a functionality to analyze the Activity Data Collector traces with the help of SAP Solution Manager Diagnostics very soon. Please stay tuned and watch out for further announcements about availability and features on SAP Service Marketplace and SDN.

3. Prerequisites

3.1 Software Requirements

Like mentioned above you need at least:

- SAP NetWeaver '04 Enterprise Portal 6.0 SPS21 (please see SAP Note 1178889)

or

- SAP NetWeaver 7.0 Enterprise Portal 7.0 SPS14

3.2 Required Documentation / Knowledge

You should be familiar with the administration of SAP Enterprise Portal and SAP Web Application Server Java to follow the instructions in this How-To Guide.

3.2.1 Related SAP Notes

SAP Note	Title
1178889	Activity Data Collector
1251041	Activity Data Collector stops generating files

3.2.2 Related Documentation

Source	Title
Help Portal	SAP NetWeaver 04 – Activity Data Collector
Help Portal	SAP NetWeaver 7.0 – Activity Data Collector
Help Portal	SAP NetWeaver 7.0 – Knowledge Management Activity Reporting
SAP Community Network	Blog: Recording Access to Documents in KM
SAP Community Network	Blog: How to use the Activity Data Collector

4. Step-by-Step Procedure

This step-by-step procedure will show how you can collect Portal activity data with the help of the Activity Data Collector Service, and it will show you as an example how you can analyze this data to get the unique users per day.

Here is a rough overview of what has to be done to get the unique users per day:

1. The first thing to do is to activate and configure the ADC service.
2. After the ADC service is activated trace files with activity data for each server process of the SAP Web Application Server Java are written. So we have to gather all the trace files from all the instances and server processes in one place and combine them.
3. Now we can start with the analyzing of the activity data and extract the information we are interested in. In this case I would like to know the unique user per day. To get the number of the unique users we have to extract the user-ID's from the activity trace, remove all the duplicated entries, and last but not least count the number the unique user-ID's from the activity traces.
4. We now have the number of unique users. Last step could be to generate a report with the data.

So this is basically what we have to do. So let's start with the implementation of this scenario.

4.1 Activation and Configuration of the Activity Data Collector Service

At first we activate and configure the ADC service. The ADC service is installed with a preconfigured set of parameters which I am going to adjust to my needs.

> **Note**
>
> You can find more details about the activation and configuration of the Activity Data Collector on the SAP Help Portal.

1. Log on to the portal with your administrator user (you need access to System Administration).
2. Navigate to *System Administration → System Configuration → Service Configuration → Applications → com.sap.portal.activitydatacollector → Services → DataCollectionHook*
3. Open the DataCollectionHook for editing
4. Change the property "`Activate Data Collection`" from "**`false`**" (default) to "**`true`**" to make sure that the ADC service will be started. Additionally I adjust some of the settings to my needs.

   `Additional File Formats`
   Here you can tell the ADC service to write some additional information which you do not want to see in the main file to separate trace files. In my case here I want to have everything in the main file so I delete the entry completely.

   `Hour in the day to close all files, in GMT`
   Here I want to make sure that the trace files are closed at midnight. The time entered here is in GMT. Since I want the files to be closed at midnight German time I have to enter **23** here because Germany is in time zone GMT+1. So 23 o'clock GMT will be 24 o'clock German time.

   `Main File Format`
   Here I can define what data I want to have in the main trace file. To get the unique users the

date and the user would be enough. But I put in a little more to show you how it can look like. Here is the main file format I used for this example:
%Orfo.t(dd-MM-yyyy HH:mm:ss,GMT+1)%%Stab%%Orfo.un%%Stab%↙
%Orfo.pu%%Stab%%Orfo.rh(referer)%%Stab%%Orfo.qs%%Stab%%Oadc.pt%%Stab%↙
%Orfo.nid%%Snl%

Important

The symbol "↙" in the string above just indicates that I added a line break here to make it more readable. The string in the field "`Main File Format`" must not include line breaks or this symbol!

So in this string I defined a date and time stamp in the format "31-12-2008 23:58:25" with German time (GMT+1), user-ID, PCD URL, request header, query string, time to process the request in milliseconds, server process ID and a line feed at the end.

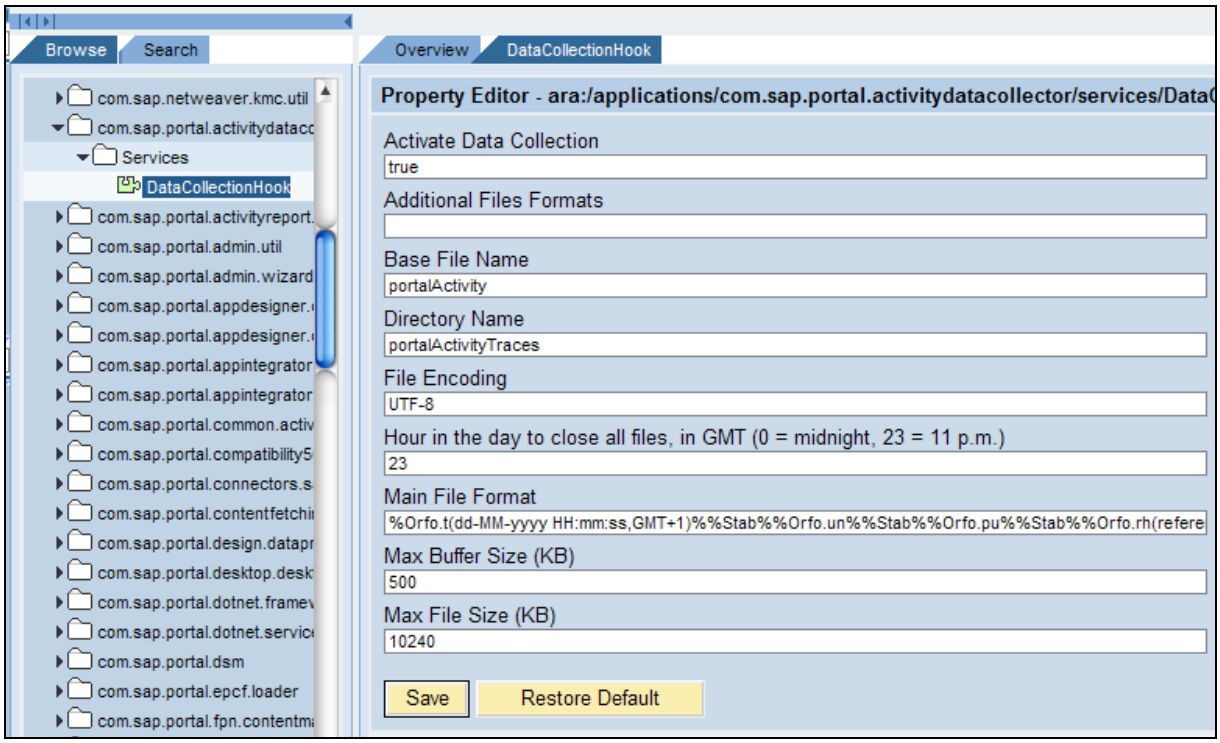
Example

This is how a single row of the trace file looks like with this configuration:

27-11-2008 15:00:36 test00001 ↙
pcd:portal_content/every_user/general/eu_role/com.sap.portal.eu_ws/com.sap.portal.portal_information http://p124134.wdf.sap.corp:50000/irj/portal ↙
NavPathUpdate=false&windowId=WID1194269623596 12 1971550

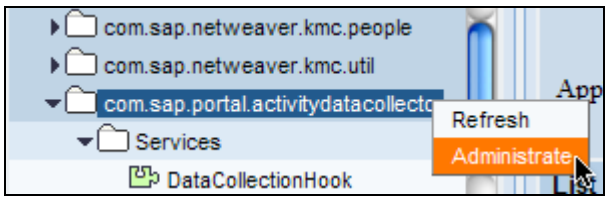
Note

You can find the complete file format definition on the SAP Help Portal.

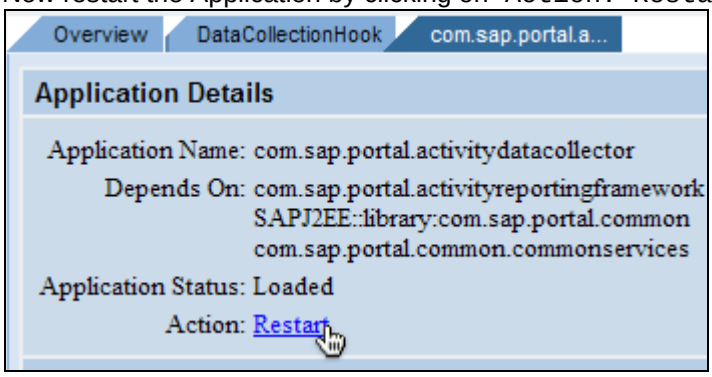


Please make sure to save all your changes!

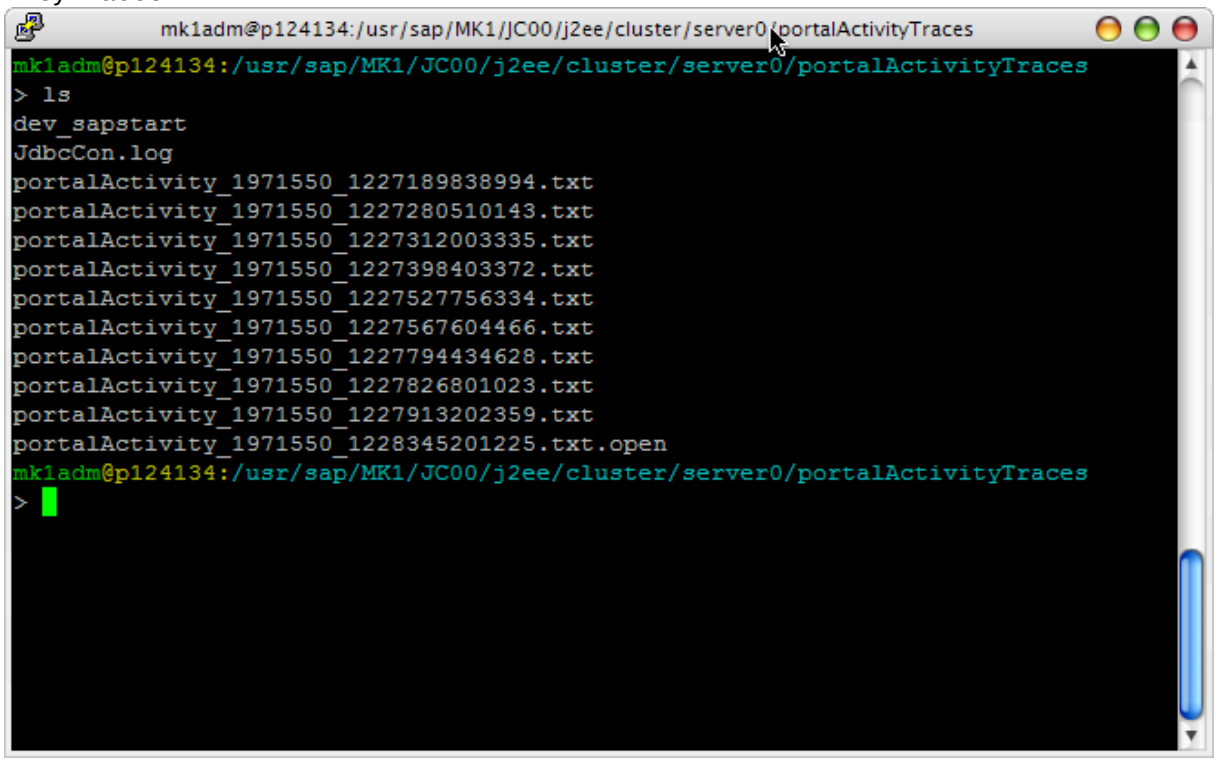
5. Next right click on "`com.sap.portal.activitydatacollector`" and choose "`Administrate`" from the context menu.



6. Now restart the Application by clicking on "`Action: Restart`"



7. With that you configured and activated the ADC service. The activity traces are now written to the following directory:
`/usr/sap/<SID>/J[C]<instance_#>/j2ee/cluster/server<number>/portalActivityTraces`



Note

Please note that depending on the Max Buffer Size you defined (default: 500kB) it can take some time until the first data is written.

Recommendation

You can set the Max Buffer Size to 1kB for test purposes. By doing so you can make sure that new activity data is written to the trace files almost immediately.
But please keep in mind that a Max Buffer Size of 1kB is not recommended for productive use!

Important

Up to now there is no housekeeping functionality in the Activity Data Collector. This means that the ADC continuously creates and writes log files, so you have to make sure that you clean up the ADC trace file directory from time to time and delete old trace files.

4.1.1 Determine what to be traced

The Activity Data Collector is now activated and writes traces with the log format you defined. Now you can even decide which Portal and Knowledge Management objects are to be traced and which one not. You may have some documents in the Knowledge Management where you do not want to trace the access, or you have some iViews or pages in the Portal which you do not want to trace because of security reasons or whatever.

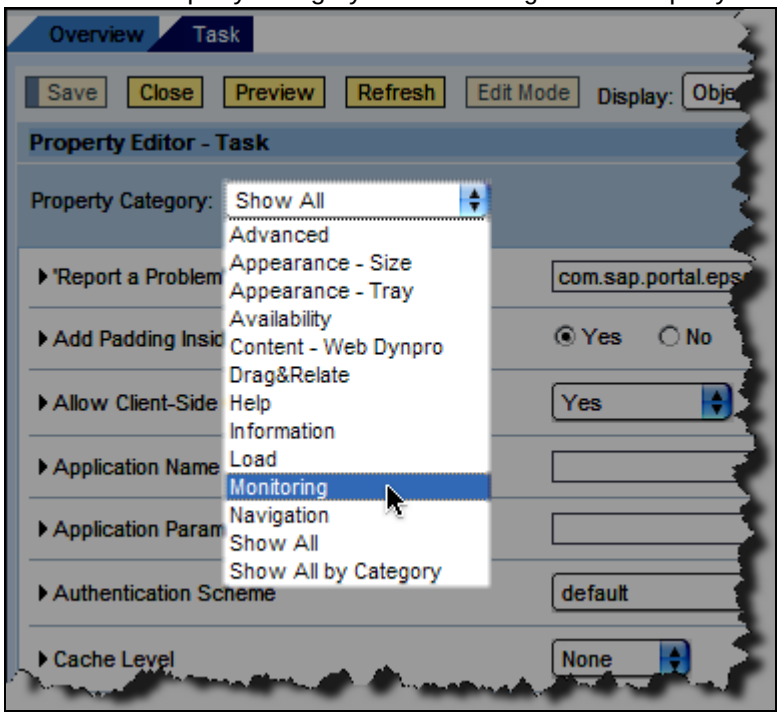
Here you will see how you can change the setting for the ADC tracing of Portal and Knowledge Management objects.

4.1.1.1 Portal PCD Objects

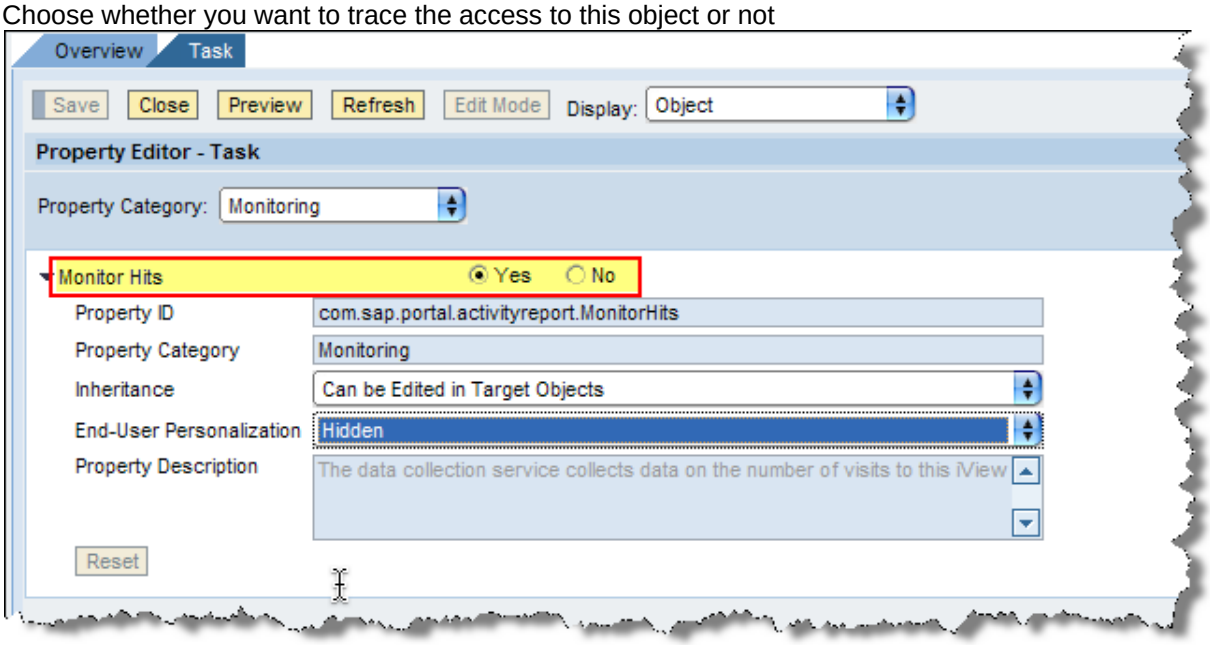
You can decide whether you want to trace the access of an Portal Content Directory object or not. Here is the step by step procedure for turning monitoring on and off:

1. Make sure you are logged on to the portal with your administrative user (access to Content Administration needed)
2. Navigate to: *Content Administration* → *Portal Content*
3. Open the PCD object you want to change for editing

4. Choose the Property Category "`Monitoring`" in the Property Editor



5. Choose whether you want to trace the access to this object or not



6. Save the settings

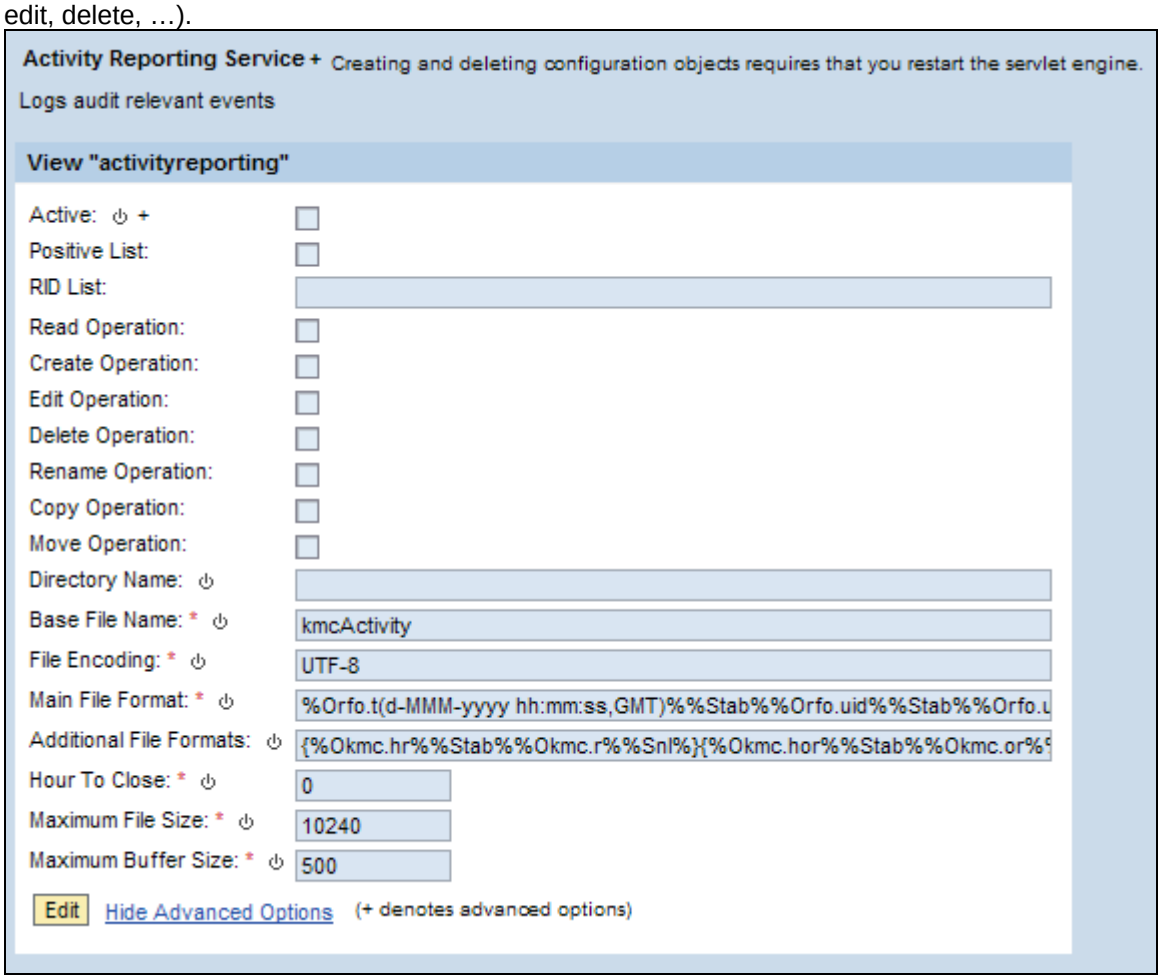
4.1.1.2 Knowledge Management

To configure the activity reporting for Knowledge Management proceed as follows:

Important

This functionality is only available starting with SAP NetWeaver 7.0 SPS14. There are no plans to release this functionality for SAP NetWeaver '04.

1. Make sure to be logged on as administrative user to the portal (access to System Administration needed)
2. Navigate to: *System Administration* → *System Configuration* → *Knowledge Management* → *Content Management* → *Global Services* → *click on "Show Advanced Options"* → *Activity Reporting Service*
3. Here you can activate and configure the activity reporting for KM by providing the RID you want to trace. You can also define which operations you want to trace (for Example: read, create, edit, delete, …).



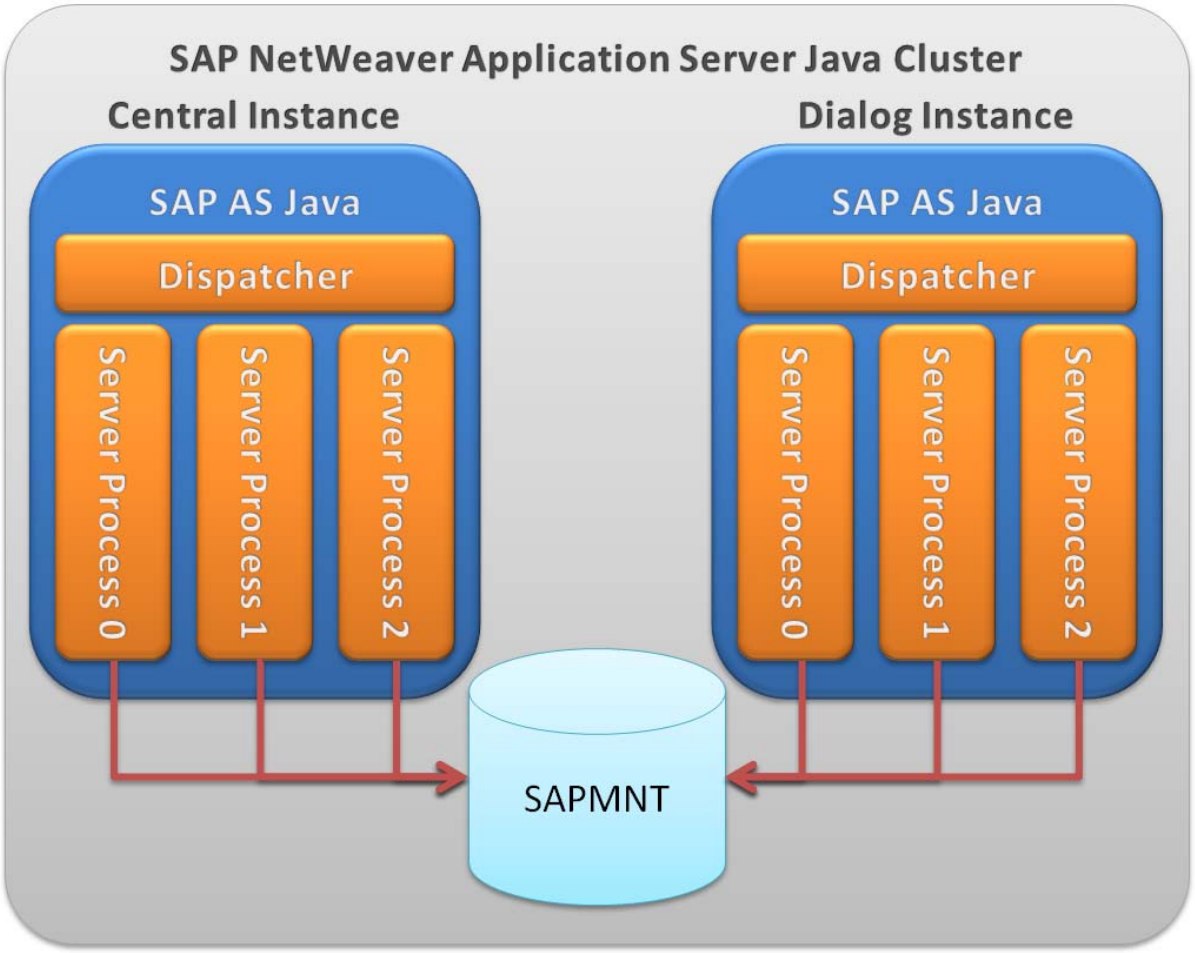
> **Note**
>
> You can find a detailed description of this configuration dialog in the SAP Help Portal.

4. Make sure to save your changes.

> **Important**
>
> Please note that a change of the parameters marked with the ⏻ icon requires a restart of the J2EE engine.

4.2 Gather Trace Files

The goal now is to gather all the trace files from all the instances and server processes in one central place.



If you have a SAP NetWeaver Application Server Java Cluster in place then you already have a central file share for all the systems of this cluster in place – SAPMNT.

So you can think of a scheduled job which copies all the trace files every night to this central share. In this example here I create a Linux shell script which copies all the trace files from the different server processes to this central place.

1. At first create a directory where you place all the trace files and make sure that your user (<SID>adm) can access it. I created a directory called: "`/sapmnt/portalActivityTraces`".
2. After that create a shell script which copies all the trace files for you. You can then run this shell script via cronjob every night after midnight. Here's what I do with my shell script:
    a. Create a new directory with the name the of the date from the trace files
    Example: `/sapmnt/portalActivityTraces/2008-11-30`
    b. Copy all the trace files with a time stamp from yesterday and today (today because the last trace file will be closed at midnight and therefore gets the time stamp from today) to the newly created directory.

Here is what my shell script looks like:

Example: copyCurrentActivityDataTracefiles.sh

```
#!/bin/sh

#*****************************************************************************
#   Copy current Portal Activity Traces to central place for analysis
#
#*****************************************************************************

#*******************************
# Define some needed variables
#*******************************
# get date from today
today=`date +%Y-%m-%d`
# get date from yesterday
TZ=`date +%Z`+24
export TZ
yesterday=`date +%Y-%m-%d`
# define target directory for Portal Activity Traces
target="/sapmnt/portalActivityTraces"
SID=MK1

#*******************************
# Create target directory
#*******************************
if test -d $target/$yesterday
  then
    # if directory already exists do nothing
    echo "Directory "$target/$yesterday" already exists."
  else
    # if directory doesn't exist create it
    mkdir $target/$yesterday
    echo "Directory "$target/$yesterday" created."
fi

#*******************************
# Copy Portal Activity Traces from yesterday and today to target directory
#*******************************
TZ=""
export TZ
echo "Copying Portal Activity Traces from yesterdy ("$yesterday")"
# copy all Portal Activity Traces from yesterday to target directory
for yesterdaylogs in `ls -ltr --time-style=+%Y-%m-%d
/usr/sap/$SID/J*/j2ee/cluster/server*/portalActivityTraces/portalActivity*
| grep $yesterday | cut -f7 -d" "`
  do
    cp $yesterdaylogs $target/$yesterday
    echo "Copied "$yesterdaylogs" to "$target/$yesterday
  done

echo "Copying Portal Activity Traces from today ("$today")"
# copy all Portal Activity Traces from today to target directory
for todaylogs in `ls -ltr --time-style=+%Y-%m-%d
/usr/sap/$SID/J*/j2ee/cluster/server*/portalActivityTraces/portalActivity*
| grep $today | cut -f7 -d" "`
  do
    cp $todaylogs $target/$yesterday
    echo "Copied "$todaylogs" to "$target/$yesterday
  done
```

3. Adopt this script to your needs (target directory, SID) and put this shell script on every server of the SAP NetWeaver Application Server Java Cluster (Central Instance and Dialog Instances) and make sure that this script is executed every night after midnight for example at 2 o'clock.

With that we have all the trace files from yesterday in a folder with the name of the date from yesterday.

> **Tip**
>
> Of course you can do something similar with Windows batch files to copy the trace files to a central share and schedule this as well.

4.3 Analyze the Trace Files

So all the trace files from yesterday and today are now collected in one central place. Now we can combine all the traces and extract the information we are looking for.

4.3.1 Analyzing on Linux

All trace files from yesterday and today and from all server processes are now collected in the folder "`/sapmnt/portalActivityTraces/<Year>-<Month>-<Day>`". Now we can start to combine them and extract the unique users.

The next Linux shell script takes care of:

1. Remove all trace lines where the date is different from yesterdays date
2. Extract all Users from all trace files
3. Sort Users and remove duplicates. With that we make sure that each user is only counted once.
4. Count the number of unique users and write the result to a log file

Like before you can run this shell script every night via cronjob. Just make sure that the shell script which copies the traces to the central place has been executed before.

Example: extractUniqueUsers.sh

```
#!/bin/sh

#*****************************************************************************#
#   get Unique Users from Activity Traces
#
#*****************************************************************************#

#*******************************
# Define some needed variables
#*******************************
# get date from today
today=`date +%Y-%m-%d`
# get date from yesterday
TZ=`date +%Z`+24
export TZ
yesterday=`date +%Y-%m-%d` # date format from filesystem
yesterlog=`date +%d-%m-%Y` # date format used in logs
# define target directory for Portal Activity Traces
target="/sapmnt/portalActivityTraces"

#*******************************
# Extract Unique Users out of Portal Activity Traces
#
# 1. filter all Portal Activity Traces for yesterdays date
# 2. extract all users
# 3. sort users and remove duplicates
# 4. count unique users
# 5. write unique users to log file
#*******************************
cd $target/$yesterday
# get the unique users out of all Portal Activity Traces
UniqueUsers=`cat portalActivity* | grep $yesterlog | cut -f2 | sort -u | wc
-w`
# create Log file with date and unique users for that date
```

```
echo $yesterlog $UniqueUsers > $yesterday"_UniqueUsers.txt"
echo "Unique Users extracted"
cat $yesterday"_UniqueUsers.txt"

# clean up no more needed Portal Activity Traces
rm portalActivity*
```

4.3.2 Analyzing on Windows

Of course you can also extract the unique users on Windows servers. For that reason I prepared some Visual Basic programs. It is doing exactly the same as the Linux shell script. So all the traces copied will be filtered for the date you are interested in, after that the users will be extracted and duplicated users will be removed. And last but not least the unique users will be counted and written to a log file.

Tip

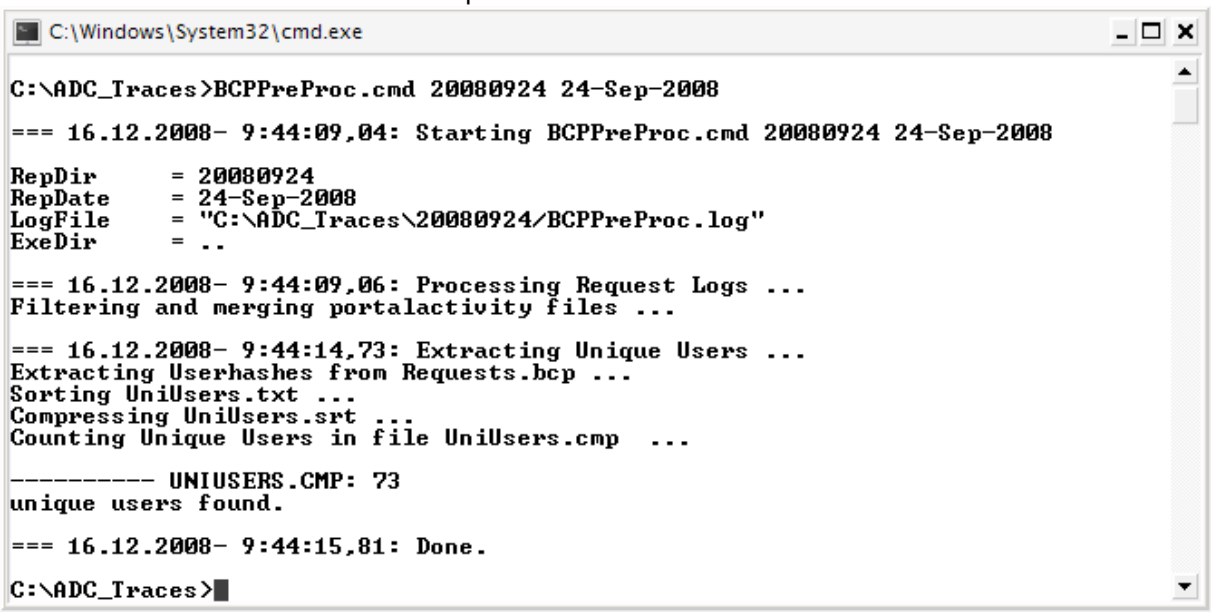
The executables, batch file and the source code is available for download on SDN. You can find the download on the same page where you can get this How-To Guide.

Let's assume that all the trace files from yesterday are copied to folder called "`C:\ADC_Traces\<Year><Month><Day>`" for example "`C:\ADC_Traces\20080924`"

1. Place the executables "Compressor.exe", "DelUsrName.exe" and "ExtUniUsers.exe" as well as the batch file "BCPPreProc.cmd" in the folder "`C:\ADC_Traces\`".
2. Now start the batch file as follows: "`BCPPreProc.cmd <Folder with trace files> <Date to filter for in trace files>`".

   Example

   `BCPPreProc.cmd 20080924 24-Sep-2008`



3. You can see that in this case 73 unique users have been found in the Activity Data Collector traces.

4.4 Generating a Custom Report

We already reached the goal of extracting the unique users per day from the Activity Data Collector traces. But wouldn't it be nice to have a little report about the results which you could access with your browser?

Here is an example how you can create a little HTML report with the results. I created a Linux shell script which generates a HTML report from the log files with the unique users. After the creation the HTML report is copied into a folder of a web server for publication.

Here is what you have to do:

1. Create a text file which contains the header of the HTML Report. In this file you can place the static information of the HTML which will always stay the same like Style Sheets, Title and the basic construction for the table which will then be filled with the results.

Example: UniqueUsersHTMLReportHead.txt

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
  <meta http-equiv="content-type" content="text/html; charset=windows-
1250">
  <title>Portal Activity Report - Unique Users</title>
  <style type="text/css">
    body {
      font-family: sans-serif;
     text-align: center;
      }
    h1 {
      color:navy;
      text-align:center;
      }
    table {
      font: 13px/24px Verdana, Arial, Helvetica, sans-serif;
      border-collapse: collapse;
      margin-left: auto;
      margin-right: auto;
      }

    th {
      padding: 0 0.5em;
      text-align: center;
      background: #FFC;
      border: 1px solid
      }

    tr {
      background: #EEE;
      }

    td {
      border: 1px solid;
      text-align: center;
      padding: 1 2.5 1 0.5em;
      }

    td td {
```

```
      border: 0;
    }
  </style>
  </head>
  <body>

<h1>Portal Activity Report - Unique Users</h1>

<table>
  <tr>
    <th>Date</th>
    <th>Unique Users</th>
    <th>Graph</th>
  </tr>
```

2. Create a text file which contains the footer of the HTML report. This file here only contains 3 lines of code to close the table, body and html tag.

 Example: UniqueUsersHTMLReportFoot.txt

```
    </table>
  </body>
</html>
```

3. In the HTML header file I already defined a table with the columns: “Date”, “Unique Users” and “Graph”. The next shell script will now:
   a) read the HTML header text file and write it to a plain HTML text file on disk
   b) create the HTML code for the table rows containing the data from the log files created before, and append this code to the created HTML file
   c) read a HTML footer text file and append it to the created HTML file.
   d) copy the HTML file into a folder of a web server

 Example

```
#!/bin/sh

target="/sapmnt/portalActivityTraces"

# gather all Unique Users files and sort them: latest entry first to have
it on top
cat $target/*/*UniqueUsers.txt | sort -t "-" -k 3nr -k 2nr -k 1nr >
$target/UniqueUsersReport/UniqueUsersReport.txt

# echo HTML Head
cat $target/UniqueUsersReport/UniqueUsersHTMLReportHead.txt >
$target/UniqueUsersReport/UniqueUsers.html

echo "<tr><td colspan=3>Report created: "`date +"%d-%m-%Y %H:%M
%Z"`"</td></tr>"  >> $target/UniqueUsersReport/UniqueUsers.html
# get max user count
maxCount=`cat $target/UniqueUsersReport/UniqueUsersReport.txt | cut -f2 -d
" " | sort -n | tail -n1`

# create table
counter=1
for entry in `cat $target/UniqueUsersReport/UniqueUsersReport.txt`
  do
    if test `expr $counter % 2` -eq 1
    then
```

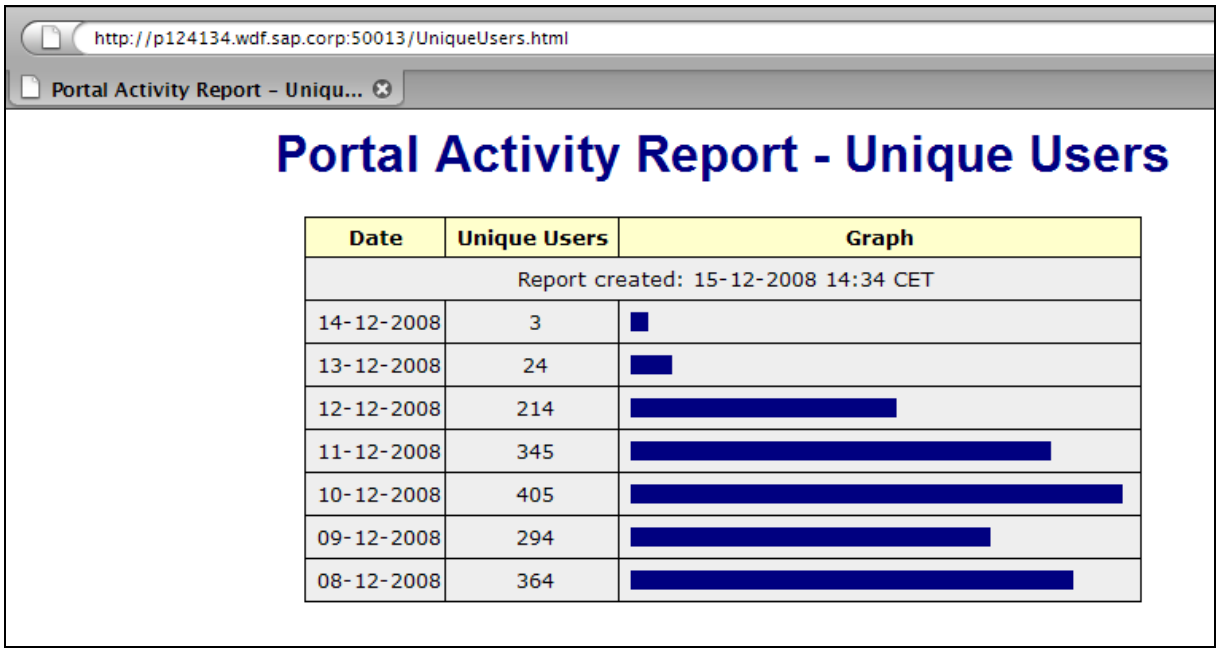
```
      echo "<tr>" >> $target/UniqueUsersReport/UniqueUsers.html
      echo "<td>"$entry"</td>" >>
$target/UniqueUsersReport/UniqueUsers.html
    else
      echo "<td>"$entry"</td>" >>
$target/UniqueUsersReport/UniqueUsers.html
      echo "<td>" >> $target/UniqueUsersReport/UniqueUsers.html
      echo "<table width=100%>" >>
$target/UniqueUsersReport/UniqueUsers.html
      echo "<tr>" >> $target/UniqueUsersReport/UniqueUsers.html
     barwidth=`echo "$entry * 300 / $maxCount" | bc`
     if test `expr $barwidth = 0` -eq 1
      then
        echo "<td height=12 title="$entry"></td>" >>
$target/UniqueUsersReport/UniqueUsers.html
      else
        echo "<td bgcolor=navy width="$barwidth" height=12
title="$entry"></td>" >> $target/UniqueUsersReport/UniqueUsers.html
     fi
      echo "<td></td>" >> $target/UniqueUsersReport/UniqueUsers.html
      echo "</tr>" >> $target/UniqueUsersReport/UniqueUsers.html
      echo "</table>" >> $target/UniqueUsersReport/UniqueUsers.html
      echo "</td>" >> $target/UniqueUsersReport/UniqueUsers.html
      echo "</tr>" >> $target/UniqueUsersReport/UniqueUsers.html
    fi
    counter=`expr $counter + 1`
  done

# echo table foot
cat $target/UniqueUsersReport/UniqueUsersHTMLReportFoot.txt >>
$target/UniqueUsersReport/UniqueUsers.html

cp $target/UniqueUsersReport/UniqueUsers.html
/usr/sap/MK1/SYS/exe/run/servicehttp/
```

By doing so you can create automatically a report every night which you can access easily with your browser.

Example: Screenshot of Report



Like mentioned at the beginning, this How-To Guides intention is to get you started and to give you an basic idea of what you can do with the Activity Data Collector. Now you can start to think of your own scenarios and reports you would like to see and implement them on your own. Or you think of other possibilities to analyze the trace files.

Tip

You can find the example scripts from this How-To Guide located on SDN for download on the same page where you can also get this How-To Guide.

www.sdn.sap.com/irj/sdn/howtoguides