Webtrekk
Drive Your Profit.

Webtrekk SOAP Services

Version 3.2

Webtrekk
Drive Your Profit.

1 Content

2 General

Webtrekk SOAP Service gives you the opportunity to automatically recall data from Webtrekk and to upload new data. The connection with the Webtrekk SOAP Service takes place via HTTP or HTTPS protocols.

Service	URL
Webtrekk Analysis SOAP Service	http://report2.webtrekk.de/cgi-bin/wt/SOAPv3.cgi
WSDL-File (rpc/encoded)	http://report2.webtrekk.de/SOAPv3.wsdl
WSDL-File (document/literal)	http://soap1.webtrekk.com/soapv3/services/WTAnalysisGeneratorWS?wsdl

2.1 Methods

Following methods are available by the SOAP Service:

Method	Description
getAnalysisDataTest	Controls the connection to Webtrekk SOAP Service
getAnalysisData	Gets output data of analysis, head of table, foot of table and image data (see chapter 3)
getReportData	Gets output data of reports (see chapter 4)
exportData	Export data (see chapter 5)
importData	Import data (see chapter 6)

2.2 Connection Test

In order to test the connection, you simply use the method „getAnalysisDataTest". The method works without any login data or other parameters. The string „Connection successful" obtains the test result.

Request
Client → Webtrekk SOAP Service

Parameter	Type	Length	Optional	Description
none				

Response
Webtrekk SOAP Service → Client

Parameter	Type	Length	Optional	Description
	string		no	Submits string „Connection successful“

2.3 Error Messages

Errors can occur during communication with Webtrekk. This is a list of error messages which can appear (under certain circumstances):

Error Code	Error Message	Description
Server.AuthError	Authorization failed	Attempt to log in failed, wrong login, wrong password, no access permission
Server.ExecError	Max querylimit reached	Request/Query-limit reached, Default: 100 requests per day per account
Server.ExecError	Unknown Analysis 'analysisName'	Name of analysis unknown. (When using Analysis Requests.)
Server.ExecError	Analysis is not activated!	The requested analysis exists but isn‘t activated for the chosen account.
Server.ExecError	No data available for this analysis and your configuration	No data for the requested analysis with given configuration available. (When retrieving analyses.)
Server.ExecError	Max row limit reached	Max. 5000 rows of one request. (When uploading retour data.)
Server.ExecError	Report not found	Report with entered name does not exist. (When retrieving a report.)

3 Analyses Requests

By using the method „getAnalysisData" you retrieve data of various analyses per SOAP. In order to do so it is necessary to add a row of parameters.

3.1 Parameters

The parameter handover of this method takes place as a Hashtable (associative array). Following parameters are expected resp. can optionally be specified:

Request
Client → Webtrekk SOAP Service

Parameter	Type	Length	Optional	Description
customerId	string	15	no	ID of wanted customer (Customer-ID)
login	string		no	Login Webtrekk Frontend
pass	string		no	Password Webtrekk Frontend
analysis	string		no	Wanted analysis, see 3.3 Analyses
time_start	string		no	Start time, format: YYYY-MM-DD or YYYY-MM-DD hh:mm:ss
time_stop	string		no	Stop time, format: YYYY-MM-DD or YYYY-MM-DD hh:mm:ss
visitor_group	string		yes	Wanted Visitor Groups, default: "all visitors" (all visitors)
output_row_start	int		yes	Start of data output from line x, default: Line 0 (first line). The data output is arranged in descending order of e.g. visits.
limit_rows	int		yes	Max. output lines (maximum: 10000), default: adjusted account standard limit
search_string	string	255	yes	Searchstring, e.g. *products*, see 3.3 Analyses
build_pic	int	1	yes	Should image data be generated? Yes: 1, default: 0 (no)

pic_width	int		yes	Picture width (minimum: 200px, maximum: 1000px), default: 300
pic_height	int		yes	Picture height (minimum: 150px, maximum: 1000px), default: 200
Language	string		yes	Language (‚de', ‚en', ‚fr', ‚es')
Column	array		yes	Output of specific columns. Input as array, e.g. [„Page Impressions", „Visitors"]
analysis_filter	array	255	yes	See 3.3 Analyses
decodeUtf8	Int	1	yes	Should the Soap response be utf8 decoded? Yes: 1, No: 0, Default 0 (No)*

* By some clients, the Soap response is utf8 coded twice. By setting this parameter, the utf8 coding is rebuilt.

3.2 Return Values

The return of values also takes place as a Hashtable. For most analyses only one table of results will be submitted. The results look as follows:

Response
Webtrekk SOAP Service → Client

Parameter	Type	Optional	Description
count	int	no	Number of table results
analysisTitle	string	no	Title of analysis
analysisData	array of arrays	no	Analysis data as 2 dimensional array
analysisTabHead	array	no	Table Head
analysisTabFoot	array of arrays	yes	Table Foot
analysisPic3d	string	yes	3-D picture data as PNG (Portable Network Graphics) coded in Base64

analysisPic2d	string	yes	2-D picture data as PNG (Portable Network Graphics) coded in Base64
totalRowCount	int	yes	Number of possible rows whereas not all rows have to be read out. Note: Not all analyses will be calculated!
analysisWarnings	array	yes	Analysiswarnings (e.g. missing rights)
dailyUnique	string	yes	Contains an advice if the analysis includes only metrics that are calculated uniquely per day
calculationTime	string	no	Time of Calculation

If there is more than one table, following structure will be submitted:

Response
Webtrekk SOAP Service → Client

Parameter	Type	Optional	Description
count	int	no	Number of table results
array	array of single response	no	Per table of the results one array-entry which contains a hash together with the data analysisTitle, analysisData, analysisTabHead, analysisTabFoot, analyisPic3d and analysisPic2s (See response with only one table of results.)

The parameter „count" in the result shows you which option was used.

3.3 Analyses

All analyses are available via the Webtrekk SOAP Service. The format of the „analysis" parameter is „Level1|Level2|Analysis", so e.g. „Navigation|Pages|Pages" or „Visitor|Traffic". The levels are equivalent to the navigation menue of the Webtrekk-Tool.

> Please note that the parameter „search_string" is not available for every anlysis.

3.3.1 Special Parameters

3.3.3.1 Parameter “search_string”

You can use the parameter „search_string” if you would like to display the filter-rule (example of analysis Navigation|Pages|Pages“) in the following way: " *shirts* " or "de.women.t_shirts".

Using placeholder like " * " is also possible. The filter "de.women.t_shirts" shows the results which match this model only. The filter " *shirts* " for example shows contains all pages whose names include „shirts“, e.g. "de.men.shirts" or "de.women.shirts.tees" etc.

3.3.3.2 Parameter „analysis_filter“

Complex filters are composed of one or more „basic filters“ which are logically connected with each other. The chosen filters are connected by „OR“ or „AND“.

A basic filter is composed of the components

Connection:	„AND“ / „OR“
Filter object:	Analysis object or metric (e.g. „Pages“, „Page Impressions“)
Attribute:	„LIKE“ / „NOT LIKE“ („like“ / „unlike“ for strings)
	„gt“ / „lt“ / „between“ („greater“ / „less“ / „between“ for numbers)
Search term:	String or number, equivalent to „search_string“

Example!:

```
[
        [„“, „Pages“,“LIKE“,“*index*          ],
        [„AND“, „Page Impressions“, „gt“, 1000        ]
]
```

3.3.3.3 Parameter „column_sort_order“

With this parameter, a sort order on a column within the data table can be given. At the moment, exactly one one sort order on a metric (no formula) is supported. It is mandatory to set the parameter “column” in which the column is also to be set.

The name of the column as well as the sort order have to be given. Valid values for the sort order are “ASC” for ascending sort order and “DESC” for descending sort order.

Example!:

```
[
        [„Page Impressions“, „DESC”          ],
]
```

Webtrekk
Drive Your Profit.

Code Examples (Client)

The following chapters show examples of how to retrieve analyses by using the SOAP-interface in different languages

3.3.2 Perl

```
#!/usr/bin/perl

use SOAP::Lite;

my $endpoint = "http://report2.webtrekk.de/cgi-bin/wt/SOAPv3.cgi";
my $soapaction = "urn:reportSOAP#getAnalysisData";
my $method = 'getAnalysisData';
my $method_urn = 'urn:objects::reportSOAP';

my %parameter = (
        customerId          => '111111111111111',
        login               => 'mylogin',
        pass                => 'mypass',
        analysis            => 'Navigation|Seiten|Seiten',
        time_start          => '2005-01-01',
        time_stop           => '2005-01-31',
        visitor_group       => 'buyer',
        output_row_start    => 0,
        limit_rows          => 10,
        search_string       => '*product*',
        build_pic           => 1,
        pic_width           => 500,
        pic_height          => 300
);

my $soap = SOAP::Lite->new(uri => $soapaction, proxy => $endpoint);
my $response = $soap->call(SOAP::Data->name($method)->attr(
                { xmlns => $method_urn }
        ) => (\%parameter) );

my $data = $response->result;

# print data
use Data::Dumper;
print Dumper($data);

# print picture
use MIME::Base64;

open(IMG, ">pic3d.png");
binmode IMG;
print IMG decode_base64($data->{analysisPic3d});
close IMG;
```

3.3.3 PHP

```
<?php
require_once ('nusoap.php');

// Configuration
$wsdl_path = 'http://report2.webtrekk.de/SOAPv3.wsdl';
$method = 'getAnalysisData';
$parameter = array (
        'customerId' => '111111111111111',
        'login' => 'mylogin',
        'pass' => 'mypass',
        'analysis' => 'Navigation|Pages|Pages',
        'analysis_filter' => array(array("","Country", "LIKE", "Spain")),
        'time_start' => '2007-02-01',
        'time_stop' => '2007-02-28',
        'visitor_group' => "",
        'limit_rows' => 5,
        'search_string' => '',
        'build_pic' => 1,
        'pic_width' => 500,
        'pic_height' => 300
);

// Create nusoap Client instance and check for errors
$client = new soapclient($wsdl_path, true); // possibly new nusoap_client(...)
$err = $client->getError();
if ($err) {
        echo '<h2>Error while generating the SOAP-instance</h2>';
echo '<pre>' . $err . '</pre>';
        exit(1);
}

// retrieve data and check for errors
$result = $client->call($method, array ($parameter));
if ($client->fault) {
        echo '<h2>Server reports error</h2><pre>';
        print_r($result);
        echo '</pre>';
        exit(1);
}
if ($client->getError()) {
        echo '<h2>Error while handing out data</h2><pre>' . $err . '</pre>';
        exit(1);
}

// Hand out result
echo '<h2>Ergebnis</h2><pre>';
print_r($result);
echo '</pre>';

// Write image in file
$fp = fopen("pic3d.png", "w+");
fwrite($fp, base64_decode($result['analysisPic3d']));
fclose($fp);
?>
```

3.3.4 Java (Axis 1.4)

In order to use Java it is necessary to use the Java packages „Apache Axis“ (of version 1.4) as well as „Apache Common Codecs“. Please install both packages into the Java-„Lib“ folder, resp. into a folder which is listed in the system environment variable CLASSPATH (e.g. „C:\Java\Lib“). Switch with the command row into the folder and implement the following:

```
java org.apache.axis.wsdl.WSDL2Java http://report2.webtrekk.de/SOAPv3.wsdl
```

By doing that two other packages (“de.webtrekk.report2.SOAPv3_wsdl” and “xml_soap_wt”) are created, by which you get access tot he SOAP-interface.

```
// RequestAnalysisData.java
import java.io.*;
import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;
import de.webtrekk.report2.SOAPv3_wsdl.*;
import org.apache.commons.codec.binary.Base64;
import xml_soap_wt.Webtrekkoutput;

public class RequestAnalysisData {
	/**
	 * Implements a request for the SOAP-interface
	 * @param args command row parameter (will not be analysed)
	 */
	public static void main(String[] args) {
		/* *** set input-parameter *** */
		xml_soap_wt.Webtrekkinput input = new xml_soap_wt.Webtrekkinput();
		input.setCustomerId("111111111111111");
		input.setLogin("mylogin");
		input.setPass("mypass");
		input.setAnalysis("Navigation|Pages|Pages");
		input.setTime_start("2007-02-01");
		input.setTime_stop("2007-02-28");
		input.setVisitor_group("");
		input.setOutput_row_start(0);
		input.setLimit_rows(10);
		input.setSearch_string("*home*");
		input.setBuild_pic(1);
		input.setPic_width(500);
		input.setPic_height(300);
		input.setLanguage("de");


		/* *** Start requests – hand out results *** */
		Webtrekkoutput result;
		try {
			AnalysisGeneratorService myService = new AnalysisGeneratorServiceLocator();
			AnalysisGeneratorPortType myPort = myService.getanalysisGeneratorPort();
```

```
            result = myPort.getAnalysisData(input);
        } catch (ServiceException e) {
            System.err.println("Connection failed: "
                                                    + e.getMessage());
            System.exit(1);
            return;
        } catch (RemoteException e) {
            System.err.println("Error during data call: " + e.getMessage());
            System.exit(1);
            return;
        }
        Object[][] analysisData = result.getAnalysisData();
        Object[] analysisTabHead = result.getAnalysisTabHead();
        Object[][] analysisTabFoot = result.getAnalysisTabFoot();
        String analysisPic3d = result.getAnalysisPic3D();
        String analysisPic2d = result.getAnalysisPic2D();
        Object[] pageUrls = result.getPageUrls();

        /* *** hand out results *** */
        // The table head
        for (int spalte = 0; column < analysisTabHead.length; cokumn++) {
            System.out.print(analysisTabHead[spalte] + "\t");
        }
        System.out.println();


    System.out.println("=====================================================");

        // The table content
        for (int row = 0; row < analysisData.length; row++) {
            for (int column = 0; column < analysisData[zeile].length; column++) {
                System.out.print(analysisData[row][column] + "\t");
            }
            System.out.println();
        }

        // The table foot
        if (analysisTabFoot != null) {

    System.out.println("=====================================================");
            for (int row = 0; row < analysisTabFoot.length; row++) {
                for (int column = 0; column < analysisTabFoot[zeile].length; column++) {
                        System.out.print(analysisTabFoot[row][column] + " ");
                }
                System.out.println();
            }
        }
        System.out.println();
        System.out.println();

        // The 2D-image
        if (analysisPic2d != null) {
            String filename = "Pic2D.png";
            byte[] bytes = analysisPic2d.getBytes();
            byte[] binary = new Base64().decode(bytes);
            try {
                FileOutputStream out = new FileOutputStream(filename);
                out.write(binary);
                out.flush();
```

```
                    out.close();
                    System.out.println("AnalysisPic2D: Datei '" + filename
                                                        + "' erfolgreich geschrieben.");
                } catch (IOException e) {
                    System.err.println("Error while saving the 2D-image: "
                                                        + e.getMessage());
                }
            }
        }
    }
```

3.3.5 Java (Axis 2.0)

In order to use Java it is necessary to use the Java package „Apache Axis 2.0“, for example. Do the following on the command line:

```
%AXIS2_HOME%\bin\WSDL2Java -uw -o . -d adb –uri
http://soap1.webtrekk.com/soapv3/services/WTAnalysisGeneratorWS?wsdl

// RequestAnalysisData.java
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.rmi.RemoteException;
import org.apache.commons.codec.binary.Base64;
import de.webtrekk.ws.doc.WTAnalysisGeneratorWSStub;
import de.webtrekk.ws.doc.WTAnalysisGeneratorWSStub.*;

public class RequestAnalysisData
{

	public static void main(String args[])
	{
		try {
			WTAnalysisGeneratorWSStub stub = new WTAnalysisGeneratorWSStub();

			WTInput input = new WTInput();
			input.setCustomerId("111111111111111");
			input.setLogin("mylogin");
			input.setPassword("mypass");
			input.setAnalysis("Visitors|Traffic");
			input.setTimeStart("2009-08-18 00:00:00");
			input.setTimeStop("2009-09-01 23:59:59");
			input.setVisitorGroup("");
			input.setPeriod("");
			input.setOutputRowStart(0);
			input.setLimitRows(5);
			input.setSearchString("");
			input.setBuildPic(0);
			input.setPicWidth(500);
			input.setPicHeight(300);
			input.setLanguage("en");

			WTOutput output = stub.getAnalysisData(input);
			printAnalysis(output, null);
		} catch (RemoteException e) {
			e.printStackTrace();
		}
	}

	private static void printAnalysis(WTOutput result, String filenameAdd) {
		// Does data exist at all?
		if (result == null) {
			return;
		}
```

```
        // Check all table results

        WTAnalysisData[] data = result.getData();

        for (int i = 0; i < data.length; i++)
        {
            WTAnalysisData analysis = data[i];
            // Separate data
            DataRecord[] analysisData = analysis.getAnalysisData();
            DataRecord analysisTabHead = analysis.getAnalysisTabHead();
            DataRecord[] analysisTabFoot = analysis.getAnalysisTabFoot();
            String analysisPic3d = analysis.getAnalysisPic3D();
            String analysisPic2d = analysis.getAnalysisPic2D();

            // Output
            System.out.println("Schreibe: " + analysis.getAnalysisTitle().getValue());
            printTable(null, analysisData, analysisTabHead, analysisTabFoot);
            printPictures(null, analysisPic3d, analysisPic2d);
        }
    }

    private static void printPictures(String filenameAdd, String analysisPic3d, String analysisPic2d) {
        if (filenameAdd == null) {
            filenameAdd = "";
        }
        // The 2D Picture
        if (analysisPic2d != null && analysisPic2d.length() > 0) {
            String filename = "Pic2D_" + filenameAdd + ".png";
            byte[] bytes = analysisPic2d.getBytes();
            byte[] binary = new Base64().decode(bytes);
            try {
              FileOutputStream out = new FileOutputStream(filename);
              out.write(binary);
              out.flush();
              out.close();
            } catch (IOException e) {
              System.err.println("Error while saving the 2D picture: " + e.getMessage());
            }
        }

        // The 3D Picture
        if (analysisPic2d != null && analysisPic2d.length() > 0) {
            String filename = "Pic3D_" + filenameAdd + ".png";
            byte[] bytes = analysisPic3d.getBytes();
            byte[] binary = new Base64().decode(bytes);
            try {
              FileOutputStream out = new FileOutputStream(filename);
              out.write(binary);
              out.flush();
              out.close();
            } catch (IOException e) {
              System.err.println("Error while saving the 3D picture: " + e.getMessage());
            }
        }
    }

    private static void printTable(String filenameAdd, DataRecord[] analysisData, DataRecord
```

```
analysisTabHead, DataRecord[] analysisTabFoot) {
    if (analysisData.length == 0) {
        System.out.println("  No data!");
        return;
    }
    // Open output file
    PrintStream textOut;
    String filename = null;
    if (filenameAdd == null) {
        textOut = System.out;
    } else {
        filename = "Analysis" + filenameAdd + ".txt";
        try {
          textOut = new PrintStream(new FileOutputStream(filename));
        } catch (IOException e) {
          System.err.println(" Can not open file: " + e.getMessage());
          return;
        }
    }

    // The table head
    if (analysisTabHead != null) {
        for (int column = 0; column < analysisTabHead.getValues().length; column++) {
          textOut.print(analysisTabHead.getValues()[column].getValue() + "\t");
        }
        textOut.println();
        textOut.println("=================================================");
    }

    // The table content
    for (int row = 0; row < analysisData.length; row++) {
        for (int column = 0; column < analysisData[row].getValues().length; column++) {
            if(analysisData[row].getValues()[column] != null)
                textOut.print(analysisData[row].getValues()[column].getValue() + "\t");
            else
                textOut.print("\t");
        }
        textOut.println();
    }

    // The table foot
    if (analysisTabFoot != null) {
    textOut.println("=================================================");
        for (int row = 0; row < analysisTabFoot.length; row++) {
            for (int column = 0; column < analysisTabFoot[row].getValues().length; column++) {
                if(analysisTabFoot[row].getValues()[column] != null)
                    textOut.print(analysisTabFoot[row].getValues()[spalte].getValue() + " ");
                else
                    textOut.print("\t");
            }
            textOut.println();
        }
    }
    textOut.println();

    // Success message and close file
    if (filename != null) {
        textOut.close();
```

```
                System.out.println("  File '" + filename + "' writing successful.");
            }
        }
    }
```

4 Request of Reports

By using the method „getReportData“ you can retrieve data of reports per SOAP. In order to do so you have to set a row of parameters.

4.1 Parameter

The hand over of parameters to this method takes place as a Hashtable (associative array). Following parameters will be expected resp. can be set optionally:

Request
Client ⟶ Webtrekk SOAP Service

Parameter	Type	Length	Optional	Description
customerId	string	15	no	ID of wanted customer (Customer-ID)
login	string		no	Login Webtrekk Frontend
pass	string		no	Password Webtrekk Frontend
build_pic	int	1	yes	Should image data be generated? Yes: 1, no: 0, default: 0 (no)
pic_width	int		yes	Image width (minimum: 200px, maximum: 1000px), default: 300
pic_height	int		yes	Image height (minimum: 150px, maximum: 1000px), default: 200
language	string		yes	Language (‚de‘, ‚en‘, ‚fr‘, ‚es‘)
report_name	string		no	Name of wanted reports
time_start	string		yes	Start time, format: YYYY-MM-DD or YYYY-MM-DD hh:mm:ss, not applicable in Java
time_stop	string		yes	Stop time, format: YYYY-MM-DD or YYYY-MM-DD hh:mm:ss, not applicable in Java
decodeUtf8	int	1	yes	Should the Soap response be utf8 decoded? Yes: 1, No: 0, Default 0 (No)*

* By some clients, the Soap response is utf8 coded twice. By setting this parameter, the utf8 coding is rebuilt.

4.2 Return Values

The return takes place in an array. This array contains single analyses in the same order like they are listed in the reports. The structure of the single analyses is identical to the structure which is used when retrieving single analyses. This structure is described in chapter 3.2.

4.3 Code Examples (Client)

The following chapters show examples of how to retrieve reports by using the SOAP-interface in different languages.

4.3.1 Perl

```
#!/usr/bin/perl

use SOAP::Lite;
use MIME::Base64;
use strict;

# Connection parameter
my $endpoint = "http://report2.webtrekk.de/cgi-bin/wt/SOAPv3.cgi";
my $soapaction = "urn:reportSOAP#getReportData";
my $method = 'getReportData';
my $method_urn = 'urn:objects::reportSOAP';

# Configuration of requested reports
my %parameter = (
	customerId		=> '111111111111111',
	login			=> 'mylogin',
	pass			=> 'mypass',
	output_row_start	=> 0,
	limit_rows		=> 5,
	build_pic		=> 1,
	pic_width		=> 500,
	pic_height		=> 300,
	language		=> 'de',
	report_name		=> 'Technicreport',
	time_start		=> '2005-01-01',
	time_stop		=> '2005-01-31',
);

# Prepare request
my $soap = SOAP::Lite->new(
		uri => $soapaction,
		proxy => $endpoint,
		on_fault => sub {
			my($soap, $res) = @_;
			print "\nFault:\n------";
			print "\nfaultcode: ".$res->faultcode;
			print "\nfaultstring: ".$res->faultstring."\n";
			return;
		}
);

# Set SOAP-request
my $response = $soap->call(SOAP::Data->name($method)->attr(
		{ xmlns => $method_urn }
	) => (\%parameter) );
```

```
# Retrieve results
my $data = $response->result;

# Counter for number of analyses
my $count = 0;

# Peruse all analyses of report
foreach my $ana (@{$data}) {
	$count++;

	# Peruse all tables of analyses
	for (my $i = 0; $i < $ana->{count}; $i++) {

		# Determine data source – determine name add-on for retention
		my $data = $ana;			# Data source
		my $name = $count;			# Namen add-on
		if ($ana->{count} > 1) {	# If multiple tables, change data source + name
			$data = $ana->{array}->[$i];
			$name .= "_".($i+1);
		}

		# Write table data in file
		open(DATA, ">out_$name.txt");
		foreach(@{$data->{analysisData}}) {
			print DATA join("\t",@{$_})."\n";
		}
		close DATA;

		# If defined, write image in file
		if (defined $data->{analysisPic3d} && $data->{analysisPic3d} ne "") {
			open(IMG, ">pic3d_$name.png");
			binmode IMG;
			print IMG decode_base64($data->{analysisPic3d});
			close IMG;
		}
	}
}

# Retrieve success message
print "$count Analyses saved!\n";
```

4.3.2 PHP

```
<?php
require_once('nusoap.php');

// Configuration
$wsdl_path = 'http://report2.webtrekk.de/SOAPv3.wsdl';
$method = 'getReportData';
$parameter = array (
        'customerId' => '111111111111111',
        'login' => 'mylogin',
        'pass' => 'mypass',
        'output_row_start' => 0,
        'limit_rows' => 5,
        'build_pic' => 1,
        'pic_width' => 500,
        'pic_height' => 300,
        'language' => 'de',
        'report_name' => 'Technicreport',
        'time_start' => '2007-02-01',
        'time_stop' => '2007-02-28',
);

// Create nusoap Client instance and check for errors
$client = new soapclient($wsdl_path, true); // possibly new nusoap_client(…)
$err = $client->getError();
if ($err) {
        echo '<h2>Error while creating SOAP-instance </h2><pre>' . $err . '</pre>';
        exit(1);
}

// Retrieve data and check for errors
$result = $client->call($method, array ($parameter));
if ($client->fault) {
        echo '<h2>The server reports errors</h2><pre>';
        print_r($result);
        echo '</pre>';
        exit(1);
}
$err = $client->getError();
if ($err) {
        echo '<h2>Error while retrieving data</h2><pre>' . $err . '</pre>';
        exit(1);
}


// Hand out result
echo '<h2>Result</h2><pre>';
print_r($result);
echo '</pre>';
```

4.3.3 Java (Axis 1.4)

In order to use Java it is necessary to use the Java packages „Apache Axis“ (of version 1.4) as well as „Apache Common Codecs“. Please install both packages into the Java-„Lib“ folder, resp. into a folder which is listed in the system environment variable CLASSPATH (e.g. „C:\Java\Lib“). Switch into the folder by using the console and implement the following:

```
java org.apache.axis.wsdl.WSDL2Java http://report2.webtrekk.de/SOAPv3.wsdl
```

By doing this two other packages (“de.webtrekk.report2.SOAPv3_wsdl” and “xml_soap_wt”) are created, by which you get access to the SOAP-interface.

```
// RequestReportData.java
import java.io.*;
import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;
import de.webtrekk.report2.SOAPv3_wsdl.*;
import org.apache.commons.codec.binary.Base64;
import xml_soap_wt.*;

public class RequestReportData {
	/**
	 * Implements a request for the SOAP-interface
	 * @param args command row parameter (will not be analysed)
	 */
	public static void main(String[] args) {
		/*
		 *  Set input parameter
		 */
		WebtrekkReportInput input = new WebtrekkReportInput();
		input.setCustomerId("111111111111111");
		input.setLogin("mylogin");
		input.setPass("mypass");
		input.setPic_width(500);
		input.setPic_height(300);
		input.setLanguage("de");
		input.setIs_java(1);
		input.setReport_name("Technicreport");

		/*
		 *  Start requests – hand out results
		 */
		Webtrekkoutput [] results;
		try {
			AnalysisGeneratorService myService = new AnalysisGeneratorServiceLocator();
			AnalysisGeneratorPortType myPort = myService.getanalysisGeneratorPort();
			results = myPort.getReportData(input);
		} catch (ServiceException e) {
			System.err.println("Connection failed: " + e.getMessage());
			System.exit(1);
			return;
```

```
        } catch (RemoteException e) {
            System.err.println("Error during data call: " + e.getMessage());
            System.exit(1);
            return;
        }

        for (int i = 0; i < results.length; i++) {
            printAnalysis(results[i], Integer.toString(i+1));
        }
    }

    /**
     * Hands out results of an analysis
     * @param result Analysis Data (with Image Data)
     * @param filenameAdd Part of the used data name or zero,
     *                                  if the table schould be handed out on System.out
     */
    private static void printAnalysis(Webtrekkoutput result, String filenameAdd) {
        // Does data exist at all?
        if (result == zero) {
            return;
        }

        // Check all table results
        if (result.getCount() > 1) {
            for (int i = 0; i < result.getCount(); i++) {
              AnalysisData analysis = result.getArray()[i];
              // Separate data
              Object[][] analysisData = analysis.getAnalysisData();
              Object[] analysisTabHead = analysis.getAnalysisTabHead();
              Object[][] analysisTabFoot = analysis.getAnalysisTabFoot();
              String analysisPic3d = analysis.getAnalysisPic3D();
              String analysisPic2d = analysis.getAnalysisPic2D();

              // Output
              System.out.println("Schreibe: " + analysis.getAnalysisTitle());
              printTable(filenameAdd + "_" + i, analysisData,
                                  analysisTabHead, analysisTabFoot);
              printPictures(filenameAdd + "_" + i, analysisPic3d, analysisPic2d);
            }
        } else {
            // Separate data
            Object[][] analysisData = result.getAnalysisData();
            Object[] analysisTabHead = result.getAnalysisTabHead();
            Object[][] analysisTabFoot = result.getAnalysisTabFoot();
            String analysisPic3d = result.getAnalysisPic3D();
            String analysisPic2d = result.getAnalysisPic2D();

            // Output
            System.out.println("Schreibe: " + result.getAnalysisTitle());
            printTable(filenameAdd, analysisData, analysisTabHead, analysisTabFoot);
            printPictures(filenameAdd, analysisPic3d, analysisPic2d);
        }
    }
```

```
/**
 * Writes pictures in one file
 * @param filenameAdd Part of the used output file name or zero
 * @param analysisPic3d 3D Picture
 * @param analysisPic2d 2D Picture
 */
private static void printPictures(String filenameAdd,
                                  String analysisPic3d, String analysisPic2d) {
    if (filenameAdd == null) {
        filenameAdd = "";
    }
    // The 2D-Picture
    if (analysisPic2d != null && analysisPic2d.length() > 0) {
        String filename = "Pic2D_" + filenameAdd + ".png";
        byte[] bytes = analysisPic2d.getBytes();
        byte[] binary = new Base64().decode(bytes);
        try {
          FileOutputStream out = new FileOutputStream(filename);
          out.write(binary);
          out.flush();
          out.close();
          System.out.println("  AnalysisPic2D: Datei '" + filename
                                                  + "' successfully written.");
        } catch (IOException e) {
          System.err.println("  Error while saving the 2D-Picture:"
                                                  + e.getMessage());
        }
    }

    // The 3D-Picture
    if (analysisPic2d != zero && analysisPic2d.length() > 0) {
        String filename = "Pic3D_" + filenameAdd + ".png";
        byte[] bytes = analysisPic3d.getBytes();
        byte[] binary = new Base64().decode(bytes);
        try {
          FileOutputStream out = new FileOutputStream(filename);
          out.write(binary);
          out.flush();
          out.close();
          System.out.println("  AnalysisPic3D: Datei '" + filename
                                                  + "' successfully written.");
        } catch (IOException e) {
          System.err.println("  Error while saving the 3D-Picture: "
                                                  + e.getMessage());
        }
    }
}

/**
 * Hands out data in text format or writes data in a file
 * @param filenameAdd Part of the output data name or zero in order to hand out to
 *                                  System.out
 * @param analysisData The analysis data
 * @param analysisTabHead The table head, matching the analysis data
 * @param analysisTabFoot The table foot, matching the analysis data
 */
```

```
    private static void printTable(String filenameAdd,
                                   Object[][] analysisData, Object[] analysisTabHead,
                                   Object[][] analysisTabFoot) {
        if (analysisData.length == 0 || analysisData[0].length == 0) {
            System.out.println("  No data!");
            return;
        }
        // Open output file
        PrintStream textOut;
        String filename = null;
        if (filenameAdd == null) {
            textOut = System.out;
        } else {
            filename = "Analysis" + filenameAdd + ".txt";
            try {
              textOut = new PrintStream(new FileOutputStream(filename));
            } catch (IOException e) {
              System.err.println("  Can not open file: " + e.getMessage());
              return;
            }
        }

        // The table head
        if (analysisTabHead != zero) {
            for (int column = 0; column < analysisTabHead.length; column++) {
              textOut.print(analysisTabHead[column] + "\t");
            }
            textOut.println();
            textOut.println("================================================");
        }

        // The table content
        for (int row = 0; row < analysisData.length; row++) {
            for (int column = 0; column < analysisData[row].length; column++) {
              textOut.print(analysisData[row][column] + "\t");
            }
            textOut.println();
        }

        // The table foot
        if (analysisTabFoot != null) {
            textOut.println("================================================");
            for (int row = 0; row < analysisTabFoot.length; row++) {
              for (int column = 0; column < analysisTabFoot[row].length; column++) {
                        textOut.print(analysisTabFoot[row][column] + " ");
              }
              textOut.println();
            }
        }
        textOut.println();

        // Success message and close file
        if (filename != null) {
            textOut.close();
            System.out.println("  File '" + filename + "' writing successful.");
        }
    }
}
```

4.3.4 Java (Axis 2.0)

Also see chapter 3.4.4.

```
// RequestReportData.java
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintStream;
import java.rmi.RemoteException;
import org.apache.commons.codec.binary.Base64;
import de.webtrekk.ws.doc.WTAnalysisGeneratorWSStub;
import de.webtrekk.ws.doc.WTAnalysisGeneratorWSStub.*;

public class RequestReportData
{

	public static void main(String args[])
	{
		try {
			WTAnalysisGeneratorWSStub stub = new WTAnalysisGeneratorWSStub();

			WTReportInput input = new WTReportInput();
			input.setCustomerId("111111111111111");
			input.setLogin("mylogin");
			input.setPassword("mypass");
			input.setIsJava(1);
			input.setReportName("Technikreport");

			WTReportOutput result = stub.getReportData(input);
			WTOutput[] outputs  = result.getOutputs();
			if(outputs != null) {
				for(int i=0; i<outputs.length; i++)
				printAnalysis(outputs[i], null);
			}
		} catch (RemoteException e) {
			e.printStackTrace();
		}
	}

	private static void printAnalysis(WTOutput result, String filenameAdd) {
		// Does data exist at all?
		if (result == null) {
			return;
		}

		// Check all table results

		WTAnalysisData[] data = result.getData();

		for (int i = 0; i < data.length; i++)
		{
			WTAnalysisData analysis = data[i];
			// Separate data
			DataRecord[] analysisData = analysis.getAnalysisData();
			DataRecord analysisTabHead = analysis.getAnalysisTabHead();
			DataRecord[] analysisTabFoot = analysis.getAnalysisTabFoot();
```

```
            String analysisPic3d = analysis.getAnalysisPic3D();
            String analysisPic2d = analysis.getAnalysisPic2D();

            // Output
            System.out.println("Schreibe: " + analysis.getAnalysisTitle().getValue());
            printTable(null, analysisData, analysisTabHead, analysisTabFoot);
            printPictures(null, analysisPic3d, analysisPic2d);
        }
    }

    private static void printPictures(String filenameAdd, String analysisPic3d, String analysisPic2d) {
        if (filenameAdd == null) {
            filenameAdd = "";
        }
        // The 2D Picture
        if (analysisPic2d != null && analysisPic2d.length() > 0) {
            String filename = "Pic2D_" + filenameAdd + ".png";
            byte[] bytes = analysisPic2d.getBytes();
            byte[] binary = new Base64().decode(bytes);
            try {
                FileOutputStream out = new FileOutputStream(filename);
                out.write(binary);
                out.flush();
                out.close();
            } catch (IOException e) {
                System.err.println("  Error while saving the 2D picture: " + e.getMessage());
            }
        }

        // The 3D Picture
        if (analysisPic2d != null && analysisPic2d.length() > 0) {
            String filename = "Pic3D_" + filenameAdd + ".png";
            byte[] bytes = analysisPic3d.getBytes();
            byte[] binary = new Base64().decode(bytes);
            try {
                FileOutputStream out = new FileOutputStream(filename);
                out.write(binary);
                out.flush();
                out.close();
            } catch (IOException e) {
                System.err.println("  Error while saving the 2D picture: " + e.getMessage());
            }
        }
    }

    private static void printTable(String filenameAdd, DataRecord[] analysisData, DataRecord
      analysisTabHead, DataRecord[] analysisTabFoot) {
        if (analysisData.length == 0) {
            System.out.println("  No data!");
            return;
        }
        // Open output file
        PrintStream textOut;
        String filename = null;
        if (filenameAdd == null) {
            textOut = System.out;
        } else {
            filename = "Analyse" + filenameAdd + ".txt";
```

```
            try {
                textOut = new PrintStream(new FileOutputStream(filename));
            } catch (IOException e) {
                System.err.println("  Can not open file: " + e.getMessage());
                return;
            }
        }

        // The table head
        if (analysisTabHead != null) {
            for (int column = 0; column < analysisTabHead.getValues().length; column++) {
                textOut.print(analysisTabHead.getValues()[column].getValue() + "\t");
            }
            textOut.println();
            textOut.println("================================================");
        }

        // The table content
        for (int row = 0; row < analysisData.length; row++) {
            for (int column = 0; column < analysisData[row].getValues().length; column++) {
                if(analysisData[row].getValues()[column] != null)
                    textOut.print(analysisData[row].getValues()[column].getValue() + "\t");
                else
                    textOut.print("\t");
            }
            textOut.println();
        }

        // The table foot
        if (analysisTabFoot != null) {
            textOut.println("================================================");
            for (int row = 0; row < analysisTabFoot.length; row++) {
                for (int column = 0; column < analysisTabFoot[row].getValues().length; column++) {
                    if(analysisTabFoot[row].getValues()[column] != null)
                        textOut.print(analysisTabFoot[row].getValues()[spalte].getValue() + " ");
                    else
                        textOut.print("\t");
                }
                textOut.println();
            }
        }
        textOut.println();

        // Success message and close file
        if (filename != null) {
            textOut.close();
        }
    }
}
```

5 Data Export

You can export data per SOAP by using the method „exportData“.

5.1 Parameters

The hand over of parameters to this method takes place as a Hashtable (associative array). Following parameters are thereby expected:

Request
Client ⟶ Webtrekk SOAP Service

Parameter	Type	Length	Optional	Description
customerId	string	15	no	ID of wanted customer (Customer-ID)
Login	string		no	Login Webtrekk Frontend
Pass	string		no	Password Webtrekk Frontend
startRow	int		no	Initial line
endRow	int		no	End line
Type	string		No	Possible values: content_categories, content_categories_time, customer_categories (CRM Categories), page_urls, media_categories, media_playtime, basket_categories, basket_categories_time, product_urls, campaigns, ecommerce_parameters, time_categories
startDate	string		Yes	Format YYYY-MM-DD Mandatory for parameter export and timedependent categories.
endDate	string		yes	Format YYYY-MM-DD Mandatory for parameter export and timedependent categories.

title	string		yes	Title of own parameter Mandatory for parameter export
rowsWithoutData	int		yes	Possible values: 0,1 Export only rows without data Standard: 0
filter	string		ja	Filters the results. Only supported for Content, Product and Media Categories. The filtered column is Pages, Products or Media.
allDataSources	int	1	yes	****DEPRECATED**** Instead, please use the dataSourceTypeFilter with the value "secondary". Only for campaign categories. Should all data sources be exported? Default data source of type Search Engine, Referrer and Direct entry are not included in the export. Yes: 1, No: 0 (Default) Data from interfaces can only be exported separately (see 'interfaceOnly').
interfaceOnly	int	1	yes	****DEPRECATED**** Instead, please use the dataSourceTypeFilter with the value "interfaces". Only for campaign categories. Should data from interfaces be exported (for example, Google AdWords API)? Yes: 1, No: 0 (Default) Data for other data sources must be exported separately (see 'allDataSources').
decodeUtf8	int	1	yes	Should the Soap response be utf8 decoded? Yes: 1, No: 0, Default 0 (No)*
activeCampaigns	Int	1	yes	Should only active campaigns be exported? Valid only for the type "campaign". Yes: 1, No: 0, Default 0 (No)
dataSourceTypeFilter	string		yes	Only for campaign categories. Filtering of the data source type. Permitted values: "all": all data source types "interfaces": only advertising media of interfaces (e.g. Google AdWords), "regular": all regular advertising media without interfaces, "secondary": all regular and secondary advertising media (secondary advertising media have one of the data source type social media sources, search engine, direct input, or other sources). Default "regular"

recordIntervalBegin	string	1	yes	Format YYYY-MM-DD. If set then also recordIntervalEnd must be set.
recordIntervalEnd	string	1	yes	Format YYYY-MM-DD. If set then also recordIntervalBegin must be set.
campaignsStartTime IntervalBegin	string	1	yes	Only those objects are exported the start times of which are after this value. Format YYYY-MM-DD. If set also campaignsStartTimeIntervalEnd must be set. Only supported for campaign categories.
campaignsStartTime IntervalEnd	string	1	yes	Only those objects are exported the start times of which are before this value. Format YYYY-MM-DD. If set also campaignsStartTimeIntervalBegin must be set. Only supported for campaign categories.
campaignsStopTime IntervalBegin	string	1	yes	Only those objects are exported the stop times of which are after this value. Format YYYY-MM-DD. If set also campaignsStopTimeIntervalEnd must be set. Only supported for campaign categories.
campaignsStopTime IntervalEnd	string	1	yes	Only those objects are exported the stop times of which are before this value. Format YYYY-MM-DD. If set also campaignsStopTimeIntervalBegin must be set. Only supported for campaign categories.

* By some clients, the Soap response is utf8 coded twice. By setting this parameter, the utf8 coding is rebuilt.

Parameters that are marked ****DEPRECATED**** should be used anymore because in future versions they are not supported anymore.

The limit of requested rows is 10.000 by default.

5.2 Return values

As a return value SOAP is sending a two-dimensional array of the type string, provided everything processed correctly. If an error occurs it's going to be returned as Server.ExecError.

5.3 Code Examples (Client)

The following chapters show examples of how to retrieve reports by using the SOAP-interface in different languages.

5.3.1 Perl

```
#!/usr/bin/perl

use SOAP::Lite;

my $endpoint   = "http://report2.webtrekk.de/cgi-bin/wt/SOAPv3.cgi";
my $soapaction = "urn:reportSOAP#exportData";
my $method     = 'exportData';
my $method_urn = 'urn:objects::reportSOAP';

my %parameter = (
      customerId          => '111111111111111',
      login               => 'mylogin',
      pass                => 'mypass',
      startRow            => 1,
      endRow              => 1000,
      type                => 'campaigns',
);

my $soap = SOAP::Lite->new(uri => $soapaction, proxy => $endpoint);
my $response = $soap->call(SOAP::Data->name($method)->attr(
         { xmlns => $method_urn }
      ) => (\%parameter) );

my $data = $response->result;

# print data
use Data::Dumper;
print Dumper($data);
```

5.3.2 PHP

```
<?php
require_once ('nusoap.php');

// Configuration
$wsdl_path = 'http://report2.webtrekk.de/SOAPv3.wsdl';
$method = 'exportData';
$parameter = array (
	customerId		=> '111111111111111',
	login			=> 'mylogin',
	pass			=> 'mypass',
	startRow		=> 1,
	endRow			=> 1000,
	type			=> 'campaigns',
);

// Create nusoap Client instance and check for errors
$client = new soapclient($wsdl_path, true); // possibly new nusoap_client(…)
$err = $client->getError();
if ($err) {
	echo '<h2>Error while creating SOAP-instance</h2>';
	echo '<pre>' . $err . '</pre>';
	exit(1);
}

// Request data and check for errors
$result = $client->call($method, array ($parameter));
if ($client->fault) {
	echo '<h2>Server reports error</h2><pre>';
	print_r($result);
	echo '</pre>';
	exit(1);
}
if ($client->getError()) {
	echo '<h2>Error while data request</h2><pre>' . $err . '</pre>';
	exit(1);
}

// Hand out result
echo '<h2>Ergebnis</h2><pre>';
print_r($result);
echo '</pre>';
?>
```

5.3.3 Java (Axis 1.4)

In order to use Java it is necessary to use the Java packages „Apache Axis“ (of version 1.4) as well as „Apache Common Codecs“. Please install both packages into the Java-„Lib“ folder, resp. into a folder which is listed in the system environment variable CLASSPATH (e.g. „C:\Java\Lib“). Switch into the folder by using the console and implement the following:

```
java org.apache.axis.wsdl.WSDL2Java http://report2.webtrekk.de/SOAPv3.wsdl
```

By doing that two other packages (“de.webtrekk.report2.SOAPv3_wsdl” and “xml_soap_wt”) are created, by which you get access tot he SOAP-interface.

```
// RequestDataExport.java
import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;
import de.webtrekk.report2.SOAPv3_wsdl.AnalysisGeneratorPortType;
import de.webtrekk.report2.SOAPv3_wsdl.AnalysisGeneratorService;
import de.webtrekk.report2.SOAPv3_wsdl.AnalysisGeneratorServiceLocator;
import xml_soap_wt.WebtrekkExportInput;

public class RequestDataExport {

 /**
       * Implements a request for the SOAP-interface
       * @param args command row parameter (will not be analysed)
 */
 public static void main(String[] args) {
  /* Set input-parameter */
  WebtrekkExportInput dataExport = new WebtrekkExportInput();
  dataExport.setCustomerId("111111111111111");
  dataExport.setLogin("mylogin");
  dataExport.setPass("mypass");
  dataExport.setStartRow(1);
  dataExport.setEndRow(1000);
  dataExport.setType("ecommerce_parameters");

  /* Start requests – hand out results */
  String [][] result;
  try {
   AnalysisGeneratorService myService = new AnalysisGeneratorServiceLocator();
   AnalysisGeneratorPortType myPort = myService.getanalysisGeneratorPort();
   result = myPort.exportData(dataExport);
  } catch (ServiceException e) {
   System.err.println("Connection failed: " + e.getMessage());
   System.exit(1);
   return;
  } catch (RemoteException e) {
   System.err.println("Error while data request: " + e.getMessage());
   System.exit(1);
   return;
  }
```

```
    // Output
    for (int zeile = 0; row < result.length; row++) {
      for (int column = 0; column < result[row].length; column++) {
        System.out.print(result[row][column] + "  ");
      }
      System.out.print("\n");
    }
  }
}
```

5.3.4 Java (Axis 2.0)

Also see chapter 3.4.4.

```
// RequestDataExport.java
import java.rmi.RemoteException;
import de.webtrekk.ws.doc.WTAnalysisGeneratorWSStub;
import de.webtrekk.ws.doc.WTAnalysisGeneratorWSStub.*;

public class RequestDataExport
{

	public static void main(String args[])
	{
		try {
			WTAnalysisGeneratorWSStub stub = new WTAnalysisGeneratorWSStub();

			WTExportInput input = new WTExportInput();
			input.setCustomerId("111111111111111");
			input.setLogin("mylogin");
			input.setPassword("mypass");
			input.setType("ecommerce_parameters");
			input.setStartRow(1);
			input.setEndRow(1000);

			WTExportOutput output = stub.exportData(input);
			DataRecord[] records  = output.getData();
			for(DataRecord record : records)
			{
				DataWrapper data[] = record.getValues();
				for(int i=0; i<data.length; i++)
					System.out.print(data[i].getValue() + "\t");
				System.out.println();
			}
		} catch (RemoteException e) {
			e.printStackTrace();
		}
	}
}
```

6 Data Import

You can import data per SOAP by using the method „importData“.

6.1 Parameter

The hand over of parameters to this method takes place as a Hashtable (associative array). Following parameters are thereby expected:

Request
Client → Webtrekk SOAP Service

Parameter	Type	Length	Optional	Description
customerId	string	15	no	ID of wanted customer (Customer-ID)
Login	string		no	Login Webtrekk Frontend
Pass	string		no	Password Webtrekk Frontend
uploadData	Array		no	Two-dimensional array
uploadType	String		no	Import type, possible values: content_categories (including Page URLs), content_categories_time, customer_categories (CRM Categories), media_categories (including Media Playtime) basket_categories (including Product URLs) basket_categories_time, campaigns, ecommerce_parameters time_categories session_parameter_tv

6.1.1 Special Parameters

6.1.1.1 Parameter “uploadData”

The parameter “uploadData” is handed over as a two-dimensional array. The first row of the array contains the headlines of data, all other rows the data itself. The decision which data will be imported is based on the headlines. All data which can be imported, can also be exported (see point 5 Data Export).

Example:

```
[
    ['Pages','Category (Text) – Main category', 'Category (Number) – ownC.'],
    ['index','archive','1'],
    ['home','archive','2']
]
```

The Limit for importable rows is 10.000 by default.

6.1.1.1.1 Upload Type – Session_Paramter_TV

The upload type Session_parameter_tv allows a backdated import of TV or radio spots.

Region and Flagrate are optional.
The Flagrate represents the percentage of flagged users.

Example: Session Parameter TV

```
[
    ['Spotname','BroadcastTime','Region','Flagrate'],
    ['SpotXY','2013-03-12 10:12:00','Berlin','2']
]
```

6.2 Return Values

As a return value SOAP is sending the string „upload successful“, provided everything processed correctly. If an error occurs it’s going to be returned as Server.ExecError.

6.3 Code Examples (Client)

The following chapters show examples of how to retrieve reports by using the SOAP-interface in different languages.

6.3.1 Perl

```
#!/usr/bin/perl

use SOAP::Lite;
use MIME::Base64;
use strict;

# Connection parameter
my $endpoint = "http://report2.webtrekk.de/cgi-bin/wt/SOAPv3.cgi";
my $soapaction = "urn:reportSOAP#importData";
my $method = 'importData';
my $method_urn = 'urn:objects::reportSOAP';

# Configuration of requested report
my %parameter = (
	customerId	=> '111111111111111',
	login		=> 'mylogin',
	pass		=> 'mypass',
	uploadType	=> 'content_categories',
	uploadData => [
		['Pages','Category (Text) – Main Category', 'Category (Number) – ownC.'],
		['index','archive','1'],
		['home','archive','2']
	]);

# Prepare request
my $soap = SOAP::Lite->new(
	uri => $soapaction,
	proxy => $endpoint,
	on_fault => sub {
		my($soap, $res) = @_;
		print "\nFault:\n------";
		print "\nfaultcode: ".$res->faultcode;
		print "\nfaultstring: ".$res->faultstring."\n";
		return;
	}
);

# Start SOAP-request
my $response = $soap->call(SOAP::Data->name($method)->attr(
		{ xmlns => $method_urn }
	) => (\%parameter) );

# Retrieve results
my $data = $response->result;
```

6.3.2 PHP

```
<?php
require_once('nusoap.php');

// Configuration
$wsdl_path = 'http://report2.webtrekk.de/SOAPv3.wsdl';
$method = 'getReportData';
$parameter = array (
        'customerId' => '111111111111111',
        'login' => 'mylogin',
        'pass' => 'mypass',
        uploadType      => 'content_categories',
        uploadData => array (
             array('Pages','Category (Text) – Main Category', 'Category (Number) – ownC.'),
             array('index','archive','1'),
             array('home','archive','2'),
        )
);

// Create nusoap Client instance and check for errors
$client = new soapclient($wsdl_path, true); // possibly new nusoap_client(…)
$err = $client->getError();
if ($err) {
        echo '<h2>Error while creating SOAP-instance</h2><pre>' . $err . '</pre>';
        exit(1);
}

// Retrieve data and check for errors
$result = $client->call($method, array ($parameter));
if ($client->fault) {
        echo '<h2>Server reports errors</h2><pre>';
        print_r($result);
        echo '</pre>';
        exit(1);
}
$err = $client->getError();
if ($err) {
        echo '<h2>Error while retrieving data</h2><pre>' . $err . '</pre>';
        exit(1);
}

// Hand out results
echo '<h2>Ergebnis</h2><pre>';
print_r($result);
echo '</pre>';
```

6.3.3 Java (Axis 1.4)

In order to use Java it is necessary to use the Java packages „Apache Axis“ (of version 1.4) as well as „Apache Common Codecs“. Please install both packages into the Java-„Lib“ folder, resp. into a folder which is listed in the system environment variable CLASSPATH (e.g. „C:\Java\Lib“). Switch into the folder by using the console and implement the following:

```
java org.apache.axis.wsdl.WSDL2Java http://report2.webtrekk.de/SOAPv3.wsdl
```

By doing that two other packages (“de.webtrekk.report2.SOAPv3_wsdl” and “xml_soap_wt”) are created, by which you get access tot he SOAP-interface.

```
// RequestDataImport.java
import java.rmi.RemoteException;
import javax.xml.rpc.ServiceException;
import de.webtrekk.report2.SOAPv3_wsdl.AnalysisGeneratorPortType;
import de.webtrekk.report2.SOAPv3_wsdl.AnalysisGeneratorService;
import de.webtrekk.report2.SOAPv3_wsdl.AnalysisGeneratorServiceLocator;
import xml_soap_wt.WebtrekkImportInput;

public class RequestDataImport {

 /**
       * Implements a request for the SOAP-interface
       * @param args command row parameter (will not be analysed)
 */
 public static void main(String[] args) {
  /*
   *  Set input parameter
   */
  WebtrekkImportInput dataImport = new WebtrekkImportInput();
  dataImport.setCustomerId("111111111111111");
  dataImport.setLogin("mylogin");
  dataImport.setPass("mypass");
  dataImport.setUploadType("content_categories");

  Object [][] data = new Object[3][3];
  data[0][0] = "Pages";
  data[0][1] = "Category (Text) – Main category";
  data[0][2] = "Category (Number) – ownC.";
  data[1][0] = "index";
  data[1][1] = "archive";
  data[1][2] = "1";
  data[2][0] = "home";
  data[2][1] = "archive";
  data[2][2] = "2";
  dataImport.setUploadData(data);

  /*
   *  Start request – Retrieve results
   */
  String result;
```

```
    try {
      AnalysisGeneratorService myService = new AnalysisGeneratorServiceLocator();
      AnalysisGeneratorPortType myPort = myService.getanalysisGeneratorPort();
      result = myPort.importData(dataImport);
    } catch (ServiceException e) {
      System.err.println("Connection failed: " + e.getMessage());
      System.exit(1);
      return;
    } catch (RemoteException e) {
      System.err.println("Error while retrieving data: " + e.getMessage());
      System.exit(1);
      return;
    }
    System.out.println(result);
  }
}
```

6.3.4 Java (Axis 2.0)

Also see chapter 3.4.4.

```
// RequestDataImport.java
import java.rmi.RemoteException;
import de.webtrekk.ws.doc.WTAnalysisGeneratorWSStub;
import de.webtrekk.ws.doc.WTAnalysisGeneratorWSStub.*;

public class RequestDataImport
{

    public static void main(String args[])
    {
        try {
            WTAnalysisGeneratorWSStub stub = new WTAnalysisGeneratorWSStub();

            WTImportInput input = new WTImportInput();
            input.setCustomerId("111111111111111");
            input.setLogin("mylogin");
            input.setPassword("mypass");
            input.setUploadType("content_categories");

            DataRecord[] data = new DataRecord[3];
            data[0] = new DataRecord();
            data[0].setValues(
                createArrayOfDataWrapper(new Object[]{
                    "Pages",
                    "Categorie (Text) – Main category",
                    "Categorie (Number) – ownC."
                })
            );
            data[1] = new DataRecord();
            data[1].setValues(
                createArrayOfDataWrapper(new Object[]{
                    "index",
                    "archiv",
                    "1"
                }
            ));
            data[2] = new DataRecord();
            data[2].setValues(
                createArrayOfDataWrapper(new Object[]{
                    "home",
                    "archiv",
                    "2"
                }
            ));
            input.setUploadData(data);

            WTImportOutput output = stub.importData(input);
            System.out.println(output.getResult());
        } catch (RemoteException e) {
            e.printStackTrace();
        }
```

```
}

private static DataWrapper[] createArrayOfDataWrapper(Object[] data)
{
    int n = data.length;
    DataWrapper[] retVal = new DataWrapper[n];

    for(int i=0; i<n; i++)
    {
        DataWrapper dWrapper = new DataWrapper();
        dWrapper.setValue(data[i].toString());
        if(data[i] != null)
            dWrapper.setType(data[i].getClass().getSimpleName());
        retVal[i] = dWrapper;
    }
    return retVal;
}
```

7 Contact

If you have any questions please feel free to contact us:

Webtrekk GmbH
Robert-Koch-Platz 4
10115 Berlin
Germany

fon +49 (0)30 - 755 415 - 0
fax +49 (0)30 - 755 415 - 100
info@webtrekk.com

http://www.webtrekk.com