# Instruction Manual for the Qualified Timestamping Service of the BEV

**Version 1.4**

**March 2009**

BEV - Bundesamt für Eich- und Vermessungswesen

# 1 Document Information

## 1.1 Purpose and Validity

This document contains the user guide for the Qualified Timestamping Service of the BEV and is published on the BEV website.

## 1.2 Document History

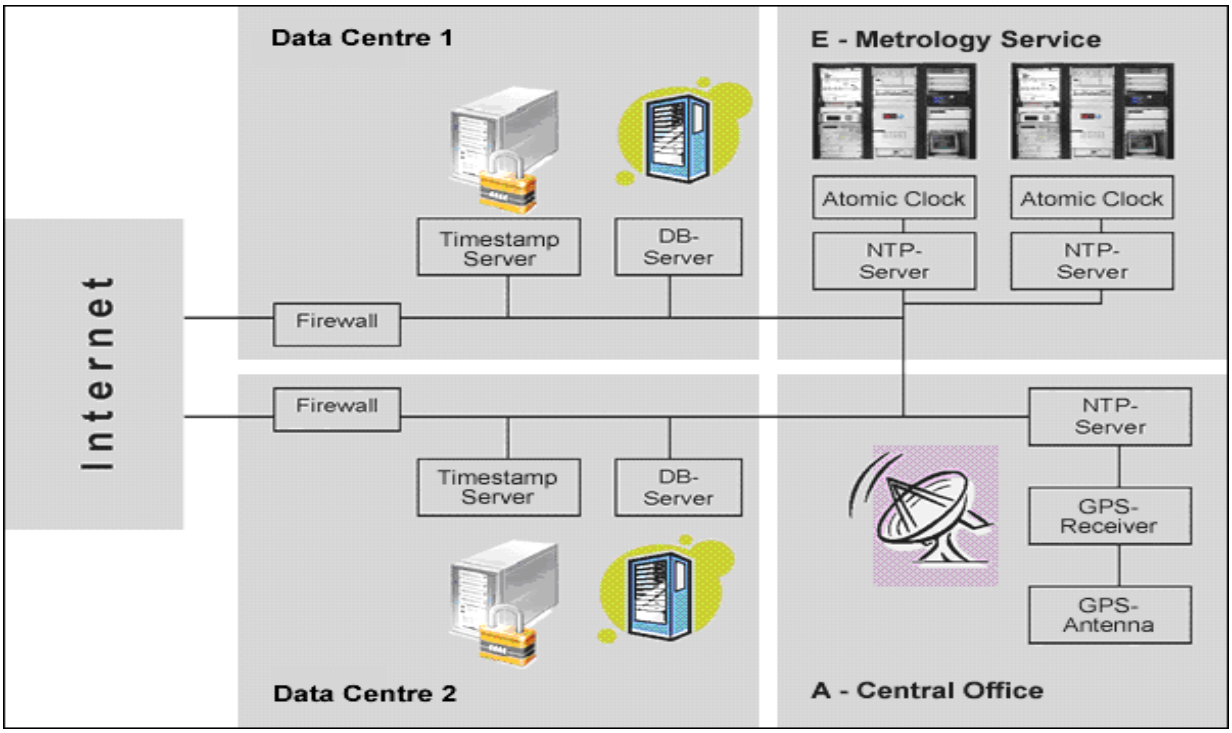
| Version | Date | Reason for change in this document |
|---|---|---|
| 1.0 | 19.01.2007 | Version for starting the test run |
| 1.1 | 16.02.2007 | Version for starting the free of charge run for public administration, free of charge test version for private users. Simplification of the graphic page 1. Correction page 3: output 10 timestamps/second. Information about availability deleted, is in the policy document. |
| 1.2 | 28.03.2007 | Page 4: Hyperlink to the certificate inserted. |
| 1.3 | 18.06.2007 | Page 2: Table 1 TimeStampReq added. |
| 1.4 | 03.2009 | Change of the name of the service.<br>Phasing out of the free of charge run. Adjustment of the graphic on page 4.<br>Adjustment of page 7 of the document – Glossary: Amendment to the Signaturgesetz BGBl I Nr. 8/2008 and BGBl I Nr. 59/2008 as well as of the Signaturverordnung BGBl II Nr. 3/2008. Hyperlinks to these legal norms.<br>Change of contact information. |

# 2 Content

**1 Document Information ..... 2**
1.1 Purpose and Validity ..... 2
1.2 Document History ..... 2
**2 Content ..... 3**
**3 General Aspects ..... 4**
3.1 Time Information ..... 4
3.2 Storage of the Records about the Services of the QZSD ..... 4
3.3 Request/Response Format ..... 5
3.4 Transport Protocol ..... 6
3.5 Response Time/Output ..... 6
**4 HTTP ..... 6**
**5 Contact Information ..... 6**
**6 References ..... 7**
**7 Glossary ..... 7**
**8 Appendix ..... 8**

# 3 General Aspects

Since the 2nd of January 2007 the BEV has established a Qualified Timestamping Service (QZSD). This service is operated and publicly provided under the Policy OID [1.2.40.0.10.1.8.1]. It is possible to install a free of charge test version.




Schematic Presentation

## 3.1 Time Information

The time information is provided by the atomic clocks of the BEV via an NTP server.
The accuracy of the atomic clock is $5 \cdot 10^{-13}$ s.
The accuracy of the NTP server is 20 ms.
The time information of the timestamp is given with an accuracy of 1 second.
If the atomic clocks respectively the NTP server do not work this has no influence on the QZSD, as then the time information is taken from a GPS receiver.

## 3.2 Storage of the Records about the Services of the QZSD

The QZSD records log entries; they are stored for 3 years.
The following data are logged into an Oracle db:

- time/date
- IP address of the sender
- incoming request and returned response
- synchronisation events of the NTP clients

## 3.3 Request/Response Format

The QZSD implements the timestamp functionality in accordance with [RFC 3161] and the specified profile of the ETSI [ETSI TS 101 861].
The QZSD accepts requests with the following formats:

| **Format of TimeStampReq** | |
|---|---|
| **Field** | **Value** |
| version | V1 (1) |
| messageImprint/ hashAlgorithm | SHA-1, SHA-256, SHA-384, SHA-512, RIPEMD-160 |
| messageImprint/ hashedMessage | Hash value for the timestamp |
| reqPolicy | s. [RFC 3161] |
| Nonce | s. [RFC 3161] |
| certReq | s. [RFC 3161] |

Table 1 TimeStampReq

Any extensions contained in the TimeStampReq will not be evaluated and an error code will be returned. No timestamp will be generated (see RFC 3161). The QZSD server returns responses (TimeStampResp) in the format in accordance with [RFC 3161].
The structure TSTInfo which is included in the TimeStampResp and signed by the QZSD corresponds with the following format:

| **Format of TSTInfo** | |
|---|---|
| **Field** | **Value** |
| version | V1 (1) |
| policy | [OPEN] |
| messageImprint | Value of Request |
| serialNumber | Value of a SEQUENCE in the Oracle Cluster staring with 1 (up to 160 Bit) |
| genTime | Time format YYYYMMDDhhmmss.sssZ |
| accuracy | millis=500 |
| ordering | False |
| Nonce | Value of Request |
| Tsa | CN=“Sicherer Zeitstempeldienst“<br>OU=“Bundesamt für Eich- und Vermessungswesen“<br>O=“Republik Österreich“<br>C=“AT“ |

Table 2 TSTInfo

| **Format of CMS signature** | |
|---|---|
| Signature Algorithm | SHA-1 with RSA |

Table 3 CMS signature

### 3.4 Transport Protocol

HTTP without server/client authentication is used as transport protocol in accordance with [RFC3161].
Notwithstanding [ETSI TS 101 861] paragraph 6 **no** *store and forward* protocol is realised.

### 3.5 Response Time/Output

It is possible to create a minimum of 10 timestamps/second.

# 4 HTTP

For the certificate please see:
http://www.signatur.rtr.at/SearchCertificatesServlet?DN=O%3DBundesamt+f%FCr+Eich-+und+Vermessungswesen%2C+C%3DAT&CA=alle&validFrom=&validTo=&serialNumberFrom=&serialNumberTo=&revokedDateFrom=&revokedDateTo=&status=alle&Submit=Suchen&locale=de

For the QZSD service please see:
http://szsd.bev.gv.at/szsd/service

Please see a JAVA sample program in the appendix.

# 5 Contact Information

For technical requests and concrete service requests please contact:

**BEV – Kundenservice**

Tel.: +43 1 211 10-2160
Fax: +43 1 211 10-2162
E-Mail: kundenservice@bev.gv.at

For activation you need the following data:

- IP address of the request client (e.g.: 192.168.0.34)
- client operating system

For general and commercial requests please contact:

**BEV – Physikalisch-technischer Prüfdienst (PTP)**
Tel.: +43 1 211 10-6327
Fax: +43 1 211 10-6000
E-Mail: ptp@bev.gv.at

# 6 References

| Reference | Titel | Author | Date |
|---|---|---|---|
| [ETSI TS 101 861] | Time Stamping Profile, V1.2.1; http://webapp.etsi.org/WorkProgram/Report_WorkItem.asp?WKI_ID=14947 | ETSI | 03.2002 |
| [ETSI TS 102 023] | Electronic Signatures and Infrastructures (ESI): Policy requirements for time-stamping authorities, V1.2.1; http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=17661 | ETSI | 01.2003 |
| [PVP] | Spezifikation Portal Verbund Protokoll, V1.5.3; http://zmr.bmi.gv.at/files/PortalVerbundProtokollV1.5.pdf | BMI | 04.2002 |
| [RFC 2030] | Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI; http://www.ietf.org/rfc/rfc1305.txt | IETF | 10.1996 |
| [RFC 2246] | The TLS Protocol – Version 1.0; http://www.ietf.org/rfc/rfc2246.txt | IETF | 01.1999 |
| [RFC 3161] | Internet X.509 Public Key Infrastructure – Time-Stamp Protocol; http://www.ietf.org/rfc/rfc3161.txt | IETF | 08.2001 |
| [RFC 3628] | Policy Requirements for Time-Stamping Authorities (TSAs); http://www.ietf.org/rfc/rfc3628.txt | IETF | 11.2003 |
| [ZERT] | X.509 Zertifikatserweiterungen für die Verwaltung, V 1.0.2; http://www.cio.gv.at/it-infrastructure/pki/ X509ext-20030218.pdf | CIO | 02.2003 |

# 7 Glossary

| Abbreviation | Name | Description |
|---|---|---|
| **BEV** | Bundesamt für Eich- und Vermessungswesen | Federal authority responsible for surveying and geoinformation and metrology, subordinate authority of the Federal Ministry for Economy, Family and Youth. |
| **HSM** | Hardware Security Module | A separate hardware componenent, used for secure generation and management of cryptographic keys. (applied in phase 2) |
| **NTP** | Network Time Protocol | IP protocol for the transmission of time information. |
| **PVP** | Portal Verbund Protokoll | The protocol defines the communication between the portals within the portal network of the Austrian administration. |
| **SigG** | Signaturgesetz | Federal Electronic Signature Law Bundesgesetz über elektronische Signaturen (Signaturgesetz - SigG), BGBl. I Nr. 190/1999 in der gültigen Fassung |
| **SigV** | Signaturverordnung | Ordinance of the Federal Chancellor on Electronic Signatures Verordnung des Bundeskanzlers über elektronische Signaturen (Signaturverordnung 2008 – SigV 2008), BGBl. II Nr. 3/2008 in der gültigen Fassung |

| | | |
|---|---|---|
| **QZSD** | Qualifizierter Zeitstempeldienst | A service that combines data with reliable time information by means of cryptographic methods. |

# 8 Appendix

IAIK-Library, please contact:

Mario Ivkovic,
IAIK – Graz University of Technology
Inffeldgasse 16a, 8010 Graz, Austria
Tel: +43 316 873 5528
http://jce.iaik.tugraz.at

SampleClient.java

```
 package demo;

import iaik.asn1.ObjectID;
import iaik.asn1.structures.AlgorithmID;
import iaik.asn1.structures.GeneralName;
import iaik.asn1.structures.GeneralNames;
import iaik.tsp.MessageImprint;
import iaik.tsp.PKIStatus;
import iaik.tsp.TSTInfo;
import iaik.tsp.TimeStampReq;
import iaik.tsp.TimeStampResp;
import iaik.tsp.TimeStampToken;
import iaik.tsp.transport.http.TspHttpConstants;

import java.io.ByteArrayOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.math.BigInteger;
import java.net.HttpURLConnection;
import java.net.URL;

import sun.misc.BASE64Encoder;

/**
 * This class demonstrates the use of the IAIK TSP 2.0 library for
 * generating a time-stamp by using the BEV QZSD time-stamping service.
 *
 * For executing this demo, the IAIK TSP 2.0 and all the JARs it depends
 * upon (IAIK-JCE and IAIK-CMS) must be on the classpath.
 */
public class SampleClient {
    private static final String SERVICE_URL = "http://szsd.bev.gv.at/service";
    //private static final String SERVICE_URL = "http://praktiker:8888/szsd/service";
    //private static final String USER = "Praktiker";
    //private static final String PASSWORD = "xYlCB35";
    private static final String POLICY_ID = "1.2.40.0.10.1.8.1";
    private static final byte[] DATA_TO_HASH = "sample data".getBytes();
    private static final String ISSUER =
        "CN=Test CA - Zeitstempeldienste,OU=Technik und Standards,O=Stabstelle IKT-Strategie des Bundes,C=AT";
    private static final BigInteger SERIAL = new BigInteger("1");


    public static void main(String[] args) throws Exception {

        // build the TimeStampReq
        byte[] hashedData =
            MessageImprint.calculateHash(DATA_TO_HASH, AlgorithmID.sha1);
        MessageImprint msg = new MessageImprint(AlgorithmID.sha1, hashedData);
        TimeStampReq req = new TimeStampReq(msg);
        req.setCertReq(false);
        req.setNonce(new BigInteger("12345678")); // should be random
        req.setTSAPolicyID(new ObjectID(POLICY_ID));

        // set up an HTTP connection for sending the request
```

```
URL url = new URL(SERVICE_URL);
HttpURLConnection connection = (HttpURLConnection) url.openConnection();
OutputStream out;
InputStream in;

connection.setRequestMethod("POST");
connection.setDoOutput(true);
connection.addRequestProperty("Content-Type",
        TspHttpConstants.CONTENT_TYPE_REQUEST);
if (USER != null && PASSWORD != null) {
    // add HTTP authentication
    String encodedAuth = new BASE64Encoder().encode((USER + ":" + PASSWORD).getBytes());
    connection.setRequestProperty("Authorization", "Basic " + encodedAuth);
}

// write request
out = connection.getOutputStream();
out.write(req.getEncoded());
out.flush();

// read response
ByteArrayOutputStream responseBytes = new ByteArrayOutputStream();
byte[] buf = new byte[1024];
int bytesRead;
TimeStampResp resp;

in = connection.getInputStream();
while ((bytesRead = in.read(buf)) > 0) {
    responseBytes.write(buf, 0, bytesRead);
}
responseBytes.flush();
resp = new TimeStampResp(responseBytes.toByteArray());

// check for correct HTTP response type
if (!TspHttpConstants.CONTENT_TYPE_REPLY.equals(connection.getContentType())) {
    System.err.println("Wrong Content-Type in response");
    System.exit(1);
}

// check for errors or bad data in response
PKIStatus pkiStatus = resp.getPKIStatusInfo().getPKIStatus();

if (pkiStatus.getStatus() != PKIStatus.GRANTED) {
    System.err.println("PKIStatus not GRANTED");
    System.exit(1);
}

TimeStampToken token = resp.getTimeStampToken();
TSTInfo tstInfo = token.getTSTInfo();


if (!token.getTSTInfo().getNonce().equals(req.getNonce())) {
    System.err.println("Nonce does not match");
    System.exit(1);
}

if (!tstInfo.getTSAPolicyID().getName().equals(POLICY_ID)) {
    System.err.println("Wrong policy ID");
    System.exit(1);
}

if (!tstInfo.getMessageImprint().equals(req.getMessageImprint())) {
    System.err.println("Message Imprint does not match");
    System.exit(1);
}

GeneralNames names = token.getCertID().getIssuerSerial().getIssuer();
GeneralName name = (GeneralName) names.getNames().nextElement();

if (!name.getName().toString().equals(ISSUER)) {
    System.err.println("Issuer does not match");
    System.exit(1);
}

if (!token.getCertID().getIssuerSerial().getSerialNumber().equals(SERIAL)) {
```

```
            System.err.println("Serial number does not match");
            System.exit(1);
        }

        System.out.println("Sent TimeStampReq OK, Got TimeStampResp OK");
    }

}
```