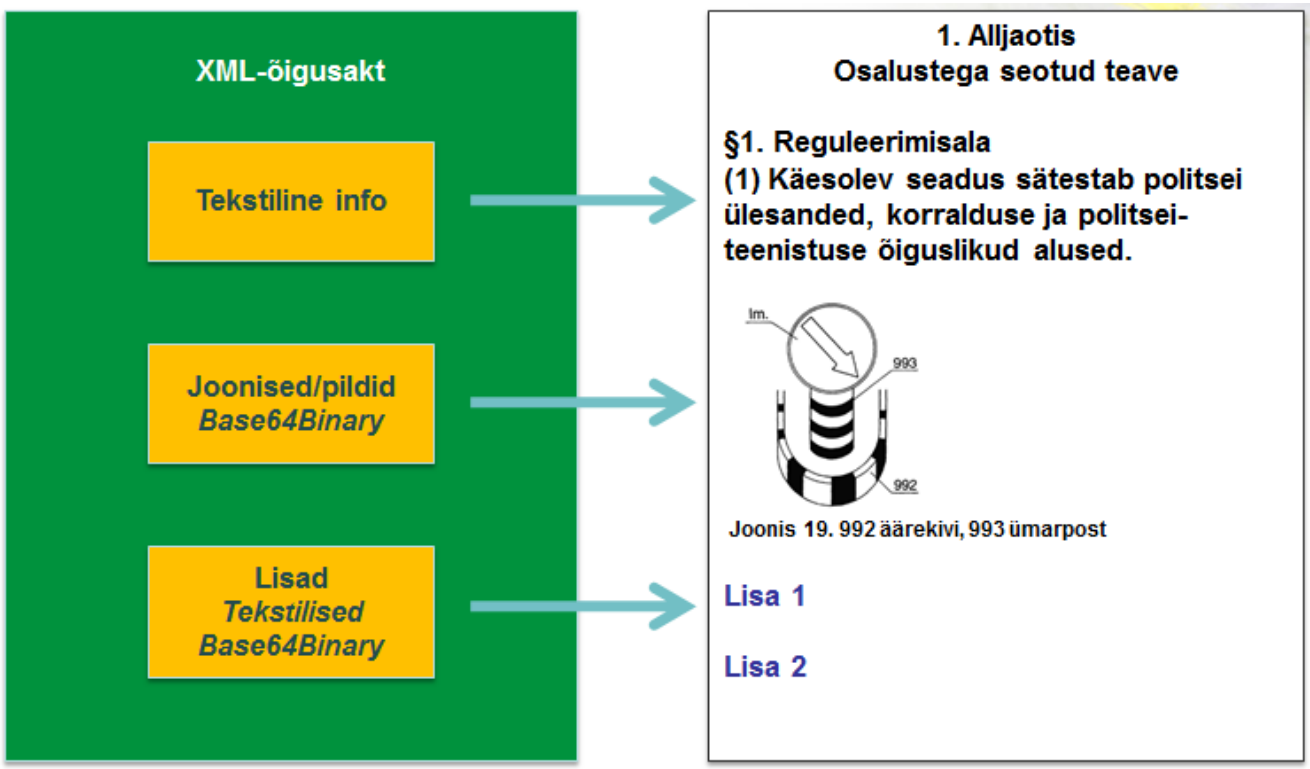
Stiilifailide kasutamise juhend

Versioon	Autor	Kuupäev	Kommentaar
1.1	Kaarel Kuddu	15.12.2010	Algversioon

1 Üldine kirjeldus

XML akti osad



XML akt koosneb tavalisest tekstist ja Base64Binary formaadis kokku pakitud piltidest ja lisadest. Teksti välja kuvamise jaoks ei ole vaja midagi lisaks teha kuid Base64Binary formaadis olevad koostisosad on vaja enne eraldi töödelda ja maha salvestada.

Webmedia Tallinn | Lõõtsa St 6C, Tallinn 11415 | Phone: +372 6 101 990 Fax: +372 6 101 991
Webmedia Tartu | Raatuse St. 20, Tartu 51009 | Phone: +372 7 309 399 Fax: +372 7 309 398
Webmedia Narva | Kerese str. 3/Kosmonaudi St. 4, Narva 20309 | Phone: +372 7 309 399 Fax: +372 7 309 399
| *www.webmedia.ee* | E-mail: *webmedia@webmedia.ee*

2 TEHNILINE KIRJELDUS

2.1 XSL TRANSFORMATSIOON JA BASE64 DEKODEEERIMINE

- Kõigepealt peab iga süsteemi jaoks kirjutama funktsionaalsuse, mis eraldaks XML failist pildid ja binaarsed lisad ning dekodeeriks base64 kodeeringu.
  - Sõltuvalt kasutatavast tehnoloogiast ja infosüsteemi vajadustest võib see erineda. Kasutades tehnoloogiat XmlBeans, mis on kasutusel eRT-s, tuleb eelnevalt skeemi põhjal luua skeemile vastavad java klassid, luua õigusaktist XmlBeans dokument ning selle käest küsida kõik fail elemendid (näiteks XPath'iga). Base64 kodeeringu oskab XmlBeans dekodeerida automaatselt. Teiste tehnoloogiate puhul võib failide eraldamine rohkem tööd nõuda.
- Eraldatud pildid ja failed tuleb salvestada samasse kausta kus asub akt ise
- Samuti stiilifaile kasutava infosüsteemi ülesandeks jääb XSL transformatsiooni tegemine ning transformeeritud õigusakti ja eraldatud failide kuvamine.

2.2 STIILIFAILIDE KASUTAMINE

- Antud juhendiga kaasas käivad stiilifailid (XSL, CSS, Javascript ja nendega kaasas käivad kujunduselemendid) tuleb süsteemi juurutada
  - XSL failed tuleb kopeerida süsteemi teiste XSL failidega samasse kohta ja transformatsioonis tuleb nendega arvestada
  - CSS ja JS failed koos kaasas käivate kujunduselementidega tuleb integreerida süsteemi ülejäänud stiilifailidega. Arvesse tuleb võtta, et antud stiilifailid ei arvesta süsteemi üldise disainiga ja kuvab akti alati sama moodi nagu see hetkel eRT-is on lahendatud. (näiteks kui ülejäänud süsteem on tehtud mingis teises fondis jne siis need stiilifailid seda arvesse ei võta)

2.3 BASE64 TRANSFORMATSIOONI JAVA KOODI NÄIDE

- Järgneb standardne näide dekodeerimisest Java koodis

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.logging.Logger;

import org.jdom.Document;
import org.jdom.Element;
import org.jdom.JDOMException;
import org.jdom.input.SAXBuilder;

public class Base64DecodeFromXML {

        static public Logger logger = Logger.getLogger("Base64DecodeFromXML");
        static private Document document = null;  // the doc

        /**
         * @param args inputFile outputFile "path/to/base64element"
         */
        public static void main(String[] args) {
                // startup
                logger.info("Base64DecodeFromXML 1.0 - starting up");
                if (args.length < 3) {
                        logger.severe("Base64DecodeFromXML got not enough parameters -
filenames needed");
```


Webmedia Tallinn | Lõõtsa St 6C, Tallinn 11415 | Phone: +372 6 101 990 Fax: +372 6 101 991
Webmedia Tartu | Raatuse St. 20, Tartu 51009 | Phone: +372 7 309 399 Fax: +372 7 309 398
Webmedia Narva | Kerese str. 3/Kosmonaudi St. 4, Narva 20309 | Phone: +372 7 309 399 Fax: +372 7 309 399
| *www.webmedia.ee* | E-mail: *webmedia@webmedia.ee*

```
                        System.err.println("Usage: Base64DecodeFromXML inputFile
outputFile path/to/base64element");
                        System.exit(-1);
                }
                // load xml-doc into document object
                logger.info("loading input file " + args[0]);
                SAXBuilder builder = new SAXBuilder();
                // try to build the doc
                try {
                        Base64DecodeFromXML.document = builder.build( args[0] );
                } catch (JDOMException e) {
                        e.printStackTrace();
                        document = null;
                } catch (IOException e) {
                        e.printStackTrace();
                        document = null;
                }
                if (null == Base64DecodeFromXML.document) {
                        logger.severe("Base64DecodeFromXML got bad XML file - not
exiting or not wellformed");
                        System.exit(-1);
                }
                // extract the path to element that contains the base64 encoded
binary from the commandline
                String path[] = args[2].split("/");
                // get the root element of the XML Doc
                Element ele = Base64DecodeFromXML.document.getRootElement();
                // recurse into the xml structure until we find the base64 element
                int i = 0;
                while ((null != ele) && (i < path.length)) {
                        logger.info("iterating into " + path[i]);
                        ele = ele.getChild(path[i]);
                        if (null == ele) {
                                logger.severe("Element not found: " + path[i]);
                                break;
                        }
                        ++i;
                }
                if (null != ele) {
                        // get the base64 content into String
                        String base64 = ele.getTextNormalize();

                        // and decode the content
                        try {
                                byte decoded[] = new
sun.misc.BASE64Decoder().decodeBuffer(base64);
                                // save it to a binary stream
                                logger.info("saving output file " + args[1]);
                                FileOutputStream  fos = null;
                                try {
                                        fos = new FileOutputStream(args[1]);
                                } catch (IOException e) {
                                        e.printStackTrace();
                                }
                                fos.write(decoded);
                                fos.close();
                        } catch (IOException e1) {
                                e1.printStackTrace();
                        }
                }
        }
```


Webmedia Tallinn | Lõõtsa St 6C, Tallinn 11415 | Phone: +372 6 101 990 Fax: +372 6 101 991
Webmedia Tartu | Raatuse St. 20, Tartu 51009 | Phone: +372 7 309 399 Fax: +372 7 309 398
Webmedia Narva | Kerese str. 3/Kosmonaudi St. 4, Narva 20309 | Phone: +372 7 309 399 Fax: +372 7 309 399
| *www.webmedia.ee* | E-mail: *webmedia@webmedia.ee*

Webmedia Tallinn | Lõõtsa St 6C, Tallinn 11415 | Phone: +372 6 101 990 Fax: +372 6 101 991
Webmedia Tartu | Raatuse St. 20, Tartu 51009 | Phone: +372 7 309 399 Fax: +372 7 309 398
Webmedia Narva | Kerese str. 3/Kosmonaudi St. 4, Narva 20309 | Phone: +372 7 309 399 Fax: +372 7 309 399
| *www.webmedia.ee* | E-mail: *webmedia@webmedia.ee*