

Digital Signature Service

Table of Contents

Introduction	3
Purpose of the document	3
Scope of the document	3
Abbreviations and Acronyms	4
References	7
Useful links	8
Build instructions	9
DSS Core	9
DSS Demonstrations	12
General framework structure	13
Maven modules	13
DSS Utils	19
DSS CRL Parser	19
DSS PAdES	19
Available demonstrations	20
Signature's profile simplification	21
Signature profile guide	22
The XML Signature (XAdES)	25
XAdES Profiles	25
Versions support	38
Reference Transformations	39
Multiple signatures	41
XAdES and specific schema version	41
Sign a Trusted List	42
Signature Extension	42
BASELINE-T	43
BASELINE-LT and -LTA	44
Signature Validation	44
Validation Process	44
Validation Result Materials	49
Validation Policy	77
CAdES signature (CMS)	86
PAdES signature (PDF)	87
PAdES Visible Signature	90
Shadow attack detection	97
JAdES signature (JWS)	98

JWS Serialization type	99
JAdES Signature Packaging	100
Base64Url encoding	101
ASiC signature (containers)	102
Counter signatures	105
Various parameters	106
Signature policy	106
Signature Policy Store	111
Trust anchor inclusion policy	112
Timestamps	112
Timestamp creation	112
Timestamp validation	113
Available implementations of DSSDocument	119
Management of signature tokens	119
PKCS#11	120
PKCS#12	120
MS CAPI	121
Other Implementations	121
Management of certificates sources	122
Management of CRL and OCSP sources	123
Repository Revocation Source	124
Other implementations of CRL and OCSP Sources	126
CertificateVerifier configuration	128
Trust Anchor(s) configuration	130
Trust store initialization	130
Trusted List Certificate Source	131
TLValidationJob	132
TLSource and LOTLSource	132
DSSFileLoader	134
The SynchronizationStrategy	135
The CacheCleaner	136
Alerting from TL Loading	136
LOTL/TL filter predicates	137
Executor Service	138
Complete configuration for the European LOTL	138
The TL / LOTL refresh	141
TLValidationJobSummary	143
TSP Sources	144
Time-stamp policy	144
Composite TSP Source	144
Supported algorithms	145

Implementation management with ServiceLoader	146
Document Validation Factory	147
Signature Policy Validator	148
Multi-threading	148
Resource sharing	148
Caching	148
XML Securities	149
JAXB modules	150
Report stylesheets	152
Alerts	153
Exceptions	153
I18N (Internationalization)	154
Additional features	155
Certificate validation	155
SSL Certificate validation (QWAC)	156
Extract the signed data from a signature	158
REST and SOAP Services	158
REST signature service	159
REST server signature service	274
REST validation service	280
REST certificate validation service	309
REST remote timestamp service	315

Introduction

Purpose of the document

This document describes some examples of how to develop in Java using the DSS framework. The aim is to show to the developers, in a progressive manner, the different uses of the framework. It will familiarize them with the code step by step.

Scope of the document

This document provides examples of code which allow easy handling of digital signatures. The examples are consistent with the Release 5.9 of DSS framework which can be downloaded via <https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/DSS+releases>

Three main features can be distinguished within the framework :

- The digital signature;
- The extension of a digital signature and;
- The validation of a digital signature.

On a more detailed manner the following concepts and features are addressed in this document:

- Formats of the signed documents: XML, JSON, PDF, DOC, TXT, ZIP...;
- Packaging structures: enveloping, enveloped, detached and internally-detached;
- Forms of digital signatures: XAdES, CAdES, PAdES, JAdES and ASiC-S/ASiC-E;
- Profiles associated to each form of the digital signature;
- Trust management;
- Revocation data handling (OCSP and CRL sources);
- Certificate chain building;
- Signature validation and validation policy;
- Signature qualification;
- Validation reports (Simple, Detailed, ETSI Validation report);
- Management of signature tokens;
- Validation of the signing certificate;
- Timestamp creation;
- Timestamp validation and qualification;
- REST and SOAP webservices.

This is not an exhaustive list of all the possibilities offered by the framework and the proposed examples cover only the most useful features. However, to discover every detail of the operational principles of the framework, the JavaDoc is available within the source code.

Please note that the DSS framework is still under maintenance and new features will be released in the future.

Abbreviations and Acronyms

Table 1. Abbreviations and Acronyms

Code	Description
AdES	Advanced Electronic Signature
API	Application Programming Interface
ASiC	Associated Signature Containers
BB	Building Block (CEF)
CA	Certificate authority
CAdES	CMS Advanced Electronic Signatures
CD	Commission Decision
CEF	Connecting Europe Facility
CMS	Cryptographic Message Syntax

CRL	Certificate Revocation List
CSP	Core Service Platform (CEF)
CSP	Cryptographic Service Provider
DER	Distinguished Encoding Rules
DSA	Digital Signature Algorithm - an algorithm for public-key cryptography
DSI	Digital Service Infrastructure (CEF)
DSS	Digital Signature Service
EC	European Commission
eID	Electronic Identity Card
ESI	Electronic Signatures and Infrastructures
ETSI	European Telecommunications Standards Institute
EUPL	European Union Public License
FSF	Free Software Foundation
GS	Generic Service (CEF)
GUI	Graphical User Interface
HSM	Hardware Security Modules
HTTP	Hypertext Transfer Protocol
I18N	Internationalization
JAdES	JSON Advanced Electronic Signatures
Java EE	Java Enterprise Edition
JavaDoc	JavaDoc is developed by Sun Microsystems to create API documentation in HTML format from the comments in the source code. JavaDoc is an industrial standard for documenting Java classes.
JAXB	Java Architecture for XML Binding
JCA	Java Cryptographic Architecture
JCE	Java Cryptography Extension
JDBC	Java DataBase Connectivity
JWS	JSON Web Signatures
LGPL	Lesser General Public License
LOTL	List of Trusted List or List of the Lists
LSP	Large Scale Pilot
MIT	Massachusetts Institute of Technology

MOCCA	Austrian Modular Open Citizen Card Architecture; implemented in Java
MS / EUMS	Member State
MS CAPI	Microsoft Cryptographic Application Programming Interface
OCF	OEBPS Container Format
OCSP	Online Certificate Status Protocol
ODF	Open Document Format
ODT	Open Document Text
OEBPS	Open eBook Publication Structure
OID	Object Identifier
OOXML	Office Open XML
OSI	Open Source Initiative
OSS	Open Source Software
PAdES	PDF Advanced Electronic Signatures
PC/SC	Personal computer/Smart Card
PDF	Portable Document Format
PDFBox	Apache PDFBox - A Java PDF Library: http://pdfbox.apache.org/
PKCS	Public Key Cryptographic Standards
PKCS#12	It defines a file format commonly used to store X.509 private key accompanying public key certificates, protected by symmetrical password
PKIX	Internet X.509 Public Key Infrastructure
RSA	Rivest Shamir Adleman - an algorithm for public-key cryptography
SCA	Signature Creation Application
SCD	Signature Creation Device
SME	Subject Matter Expert
SMO	Stakeholder Management Office (CEF)
SOAP	Simple Object Access Protocol
SSCD	Secure Signature-Creation Device
SVA	Signature Validation Application
TL	Trusted List
TLManager	Application for managing trusted lists.
TSA	Time Stamping Authority

TSL	Trust-service Status List
TSP	Time Stamp Protocol
TSP	Trusted Service Provider
TST	Time-Stamp Token
UCF	Universal Container Format
URI	Uniform Resource Identifier
WSDL	Web Services Description Language
WYSIWYS	What you see is what you sign
XAdES	XML Advanced Electronic Signatures
XML	Extensible Markup Language
ZIP	File format used for data compression and archiving

References

Table 2. References

Ref.	Title	Reference	Version
R01	ESI - XAdES digital signatures	ETSI EN 319 132 part 1-2	1.1.1
R02	ESI - CADES digital signatures	ETSI EN 319 122 part 1-2	1.1.1
R03	ESI - PAdES digital signatures	ETSI EN 319 142 part 1-2	1.1.1
R04	ESI - Associated Signature Containers (ASiC)	ETSI EN 319 162 part 1-2	1.1.1
R05	ESI - JAdES digital signatures	ETSI TS 119 182 part 1	1.1.1
R06	Document management - Portable document format - Part 1: PDF 1.7	ISO 32000-1	1
R07	Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures.	DIRECTIVE 1999/93/EC	

Ref.	Title	Reference	Version
R08	Internet X.509 Public Key Infrastructure - Time-Stamp Protocol (TSP)	RFC 3161	
R09	ESI - Procedures for Creation and Validation of AdES Digital Signatures	ETSI EN 319 102-1	1.1.1
R10	ESI - Signature validation policy for European qualified electronic signatures/seals using trusted lists	ETSI TS 119 172-4	draft
R11	ESI - Trusted Lists	ETSI TS 119 612	2.1.1
R12	eIDAS Regulation No 910/2014	910/2014/EU	
R13	ESI - Procedures for Creation and Validation of AdES Digital Signatures	ETSI TS 119 102-2	1.2.1
R14	ESI - Procedures for using and interpreting EU Member States national trusted lists	ETSI TS 119 615	draft

Useful links

- [CEF Digital](#)
- [eSignature FAQ](#)
- [Trust Services Dashboard](#)
- [eSignature validation tests](#)
- [Trusted List Manager non-EU](#)
- [Source code \(GitHub\)](#)
- [Source code \(EC Bitbucket\)](#)
- [Source code demonstrations \(EC Bitbucket\)](#)
- [Report an issue \(EC Jira\)](#)
- [Old Jira](#)

Build instructions

The section explains the basic steps required to successfully build the DSS components.

DSS Core

This section explains the build and usage requirements for [DSS framework](#).

Requirements

The latest version of DSS framework has the following minimal requirements:

- Java 11 and higher (tested up to Java 17) for the build is required. For usage Java 8 is a minimum requirement;
- Maven 3.6 and higher;
- Memory and Disk: see minimal requirements for the used JVM. In general the higher available is better;
- Operating system: no specific requirements (tested on Windows and Linux).



We strongly recommend using the latest available version of JDK, in order to have the latest security fixes and cryptographical algorithm updates.



Before processing the build steps, please, ensure you have successfully installed Maven and JVM with a required version.

Adding as Maven dependency

The simplest way to include DSS to your Maven project is to add a repository into pom.xml file in the root directory of your project as following:

```
<repositories>
  ...
  <repository>
    <id>cefdigital</id>
    <name>cefdigital</name>
    <url>https://ec.europa.eu/cefdigital/artifact/content/repositories/esignaturedss/</url>
  </repository>
</repositories>
```

After that specify a list of dependencies required for your project.

Refresh your project, in order to download the dependency and after you will be able to use all modules of DSS framework.

Maven build and profiles

In order to use a customized bundle of DSS, you may want to build the DSS Core framework modules.



If you have implemented a new feature or fixed a bug issue, your pull requests are welcome at our [GitHub Repository](#)

A simple build of the DSS Maven project can be done with the following command:

```
mvn clean install
```



All listed commands must be executed from the project directory via a Command Line Interface (CLI).

This installation will run all unit tests present in the modules, which can take more than one hour to do the complete build.

In addition to the general build, the framework provides a list of custom profiles, allowing a customized behavior:

- quick - disables unit tests and java-doc check, in order to process the build as quick as possible (takes 2-3 minutes).
- slow-tests - executes all tests, including time-consuming unit tests.
- owasp - runs validation of the project and using dependencies according to the [National Vulnerability Database \(NVD\)](#).
- jdk19-plus - executed automatically for JDK version 9 and higher. Provides a support of JDK 8 with newer versions.
- spotless - used to add a licence header into project files.



Some modules (e.g. `dss-utils`, `dss-crl-parser`, etc., see below) still have to be built completely when using the `quick` profile.

In order to run a build with a specific profile, the following command must be executed:

```
mvn clean install -P *profile_name*
```

Specific modules

Some modules of DSS framework have a specific behavior and need to be handled accordingly.

DSS contains a bundle of JAXB-based modules, generation Java classes in runtime based on XSD-schema. When any change is made in the XSD, the classes of the module are being re-generated according to the change. The following modules represent this behavior:

- specs-xmlsig;
- specs-xades;
- specs-trusted-list;
- specs-validation-report;
- specs-asic-manifest;
- specs-saml-assertion;
- dss-policy-jaxb;
- dss-diagnostic-jaxb;
- dss-detailed-report-jaxb;
- dss-simple-report-jaxb;
- dss-simple-certificate-report-jaxb.

Specific modules with JWS and JAdES specifications exist. These modules allow to validate the generated JSON against the related JSON Schema :

- specs-jws;
- specs-jades.

Also, as it was explained in the previous section, some modules are required to be built completely for a building of their dependent modules when using a quick profile, namely:

- dss-utils;
- dss-crl-parser;
- dss-test;
- dss-pades;
- dss-asic-common.

The modules contain common interfaces, used in other DSS modules, as well as unit tests to ensure the equal behavior between their implementations.

Documentation generation

In order to generate HTML and PDF documentation for DSS project, the module [dss-cookbook](#) of DSS Core must be build with the following command (please, ensure that you are located in the [/dss-cookbook](#) directory):

```
mvn clean install -P asciidoc
```

Javadoc generation

In order to generate [HTML Javadoc](#), you will need to build completely the DSS Core.

DSS Demonstrations

This section explains the build and usage requirements for [DSS Demonstration Applications](#).

Requirements

The minimal requirements to build/run DSS Demonstrations:

- Java 8 and higher (tested up to Java 17) is required;
- Maven 3.6 and higher (if build required);
- Tomcat 8.5+ for Java 8 and Tomcat 9+ for Java 9 and higher (for Web-application);
- Memory and Disk: see minimal requirements for the used JVM. In general the higher available is better;
- Operating system: no specific requirements (tested on Windows and Linux).

Maven build

The build of the project can be done similarly to DSS Core framework build with the command `mvn clean install`.

Please, ensure, that you build modules what you really need. Ignore, build failures for non-required modules.

DSS Standalone Application

In order to build the standalone application, the following modules are required:

- dss-mock-tsa;
- dss-standalone-app;
- dss-standalone-package.

If the build is successfull, you will be able to find out the following containers in the directory `/dss-standalone-app-package/target/`:

- dss-standalone-app-package-minimal.zip - contains the application code. Requires JDK and JavaFX installed on a target machine in order to run the application;
- dss-standalone-app-package-complete.zip - contains the application code, as well as JDK and JavaFX library code. Can be run on a machine whithout pre-installed libraries.

In order to launch the application, you will need to extract the archive and run the file `dss-run.bat`.

DSS Web Application

To build the DSS Web Application the following modules are required:

- dss-mock-tsa;
- dss-demo-webapp;

- dss-demo-bundle.

After a successful build, in the directory `/dss-demo-bundle/target/` you will be able to find out two containers: `dss-demo-bundle.zip` and `dss-demo-bundle.tar.gz`. Despite the container type, the content of both files is the same. After extracting the content, you will need to run the file `Webapp-Startup.bat` in order to launch the server and the file `Webapp-Shutdown.bat` to stop the server. After running the server, the web-application will be available at the address `http://localhost:8080/`.

If during TL/LOTL loading you experience problems with some particular Trusted Lists, please refer the chapter [Java Keystore Management](#) for a resolution.

The documentation and javadoc will be copied automatically from built DSS Core and available on the following addresses respectively:

- HTML documentation : <http://localhost:8080/doc/dss-documentation.html>;
- PDF documentation : <http://localhost:8080/doc/dss-documentation.pdf>;
- Javadoc : <http://localhost:8080/apidocs/index.html>.

In order to build a bundle for JDK 16, the following profile can be used from `dss-demo-bundle` module:

```
mvn clean install -P java16
```

This will create a bundle with Tomcat 9.

Integration tests

The `dss-demo-webapp` module provides a collection of integration tests in order to test the behavior of REST/SOAP web-services. In order to run the tests, a web-server with DSS Web Application shall be launched and the following profile need to be executed from the module:

```
mvn clean install -P run-integration-test
```

General framework structure

DSS framework is a multi-modules project which can be built with Maven.

Maven modules

Shared modules

dss-enumerations

Contains a list of all used enumerations in the DSS project.

dss-alerts

Allows configuration of triggers and handers for arbitrary defined events.

dss-jaxb-parsers

Contains a list of all classes used to transform JAXB objects/strings to Java objects and vice versa.

JAXB model modules

specs-xmlsig

W3C XSD schema for signatures <http://www.w3.org/2000/09/xmldsig>

specs-xades

ETSI EN 319 132-1 XSD schema for XAdES.

specs-trusted-list

ETSI TS 119 612 XSD schema for parsing Trusted Lists.

specs-validation-report

ETSI TS 119 102-2 XSD schema for the Validation report.

specs-asic-manifest

ETSI EN 319 162 schema for ASiCManifest.

specs-saml-assertion

OASIS schema for SAML Assertions.

dss-policy-jaxb

JAXB model of the validation policy.

dss-diagnostic-jaxb

JAXB model of the diagnostic data.

dss-detailed-report-jaxb

JAXB model of the detailed report.

dss-simple-report-jaxb

JAXB model of the simple report.

dss-simple-certificate-report-jaxb

JAXB model of the simple report for certificates.

JSON validation modules

specs-jws

JSON Schemas based on the RFC 7515 specifications (not official)

Utils modules

dss-utils

API with utility methods for String, Collection, I/O,...

dss-utils-apache-commons

Implementation of dss-utils with Apache Commons libraries.

dss-utils-google-guava

Implementation of dss-utils with Google Guava.

i18n

dss-i18n

a module allowing internationalization of generated reports.

Core modules

dss-model

Data model used in almost every module.

dss-crl-parser

API to validate CRLs and retrieve revocation data

dss-crl-parser-stream

Implementation of dss-crl-parser which streams the CRL.

dss-crl-parser-x509crl

Implementation of dss-crl-parser which uses the java object X509CRL.

dss-spi

Interfaces, util classes to manipulate ASN1, compute digests,...

dss-document

Common module to sign and validate document. This module doesn't contain any implementation.

dss-service

Implementations to communicate with online resources (TSP, CRL, OCSP).

dss-token

Token definitions and implementations for MS CAPI, PKCS#11, PKCS#12.

validation-policy

Business of the signature's validation (ETSI EN 319 102 / TS 119 172-4).

dss-xades

Implementation of the XAdES signature, extension and validation.

dss-cades

Implementation of the CAdES signature, extension and validation.

dss-jades

Implementation of the JAdES signature, extension and validation.

dss-pades

Common code which is shared between dss-pades-pdfbox and dss-pades-openpdf.

dss-pades-pdfbox

Implementation of the PAdES signature, extension and validation with [PDFBox](#).

dss-pades-openpdf

Implementation of the PAdES signature, extension and validation with [OpenPDF \(fork of iText\)](#).

dss-asic-common

Common code which is shared between dss-asic-xades and dss-asic-cades.

dss-asic-cades

Implementation of the ASiC-S and ASiC-E signature, extension and validation based on CAdES signatures.

dss-asic-xades

Implementation of the ASiC-S and ASiC-E signature, extension and validation based on XAdES signatures.

dss-tsl-validation

Module which allows loading / parsing / validating of LOTL and TSLS.

WebServices

dss-common-remote-dto

Common classes between all remote services (REST and SOAP).

dss-common-remote-converter

Classes which convert the DTO to DSS Objects.

dss-signature-dto

Data Transfer Objects used for signature creation/extension (REST and SOAP).

dss-signature-remote

Common classes between dss-signature-rest and dss-signature-soap.

dss-signature-rest-client

Client for the REST webservices.

dss-signature-rest

REST webservices to sign (getDataToSign, signDocument methods), counter-sign and extend a signature.

dss-signature-soap-client

Client for the SOAP webservices.

dss-signature-soap

SOAP webservices to sign (getDataToSign, signDocument methods), counter-sign and extend a signature.

dss-server-signing-dto

Data Transfer Objects used for the server signing module (REST and SOAP).

dss-server-signing-common

Common classes for server signing.

dss-server-signing-rest

REST webservice for server signing.

dss-server-signing-rest-client

REST client for server signing (sign method).

dss-server-signing-soap

SOAP webservice for server signing.

dss-server-signing-soap-client

SOAP client for server signing (sign method).

dss-validation-dto

Data Transfer Objects used for signature validation (REST and SOAP).

dss-validation-common

Common classes between dss-validation-rest and dss-validation-soap.

dss-validation-rest-client

Client for the REST signature-validation webservices.

dss-validation-soap-client

Client for the SOAP signature-validation webservices.

dss-validation-rest

REST webservices to validate a signature.

dss-validation-soap

SOAP webservices to validate a signature.

dss-certificate-validation-dto

Data Transfer Objects used for certificate validation (REST and SOAP).

dss-certificate-validation-common

Common classes between dss-certificate-validation-rest and dss-certificate-validation-soap.

dss-certificate-validation-rest-client

Client for the REST certificate-validation webservice.

dss-certificate-validation-soap-client

Client for the SOAP certificate-validation webservice.

dss-certificate-validation-rest

REST webservice to validate a certificate.

dss-certificate-validation-soap

SOAP webservice to validate a certificate.

dss-timestamp-dto

Data Transfer Objects used for timestamp creation.

dss-timestamp-remote-common

Common classes between dss-timestamp-remote-rest and dss-timestamp-remote-soap.

dss-timestamp-remote-rest-client

Client for the REST timestamp webservice.

dss-timestamp-remote-soap-client

Client for the SOAP timestamp webservice.

dss-timestamp-remote-rest

REST webservice to create a timestamp.

dss-timestamp-remote-soap

SOAP webservice to create a timestamp.

Other modules

dss-test

Mocks and util classes for unit tests.

dss-cookbook

Samples and documentation of DSS used to generate this documentation.

dss-bom

Module which helps the integration with all DSS modules and the version.

DSS Utils

The module dss-utils offers an interface with utility methods to operate on String, Collection, I/O,... DSS framework provides two different implementations with the same behaviour :

- dss-utils-apache-commons : this module uses Apache Commons libraries (commons-lang3, commons-collection4, commons-io and commons-codec).
- dss-utils-google-guava : this module only requires Google Guava (recommended on Android).

If your integration include dss-utils, you will need to select an implementation.

DSS CRL Parser

DSS contains two ways to parse/validate a CRL and to retrieve revocation data. An alternative to the X509CRL java object was developed to face memory issues in case of large CRLs. The X509CRL object fully loads the CRL in memory and can cause OutOfMemoryError.

- dss-crl-parser-x509crl : this module uses the X509CRL java object.
- dss-crl-parser-streams : this module offers an alternative with a CRL streaming (experimental).

If your integration require dss-crl-parser, you will need to choose your implementation.

DSS PAdES

Since the version 5.4, DSS allows generation/extension/validation PAdES signatures with two different frameworks : PDFBox and OpenPDF (fork of iText). The dss-pades module only contains the common code and requires an underlying implementation :

- dss-pades-pdfbox : Supports drawing of custom text, images, as well as text+image, in a signature field.
- dss-pades-openpdf : Supports drawing of custom text OR images in a signature field.

DSS permits to override the visible signature generation with these interfaces :

- eu.europa.esig.dss.pdf.IPdfObjFactory
- eu.europa.esig.dss.pdf.visible.SignatureDrawerFactory (selects the SignatureDrawer depending on the SignatureImageParameters content)

- eu.europa.esig.dss.pdf.visible.SignatureDrawer

A new instance of the IPdfObjFactory can be created with its own SignatureDrawerFactory and injected in the padesservice.setPdfObjFactory(IPdfObjFactory). By default, DSS uses an instance of ServiceLoaderPdfObjFactory. This instance checks for any registered implementation in the classpath with the ServiceLoader (potentially a service from dss-pades-pdfbox, dss-pades-openpdf or your own(s)).

DSS PDFBox

Since the version 5.5, DSS allows switching between two implementations of the framework PDFBox : default (original) and native.

- Default Drawer : The original drawer implemented on the PDFBox framework, supports displaying of custom text, images, text+image combination in a signature field. The implementation does not include the provided custom text to the inner PDF structure, instead of it, the drawer creates an image representation of the provided text, which is added to the signature field (i.e. the text is not selectable and not searchable).
- Native Drawer : Since the version 5.5, DSS includes a new implementation of PDFBox Drawer, that allows a user to add a real custom text, image or combination of text and image to a visible signature field. The native implementation embeds the provided custom text to the inner PDF structure, that makes the text selectable and searchable, and also clearer and smoother in comparison with the original implementation.

By default, DSS uses "Default Drawer" as the PDFBox implementation. In order to switch the implementation, that allowed in runtime, you have to set a new instance for PdfObjFactory as following:

Runtime PDF Object Factory changing

```
service.setPdfObjFactory(new PdfBoxNativeObjectFactory());
```

Available demonstrations

With the framework, some demonstrations are provided.

dss-mock-tsa	The class which generate false timestamps from a self-signed certificate.
sscd-mocca-adapter	Adapter for the MOCCA connection.
dss-standalone-app	Standalone application which allows signing a document with different formats and tokens (JavaFX).
dss-standalone-app-package	Packaging module for dss-standalone-app.

dss-demo-webapp

Demonstration web application which presents a part of the DSS possibilities.

dss-demo-bundle

Packaging module for dss-demo-webapp.



The demonstrations use a simulated timestamp service (Mock) so that is not recommended for a production usage.

The requirements and build instructions for DSS demonstrations can be found in the chapter [DSS Demonstrations](#).

Signature's profile simplification

The different formats of the digital signature make possible to cover a wide range of real live cases of use of this technique. Thus we distinguish the following formats: XAdES, CAdES, PAdES, JAdES and ASIC. To each one of them a specific standard is dedicated. The wide variety of options, settings and versions of the standards makes their interoperability very difficult. This is the main reason for which new standards commonly called "baseline profiles" were published. Their goal is to limit the number of options and variants thereby making possible a better interoperability between different actors.

In general can be said that for each format of the digital signature the number of security levels defined in the new standards has been reduced. Below is a comparative table of old and new levels for each format of the signature:

Table 3. Signature supported profiles

XAdES		CAdES		PAdES		JAdES	
STANDARD	BASELINE	STANDARD	BASELINE	STANDARD	BASELINE	BASELINE	
XAdES-BES	XAdES-B	CAdES-BES	CAdES-B	PAdES-BES	PAdES-B	JAdES-B	
XAdES-EPES		CAdES-EPES		PAdES-EPES			
XAdES-T	XAdES-T	CAdES-T	CAdES-T	PAdES-T	PAdES-T	JAdES-T	
XAdES-XL	XAdES-LT	CAdES-XL	CAdES-LT	PAdES-XL	PAdES-LT	JAdES-LT	
XAdES-A	XAdES-LTA	CAdES-A	CAdES-LTA	PAdES-LTV	PAdES-LTA	JAdES-LTA	

Note that the new version (v4) of the DSS framework is compatible with the baseline profiles, it is no longer possible to use the standard profiles for signing purpose. The validation of the signature still takes into account the old profiles.

Signature profile guide

Below you can find a table specifying various signature possibilities with available in DSS signature's profiles/formats. The vertical column specifies available signature profiles and their extensions. The horizontal row specifies types of documents to be signed with the formats.

Table 4. File formats and Signature types conformance

Signature profiles			XML	JSON	PDF	Binary	Digest	Multiple files	Multiple signatures	Counter signature	Stand-alone timestamp
XAdES	Enveloping	Base64 encoded	✓	✓	✓	✓	✗	✓	✗	✓	✗
		Embed XML	✓	✗	✗	✗	✗	XML only	✗	✓	✗
		Manifest	✓	✓	✓	✓	✓	✓	✗	✓	✗
		Canonicalization	✓	✗	✗	✗	✗	XML only	✗	✓	✗
	Enveloped	enveloped transformation	✓	✗	✗	✗	✗	✗	✗	✓	✗
		based on XPath	✓	✗	✗	✗	✗	✗	✓	✓	✗
		based on Filter2	✓	✗	✗	✗	✗	✗	✓	✓	✗
		Canonicalization	✓	✗	✗	✗	✗	XML only	✓	✓	✗
	Detached		✓	✓	✓	✓	✓	✓	✗	✓	✗
	Internally Detached		✓	✗	✗	✗	✗	XML only	✓	✓	✗

Signature profiles		XML	JSON	PDF	Binary	Digest	Multiple files	Multiple signatures	Counter signature	Stand-alone timestamp
CAdES	Enveloping	✓	✓	✓	✓	✗	✗	✓	✓	✗
	Detached	✓	✓	✓	✓	✓	✗	✓	✓	✗
PAdES	Enveloped	✗	✗	✓	✗	✗	✗	✓	✗	✓
JAdES	Enveloping	Compact Serializatio n	✓	✓	✓	✓	✗	✗	✗	✗
		Flattened JSON Serializatio n	✓	✓	✓	✓	✗	✗	✓	✗
		JSON Serializatio n	✓	✓	✓	✓	✗	✗	✓	✗
	Detached	Compact Serializatio n	✓	✓	✓	✓	✓	SigD only	✗	✗
		Flattened JSON Serializatio n	✓	✓	✓	✓	✓	SigD only	✗	✓
		JSON Serializatio n	✓	✓	✓	✓	✓	SigD only	✓	✗

Signature profiles			XML	JSON	PDF	Binary	Digest	Multiple files	Multiple signatures	Counter signature	Stand-alone timestamp
ASiC	ASiCS	CAdES / XAdES	✓	✓	✓	✓	✗	✓	✓	✓	✓
	ASiCE	CAdES / XAdES	✓	✓	✓	✓	✗	✓	✓	✓	✓

The XML Signature (XAdES)

The simplest way to address the digital signature passes through the XAdES format. Indeed, it allows visualization of the signature content with a simple text editor. Thus it becomes much easier to make the connection between theoretical concepts and their implementation. Before embarking on the use of the DSS framework, it is advisable to read the following documents:

- XAdES Specifications (cf. [\[R01\]](#))

After reading these documents, it is clear that:

- To electronically sign a document, a signing certificate (that proves the signer's identity) and the access to its associated private key is needed.
- To electronically validate a signed document the signer's certificate containing the public key is needed. To give a more colourful example: when a digitally signed document is sent to a given person or organization in order to be validated, the certificate with the public key used to create the signature must also be provided.

XAdES Profiles

The new ETSI standard defines four conformance levels to address the growing need to protect the validity of the signature in time. Henceforth to denote the level of the signature the word "level" will be used. Follows the list of levels defined in the standard:

- XAdES-BASELINE-B: *Basic Electronic Signature* The lowest and simplest version just containing the SignedInfo, SignatureValue, KeyInfo and SignedProperties. This level combines the old -BES and -EPES levels. This form extends the definition of an electronic signature to conform to the identified signature policy.
- XAdES-BASELINE-T: *Signature with a timestamp* A timestamp regarding the time of signing is added to protect against repudiation.
- XAdES-BASELINE-LT: *Signature with Long Term Data* Certificates and revocation data are embedded to allow verification in future even if their original source is not available. This level is equivalent to the old -XL level.
- XAdES-BASELINE-LTA: *Signature with Long Term Data and Archive timestamp* By using periodical timestamping (e.g. each year) compromising is prevented which could be caused by weakening previous signatures during a long-time storage period. This level is equivalent to the old -A level.



Old levels: -BES, -EPES, -C, -X, -XL, -A are not supported any more when signing.

XAdES-BASELINE-B

To start, let's take a simple XML document:

```
<?xml version="1.0"?>
<test>Hello World !</test>
```

Since this is an XML document, we will use the XAdES signature and more particularly XAdES-BASELINE-B level, which is the lowest level of protection: just satisfying Directive (cf. [R07]) legal requirements for advanced signature. The normal process of signing wants to sign first with the level -B or level-T, and then later when it becomes necessary to complete the signature with superior levels. However, the framework allows signing directly with any level. When signing data, the resulting signature needs to be linked with the data to which it applies. This can be done either by creating a data set which combines the signature and the data (e.g. by enveloping the data with the signature or including a signature element in the data set) or placing the signature in a separate resource and having some external means for associating the signature with the data. So, we need to define the packaging of the signature, namely ENVELOPED, ENVELOPING, DETACHED or INTERNALLY-DETACHED. More information about supported reference transformations for each signature packaging (except 'Detached'), can be found in the section [Reference Transformations](#)

- **ENVELOPED** : when the signature applies to data that surround the rest of the document;
- **ENVELOPING** : when the signed data form a sub-element of the signature itself;
 - Base64 encoded binaries;
 - Embed XML object(s);
 - Embed [Manifest](#) object(s).
- **DETACHED** : when the signature relates to the external resource(s) separated from it.
- **INTERNALLY-DETACHED** : when the signature and the related signed data are both included in a parent element (only XML).

For our example, we will use ENVELOPED packaging.

The DSS framework uses 3 atomic steps to sign a document :

1. Compute the digest to be signed;
2. Sign the digest;
3. Sign the document (add the signed digest).

The DSS fully manages the steps 1 and 3. We need to specify how to do the signature operation. DSS offers some implementations in the dss-token module

To write our Java code, we still need to specify the type of KeyStore to use for signing our document, more simply, where the private key can be found. In the package "eu.europa.esig.dss.token", we can choose between different connection tokens :

- **Pkcs11SignatureToken** : allows communicating with SmartCards with the PKCS#11 interface. It requires some installed drivers (dll, sso,...) .
- **Pkcs12SignatureToken** : allows signing with a PKC#12 keystore (.p12 file).

- **MSCAPISignatureToken** : handles the signature with MS CAPI (the Microsoft interface to communicate with SmartCards).
- **JKSSignatureToken** : allows signing with a Java Key Store (.jks file).



The DSS also provides the support for MOCCA framework to communicate with the Smartcard with PC/SC, but it involves the installation of the MOCCA and IAIK libraries.

To know more about the use of the different signature tokens, please consult "Management of Signature Tokens" chapter.

In our example the class: "Pkcs12SignatureToken" will be used. A file in PKCS#12 format must be provided to the constructor of the class. It contains an X.509 private key accompanying the public key certificate and protected by symmetrical password. The certification chain can also be included in this file. It is possible to generate dummy certificates and their chains with OpenSSL. Please visit <http://www.openssl.org/> for more details.

This is the complete code that allows you to sign our XML document.

Create a XAdES signature

```
// Preparing parameters for the XAdES signature
XAdESSignatureParameters parameters = new XAdESSignatureParameters();
// We choose the level of the signature (-B, -T, -LT, -LTA).
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_B);
// We choose the type of the signature packaging (ENVELOPED, ENVELOPING, DETACHED).
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPED);
// We set the digest algorithm to use with the signature algorithm. You must use the
// same parameter when you invoke the method sign on the token. The default value is
// SHA256
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();

// Create XAdES service for signature
XAdESService service = new XAdESService(commonCertificateVerifier);

// Get the SignedInfo XML segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm
SignatureValue signatureValue = signingToken.sign(dataToSign,
parameters.getDigestAlgorithm(), privateKey);

// We invoke the service to sign the document with the signature value obtained in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);
```

What you may notice is that to sign a document we need to:

- Create an object based on SignatureParameters class. The number of specified parameters depends on the type of signature. Generally, the number of specified parameters depends on the profile of signature. This object also defines some default parameters.
- Choose the profile, packaging, signature digest algorithm.
- Indicate the private key entry to be used.
- Instantiate the adequate signature service.
- Carry out the signature process.

The encryption algorithm is determined by the private key and therefore cannot be compelled by

the setter of the signature parameters object. It will cause an inconsistency in the signature making its validation impossible. This setter can be used in a particular context where the signing process is distributed on different machines and the private key is known only to the signature value creation process. See clause "Signing process" for more information. In the case where the private key entry object is not available, it is possible to choose the signing certificate and its certificate chain as in the following example:

```
// We set the signing certificate
parameters.setSigningCertificate(certificateToken);
// We set the certificate chain
parameters.setCertificateChain(certificateChain);
```

Integrating the certificate chain in the signature simplifies the build of a prospective certificate chain during the validation process.

By default the framework uses the current date time to set the signing date, but in the case where it is necessary to indicate the different time it is possible to use the setter "setSigningDate(Date)" as in the example:

```
// We set the date of the signature.
parameters.bLevel().setSigningDate(new Date());
```

When the specific service is instantiated a certificate verifier must be set. This object is used to provide four different sources of information:

- the source of trusted certificates (based on the trusted list(s) specific to the context);
- the source of intermediate certificates used to build the certificate chain till the trust anchor. This source is only needed when these certificates are not included in the signature itself;
- the source of OCSP;
- the source of CRL.

In the current implementation this object is only used when profile -LT or -LTA are created.

Signing process

Once the parameters of the signature were identified the service object itself must be created. The service used will depend on the type of document to sign. In our case it is an XML file, so we will instantiate a XAdES service. The process of signing takes place in three stages. The first is the [getDataToSign\(\)](#) method call, passing as a parameter the document to be signed and the previously selected settings. This step returns the data which is going to be digested and encrypted. In our case it corresponds to the SignedInfo XMLDSig element.

```

// Create XAdES service for signature
XAdESService service = new XAdESService(commonCertificateVerifier);

// Get the SignedInfo XML segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

```

The next step is a call to the function `sign()` which is invoked on the object token representing the KeyStore and not on the service. This method takes three parameters. The first is the array of bytes that must be signed. It is obtained by the previous method invocation. The second is the algorithm used to create the digest. You have the choice between SHA1, SHA256, and SHA512 (this list is not exhaustive). And the last one is the private key entry.

```

DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm,
privateKey);

```

The last step of this process is the integration of the signature value in the signature and linking of that one to the signed document based on the selected packaging method. This is the method `signDocument()` on the service. We must pass to it three parameters: again the document to sign, the signature parameters and the value of the signature obtained in the previous step.

This separation into three steps allows use cases where different environments have their precise responsibilities: specifically the distinction between communicating with the token and executing the business logic.

When the breakdown of this process is not necessary, than a simple call to only one method can be done as in the following example:

```

DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);

```

Additional attributes

For this type (XAdES-BASELINE-B) of signature it is possible to identify some additional attributes.

XAdES signature with additional signed attributes

```

XAdESSignatureParameters parameters = new XAdESSignatureParameters();

// Basic signature configuration
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPED);
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_B);
parameters.setDigestAlgorithm(DigestAlgorithm.SHA512);
parameters.setSigningCertificate(privateKey.getCertificate());
parameters.setCertificateChain(privateKey.getCertificateChain());

// Configuration of several signed attributes like ...

```

```

BLevelParameters bLevelParameters = parameters.bLevel();

// Contains claimed roles assumed by the signer when creating the signature
bLevelParameters.setClaimedSignerRoles(Arrays.asList("Manager"));

// signer location
SignerLocation signerLocation = new SignerLocation();
signerLocation.setCountry("BE");
signerLocation.setStateOrProvince("Luxembourg");
signerLocation.setPostalCode("1234");
signerLocation.setLocality("SimCity");
// Contains the indication of the purported place where the signer claims to have
produced the signature
bLevelParameters.setSignerLocation(signerLocation);

// Identifies the commitment undertaken by the signer in signing (a) signed data
object(s)
// in the context of the selected signature policy
List<CommitmentType> commitmentTypeIndications = new ArrayList<>();
commitmentTypeIndications.add(CommitmentTypeEnum.ProofOfOrigin);
commitmentTypeIndications.add(CommitmentTypeEnum.ProofOfApproval);
// NOTE: CommitmentType supports also IDQualifier and documentationReferences.
// To use it, you need to have a custom implementation of the interface.
bLevelParameters.setCommitmentTypeIndications(commitmentTypeIndications);

CommonCertificateVerifier verifier = new CommonCertificateVerifier();
XAdESService service = new XAdESService(verifier);
service.setTspSource(getOnlineTSPSource());

// Allows setting of content-timestamp (part of the signed attributes)
TimestampToken contentTimestamp = service.getContentTimestamp(toSignDocument,
parameters);
parameters.setContentTimestamps(Arrays.asList(contentTimestamp));

// Signature process with its 3 stateless steps
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);
SignatureValue signatureValue = signingToken.sign(dataToSign,
parameters.getDigestAlgorithm(), privateKey);
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);

```

In XAdES format the following types of a Content Timestamp can be used:

- AllDataObjectsTimeStamp - each time-stamp token within this property covers the full set of references defined in the Signature's SignedInfo element, excluding references of type "SignedProperties".
- IndividualDataObjectsTimeStamp - each time-stamp token within this property covers selected signed data objects.

The code above produces the following signature :

XAdES signature example

```
<xades:SignedProperties Id="xades-id-ea3e16770317bb1a3e97244292931644">
  <xades:SignedSignatureProperties>
    <xades:SigningTime>2018-03-20T08:17:35Z</xades:SigningTime>
    <xades:SigningCertificateV2>
      <xades:Cert>
        <xades:CertDigest>
          <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          <ds:DigestValue>2FeANjXzi09x2877Sfc1R1RVj1E=</ds:DigestValue>
        </xades:CertDigest>

<xades:IssuerSerialV2>MD4wNKQyMDAxGzAZBgNVBAMME1Jvb3RTZWxmU2lnbmVkRmFrZTERMA8GA1UECgwI
RFNTLXRlc3QCBi7WFNe7Vw==</xades:IssuerSerialV2>
      </xades:Cert>
    </xades:SigningCertificateV2>
    <xades:SignatureProductionPlaceV2>
      <xades:City>SimCity</xades:City>
      <xades:StateOrProvince>Luxembourg</xades:StateOrProvince>
      <xades:PostalCode>1234</xades:PostalCode>
      <xades:CountryName>BE</xades:CountryName>
    </xades:SignatureProductionPlaceV2>
    <xades:SignerRoleV2>
      <xades:ClaimedRoles>
        <xades:ClaimedRole>Manager</xades:ClaimedRole>
      </xades:ClaimedRoles>
    </xades:SignerRoleV2>
  </xades:SignedSignatureProperties>
  <xades:SignedDataObjectProperties>
    <xades:DataObjectFormat ObjectReference="#r-id-1">
      <xades:MimeType>text/xml</xades:MimeType>
    </xades:DataObjectFormat>
    <xades:CommitmentTypeIndication>
      <xades:CommitmentTypeId>

<xades:Identifier>http://uri.etsi.org/01903/v1.2.2#ProofOfOrigin</xades:Identifier>
      </xades:CommitmentTypeId>
      <xades:AllSignedDataObjects />
    </xades:CommitmentTypeIndication>
    <xades:CommitmentTypeIndication>
      <xades:CommitmentTypeId>

<xades:Identifier>http://uri.etsi.org/01903/v1.2.2#ProofOfApproval</xades:Identifier>
      </xades:CommitmentTypeId>
      <xades:AllSignedDataObjects />
    </xades:CommitmentTypeIndication>
    <xades:AllDataObjectsTimeStamp Id="TS-
678B5861DBA1469B3AA3DD49DD54D7046BADA578C5561F8ABDA935CE0825279E">
      <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
```

```

<xades:EncapsulatedTimeStamp>MIAGCSqGSIB3DQEHAq...aAAAAAA=</xades:EncapsulatedTimeStamp>
</xades:AllDataObjectsTimeStamp>
</xades:SignedDataObjectProperties>
</xades:SignedProperties>

```

XAdES-BASELINE-T

XAdES-BASELINE-T is a signature for which there exists a trusted time associated to the signature. It provides the initial steps towards providing long term validity and more specifically it provides a protection against repudiation. This extension of the signature can be created as well during the generation process as validation process. However, the case when these validation data are not added during the generation process should no longer occur. The XAdES-BASELINE-T trusted time indications must be created before the signing certificate has been revoked or expired and close to the time that the XAdES signature was produced. The XAdES-BASELINE-T form must be built on a XAdES-BASELINE-B form. The DSS framework allows extending the old -BES and -EPES profiles to the new BASELINE-T profile, indeed there is no difference in the structure of the signature.

To implement this profile of signature you must indicate to the service the TSA source, which delivers from each Timestamp Request a Timestamp Response (RFC 3161 (cf. [\[R08\]](#))) containing tokens. Below is the source code that creates a XAdES-BASELINE-T signature. For our example, we will use the Belgian provider and an instance of OnlineTSPSource (see "TSP Sources" chapter for more details).

Create a XAdES-Baseline-T with an OnlineTSPSource

```
// Preparing parameters for the XAdES signature
XAdESSignatureParameters parameters = new XAdESSignatureParameters();
// We choose the level of the signature (-B, -T, -LT, -LTA).
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_T);
// We choose the type of the signature packaging (ENVELOPED, ENVELOPING, DETACHED).
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPED);
// We set the digest algorithm to use with the signature algorithm. You must use the
// same parameter when you invoke the method sign on the token. The default value is
// SHA256
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
// Create XAdES service for signature
XAdESService service = new XAdESService(commonCertificateVerifier);

// Set the Timestamp source
String tspServer = "http://dss.nowina.lu/pki-factory/tsa/good-tsa";
OnlineTSPSource onlineTSPSource = new OnlineTSPSource(tspServer);
onlineTSPSource.setDataLoader(new TimestampDataLoader()); // uses the specific
content-type
service.setTspSource(onlineTSPSource);

// Get the SignedInfo XML segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm
SignatureValue signatureValue = signingToken.sign(dataToSign,
parameters.getDigestAlgorithm(), privateKey);

// We invoke the service to sign the document with the signature value obtained in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);
```

If the timestamp source is not set a NullPointerException is thrown.

The SignatureTimeStamp mandated by the XAdES-T form appears as an unsigned property within the QualifyingProperties:

XAdES Signature Timestamp

```
<SignatureTimeStamp Id="time-stamp-28a441da-4030-46ef-80e1-041b66c0cb96">
  <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
  <EncapsulatedTimeStamp
    Id="time-stamp-token-76234ed8-cc15-46fc-aa95-9460dd601cad">
    MIAGCSqSIb3DQEHAqCAMIACQMyCzAJBgUrDgMCg
    UAMIAGCyq6SIb3DQEJEAEoIAkgARMMEoCAQEGBoIS
    ...
  </EncapsulatedTimeStamp>
</SignatureTimeStamp>
```

XAdES-BASELINE-LT

This level has to prove that the certification path was valid, at the time of the validation of the signature, up to a trust point according to the naming constraints and the certificate policy constraints from the "Signature Validation Policy". It will add to the signature the CertificateValues and RevocationValues unsigned properties. The CertificateValues element contains the full set of certificates that have been used to validate the electronic signature, including the signer's certificate. However, it is not necessary to include one of those certificates, if it is already present in the ds:KeyInfo element of the signature. This is like DSS framework behaves. In order to find a list of all the certificates and the list of all revocation data, an automatic process of signature validation is executed. To carry out this process an object called CertificateVerifier must be passed to the service. The implementer must set some of its properties (e.g. a source of trusted certificates). The code below shows how to use the default parameters with this object. Please refer to "The Signature Validation" chapter to have the further information. It also includes an example of how to implement this level of signature:

SignXmlXadesLTTest.java

```
// Preparing parameters for the XAdES signature
XAdESSignatureParameters parameters = new XAdESSignatureParameters();
// We choose the level of the signature (-B, -T, -LT, -LTA).
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_LT);
// We choose the type of the signature packaging (ENVELOPED, ENVELOPING, DETACHED).
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPED);
// We set the digest algorithm to use with the signature algorithm. You must use the
// same parameter when you invoke the method sign on the token. The default value is
// SHA256
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
```

```

CommonsDataLoader commonsHttpDataLoader = new CommonsDataLoader();
OCSPDataLoader ocspDataLoader = new OCSPDataLoader();

KeyStoreCertificateSource keyStoreCertificateSource = new
KeyStoreCertificateSource(new File("src/main/resources/keystore.p12"), "PKCS12",
    "dss-password");

LOTLSouce lotlSource = new LOTLSouce();
lotlSource.setUrl("https://ec.europa.eu/tools/lotl/eu-lotl.xml");
lotlSource.setCertificateSource(keyStoreCertificateSource);
lotlSource.setPivotSupport(true);

TrustedListsCertificateSource tslCertificateSource = new
TrustedListsCertificateSource();

FileCacheDataLoader onlineFileLoader = new FileCacheDataLoader(commonsHttpDataLoader);

CacheCleaner cacheCleaner = new CacheCleaner();
cacheCleaner.setCleanFileSystem(true);
cacheCleaner.setDSSFileLoader(onlineFileLoader);

TLValidationJob validationJob = new TLValidationJob();
validationJob.setTrustedListCertificateSource(tslCertificateSource);
validationJob.setOnlineDataLoader(onlineFileLoader);
validationJob.setCacheCleaner(cacheCleaner);
validationJob.setListOfTrustedListSources(lotlSource);
validationJob.onlineRefresh();

commonCertificateVerifier.setTrustedCertSources(tslCertificateSource);

OnlineCRLSource onlineCRLSource = new OnlineCRLSource();
onlineCRLSource.setDataLoader(commonsHttpDataLoader);
commonCertificateVerifier.setCrlSource(onlineCRLSource);

OnlineOCSPSource onlineOCSPSource = new OnlineOCSPSource();
onlineOCSPSource.setDataLoader(ocspDataLoader);
commonCertificateVerifier.setOcspSource(onlineOCSPSource);

// For test purpose
// Will request unknown OCSP responder / download untrusted CRL
commonCertificateVerifier.setCheckRevocationForUntrustedChains(true);

// Create XAdES service for signature
XAdESService service = new XAdESService(commonCertificateVerifier);
service.setTspSource(getOnlineTSPSource());

// Get the SignedInfo XML segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm

```

```

SignatureValue signatureValue = signingToken.sign(dataToSign,
parameters.getDigestAlgorithm(), privateKey);

// We invoke the service to sign the document with the signature value obtained in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);

```

The following XML segment will be added to the signature qualified and unsigned properties:

Validation data values

```

<CertificateValues>
  <EncapsulatedX509Certificate>
    MIIFNTCCBB2gAwIBAgIBATANB...
  </EncapsulatedX509Certificate>
  <EncapsulatedX509Certificate>
    MIIFsjCCBJqgAwIBAgIDAMoBM...
  </EncapsulatedX509Certificate>
  <EncapsulatedX509Certificate>
    MIIFRjCCBC6gAwIBAgIBATANB...
  </EncapsulatedX509Certificate>
</CertificateValues>
<RevocationValues>
  <OCSPValues>
    <EncapsulatedOCSPValue>
      MIIGzAoBAKCCBsUwggBBgkr...
    </EncapsulatedOCSPValue>
  </OCSPValues>
</RevocationValues>

```



The use of online sources can significantly increase the execution time of the signing process. For testing purpose you can create your own source of data.

In last example the CommonsHttpDataLoader is used to provide the communication layer for HTTP protocol. Each source which need to go through the network to retrieve data need to have this component set.

XAdES-BASELINE-LTA

When the cryptographic data becomes weak and the cryptographic functions become vulnerable the auditor should take steps to maintain the validity of the signature. The XAdES-BASELINE-A form uses a simple approach called "archive validation data". It adds additional time-stamps for archiving signatures in a way that they are still protected, but also to be able to prove that the signatures were validated at the time when the used cryptographic algorithms were considered safe. The time-stamping process may be repeated every time the protection used becomes weak. Each time-stamp needs to be affixed before either the signing key or the algorithms used by the TSA are no longer secure. XAdES-A form adds the ArchiveTimestamp element within the UnsignedSignatureProperties and may contain several ArchiveTimestamp elements.

Below is an example of the implementation of this level of signature (but in practice, we will rather extend the signature to this level when there is a risk that the cryptographic functions become vulnerable or when one of certificates arrives to its expiration date):

Signature level setting

```
// Allows to set a final signature level
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_LTA);
```

The following XML segment will be added to the signature qualified and unsigned properties:

XAdES Archive Timestamp

```
<ns4:ArchiveTimeStamp
  Id="time-stamp-22b92602-2670-410e-888f-937c5777c685">
  <ds:CanonicalizationMethod
    Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
  <EncapsulatedTimeStamp
    Id="time-stamp-token-0bd5aaf3-3850-4911-a22d-c98dcaca5cea">MIAGCSqGSDHAqCAM...
  </EncapsulatedTimeStamp>
</ns4:ArchiveTimeStamp>
```

Versions support

DSS supports the following XAdES formats :

Table 5. Supported XAdES versions

	B-level	T-level	LT-level	LTA-level
XAdES 1.1.1	✓	✓	✓	✗
XAdES 1.2.2	✓	✓	✓	✗
XAdES 1.3.2	✓	✓	✓	✓
XAdES 1.4.1	The format contains qualifying properties for XAdES 1.3.2 LTA level			

The XAdES Profile, as well as a customizable prefixes can be set with following methods :

XAdES formats and prefixes

```
// Allows setting of a XAdES namespace (changes a XAdES format)
// Default : XAdESNamespaces.XADES_132 (produces XAdES 1.3.2)
parameters.setXadesNamespace(XAdESNamespaces.XADES_132);

// Defines an XmlDSig prefix
// Default : XAdESNamespaces.XMLDSIG
parameters.setXmlsigNamespace(new DSSNamespace(XMLSignature.XMLNS, "myPrefix"));

// Defines a XAdES 1.4.1 format prefix
// Default : XAdESNamespaces.XADES_141
parameters.setXades141Namespace(XAdESNamespaces.XADES_141);
```

Reference Transformations

In case of 'Enveloping', 'Enveloped' and 'Internally Detached' signatures, it is possible to apply custom transformations for signing references in order to compute proper digest result. Example of a definition reference transformations, you can find below:

Custom transformations definition

```
// Prepare transformations in the proper order
List<DSSTransform> transforms = new ArrayList<>();
DSSTransform envelopedTransform = new EnvelopedSignatureTransform();
transforms.add(envelopedTransform);
DSSTransform canonicalization = new
CanonicalizationTransform(CanonicalizationMethod.EXCLUSIVE_WITH_COMMENTS);
transforms.add(canonicalization);

// Assign reference to the document
List<DSSReference> references = new ArrayList<>();
DSSReference dssReference = new DSSReference();
dssReference.setContents(toSignDocument);
dssReference.setId("r-" + toSignDocument.getName());
dssReference.setTransforms(transforms);
// set empty URI to cover the whole document
dssReference.setUri("");
dssReference.setDigestMethodAlgorithm(DigestAlgorithm.SHA256);
references.add(dssReference);

// Initialize signature parameters
XAdESSignatureParameters parameters = new XAdESSignatureParameters();
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPED);
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_B);
// set references
parameters.setReferences(references);
```

Current version of DSS supports the following transformations:

- Enveloped - removes the current **Signature** element from the digest calculation of the reference.



Enveloped Signature Transform does not support parallel signatures!

```
DSSTransform envelopedTransform = new EnvelopedSignatureTransform();
```

- Canonicalization - any canonicalization algorithm that can be used for 'CanonicalizationMethod' can be used as a transform:

```
DSSTransform canonicalization = new
CanonicalizationTransform(CanonicalizationMethod.EXCLUSIVE_WITH_COMMENTS);
```

- Base64 - the transform is used if application needs to sign a RAW data (binaries, images, audio or other formats). The 'Base64 Transform' is not compatible with following signature parameters:
 - Reference contains more than one transform (must be a sole element of the reference transforms);
 - `setEmbedXML(true)` - embedded setting cannot be used;
 - `setManifestSignature(true)` - As is apparent from the previous point, Manifest cannot be used with the Base64 Transform as well since it also must be embedded to the signature.

```
DSSDocument document = new InMemoryDocument("Hello World!".getBytes(), "Hello.txt",
MimeType.BINARY);
List<DSSTransform> transforms = new ArrayList<>();
DSSTransform base64Transform = new Base64Transform();
transforms.add(base64Transform);
```

- XPath - allows signing a custom nodes in a signature or embedded document. DSS contains an additional class **XPathEnvelopedSignatureTransform** allowing to exclude signatures from the digested content (used for Enveloped signatures by default). Additional information about the 'XPath Transform' can be found [by the link](#).

```
List<DSSTransform> transforms = new ArrayList<>();
DSSTransform envelopedTransform = new XPathTransform("not(ancestor-or-
self::ds:Signature)");
transforms.add(envelopedTransform);
```

- XPath-2-Filter - an alternative to 'XPath Transform'. Additional information about the 'XPath2Filter Transform' can be found [by the link](#). DSS contains an additional class **XPath2FilterEnvelopedSignatureTransform** allowing to exclude signatures from the digest calculation.



Since DSS 5.7 the XPath-2-Filter transform is used by default for ENVELOPED signature packaging.

```
List<DSSTransform> transforms = new ArrayList<>();  
DSSTransform envelopedTransform = new  
XPath2FilterTransform("descendant::ds:Signature", "subtract");  
transforms.add(envelopedTransform);
```

- XSLT Transform - This transform requires a 'org.w3.dom.Document' as an input, compatible with the normative [XSLT Specification](#). Must be a sole transform.



All transformations, except Base64, can be applied only to XML objects.

Multiple signatures

In everyday life, there are many examples where it is necessary to have multiple signatures covering the same document, such as a contract to purchase a vehicle. Independent signatures are parallel signatures where the ordering of the signatures is not important. The computation of these signatures is performed on exactly the same input but using different private keys.

XAdES and specific schema version

Some signatures may have been created with an older version of XAdES standard using different schema definition. To take into account the validation of such signatures the interface `eu.europa.esig.dss.xades.definition.XAdESPaths` was created. This interface allows to provide the different needed XPath expressions which are used to explore the elements of the signature. The DSS framework proposes 3 implementations :

- XAdES132Paths (XAdES 1.3.2 / 1.4.1)
- XAdES122Paths (XAdES 1.2.2)
- XAdES111Paths (XAdES 1.1.1)

By default, all XAdES are supported and DSS loads/parses all versions of XAdES. That's possible to restrict to only one version of XAdES with the following code :

Customize the supported XAdES version(s) at the validation

```
XMLDocumentValidator xmlDocValidator = new XMLDocumentValidator(xmlDocument);
xmlDocValidator.setCertificateVerifier(certificateVerifier);

// Restrict the current XMLDocumentValidator to XAdES 1.3.2 (and 1.4.1 for
// archival timestamps)
List<XAdESPaths> xadesPathsHolders = xmlDocValidator.getXAdESPathsHolder();
xadesPathsHolders.clear();
xadesPathsHolders.add(new XAdES132Paths());

Reports reports = xmlDocValidator.validateDocument();
```

Sign a Trusted List

The standard ETSI TS 119 612 specifies in its annex B the XML structure and the format of the signature (XAdES, enveloped signature, transformation, canonicalization, etc.). With the class [TrustedListSignatureParametersBuilder](#), DSS is able to pre-configure the signature parameters to comply with the specifications and simplify the signature creation.

Sign a Trusted List with the TrustedListSignatureParametersBuilder

```
DSSDocument trustedList = new FileDocument("src/main/resources/trusted-list.xml");

DSSPrivateKeyEntry privateKeyEntry = signingToken.getKeys().get(0);
CertificateToken signingCertificate = privateKeyEntry.getCertificate();

// This class creates the appropriated XAdESSignatureParameters object to sign a
// trusted list.
// It handles the configuration complexity and creates a ready-to-be-used
// XAdESSignatureParameters with a correct configuration.
TrustedListSignatureParametersBuilder builder = new
TrustedListSignatureParametersBuilder(signingCertificate, trustedList);
XAdESSignatureParameters parameters = builder.build();

XAdESService service = new XAdESService(new CommonCertificateVerifier());

ToBeSigned dataToSign = service.getDataToSign(trustedList, parameters);
SignatureValue signatureValue = signingToken.sign(dataToSign,
parameters.getDigestAlgorithm(), privateKeyEntry);
DSSDocument signedTrustedList = service.signDocument(trustedList, parameters,
signatureValue);
```

Signature Extension

The -B level contains immutable signed properties. Once this level is created, these properties cannot be changed.

The levels -T/-LT/-LTA add unsigned properties to the signature. This means that the properties of these levels could be added afterwards to any AdES signature. This addition helps to make the signature more resistant to cryptographic attacks on a longer period of time. The extension of the signature is incremental, i.e. when you want to extend the signature to the level -LT the lower level (-T) will also be added. The whole extension process is implemented by reusing components from signature production. To extend a signature we proceed in the same way as in the case of a signature, except that you have to call the function "extendDocument" instead of the "sign" function. Note that when the document is signed with several signatures then they are all extended.

BASELINE-T

The AdES-BASELINE-T trusted time indications have to be created before a certificate has been revoked or expired and close to the time that the AdES signature was produced. It provides a protection against repudiation. The framework adds the timestamp only if there is no timestamp or there is one but the creation of a new extension of the level-T is deliberate (using another TSA). It is not possible to extend a signature which already incorporates higher level as -LT or -LTA. In the theory it would be possible to add another -T level when the signature has already reached level -LT but the framework prevents this operation. Note that if the signed document contains multiple signatures, then all the signatures will be extended to level -T. It is also possible to sign a document directly at level -T.

Here is an example of creating an extension of type T:

Extend a XAdES signature

```
DSSDocument document = new FileDocument("src/test/resources/signature-
pool/signedXmlXadesB.xml");

XAdESSignatureParameters parameters = new XAdESSignatureParameters();
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_T);

CommonCertificateVerifier certificateVerifier = new CommonCertificateVerifier();
XAdESService xadesService = new XAdESService(certificateVerifier);
xadesService.setTspSource(getOnlineTSPSource());

DSSDocument extendedDocument = xadesService.extendDocument(document, parameters);
```

Here is the result of adding a new extension of type-T to an already existing -T level signature:

```
<UnsignedSignatureProperties>
  <SignatureTimeStamp Id="time-stamp-b16a2552-b218-4231-8982-40057525fb5">
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <EncapsulatedTimeStamp Id="time-stamp-token-39fbf78c-9cec-4cc1-ac21-a467d2238405"> MIAGCSqGSIb3DQEHAq...
    </EncapsulatedTimeStamp>
  </SignatureTimeStamp>
  <SignatureTimeStamp Id="time-stamp-5ffab0d9-863b-414a-9690-a311d3e1af1d">
    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
    <EncapsulatedTimeStamp Id="time-stamp-token-87e8c599-89e5-4fb3-a32a-e5e2a40073ad"> MIAGCSqGSIb3DQEHAq...
    </EncapsulatedTimeStamp>
  </SignatureTimeStamp>
</UnsignedSignatureProperties>
```

BASELINE-LT and -LTA

For these types of extensions, the procedure to follow is the same as the case of the extension of type T. Please refer to the chapter XAdES Profiles to know specific parameters for each level of signature and which must be positioned.

Signature Validation

Generally and following ETSI standard, the validation process of an electronic signature must provide one of these three following statuses: TOTAL-FAILED, TOTAL-PASSED or INDETERMINATE. A TOTAL-PASSED response indicates that the signature has passed verification and it complies with the signature validation policy. A TOTAL_FAILED response indicates that either the signature format is incorrect or that the digital signature value fails the verification. An INDETERMINATE validation response indicates that the format and digital signature verifications have not failed but there is an insufficient information to determine if the electronic signature is valid. For each of the validation checks, the validation process must provide information justifying the reasons for the resulting status indication as a result of the check against the applicable constraints. In addition, the ETSI standard defines a consistent and accurate way for justifying statuses under a set of sub-indications.

Validation Process

Since version 4.7 of the DSS framework the validation process is based on the latest ETSI standard [R09]. It is driven by the validation policy and allows long term signature validation. It not only verifies the existence of certain data and their validity, but it also checks the temporal dependences between these elements. The signature check is done following basic building blocks. On the simplified diagram below, showing the process of the signature validation, you can follow the relationships between each building block which represents a logic set of checks used in validation

process.

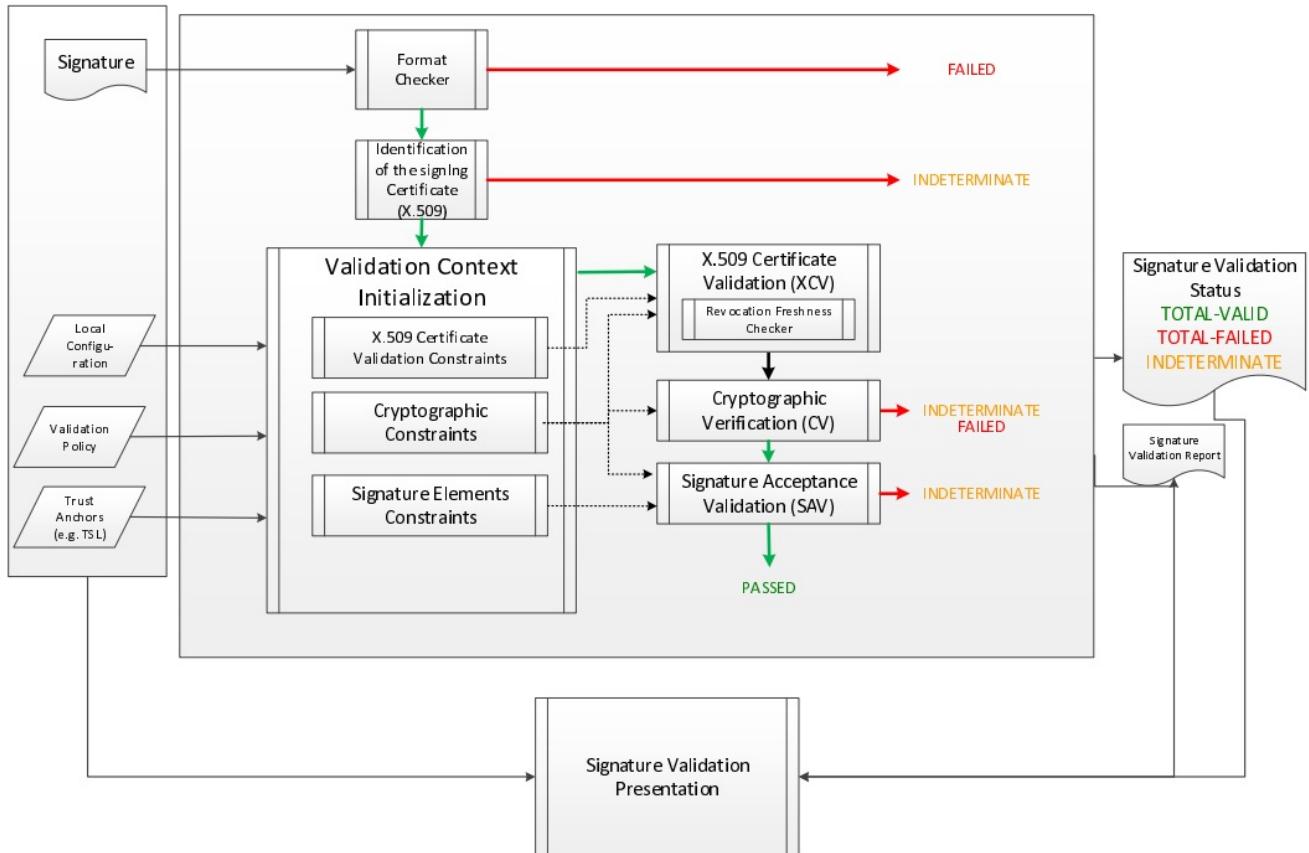


Figure 1. Signature Validation Process

Note that the current version of the framework during the validation process does not indicate what part of a document was signed. However, in a case of XAdES signature XPath transformations presented in the signature will be applied, in the case of CAdES or PAdES signature the whole document must be signed.

At the end of the validation process four reports are created. They contain the different detail levels concerning the validation result. They provide four kinds of visions for the validation process: macroscopic, microscopic, input data and ETSI Validation report conformant with the standard [\[R09\]](#). For more information about these reports, please refer to "Simple Report" chapter.

Below is the simplest example of the validation of the signature of a document. The first thing to do is instantiating an object named validator, which orchestrates the verification of the different rules. To perform this it is necessary to invoke a static method `fromDocument()` on the abstract class `SignedDocumentValidator`. This method returns the object in question whose type is chosen dynamically based on the type of source document.

The next step is to create an object that will check the status of a certificate using the Trusted List model (see "Trusted Lists of Certification Service Provider" for more information). In order to achieve this, an instance of a `CertificateVerifier` must be created with a defined source of trusted certificates. In our example, the trusted source is instantiated with `CommonTrustedCertificateSource` class. As well as a trusted source the `CertificateVerifier` object needs an OCSP and/or CRL source and a TSL source (which defines how the certificates are retrieved from the Trusted Lists). See chapter "Management of CRL and OCSP Sources" for more information concerning sources.

Validation of a signature

```
// First, we need a Certificate verifier
CertificateVerifier cv = new CommonCertificateVerifier();

// We can inject several sources. eg: OCSP, CRL, AIA, trusted lists

// Capability to download resources from AIA
cv.setAIAsource(new DefaultAIAsource());

// Capability to request OCSP Responders
cv.setOcspSource(new OnlineOCSPSource());

// Capability to download CRL
cv.setCrlSource(new OnlineCRLSource());

// Create an instance of a trusted certificate source
CommonTrustedCertificateSource trustedCertSource = new
CommonTrustedCertificateSource();
// import the keystore as trusted
trustedCertSource.importAsTrusted(keystoreCertSource);

// Add trust anchors (trusted list, keystore,...) to a list of trusted certificate
sources
// Hint : use method {@code CertificateVerifier.setTrustedCertSources(certSources)} in
order to overwrite the existing list
cv.addTrustedCertSources(trustedCertSource);

// Additionally add missing certificates to a list of adjunct certificate sources
cv.addAdjunctCertSources(adjunctCertSource);

// Here is the document to be validated (any kind of signature file)
DSSDocument document = new FileDocument(new File("src/test/resources/signature-
pool/signedXmlXadesLT.xml"));

// We create an instance of DocumentValidator
// It will automatically select the supported validator from the classpath
SignedDocumentValidator documentValidator =
SignedDocumentValidator.fromDocument(document);

// We add the certificate verifier (which allows to verify and trust certificates)
documentValidator.setCertificateVerifier(cv);

// Here, everything is ready. We can execute the validation (for the example, we use
the default and embedded
// validation policy)
Reports reports = documentValidator.validateDocument();

// We have 3 reports
// The diagnostic data which contains all used and static data
DiagnosticData diagnosticData = reports.getDiagnosticData();
```

```
// The detailed report which is the result of the process of the diagnostic data and
// the validation policy
DetailedReport detailedReport = reports.getDetailedReport();

// The simple report is a summary of the detailed report (more user-friendly)
SimpleReport simpleReport = reports.getSimpleReport();
```



When using the TrustedListsCertificateSource class, for performance reasons, consider creating a single instance of this class and initialize it only once.



In general, the signature must cover the entire document so that the DSS framework can validate it. However, for example in the case of a XAdES signature, some transformations can be applied on the XML document. They can include operations such as canonicalization, encoding/decoding, XSLT, XPath, XML schema validation, or XInclude. XPath transforms permit the signer to derive an XML document that omits portions of the source document. Consequently those excluded portions can change without affecting signature validity.

SignedDocumentValidator

For execution of the validation process, DSS uses the 'SignedDocumentValidator' class. The DSS framework provides five implementations of validator:

- **XMLDocumentValidator** - validates documents in XML format (XAdES format);
- **CMSDocumentValidator** - validates documents in CMS format (CAdES format);
- **PDFDocumentValidator** - validates documents in PDF format (PADES format);
- **JWSCompactDocumentValidator** - validates documents with base64url encoded content (JAdES compact format);
- **JWSSerializationDocumentValidator** - validates documents in JSON format (JAdES serialization formats);
- **ASICContainerWithXAdESValidator** - validates ASIC with XAdES containers;
- **ASICContainerWithCAdESValidator** - validates ASIC with CAdES containers;
- **DetachedTimestampValidator** - validates CMS timestamps provided alone.

DSS initializes a relevant validator based on specific characteristics of an input file (e.g. a PDF file version declaration for a PDF file). It checks the file format and loads the required validator from a classpath. Below you can find a list of settings that can be used for the configuration of the class.

SignedDocumentValidator usage

```
// The method allows instantiation of a related validator for a provided document
// independently on its format (the target dss module must be added as dependency)
SignedDocumentValidator documentValidator =
SignedDocumentValidator.fromDocument(document);
```

```

// Allows specifying a custom certificate verifier (online or offline)
documentValidator.setCertificateVerifier(new CommonCertificateVerifier());

// Allows specifying which tokens need to be extracted in the diagnostic data
// (Base64).
// Default : NONE
documentValidator.setTokenExtractionStrategy(TokenExtractionStrategy.EXTRACT_CERTIFICA
TES_AND_TIMESTAMPS);

// Allows providing signing certificate(s) in the explicit way, in case if the
// certificate is not provided in the signature itself (can be used for non-ASIC
// signatures)
CertificateSource signingCertificateSource = new CommonCertificateSource();
signingCertificateSource.addCertificate(DSSUtils.loadCertificateFromBase64EncodedString(
    "MIIC9TCCAd2gAwIBAgIBAjANBgkqhkiG9w0BAQUFADArMQswCQYDVQQGEwJBQTEMMAoGA1UEChMDRFNTMQ4wD
    AYDVQQDEwVJQ0EgQTAeFw0xMzEyMDIxNzMzMTBaFw0xNTEyMDIxNzMzMTBaMDAxCzAJBgNVBAYTAKFBMQuwCgY
    DVQQKEwNEU1MxEzARBgNVBAMTCnVzZXIgQSBSU0EwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJUHHAphm
    SDdQ1t62tppK+dLTANsE2nAj+HCpasS3ohLBrhteRsvTAbryDyIzCmTYWu/nVI4TGbzbESwV/Qit1koMLpYFw
    32MIBf2DLmEcGJ3vm5haw6u8S9quR1h8Vu7QWd+5KMaBZuR+j91RiSuoY0xS2ZQxJw1vhvW9hRYjAgMBAAGjg
    aIwgZ8wCQYDVR0TBAIwADAdBgNVHQ4EFgQU9ESnTWfwg13c3LQZzqqwibY5WVYwUwYDVR0jBEwwSoAUI01CDsB
    SUcEoFZxKaWf1PAL1U+uhL6QtMCsxDDAKBgNVBAoTA0RTUzELMAkGA1UEBhMCQUExDjAMBgNVBAMTBVJDQSBBg
    gEBMAsGA1UdDwQEAvIHgDARBgNVHSAECjAIMAYGBFUdIAAwDQYJKoZIhvcNAQEFBQADggEBAGnhhnoyVUhDnr/
    BSbZ/uWfSuwzFPG+2V9K6WxdIaaX0ORFGIdFwGLAwA/Qzpq9snfBxuTkAykxq0uEdhHTj0qXxWRjQ+Dop/Drmc
    coF/zDvgGusyY1YXaABd/kc3IYt7ns7z3tpiqIz4A7a/UHplBRXfqjyaZurZuJQRaSdxh6CNhdEUiUBxkbb1Sd
    Mju0gjzSDjcDjcegjvDquMKdDetvtu2Qh4ConBBo3fUImwIFRWnbudS5H2HE18ikC7gY/QIuNr7USf1PNyUgcG
    2g31cMtemj7UTBHZ2V/jPf7ZXqwfVSAyKNVm3weAI6R3PI0STjdxN6a9qjt9xld40YEdw="));
documentValidator.setSigningCertificateSource(signingCertificateSource);

// Sets the detached contents that were used for the detached signature creation
documentValidator.setDetachedContents(Arrays.asList(new InMemoryDocument("Hello
world!".getBytes())));

// Allows defining a custom Process Executor
// By default used {@code new DefaultSignatureProcessExecutor()}
documentValidator.setProcessExecutor(new DefaultSignatureProcessExecutor());

// Sets custom Signature Policy Provider
documentValidator.setSignaturePolicyProvider(new SignaturePolicyProvider());

// Sets an expected signature validation level
// The recommended level is ARCHIVAL_DATA (maximal level of the validation)
// Default : ValidationLevel.ARCHIVAL_DATA
documentValidator.setValidationLevel(ValidationLevel.ARCHIVAL_DATA);

// Sets if the ETSI validation report must be created
// If true, it will become accessible through the method below
// Default : true
documentValidator.setEnableEtsiValidationReport(true);

```

```

// Sets if the semantics for Indication / SubIndication must be included in the
// Simple Report (see table 5 / 6 of the ETSI TS 119 102-1)
// Default : false
documentValidator.setIncludeSemantics(true);

// Executes the validation process and produces validation reports:
// Simple report, Detailed report, Diagnostic data and ETSI Validation Report (if
// enabled)
Reports reports = documentValidator.validateDocument();

// Returns ETSI Validation Report (if enabled, NULL otherwise)
ValidationReportType etsiValidationReport = reports.getEtsiValidationReportJaxb();

```

Validation Result Materials

The result of the validation process consists of three elements:

- the Simple Report,
- the Detailed Report,
- the Diagnostic Data and
- the ETSI Validation Report.

All these reports are encoded using XML, which allows the implementer to easily manipulate and extract information for further analysis. For each report, XML Schema and JaxB model are available as maven dependencies.

DSS also provides XSLT to able to generate PDF or HTML reports (simple and detailed reports).

You will find below a detailed description of each of these elements.

Simple Report

This is a sample of the simple validation report:

Simple Report

```

<SimpleReport ValidationTime="2021-07-09T10:04:33Z"
xmlns="http://dss.esig.europa.eu/validation/simple-report">
  <ValidationPolicy>
    <PolicyName>QES AdESQC TL based</PolicyName>
    <PolicyDescription>Validate electronic signatures and indicates whether they
    are Advanced electronic Signatures (AdES), AdES supported by a Qualified Certificate
    (AdES/QC) or a
      Qualified electronic Signature (QES). All certificates and their related
      chains supporting the signatures are validated against the EU Member State Trusted
      Lists (this includes
        signer's certificate and certificates used to validate certificate
        validity status services - CRLs, OCSP, and time-stamps).
    </PolicyDescription>
  </ValidationPolicy>
</SimpleReport>

```

```

</ValidationPolicy>
<DocumentName>json-flattened-serialization.json</DocumentName>
<ValidSignaturesCount>0</ValidSignaturesCount>
<SignaturesCount>1</SignaturesCount>
<Signature SignatureFormat="JAdES-BASELINE-B" Id="S-EC1FD6906412EED291F00796F969D1D86786EFDF0E3700CF74BA65567B4B28">
    <CertificateChain>
        <Certificate>
            <id>C-  
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014</id>
                <qualifiedName>good-user</qualifiedName>
            </Certificate>
            <Certificate>
                <id>C-  
DD7496F8B6C574ECFA0B356A3D1A00DCFEC2111A3094B3B148A978D3C31F6126</id>
                    <qualifiedName>good-ca</qualifiedName>
                </Certificate>
                <Certificate>
                    <id>C-  
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F</id>
                        <qualifiedName>root-ca</qualifiedName>
                    </Certificate>
                </CertificateChain>
                <Indication>INDETERMINATE</Indication>
                <SubIndication>TRY_LATER</SubIndication>
                <AdESValidationDetails>
                    <Error Key="BBB_XCV_SUB_ANS">The certificate validation is not conclusive!</Error>
                    <Error Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the certificate!</Error>
                    <Error Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation data for the certificate!</Error>
                </AdESValidationDetails>
                <QualificationDetails>
                    <Error Key="QUAL_CERT_TRUSTED_LIST_REACHED_ANS">Unable to build a certificate chain up to a trusted list!</Error>
                    <Warning Key="QUAL_IS_ADES_IND">The signature/seal is an INDETERMINATE AdES digital signature!</Warning>
                </QualificationDetails>
                <SigningTime>2021-07-09T10:04:31Z</SigningTime>
                <BestSignatureTime>2021-07-09T10:04:33Z</BestSignatureTime>
                <SignedBy>good-user</SignedBy>
                <SignatureLevel description="Not applicable">N/A</SignatureLevel>
                <SignatureScope scope="FULL">Full document</SignatureScope>
            </Signature>
    </SimpleReport>

```

The result of the validation process is based on very complex rules. The purpose of this report is to make as simple as possible the information while keeping the most important elements. Thus the end user can, at a glance, have a synthetic view of the validation. To build this report the

framework uses some simple rules and the detailed report as input.

Detailed Report

This is a sample of the detailed validation report. Its structure is based on the ETSI standard [R09] and is built around Basic Building Blocks, Basic Validation Data, Timestamp Validation Data, AdES-T Validation Data and Long Term Validation Data. Some segments were deleted to make reading easier. They are marked by three dots:

Detailed Report

```
<DetailedReport ValidationTime="2021-07-09T10:04:33Z"
  xmlns="http://dss.esig.europa.eu/validation/detailed-report">
  <Signature Id="S-
  EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28">
    <ValidationProcessBasicSignature Title="Validation Process for Basic
  Signatures">
      <Constraint Id="S-
  EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28-FC">
        <Name Key="BSV_IFCRC">Is the result of the 'Format Checking' building
  block conclusive?</Name>
        <Status>OK</Status>
      </Constraint>
      <Constraint Id="S-
  EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28-ISC">
        <Name Key="BSV_IISCRC">Is the result of the 'Identification of Signing
  Certificate' building block conclusive?</Name>
        <Status>OK</Status>
      </Constraint>
      <Constraint Id="S-
  EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28-VCI">
        <Name Key="BSV_IVCIRC">Is the result of the 'Validation Context
  Initialization' building block conclusive?</Name>
        <Status>OK</Status>
      </Constraint>
      <Constraint Id="S-
  EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28-XCV">
        <Name Key="BSV_IIXCVR">Is the result of the 'X.509 Certificate
  Validation' building block conclusive?</Name>
        <Status>WARNING</Status>
        <Warning Key="BSV_IIXCVR_ANS">The result of the 'X.509 Certificate
  Validation' building block is not conclusive!</Warning>
      </Constraint>
      <Constraint Id="S-
  EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28-XCV">
        <Name Key="BSV_ISCRAVTC">Is the signing certificate not revoked at
  validation time?</Name>
        <Status>OK</Status>
      </Constraint>
      <Constraint Id="S-
  EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28-XCV">
```

```

        <Name Key="BSV_IVTAVRSC">Is the validation time in the validity range
of the signing certificate?</Name>
        <Status>OK</Status>
        </Constraint>
        <Constraint Id="S-
EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28-CV">
            <Name Key="BSV_ICVRC">Is the result of the 'Cryptographic
Verification' building block conclusive?</Name>
            <Status>OK</Status>
            </Constraint>
            <Constraint Id="S-
EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28">
                <Name Key="ADEST_ROBVPIIC">Is the result of the Basic Validation
Process conclusive?</Name>
                <Status>NOT OK</Status>
                <Error Key="ADEST_ROBVPIIC_ANS">The result of the Basic validation
process is not conclusive!</Error>
                <AdditionalInfo>Basic Signature Validation process failed with
INDETERMINATE/TRY_LATER indication</AdditionalInfo>
                </Constraint>
                <Conclusion>
                    <Indication>INDETERMINATE</Indication>
                    <SubIndication>TRY_LATER</SubIndication>
                    <Errors Key="BBB_XCV_SUB_ANS">The certificate validation is not
conclusive!</Errors>
                    <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
                    <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
                </Conclusion>
                <ProofOfExistence>
                    <Time>2021-07-09T10:04:33Z</Time>
                </ProofOfExistence>
            </ValidationProcessBasicSignature>
            <ValidationProcessLongTermData Title="Validation Process for Signatures with
Time and Signatures with Long-Term Validation Data">
                <Constraint>
                    <Name Key="LTV_ABSV">Is the result of the Basic Validation Process
acceptable?</Name>
                    <Status>OK</Status>
                </Constraint>
                <Constraint>
                    <Name Key="BBB_XCV_IARDPFC">Is an acceptable revocation data present
for the certificate?</Name>
                    <Status>NOT OK</Status>
                    <Error Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation data for the
certificate!</Error>
                    <AdditionalInfo>Certificate Id = C-
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014</AdditionalInfo>
                </Constraint>
                <Conclusion>

```

```

        <Indication>INDETERMINATE</Indication>
        <SubIndication>TRY_LATER</SubIndication>
        <Errors Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation data for
the certificate!</Errors>
    </Conclusion>
    <ProofOfExistence>
        <Time>2021-07-09T10:04:33Z</Time>
    </ProofOfExistence>
</ValidationProcessLongTermData>
<ValidationProcessArchivalData Title="Validation Process for Signatures with
Archival Data">
    <Constraint>
        <Name Key="ARCH_LTVV">Is the result of the LTV validation process
acceptable?</Name>
        <Status>NOT OK</Status>
        <Error Key="ARCH_LTVV_ANS">The result of the LTV validation process is
not acceptable to continue the process!</Error>
    </Constraint>
    <Conclusion>
        <Indication>INDETERMINATE</Indication>
        <SubIndication>TRY_LATER</SubIndication>
        <Errors Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation data for
the certificate!</Errors>
    </Conclusion>
    <ProofOfExistence>
        <Time>2021-07-09T10:04:33Z</Time>
    </ProofOfExistence>
</ValidationProcessArchivalData>
<ValidationSignatureQualification SignatureQualification="N/A"
Title="Signature Qualification">
    <Constraint>
        <Name Key="QUAL_IS_ADES">Is the signature/seal an acceptable AdES
digital signature (ETSI EN 319 102-1)?</Name>
        <Status>WARNING</Status>
        <Warning Key="QUAL_IS_ADES_IND">The signature/seal is an INDETERMINATE
AdES digital signature!</Warning>
    </Constraint>
    <Constraint>
        <Name Key="QUAL_CERT_TRUSTED_LIST_REACHED">Has a trusted list been
reached for the certificate chain?</Name>
        <Status>NOT OK</Status>
        <Error Key="QUAL_CERT_TRUSTED_LIST_REACHED_ANS">Unable to build a
certificate chain up to a trusted list!</Error>
    </Constraint>
    <Conclusion>
        <Indication>FAILED</Indication>
        <Errors Key="QUAL_CERT_TRUSTED_LIST_REACHED_ANS">Unable to build a
certificate chain up to a trusted list!</Errors>
        <Warnings Key="QUAL_IS_ADES_IND">The signature/seal is an
INDETERMINATE AdES digital signature!</Warnings>
    </Conclusion>

```

```

</ValidationSignatureQualification>
<Conclusion>
    <Indication>INDETERMINATE</Indication>
    <SubIndication>TRY_LATER</SubIndication>
</Conclusion>
</Signature>
<BasicBuildingBlocks Id="S-
EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28" Type="SIGNATURE">
    <FC Title="Format Checking">
        <Constraint>
            <Name Key="BBB_FC_IEFF">Does the signature format correspond to an
expected format?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name Key="BBB_FC_ISD">Is the signature identification not
ambiguous?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name Key="BBB_FC_ISRIA">Is the signed references identification not
ambiguous?</Name>
            <Status>OK</Status>
        </Constraint>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
    </FC>
    <ISC Title="Identification of the Signing Certificate">
        <Constraint>
            <Name Key="BBB_ICS_ISCI">Is there an identified candidate for the
signing certificate?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name Key="BBB_ICS_ISACDP">Is the signed attribute: 'cert-digest' of
the certificate present?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name Key="BBB_ICS_ICDVV">Does the certificate digest value match a
digest value found in the certificate reference(s)?</Name>
            <Status>OK</Status>
        </Constraint>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
        <CertificateChain>
            <ChainItem Id="C-
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014">
                <Source>SIGNATURE</Source>

```

```

        </ChainItem>
        <ChainItem Id="C-
DD7496F8B6C574ECFA0B356A3D1A00DCFEC2111A3094B3B148A978D3C31F6126">
            <Source>SIGNATURE</Source>
        </ChainItem>
        <ChainItem Id="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F">
            <Source>TRUSTED_STORE</Source>
        </ChainItem>
    </CertificateChain>
</ISC>
<VCI Title="Validation Context Initialization">
    <Constraint>
        <Name Key="BBB_VCI_ISPK">Is the signature policy known?</Name>
        <Status>OK</Status>
    </Constraint>
    <Conclusion>
        <Indication>PASSED</Indication>
    </Conclusion>
</VCI>
<XCV Title="X509 Certificate Validation">
    <Constraint>
        <Name Key="BBB_XCV_CCCBB">Can the certificate chain be built till a
trust anchor?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint Id="C-
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014" BlockType="SUB_XCV">
        <Name Key="BBB_XCV_SUB">Is the certificate validation
conclusive?</Name>
        <Status>NOT OK</Status>
        <Error Key="BBB_XCV_SUB_ANS">The certificate validation is not
conclusive!</Error>
        </Constraint>
        <Conclusion>
            <Indication>INDETERMINATE</Indication>
            <SubIndication>TRY_LATER</SubIndication>
            <Errors Key="BBB_XCV_SUB_ANS">The certificate validation is not
conclusive!</Errors>
            <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
            <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
        </Conclusion>
        <SubXCV Id="C-
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014" TrustAnchor="false"
SelfSigned="false" Title="Certificate">
            <Constraint>
                <Name Key="QUAL_UNIQUE_CERT">Is the certificate unique?</Name>
                <Status>OK</Status>
            </Constraint>

```

```

<Constraint>
  <Name Key="BBB_XCV_PSEUDO_USE">Is a pseudonym used?</Name>
  <Status>OK</Status>
</Constraint>
<Constraint>
  <Name Key="BBB_XCV_ISNSSC">Is certificate not self-signed?</Name>
  <Status>OK</Status>
</Constraint>
<Constraint>
  <Name Key="BBB_XCV_ICSI">Is the certificate signature
intact?</Name>
  <Status>OK</Status>
</Constraint>
<Constraint>
  <Name Key="BBB_XCV_ISCGKU">Does the signer's certificate have an
expected key-usage?</Name>
  <Status>OK</Status>
  <AdditionalInfo>Key usage : [NON_REPUDIATION]</AdditionalInfo>
</Constraint>
<Constraint>
  <Name Key="BBB_XCV_AIA_PRES">Is the authority info access
present?</Name>
  <Status>OK</Status>
</Constraint>
<Constraint>
  <Name Key="BBB_XCV_REVOC_PRES">Is the revocation info access
present?</Name>
  <Status>OK</Status>
</Constraint>
<Constraint>
  <Name Key="BBB_XCV_IRDPFC">Is the revocation data present for the
certificate?</Name>
  <Status>NOT OK</Status>
  <Error Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Error>
</Constraint>
<Conclusion>
  <Indication>INDETERMINATE</Indication>
  <SubIndication>TRY_LATER</SubIndication>
  <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
</Conclusion>
<RFC Title="Revocation Freshness Checker">
<Constraint>
  <Name Key="BBB_XCV_IARDPFC">Is an acceptable revocation data
present for the certificate?</Name>
  <Status>NOT OK</Status>
  <Error Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation data
for the certificate!</Error>
</Constraint>
<Conclusion>

```

```

        <Indication>INDETERMINATE</Indication>
        <SubIndication>TRY_LATER</SubIndication>
        <Errors Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation
data for the certificate!</Errors>
        </Conclusion>
        </RFC>
        </SubXCV>
        <SubXCV Id="C-
DD7496F8B6C574ECFA0B356A3D1A00DCFEC2111A3094B3B148A978D3C31F6126" TrustAnchor="false"
SelfSigned="false" Title="Certificate">
        <Constraint>
            <Name Key="BBB_XCV_ICSI">Is the certificate signature
intact?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name Key="BBB_XCV_IRDPFC">Is the revocation data present for the
certificate?</Name>
            <Status>NOT OK</Status>
            <Error Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Error>
        </Constraint>
        <Conclusion>
            <Indication>INDETERMINATE</Indication>
            <SubIndication>TRY_LATER</SubIndication>
            <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
        </Conclusion>
        <RFC Title="Revocation Freshness Checker">
        <Constraint>
            <Name Key="BBB_XCV_IARDPFC">Is an acceptable revocation data
present for the certificate?</Name>
            <Status>NOT OK</Status>
            <Error Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation data
for the certificate!</Error>
        </Constraint>
        <Conclusion>
            <Indication>INDETERMINATE</Indication>
            <SubIndication>TRY_LATER</SubIndication>
            <Errors Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation
data for the certificate!</Errors>
        </Conclusion>
        </RFC>
        </SubXCV>
        <SubXCV Id="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F" TrustAnchor="true"
SelfSigned="true" Title="Certificate">
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
        </SubXCV>

```

```

</XCV>
<CV Title="Cryptographic Verification">
    <Constraint>
        <Name Key="BBB_CV_IRDOF">Has the reference data object been
found?</Name>
        <Status>OK</Status>
        <AdditionalInfo>Reference : JWS_SIGNING_INPUT_DIGEST</AdditionalInfo>
    </Constraint>
    <Constraint>
        <Name Key="BBB_CV_IRDOI">Is the reference data object intact?</Name>
        <Status>OK</Status>
        <AdditionalInfo>Reference : JWS_SIGNING_INPUT_DIGEST</AdditionalInfo>
    </Constraint>
    <Constraint>
        <Name Key="BBB_CV_ISI">Is the signature intact?</Name>
        <Status>OK</Status>
    </Constraint>
    <Conclusion>
        <Indication>PASSED</Indication>
    </Conclusion>
</CV>
<SAV Title="Signature Acceptance Validation">
    <Constraint>
        <Name Key="BBB_SAV_ISSV">Is the structure of the signature
valid?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint>
        <Name Key="BBB_ICS_ISASCP">Is the signed attribute: 'signing-
certificate' present?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint>
        <Name Key="BBB_ICS_ISASCPU">Is the signed attribute: 'signing-
certificate' present only once?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint>
        <Name Key="BBB_SAV_DSCACRCC">Does the 'Signing Certificate' attribute
contain references only to the certificate chain?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint>
        <Name Key="BBB_SAV_ISQPSTP">Is the signed qualifying property:
'signing-time' present?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint>
        <Name Key="ACCM">Are cryptographic constraints met for the signature
creation?</Name>
        <Status>OK</Status>

```

```

        <AdditionalInfo>Signature algorithm RSA with SHA256 with key size 2048
at validation time : 2021-07-09 10:04</AdditionalInfo>
        </Constraint>
        <Constraint>
            <Name Key="ACCM">Are cryptographic constraints met for the jws signing
input?</Name>
            <Status>OK</Status>
            <AdditionalInfo>Digest algorithm SHA256 at validation time : 2021-07-
09 10:04 for jws signing input</AdditionalInfo>
        </Constraint>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
        <CryptographicValidation secure="true">
            <Algorithm>
                <Name>RSA with SHA256</Name>
                <Uri>http://www.w3.org/2001/04/xmldsig-more#rsa-sha256</Uri>
                <KeyLength>2048</KeyLength>
            </Algorithm>
            <NotAfter>2022-12-31T23:00:00Z</NotAfter>
            <ValidationTime>2021-07-09T10:04:33Z</ValidationTime>
            <ConcernedMaterial>S-
EC1FD6906412EED291F00796F969D1D86786EFDFFF0E3700CF74BA65567B4B28</ConcernedMaterial>
            </CryptographicValidation>
        </SAV>
        <CertificateChain>
            <ChainItem Id="C-
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014">
                <Source>SIGNATURE</Source>
            </ChainItem>
            <ChainItem Id="C-
DD7496F8B6C574ECFA0B356A3D1A00DCFEC211A3094B3B148A978D3C31F6126">
                <Source>SIGNATURE</Source>
            </ChainItem>
            <ChainItem Id="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F">
                <Source>TRUSTED_STORE</Source>
            </ChainItem>
        </CertificateChain>
        <Conclusion>
            <Indication>INDETERMINATE</Indication>
            <SubIndication>TRY_LATER</SubIndication>
            <Errors Key="BBB_XCV_SUB_ANS">The certificate validation is not
conclusive!</Errors>
            <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
            <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
        </Conclusion>
        </BasicBuildingBlocks>
    </DetailedReport>

```

For example the Basic Building Blocks are divided into seven elements:

- FC - Format Checking
- ISC - Identification of the Signing Certificate
- VCI - Validation Context Initialization
- RFC - Revocation Freshness Checker
- XCV - X.509 certificate validation
- CV - Cryptographic Verification
- SAV - Signature Acceptance Validation

The following additional elements also can be executed in case of validation in the past :

- PCV - Past Certificate Validation
- VTS - Validation Time Sliding process
- POE extraction - Proof Of Existence extraction
- PSV - Past Signature Validation

Past certificate/signature validation is used when basic validation of a certificate/signature fails at the current time with an INDETERMINATE status such that the provided proofs of existence may help to go to a determined status. The process shall initialize the *best-signature-time* either to a time indication for a related POE provided, or the current time when this parameter has not been used by the algorithm.

- **Best-signature-time** is an internal variable for the algorithm denoting the earliest time when it can be trusted by the SVA (either because proven by some POE present in the signature or passed by the DA and for this reason assumed to be trusted) that a signature has existed. [\[R09\]](#)

Each block contains a number of rules that are executed sequentially. The rules are driven by the constraints defined in the validation policy. The result of each rule is OK or NOT OK. The process is stopped when the first rule fails. Each block also contains a conclusion. If all rules are met then the conclusion node indicates PASSED. Otherwise FAILED or INDETERMINATE indication is returned depending on the ETSI standard definition.

Diagnostic Data

This is a data set constructed from the information contained in the signature itself, but also from information retrieved dynamically as revocation data and information extrapolated as the mathematical validity of a signature. All this information is independent of the applied validation policy. Two different validation policies applied to the same diagnostic data can lead to different results.

This is an example of the diagnostic data for a XAdES signature. Certain fields and certain values were trimmed or deleted to make reading easier:

Diagnostic Data

```

<DiagnosticData xmlns="http://dss.esig.europa.eu/validation/diagnostic">
    <DocumentName>json-flattened-serialization.json</DocumentName>
    <ValidationDate>2021-07-09T10:04:33Z</ValidationDate>
    <Signatures>
        <Signature Id="S-
EC1FD6906412EED291F00796F969D1D86786EFDF0E3700CF74BA65567B4B28">
            <SignatureFilename>json-flattened-serialization.json</SignatureFilename>
            <ClaimedSigningTime>2021-07-09T10:04:31Z</ClaimedSigningTime>
            <SignatureFormat>JAdES-BASELINE-B</SignatureFormat>
            <StructuralValidation valid="true"/>
            <DigestMatchers>
                <DigestMatcher type="JWS_SIGNING_INPUT_DIGEST">
                    <DigestMethod>SHA256</DigestMethod>
                </DigestMatcher>
            </DigestMatchers>
            <BasicSignature>
                <EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
                    <KeyLengthUsedToSignThisToken>2048</KeyLengthUsedToSignThisToken>
                    <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
                    <SignatureIntact>true</SignatureIntact>
                    <SignatureValid>true</SignatureValid>
                </BasicSignature>
                <SigningCertificate Certificate="C-
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014"/>
                    <CertificateChain>
                        <ChainItem Certificate="C-
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014"/>
                            <ChainItem Certificate="C-
DD7496F8B6C574ECFA0B356A3D1A00DCFEC2111A3094B3B148A978D3C31F6126"/>
                                <ChainItem Certificate="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F"/>
                            </CertificateChain>
                            <MimeType>application/jose</MimeType>
                            <CommitmentTypeIndications/>
                            <FoundCertificates>
                                <RelatedCertificate Certificate="C-
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014">
                                    <Origin>KEY_INFO</Origin>
                                    <CertificateRef>
                                        <Origin>SIGNING_CERTIFICATE</Origin>
                                        <DigestAlgoAndValue match="true">
                                            <DigestMethod>SHA256</DigestMethod>
                                        </DigestAlgoAndValue>
                                    </CertificateRef>
                                </RelatedCertificate>
                            </FoundCertificates>
                        </ChainItem>
                    </CertificateChain>
                </BasicSignature>
            </DigestMatchers>
        </Signature>
    </Signatures>
    <DigestValue>zy6Ne5uUlSS4zmic/PM3pIuuDA1fpfjIlPwdaSJ7EBQ=</DigestValue>
    </DigestValue>
</DiagnosticData>

```

```

        </RelatedCertificate>
        <RelatedCertificate Certificate="C-
DD7496F8B6C574ECFA0B356A3D1A00DCFEC2111A3094B3B148A978D3C31F6126">
            <Origin>KEY_INFO</Origin>
        </RelatedCertificate>
    </FoundCertificates>
    <FoundRevocations/>
    <FoundTimestamps/>
    <SignatureScopes>
        <SignatureScope SignerData="D-
381C1562B49EA87ECF34FB8C61B4701BC4457A12F332AAFCFB3E96EF961B1769">
            <Scope>FULL</Scope>
            <Description>Full document</Description>
        </SignatureScope>
    </SignatureScopes>
    <SignatureDigestReference>
        <DigestMethod>SHA256</DigestMethod>

<DigestValue>Vj6MSVsMjSCqnaBoXq1aYEuD/4UPU5knyqY6GEpfQXY=</DigestValue>
        </SignatureDigestReference>
        <DataToBeSignedRepresentation>
            <DigestMethod>SHA256</DigestMethod>

<DigestValue>Rjh00YrhhAMoWAxz9geHPrfCi491xPCcbtHu0r4tiPM=</DigestValue>
        </DataToBeSignedRepresentation>

<SignatureValue>aZ7BLFCchb0kwDToef/x0eaKPBUVfuQ3b3PBGDUCeKv9ieGuMtzQRDmLrsBgc0H3m/70to
aIunChbASHImSx2u9HsC15DgjBYlnmf5eG2dkSWqG1EZt3i8q8WQT1T8zBcZfP3+Ghp+fzK8d4jq8q7+oBd6KX
grCkYZnz5STaUbi5rfC9Lh/GM06L3LkHJzJ3zpiFYSvUnK+RIFFFG8/nEICRmWP7VyaAXwlHvbyFBQ1tqW41OP
VXF9T/Rx0bmSwNiJTnzd/BAglGHpuab0jGVSwaaRq32V2j4EH0Bq39z6KpcG6rn30pXf2tc+WQD3di90S+P+5
I4hT4oJdmwE4Lg==</SignatureValue>
        </Signature>
    </Signatures>
    <UsedCertificates>
        <Certificate Id="C-
CF2E8D7B9B949524B8CE689CFCF337A48BAE0C0D5FA5F8C894FC1D69227B1014">
            <SubjectDistinguishedName Format="CANONICAL">c=lu,ou=pk1-test,o=nowina
solutions,cn=good-user</SubjectDistinguishedName>
            <SubjectDistinguishedName Format="RFC2253">C=LU,OU=PKI-TEST,O=Nowina
Solutions,CN=good-user</SubjectDistinguishedName>
            <IssuerDistinguishedName Format="CANONICAL">c=lu,ou=pk1-test,o=nowina
solutions,cn=good-ca</IssuerDistinguishedName>
            <IssuerDistinguishedName Format="RFC2253">C=LU,OU=PKI-TEST,O=Nowina
Solutions,CN=good-ca</IssuerDistinguishedName>
            <SerialNumber>10</SerialNumber>
            <CommonName>good-user</CommonName>
            <CountryName>LU</CountryName>
            <OrganizationName>Nowina Solutions</OrganizationName>
            <OrganizationalUnit>PKI-TEST</OrganizationalUnit>
            <AuthorityInformationAccessUrls>
                <aiaUrl>http://dss.nowina.lu/pki-factory/crt/good-ca.crt</aiaUrl>

```

```

</AuthorityInformationAccessUrls>
<CRLDistributionPoints/>
<OCSPAccessUrls>
    <ocspServerUrl>http://dss.nowina.lu/pki-factory/ocsp/good-
ca</ocspServerUrl>
</OCSPAccessUrls>
<Sources>
    <Source>SIGNATURE</Source>
</Sources>
<NotAfter>2022-02-22T16:30:08Z</NotAfter>
<NotBefore>2020-04-22T15:30:08Z</NotBefore>
<PublicKeySize>2048</PublicKeySize>
<PublicKeyEncryptionAlgo>RSA</PublicKeyEncryptionAlgo>
<EntityKey>PK-
ADA27354317627EE223DF7F84600E1231E4CD961CD27FE30AE81DFCB6710962E</EntityKey>
<KeyUsageBits>
    <KeyUsage>nonRepudiation</KeyUsage>
</KeyUsageBits>
<ExtendedKeyUsages/>
<IdPkixOcspNoCheck>false</IdPkixOcspNoCheck>
<BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
    <KeyLengthUsedToSignThisToken>2048</KeyLengthUsedToSignThisToken>
    <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
    <SignatureIntact>true</SignatureIntact>
    <SignatureValid>true</SignatureValid>
</BasicSignature>
<SigningCertificate Certificate="C-
DD7496F8B6C574ECFA0B356A3D1A00DCFEC2111A3094B3B148A978D3C31F6126"/>
<CertificateChain>
    <ChainItem Certificate="C-
DD7496F8B6C574ECFA0B356A3D1A00DCFEC2111A3094B3B148A978D3C31F6126"/>
    <ChainItem Certificate="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F"/>
    </CertificateChain>
    <Trusted>false</Trusted>
    <SelfSigned>false</SelfSigned>
    <CertificatePolicies/>
    <TrustedServiceProviders/>
    <Revocations/>
    <DigestAlgoAndValue>
        <DigestMethod>SHA256</DigestMethod>

<DigestValue>zy6Ne5uUlSS4zmic/PM3pIuuDA1fpfjIlPwdaSJ7EBQ=</DigestValue>
        </DigestAlgoAndValue>
    </Certificate>
    <Certificate Id="C-
DD7496F8B6C574ECFA0B356A3D1A00DCFEC2111A3094B3B148A978D3C31F6126">
        <SubjectDistinguishedName Format="CANONICAL">c=lu,ou=pk-test,o=nowina
solutions,cn=good-ca</SubjectDistinguishedName>

```

```

<SubjectDistinguishedName Format="RFC2253">C=LU,OU=PKI-TEST,O=Nowina
Solutions,CN=good-ca</SubjectDistinguishedName>
    <IssuerDistinguishedName Format="CANONICAL">c=lu,ou=pki-test,o=nowina
solutions,cn=root-ca</IssuerDistinguishedName>
    <IssuerDistinguishedName Format="RFC2253">C=LU,OU=PKI-TEST,O=Nowina
Solutions,CN=root-ca</IssuerDistinguishedName>
    <SerialNumber>4</SerialNumber>
    <CommonName>good-ca</CommonName>
    <CountryName>LU</CountryName>
    <OrganizationName>Nowina Solutions</OrganizationName>
    <OrganizationalUnit>PKI-TEST</OrganizationalUnit>
    <AuthorityInformationAccessUrls>
        <aiaUrl>http://dss.nowina.lu/pki-factory/crt/root-ca.crt</aiaUrl>
    </AuthorityInformationAccessUrls>
    <CRLDistributionPoints>
        <crlUrl>http://dss.nowina.lu/pki-factory/crl/root-ca.crl</crlUrl>
    </CRLDistributionPoints>
    <OCSPAccessUrls/>
    <Sources>
        <Source>SIGNATURE</Source>
    </Sources>
    <NotAfter>2022-02-22T16:30:06Z</NotAfter>
    <NotBefore>2020-04-22T15:30:06Z</NotBefore>
    <PublicKeySize>2048</PublicKeySize>
    <PublicKeyEncryptionAlgo>RSA</PublicKeyEncryptionAlgo>
    <EntityKey>PK-
F77AA76DB7798A59FAB9A37A7CAB3CC5ED6E55D3AE47C348C13EC684009D7431</EntityKey>
    <KeyUsageBits>
        <KeyUsage>keyCertSign</KeyUsage>
        <KeyUsage>crlSign</KeyUsage>
    </KeyUsageBits>
    <ExtendedKeyUsages/>
    <IdPkixOcspNoCheck>false</IdPkixOcspNoCheck>
    <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
    <KeyLengthUsedToSignThisToken>2048</KeyLengthUsedToSignThisToken>
    <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
    <SignatureIntact>true</SignatureIntact>
    <SignatureValid>true</SignatureValid>
</BasicSignature>
    <SigningCertificate Certificate="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F"/>
    <CertificateChain>
        <ChainItem Certificate="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F"/>
    </CertificateChain>
    <Trusted>false</Trusted>
    <SelfSigned>false</SelfSigned>
    <CertificatePolicies/>
    <Revocations/>

```

```

<DigestAlgoAndValue>
    <DigestMethod>SHA256</DigestMethod>

<DigestValue>3XSW+LbFd0z6CzVqPRoA3P7CERowL0xSK1408MfYSY=</DigestValue>
    </DigestAlgoAndValue>
</Certificate>
<Certificate Id="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F">
    <SubjectDistinguishedName Format="CANONICAL">c=lu,ou=pk-i-test,o=nowina
solutions,cn=root-ca</SubjectDistinguishedName>
    <SubjectDistinguishedName Format="RFC2253">C=LU,OU=PKI-TEST,O=Nowina
Solutions,CN=root-ca</SubjectDistinguishedName>
    <IssuerDistinguishedName Format="CANONICAL">c=lu,ou=pk-i-test,o=nowina
solutions,cn=root-ca</IssuerDistinguishedName>
    <IssuerDistinguishedName Format="RFC2253">C=LU,OU=PKI-TEST,O=Nowina
Solutions,CN=root-ca</IssuerDistinguishedName>
    <SerialNumber>1</SerialNumber>
    <CommonName>root-ca</CommonName>
    <CountryName>LU</CountryName>
    <OrganizationName>Nowina Solutions</OrganizationName>
    <OrganizationalUnit>PKI-TEST</OrganizationalUnit>
    <AuthorityInformationAccessUrls/>
    <CRLDistributionPoints/>
    <OCSPAccessUrls/>
    <Sources>
        <Source>TRUSTED_STORE</Source>
    </Sources>
    <NotAfter>2022-03-22T16:30:05Z</NotAfter>
    <NotBefore>2020-03-22T16:30:05Z</NotBefore>
    <PublicKeySize>2048</PublicKeySize>
    <PublicKeyEncryptionAlgo>RSA</PublicKeyEncryptionAlgo>
    <EntityKey>PK-
349206CC34F40C66DB58BB8C0D7BF6551826FE82A9B359F872523442F9E4FE7C</EntityKey>
    <KeyUsageBits>
        <KeyUsage>keyCertSign</KeyUsage>
        <KeyUsage>crlSign</KeyUsage>
    </KeyUsageBits>
    <ExtendedKeyUsages/>
    <IdPkixOcspNoCheck>false</IdPkixOcspNoCheck>
    <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
    <KeyLengthUsedToSignThisToken>2048</KeyLengthUsedToSignThisToken>
    <DigestAlgoUsedToSignThisToken>SHA512</DigestAlgoUsedToSignThisToken>
    <SignatureIntact>true</SignatureIntact>
    <SignatureValid>true</SignatureValid>
</BasicSignature>
<CertificateChain/>
<Trusted>true</Trusted>
<SelfSigned>true</SelfSigned>
<CertificatePolicies/>

```

```

<DigestAlgoAndValue>
    <DigestMethod>SHA256</DigestMethod>

<DigestValue>3k40bmntZwrArkHIFXNbIfk5TqiAvJJdRDqr1pPmY8=</DigestValue>
    </DigestAlgoAndValue>
</Certificate>
</UsedCertificates>
<UsedRevocations/>
<UsedTimestamps/>
<OriginalDocuments>
    <SignerData Id="D-
381C1562B49EA87ECF34FB8C61B4701BC4457A12F332AAFCFB3E96EF961B1769">
        <DigestAlgoAndValue>
            <DigestMethod>SHA256</DigestMethod>

<DigestValue>f40xZX/x/F05LcGBSKHWXfwtSx+j1ncoSt3SABJtkGk=</DigestValue>
        </DigestAlgoAndValue>
    </SignerData>
</OriginalDocuments>
<TrustedLists/>
</DiagnosticData>

```

ETSI Validation Report

The ETSI Validation Report represents an implementation of TS 119 102-2 (cf. [R13]). The report contains a standardized result of an ASiC digital signature validation. It includes the original validation input data, the applied validation policy, as well as the validation result of one or more signature(s) and its(their) constraints.

This is an example of the ETSI validation report:

ETSI Validation Report (TS 119 102-2)

```

<ValidationReport xmlns="http://uri.etsi.org/19102/v1.2.1#"
    xmlns:ns2="http://www.w3.org/2000/09/xmldsig#"
    xmlns:ns4="http://uri.etsi.org/02231/v2#"
    xmlns:ns3="http://uri.etsi.org/01903/v1.3.2#">
    <SignatureValidationReport>
        <SignatureIdentifier id="S-
08113A9BAB65F6271F837FF4992635CC725B49D27B1AED0D714EAD428BE98C6E">
            <DigestAlgAndValue>
                <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <ns2:DigestValue>e3CCNW+o+NudYnfnQfn5oPXdaV5eSmYYwFwBb3TvDLs=</ns2:DigestValue>
            </DigestAlgAndValue>

            <ns2:SignatureValue>m2sMUIIfYKHDVr1IMTyVufWJcwuxwhWjGvkF/xa/r0AKieZCe4LZPa0/uwzgwM4QAb
Pstdy4gHSQzCF0R6/fT9hv639kQS3TyZedw1raMeDj9mQ0wK01Ml10IxEI7jSf7xP6n62s0wQAhTJ1ARn0Y1G5
vppziVKb1vPED27HPBB4Yljn8j6hse+EJ0bwxAN1gwufbxZBvjHYgz/U/9EHafa1oGPcoIBrXvoUdzVX76sVE3

```

```

n1DvX4psEU4eq7paIZA7AWGSfWk8/k98pPqFcP2VYJaAju9GI+uZNMfRgPd0vGPxTjUBYiEyr3satod+cMQGiA
zie80n0ovQrfn7ebcA==</ns2:SignatureValue>
    <HashOnly>false</HashOnly>
    <DocHashOnly>false</DocHashOnly>
</SignatureIdentifier>
<ValidationConstraintsEvaluationReport>
    <ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:formatChecking</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:identificationOfTheSigningCertificate</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:validationContextInitialization</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:cryptographicVerification</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

```

```

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:signatureAcceptanceValidation</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:x509CertificateValidation</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:pastSignatureValidation</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:disabled</Status>
    </ConstraintStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:pastCertificateValidation</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:disabled</Status>
    </ConstraintStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:validationTimeSliding</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:disabled</Status>
    </ConstraintStatus>
</ValidationConstraint>

```

```

</ValidationConstraintsEvaluationReport>
<ValidationTimeInfo>
    <ValidationTime>2020-12-14T11:03:21Z</ValidationTime>
    <BestSignatureTime>
        <POETime>2020-12-14T11:03:21Z</POETime>
        <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
    </BestSignatureTime>
</ValidationTimeInfo>
<SignersDocument>
    <DigestAlgAndValue>
        <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<ns2:DigestValue>XC6PrK0RnL59qX5UJCibBUw/kPLQNtqQx81H+vd1PXw=</ns2:DigestValue>
    </DigestAlgAndValue>
    <SignersDocumentRef VOReference="D-
BD598965790CA5673910D64EFEDA798485364C66B6F4E7C49D23E1FF8BAFBBE8"/>
</SignersDocument>
<SignatureAttributes>
    <SigningTime Signed="true">
        <Time>2019-08-27T14:06:11Z</Time>
    </SigningTime>
    <SigningCertificate Signed="true">
        <AttributeObject VOReference="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F"/>
        </SigningCertificate>
    <DataObjectFormat Signed="true">
        <ContentType>1.2.840.113549.1.7.1</ContentType>
    </DataObjectFormat>
    <CompleteCertificateRefs/>
    <CompleteRevocationRefs/>
    <AttributeCertificateRefs/>
    <AttributeRevocationRefs/>
    <MessageDigest Signed="true">
        <Digest>SGEPVF0j/zskv8+n1zixt+PbLxWE9SS67rkpd0V5Wi4=</Digest>
    </MessageDigest>
    <Name>
        <NameElement>Pierrick Vandenbroucke (Signature)
70a3cb70f0f4d6513fb12cf0691965c58c7e7679</NameElement>
        </Name>
    <SubFilter>
        <SubFilterElement>ETSI.CAdES.detached</SubFilterElement>
    </SubFilter>
    <ByteRange>0 5340 43230 342</ByteRange>
    <Filter>
        <Filter>Adobe.PPKLite</Filter>
    </Filter>
</SignatureAttributes>
<SignerInformation Pseudonym="false">
    <SignerCertificate VOReference="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F"/>

```

```

<Signer>Pierrick Vandenbroucke (Signature)</Signer>
</SignerInformation>
<SignatureQuality>

<SignatureQualityInformation>urn:cef:dss:signatureQualification:QESig</SignatureQualit
yInformation>
    </SignatureQuality>
    <SignatureValidationProcess>

<SignatureValidationProcessID>urn:etsi:019102:validationprocess:LTA</SignatureValidati
onProcessID>
    </SignatureValidationProcess>
    <SignatureValidationStatus>
        <MainIndication>urn:etsi:019102:mainindication:total-
passed</MainIndication>
        <AssociatedValidationReportData>
            <TrustAnchor VOReference="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            <CertificateChain>
                <SigningCertificate VOReference="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F"/>
                <IntermediateCertificate VOReference="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA"/>
                <TrustAnchor VOReference="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            </CertificateChain>
            <CryptoInformation>
                <ValidationObject Id VOReference="S-
08113A9BAB65F6271F837FF4992635CC725B49D27B1AED0D714EAD428BE98C6E"/>
                <Algorithm>http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256</Algorithm>
                <SecureAlgorithm>true</SecureAlgorithm>
                <NotAfter>2022-12-31T23:00:00Z</NotAfter>
            </CryptoInformation>
            </AssociatedValidationReportData>
        </SignatureValidationStatus>
    </SignatureValidationReport>
    <SignatureValidationObjects>
        <ValidationObject id="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA">
            <ObjectType>urn:etsi:019102:validationObject:certificate</ObjectType>
            <ValidationObjectRepresentation>
                <DigestAlgAndValue>
                    <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                    <ns2:DigestValue>KT0Lo6MeXYKo4/rhJwmTL/3aREI+D3M/sB7xI+c+tNo=</ns2:DigestValue>
                </DigestAlgAndValue>
            </ValidationObjectRepresentation>
            <POE>
                <POETime>2020-12-14T11:03:21Z</POETime>
            </POE>
        </ValidationObject>
    </SignatureValidationObjects>

```

```

        <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
        </POE>
    </ValidationObject>
    <ValidationObject id="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F">
        <ObjectType>urn:etsi:019102:validationObject:certificate</ObjectType>
        <ValidationObjectRepresentation>
            <DigestAlgAndValue>
                <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<ns2:DigestValue>T6spAncn5Y5FG00LauVU0FXwWz2R1+DRayACjSJ9Gp8=</ns2:DigestValue>
            </DigestAlgAndValue>
        </ValidationObjectRepresentation>
        <POE>
            <POETime>2020-12-14T11:03:21Z</POETime>
            <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
        </POE>
    </ValidationObject>
    <ValidationObject id="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
        <ObjectType>urn:etsi:019102:validationObject:certificate</ObjectType>
        <ValidationObjectRepresentation>
            <DigestAlgAndValue>
                <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<ns2:DigestValue>cC3VwaCTzwqdcfrdm/mnxYV9iftztxboZyKLPCvrlo8=</ns2:DigestValue>
            </DigestAlgAndValue>
        </ValidationObjectRepresentation>
        <POE>
            <POETime>2020-12-14T11:03:21Z</POETime>
            <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
        </POE>
    </ValidationObject>
    <ValidationObject id="C-
B336CAA1F3C4930E4EF9C803C12877A004991EE9206C0D4AD3891688C1E478FF">
        <ObjectType>urn:etsi:019102:validationObject:certificate</ObjectType>
        <ValidationObjectRepresentation>
            <DigestAlgAndValue>
                <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<ns2:DigestValue>szbKofPEkw50+cgDwSh3oASZHukgbA1K04kWiMHkeP8=</ns2:DigestValue>
            </DigestAlgAndValue>
        </ValidationObjectRepresentation>
        <POE>
            <POETime>2020-12-14T11:03:21Z</POETime>
            <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
        </POE>
    </ValidationObject>

```

```

<ValidationObject id="C-
C3FBF37259AF0954EEEA4282DD1C7226A54E7150F7C29A2C495BA34DBFE09CA0">
    <ObjectType>urn:etsi:019102:validationObject:certificate</ObjectType>
    <ValidationObjectRepresentation>
        <DigestAlgAndValue>
            <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<ns2:DigestValue>w/vzclmvCVTu6kKC3RxyJqV0cVD3wposSVujTb/gnKA=</ns2:DigestValue>
        </DigestAlgAndValue>
    </ValidationObjectRepresentation>
    <POE>
        <POETime>2020-12-14T11:03:21Z</POETime>
        <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
    </POE>
</ValidationObject>
<ValidationObject id="R-
46B3B0FDA4A56A2601A20B31701AF2B1D0DCAEA8A903CD1346FBA80F49DFF1BD">
    <ObjectType>urn:etsi:019102:validationObject:CRL</ObjectType>
    <ValidationObjectRepresentation>
        <DigestAlgAndValue>
            <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<ns2:DigestValue>Rr0w/aSlaiYBogsxcBrysDcrqipA80TRvu0D0nf8b0=</ns2:DigestValue>
        </DigestAlgAndValue>
    </ValidationObjectRepresentation>
    <POE>
        <POETime>2020-12-14T11:03:21Z</POETime>
        <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
    </POE>
    <ValidationReport>
        <ValidationConstraintsEvaluationReport>
            <ValidationConstraint>
<ValidationConstraintIdentifier>urn:cef:dss:bbb:formatChecking</ValidationConstraintIdentifier>
                <ConstraintStatus>
                    <Status>urn:etsi:019102:constraintStatus:disabled</Status>
                </ConstraintStatus>
            </ValidationConstraint>
            <ValidationConstraint>
<ValidationConstraintIdentifier>urn:cef:dss:bbb:identificationOfTheSigningCertificate</ValidationConstraintIdentifier>
                <ConstraintStatus>
                    <Status>urn:etsi:019102:constraintStatus:applied</Status>
                </ConstraintStatus>
            </ValidationConstraint>
<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>

```

```

        </ValidationStatus>
    </ValidationConstraint>
    <ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:validationContextInitialization</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:disabled</Status>
    </ConstraintStatus>
    </ValidationConstraint>
    <ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:cryptographicVerification</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:signatureAcceptanceValidation</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:x509CertificateValidation</ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
</ValidationConstraintsEvaluationReport>
<SignerInformation>
    <SignerCertificate V0Reference="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
    <Signer>Belgium Root CA4</Signer>

```

```

        </SignerInformation>
        <SignatureValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
            <AssociatedValidationReportData>
                <TrustAnchor VOReference="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
                    <CertificateChain>
                        <SigningCertificate VOReference="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
                            <TrustAnchor VOReference="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
                                </CertificateChain>
                                <CryptoInformation>
                                    <ValidationObjectId VOReference="R-
46B3B0FDA4A56A2601A20B31701AF2B1D0DCAEA8A903CD1346FBA80F49DFF1BD"/>
                                        <Algorithm>http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256</Algorithm>
                                    <SecureAlgorithm>true</SecureAlgorithm>
                                    <NotAfter>2025-12-31T23:00:00Z</NotAfter>
                                </CryptoInformation>
                                <AssociatedValidationReportData>
                            </SignatureValidationStatus>
                        </ValidationReport>
                    </ValidationObject>
                    <ValidationObject id="R-
83B6DB1D1C8369D90161C212DEBCE29E2608D9B117356B8AA9E531721CB1F5BF">
                        <ObjectType>urn:etsi:019102:validationObject:OCSPResponse</ObjectType>
                        <ValidationObjectRepresentation>
                            <DigestAlgAndValue>
                                <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<ns2:DigestValue>g7bbHRyDadkBYcIS3rziniYI2bEXNWuKqeUxchyx9b8=</ns2:DigestValue>
                            </DigestAlgAndValue>
                        </ValidationObjectRepresentation>
                        <POE>
                            <POETime>2020-12-14T11:03:21Z</POETime>
                            <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
                        </POE>
                    <ValidationReport>
                        <ValidationConstraintsEvaluationReport>
                            <ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:formatChecking</ValidationConstraintId
entifier>
                            <ConstraintStatus>
                                <Status>urn:etsi:019102:constraintStatus:disabled</Status>
                            </ConstraintStatus>
                        </ValidationConstraint>
                    <ValidationConstraint>

```

```

<ValidationConstraintIdentifier>urn:cef:dss:bbb:identificationOfTheSigningCertificate<
/ValidationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:validationContextInitialization</Valid
ationConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:disabled</Status>
    </ConstraintStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:cryptographicVerification</ValidationC
onstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:signatureAcceptanceValidation</Validat
ionConstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
</ValidationConstraint>
<ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:x509CertificateValidation</ValidationC
onstraintIdentifier>
    <ConstraintStatus>
        <Status>urn:etsi:019102:constraintStatus:applied</Status>
    </ConstraintStatus>
    <ValidationStatus>

```

```

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    </ValidationStatus>
    </ValidationConstraint>
    </ValidationConstraintsEvaluationReport>
    <SignerInformation>
        <SignerCertificate VOReference="C-
B336CAA1F3C4930E4EF9C803C12877A004991EE9206C0D4AD3891688C1E478FF"/>
            <Signer>Belgium OCSP Responder</Signer>
        </SignerInformation>
    <SignatureValidationStatus>

<MainIndication>urn:etsi:019102:mainindication:passed</MainIndication>
    <AssociatedValidationReportData>
        <TrustAnchor VOReference="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            <CertificateChain>
                <SigningCertificate VOReference="C-
B336CAA1F3C4930E4EF9C803C12877A004991EE9206C0D4AD3891688C1E478FF"/>
                    <IntermediateCertificate VOReference="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA"/>
                        <TrustAnchor VOReference="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
                    </CertificateChain>
                    <CryptoInformation>
                        <ValidationObjectId VOReference="R-
83B6DB1D1C8369D90161C212DEBCE29E2608D9B117356B8AA9E531721CB1F5BF"/>
                            <Algorithm>http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256</Algorithm>
                        <SecureAlgorithm>true</SecureAlgorithm>
                        <NotAfter>2022-12-31T23:00:00Z</NotAfter>
                    </CryptoInformation>
                </AssociatedValidationReportData>
            </SignatureValidationStatus>
        </ValidationReport>
    </ValidationObject>
    <ValidationObject id="D-
BD598965790CA5673910D64EFEDA798485364C66B6F4E7C49D23E1FF8BAFB8E8">
        <ObjectType>urn:etsi:019102:validationObject:signedData</ObjectType>
        <ValidationObjectRepresentation>
            <DigestAlgAndValue>
                <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <ns2:DigestValue>XC6PrK0RnL59qX5UJCibBUw/kPLQNtqQx81H+vd1Pw=</ns2:DigestValue>
            </DigestAlgAndValue>
        </ValidationObjectRepresentation>
        <POE>
            <POETime>2020-12-14T11:03:21Z</POETime>
            <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
        </POE>
    </ValidationObject>

```

```
</ValidationObject>
</SignatureValidationObjects>
</ValidationReport>
```

Validation Policy

The validation process may be driven by a set of constraints that are contained in the XML policy file.

In order to run a validation process with a custom validation policy, an XML file shall be created in compliance with the [policy.xsd](#) schema and passed to the relevant [DocumentValidator](#) as shown below.

Custom validation policy

```
Reports reports = validator.validateDocument(new
File("/path/to/validation/policy.xml"));
```

XML policy structure

The validation policy allows to define different behavior for various validating token types or signature formats. The following groups are considered:

- [ContainerConstraints](#) - defines rules for processing of ASiC containers validation;
- [SignatureConstraints](#) - defines rules for signature basic building blocks processing and the related certificate chain;
- [CounterSignatureConstraints](#) - allows to define custom rules for counter signature processing;
- [Timestamp](#) - defines rules for timestamp validation;
- [Revocation](#) - defines rules for revocation data validation;
- [Cryptographic](#) - defines common rules for cryptographic validation of used algorithms. The general constraints are used when no cryptographic constraints are defined for a particular token type;
- [Model](#) - defines the way of a certificate chain processing;
- [eIDAS](#) - defines rules for validation of Trusted Lists.

Constraints

Each constraint defined in the policy forces an execution of a relevant check in the validation process.



If a constraint is missing in the policy - the check is not processed.

The following constraint types are supported:

- [LevelConstraint](#) - a simple constraint type with a defined processing [Level](#);

- **MultiValuesConstraint** - allows to define a set of accepted values relatively to the using constraint.

Level

The **Level** attribute of a constraint defines a validation process behavior in case of a check failure. While used, the following behaviors apply in case of a check failure:

- **FAIL** - brakes the validation process and returns the relevant indication;
- **WARN** - continues the validation process and returns a warning message to the validation process output;
- **INFORM** - continues the validation process and returns an information message to the validation process output;
- **IGNORE** - processes the check in a silent mode (equivalent to a not defined constraint).

Multi Values Constraint

When using the **MultiValuesConstraint**, a list of acceptable values shall be defined in the list of `<Id>...</Id>` elements, one for each accepted value. While doing, the following rules apply:

- Empty list of values → accept only empty values for the item in question, fails otherwise;
- `"*"` constraint value → accepts all values, reject empty list of values;
- Custom values → accepts only item values matching the constraint.

Cryptographic constraints

Cryptographic constraints define a list of acceptable cryptographic algorithms and their expiration dates when needed. The following settings are possible:

- **AcceptableEncryptionAlgo** - defines a list of acceptable encryption algorithms. All tokens and signatures using other algorithms will be rejected.
- **MinPublicKeySize** - defines the minimal allowed public key size to be used with the defined encryption algorithms. An algorithm with a key size less than the defined one will be rejected. The minimal key size if required to be defined for an encryption algorithm, otherwise all used key sizes will be rejected.
- **AcceptableDigestAlgo** - defines a list of acceptable digest algorithms. All tokens and signatures using other algorithms will be rejected.
- **AlgoExpirationDate** - defines expiration dates for the algorithms. The algorithm is rejected when it is used after the defined date. If the algorithm expiration date is not defined, or set to null, the algorithm is treated as reliable for an unlimited time.

The default XML policy

The default XML validation policy is present below.

constraint.xml (default policy is provided in dss-policy-jaxb module)

```

<ConstraintsParameters Name="QES AdESQC TL based"
xmlns="http://dss.esig.europa.eu/validation/policy">
  <Description>Validate electronic signatures and indicates whether they are
Advanced electronic Signatures (AdES), AdES supported by a Qualified Certificate
(AdES/QC) or a
    Qualified electronic Signature (QES). All certificates and their related
chains supporting the signatures are validated against the EU Member State Trusted
Lists (this includes
      signer's certificate and certificates used to validate certificate validity
status services - CRLs, OCSP, and time-stamps).
  </Description>
  <ContainerConstraints>
    <AcceptableContainerTypes Level="FAIL">
      <Id>ASiC-S</Id>
      <Id>ASiC-E</Id>
    </AcceptableContainerTypes>
    <!--<ZipCommentPresent Level="WARN" />-->
    <!--<AcceptableZipComment Level="WARN">-->
    <!--<Id>mimetype=application/vnd.etsi.asic-s+zip</Id>-->
    <!--<Id>mimetype=application/vnd.etsi.asic-e+zip</Id>-->
    <!--</AcceptableZipComment>-->
    <MimeTypeFilePresent Level="FAIL" />
    <AcceptableMimeTypeFileContent Level="WARN">
      <Id>application/vnd.etsi.asic-s+zip</Id>
      <Id>application/vnd.etsi.asic-e+zip</Id>
    </AcceptableMimeTypeFileContent>
    <ManifestFilePresent Level="FAIL" />
    <SignedFilesPresent Level="FAIL" />
    <AllFilesSigned Level="WARN" />
  </ContainerConstraints>
  <SignatureConstraints>
    <StructuralValidation Level="WARN" />
    <AcceptablePolicies Level="FAIL">
      <Id>ANY_POLICY</Id>
      <Id>NO_POLICY</Id>
    </AcceptablePolicies>
    <PolicyAvailable Level="FAIL" />
    <PolicyHashMatch Level="FAIL" />
    <AcceptableFormats Level="FAIL">
      <Id>*</Id>
    </AcceptableFormats>
    <BasicSignatureConstraints>
      <ReferenceDataExistence Level="FAIL" />
      <ReferenceDataIntact Level="FAIL" />
      <ManifestEntryObjectExistence Level="WARN" />
      <SignatureIntact Level="FAIL" />
      <SignatureDuplicated Level="FAIL" />
      <ProspectiveCertificateChain Level="FAIL" />
      <SignerInformationStore Level="FAIL" />
      <PdfPageDifference Level="FAIL" />
      <PdfAnnotationOverlap Level="WARN" />
    </BasicSignatureConstraints>
  </SignatureConstraints>
</ConstraintsParameters>

```

```

<PdfVisualDifference Level="WARN" />
<!--           <TrustedServiceTypeIdentifier Level="WARN"> -->
<!--           <Id>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</Id>
-->
<!--           </TrustedServiceTypeIdentifier> -->
<!--           <TrustedServiceStatus Level="FAIL"> -->
<!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/undersupervision</Id> -->
<!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/accredited</Id> -->
<!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/supervisionincessation</Id> -->
<!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted</Id> -->
<!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/withdrawn</Id> -->
<!--           </TrustedServiceStatus> -->
<SigningCertificate>
    <Recognition Level="FAIL" />
    <Signature Level="FAIL" />
    <NotExpired Level="FAIL" />
    <AuthorityInfoAccessPresent Level="WARN" />
    <RevocationInfoAccessPresent Level="WARN" />
    <RevocationDataAvailable Level="FAIL" />
    <CRLNextUpdatePresent Level="WARN" />
    <RevocationDataFreshness Level="FAIL" />
    <KeyUsage Level="WARN">
        <Id>nonRepudiation</Id>
    </KeyUsage>
    <SerialNumberPresent Level="WARN" />
    <NotRevoked Level="FAIL" />
    <NotOnHold Level="FAIL" />
    <RevocationIssuerNotExpired Level="FAIL" />
    <NotSelfSigned Level="WARN" />
    <!--           <Qualification Level="WARN" /> -->
    <!--           <SupportedByQSCD Level="WARN" /> -->
    <!--           <QcLegislationCountryCodes Level="WARN" /> -->
    <!--           <IssuedToNaturalPerson Level="INFORM" /> -->
    <!--           <IssuedToLegalPerson Level="INFORM" /> -->
    <UsePseudonym Level="INFORM" />
    <Cryptographic />
</SigningCertificate>
<CACertificate>
    <Signature Level="FAIL" />
    <NotExpired Level="FAIL" />
    <RevocationDataAvailable Level="FAIL" />
    <CRLNextUpdatePresent Level="WARN" />
    <RevocationDataFreshness Level="FAIL" />
    <NotRevoked Level="FAIL" />
    <NotOnHold Level="FAIL" />
    <Cryptographic />

```

```

</CACertificate>
  <Cryptographic />
</BasicSignatureConstraints>
<SignedAttributes>
  <SigningCertificatePresent Level="WARN" />
  <UnicitySigningCertificate Level="WARN" />
  <SigningCertificateRefersCertificateChain Level="WARN" />
  <CertDigestPresent Level="FAIL" />
  <CertDigestMatch Level="FAIL" />
  <IssuerSerialMatch Level="WARN" />
  <SigningTime Level="FAIL" />
  <MessageDigestOrSignedPropertiesPresent Level="FAIL" />
  <!--    <ContentType Level="FAIL" value="1.2.840.113549.1.7.1" />
          <ContentHints Level="FAIL" value="*" />
          <CommitmentTypeIndication Level="FAIL">
            <Id>1.2.840.113549.1.9.16.6.1</Id>
            <Id>1.2.840.113549.1.9.16.6.4</Id>
            <Id>1.2.840.113549.1.9.16.6.5</Id>
            <Id>1.2.840.113549.1.9.16.6.6</Id>
          </CommitmentTypeIndication>
          <SignerLocation Level="FAIL" />
          <ContentTimeStamp Level="FAIL" /> -->
</SignedAttributes>
<UnsignedAttributes>
  <!--    <CounterSignature Level="IGNORE" /> check presence -->
</UnsignedAttributes>
</SignatureConstraints>
<CounterSignatureConstraints>
  <BasicSignatureConstraints>
    <ReferenceDataExistence Level="FAIL" />
    <ReferenceDataIntact Level="FAIL" />
    <SignatureIntact Level="FAIL" />
    <SignatureDuplicated Level="FAIL" />
    <ProspectiveCertificateChain Level="FAIL" />
    <!--          <TrustedServiceTypeIdentifier Level="WARN"> -->
    <!--          <Id>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</Id>
-->
    <!--          </TrustedServiceTypeIdentifier> -->
    <!--          <TrustedServiceStatus Level="FAIL"> -->
    <!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/undersupervision</Id> -->
    <!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/accredited</Id> -->
    <!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/supervisionincessation</Id> -->
    <!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted</Id> -->
    <!--
<Id>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/withdrawn</Id> -->
    <!--          </TrustedServiceStatus> -->
    <SigningCertificate>

```

```

<Recognition Level="FAIL" />
<Signature Level="FAIL" />
<NotExpired Level="FAIL" />
<AuthorityInfoAccessPresent Level="WARN" />
<RevocationInfoAccessPresent Level="WARN" />
<RevocationDataAvailable Level="FAIL" />
<CRLNextUpdatePresent Level="WARN" />
<RevocationDataFreshness Level="FAIL" />
<KeyUsage Level="WARN">
    <Id>nonRepudiation</Id>
</KeyUsage>
<SerialNumberPresent Level="WARN" />
<NotRevoked Level="FAIL" />
<NotOnHold Level="FAIL" />
<NotSelfSigned Level="WARN" />
<!--          <Qualification Level="WARN" /> -->
<!--          <SupportedByQSCD Level="WARN" /> -->
<!--          <IssuedToNaturalPerson Level="INFORM" /> -->
<!--          <IssuedToLegalPerson Level="INFORM" /> -->
<UsePseudonym Level="INFORM" />
<Cryptographic />
</SigningCertificate>
<CACertificate>
    <Signature Level="FAIL" />
    <NotExpired Level="FAIL" />
    <RevocationDataAvailable Level="FAIL" />
    <CRLNextUpdatePresent Level="WARN" />
    <RevocationDataFreshness Level="FAIL" />
    <NotRevoked Level="FAIL" />
    <NotOnHold Level="FAIL" />
    <Cryptographic />
</CACertificate>
<Cryptographic />
</BasicSignatureConstraints>
<SignedAttributes>
    <SigningCertificatePresent Level="WARN" />
    <CertDigestPresent Level="FAIL" />
    <CertDigestMatch Level="FAIL" />
    <IssuerSerialMatch Level="WARN" />
    <SigningTime Level="FAIL" />
    <MessageDigestOrSignedPropertiesPresent Level="FAIL" />
    <!--      <ContentType Level="FAIL" value="1.2.840.113549.1.7.1" />
          <ContentHints Level="FAIL" value="*" />
          <CommitmentTypeIndication Level="FAIL">
              <Id>1.2.840.113549.1.9.16.6.1</Id>
              <Id>1.2.840.113549.1.9.16.6.4</Id>
              <Id>1.2.840.113549.1.9.16.6.5</Id>
              <Id>1.2.840.113549.1.9.16.6.6</Id>
          </CommitmentTypeIndication>
          <SignerLocation Level="FAIL" />
          <ContentTimeStamp Level="FAIL" /> -->

```

```

        </SignedAttributes>
    </CounterSignatureConstraints>
    <Timestamp>
        <TimestampDelay Level="IGNORE" Unit="DAYS" Value="0" />
        <RevocationTimeAgainstBestSignatureTime Level="FAIL" />
        <BestSignatureTimeBeforeExpirationDateOfSigningCertificate Level="FAIL" />
        <Coherence Level="WARN" />
    <BasicSignatureConstraints>
        <ReferenceDataExistence Level="FAIL" />
        <ReferenceDataIntact Level="FAIL" />
        <SignatureIntact Level="FAIL" />
        <ProspectiveCertificateChain Level="FAIL" />
        <SigningCertificate>
            <Recognition Level="FAIL" />
            <Signature Level="FAIL" />
            <NotExpired Level="FAIL" />
            <RevocationDataAvailable Level="FAIL" />
            <CRLNextUpdatePresent Level="WARN" />
            <RevocationDataFreshness Level="FAIL" />
            <ExtendedKeyUsage Level="WARN">
                <Id>timeStamping</Id>
            </ExtendedKeyUsage>
            <NotRevoked Level="FAIL" />
            <NotOnHold Level="FAIL" />
            <NotSelfSigned Level="WARN" />
            <Cryptographic />
        </SigningCertificate>
        <CACertificate>
            <Signature Level="FAIL" />
            <NotExpired Level="FAIL" />
            <RevocationDataAvailable Level="WARN" />
            <CRLNextUpdatePresent Level="WARN" />
            <RevocationDataFreshness Level="FAIL" />
            <NotRevoked Level="FAIL" />
            <NotOnHold Level="FAIL" />
            <Cryptographic />
        </CACertificate>
        <Cryptographic />
    </BasicSignatureConstraints>
    <SignedAttributes>
        <SigningCertificatePresent Level="WARN" />
        <!-- <UnicitySigningCertificate Level="WARN" /> RFC 5816 -->
        <SigningCertificateRefersCertificateChain Level="WARN" />
        <CertDigestPresent Level="WARN" />
        <IssuerSerialMatch Level="WARN" />
    </SignedAttributes>
    <TSAGeneralNameContentMatch Level="WARN" />
</Timestamp>
<Revocation>
    <RevocationFreshness Level="IGNORE" Unit="DAYS" Value="0" />
    <UnknownStatus Level="FAIL" />

```

```

<SelfIssuedOCSP Level="WARN" />
<BasicSignatureConstraints>
    <ReferenceDataExistence Level="FAIL" />
    <ReferenceDataIntact Level="FAIL" />
    <SignatureIntact Level="FAIL" />
    <ProspectiveCertificateChain Level="FAIL" />
    <SigningCertificate>
        <Recognition Level="FAIL" />
        <Signature Level="FAIL" />
        <NotExpired Level="FAIL" />
        <RevocationDataAvailable Level="FAIL" />
        <CRLLNextUpdatePresent Level="WARN" />
        <RevocationDataFreshness Level="FAIL" />
        <NotRevoked Level="FAIL" />
        <NotOnHold Level="FAIL" />
        <Cryptographic />
    </SigningCertificate>
    <CACertificate>
        <Signature Level="FAIL" />
        <NotExpired Level="FAIL" />
        <RevocationDataAvailable Level="WARN" />
        <CRLLNextUpdatePresent Level="WARN" />
        <RevocationDataFreshness Level="FAIL" />
        <NotRevoked Level="FAIL" />
        <NotOnHold Level="FAIL" />
        <Cryptographic />
    </CACertificate>
    <Cryptographic />
</BasicSignatureConstraints>
</Revocation>
<Cryptographic Level="FAIL">
    <AcceptableEncryptionAlgo>
        <Algo>RSA</Algo>
        <Algo>DSA</Algo>
        <Algo>ECDSA</Algo>
        <Algo>PLAIN-ECDSA</Algo>
        <!--          <Algo>Ed25519</Algo>          Not referenced in
ETSI/SOGIS -->
        </AcceptableEncryptionAlgo>
    <MiniPublicKeySize>
        <Algo Size="1024">DSA</Algo>
        <Algo Size="1024">RSA</Algo>
        <Algo Size="160">ECDSA</Algo>
        <Algo Size="160">PLAIN-ECDSA</Algo>
        <!--          <Algo Size="24">Ed25519</Algo>          Not referenced in
ETSI/SOGIS -->
        </MiniPublicKeySize>
    <AcceptableDigestAlgo>
        <Algo>MD2</Algo>
        <Algo>MD5</Algo>
        <Algo>SHA1</Algo>

```

```

<Algo>SHA224</Algo>
<Algo>SHA256</Algo>
<Algo>SHA384</Algo>
<Algo>SHA512</Algo>
<Algo>SHA3-224</Algo>
<Algo>SHA3-256</Algo>
<Algo>SHA3-384</Algo>
<Algo>SHA3-512</Algo>
<Algo>RIPEMD160</Algo>
<Algo>WHIRLPOOL</Algo>
</AcceptableDigestAlgo>
<AlgoExpirationDate Format="yyyy">
  <!-- Digest algorithms -->
  <Algo Date="2005">MD2</Algo> <!-- The same as for MD5 -->
  <Algo Date="2005">MD5</Algo> <!-- ETSI TS 102 176-1 (Historical) V2.1.1
-->
  <Algo Date="2009">SHA1</Algo> <!-- ETSI TS 102 176-1 (Historical) V2.0.0
-->
  <Algo Date="2023">SHA224</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026">SHA256</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026">SHA384</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026">SHA512</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026">SHA3-224</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026">SHA3-256</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026">SHA3-384</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026">SHA3-512</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2011">RIPEMD160</Algo> <!-- ETSI TS 102 176-1 (Historical)
V2.0.0 -->
  <Algo Date="2015">WHIRLPOOL</Algo> <!-- ETSI 119 312 V1.1.1 -->
  <!-- end Digest algorithms -->
  <!-- Encryption algorithms -->
  <Algo Date="2013" Size="1024">DSA</Algo> <!-- ETSI TS 102 176-1
(Historical) V2.1.1 -->
  <Algo Date="2023" Size="2048">DSA</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026" Size="3072">DSA</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2009" Size="1024">RSA</Algo> <!-- ETSI TS 102 176-1
(Historical) V2.0.0 -->
  <Algo Date="2016" Size="1536">RSA</Algo> <!-- ETSI 119 312 V1.1.1 -->
  <Algo Date="2023" Size="1900">RSA</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026" Size="3000">RSA</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2013" Size="160">ECDSA</Algo> <!-- ETSI TS 102 176-1
(Historical) V2.1.1 -->
  <Algo Date="2013" Size="192">ECDSA</Algo> <!-- ETSI TS 102 176-1
(Historical) V2.1.1 -->
  <Algo Date="2016" Size="224">ECDSA</Algo> <!-- ETSI 119 312 V1.1.1 -->
  <Algo Date="2026" Size="256">ECDSA</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026" Size="384">ECDSA</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2026" Size="512">ECDSA</Algo> <!-- ETSI 119 312 V1.3.1 -->
  <Algo Date="2013" Size="160">PLAIN-ECDSA</Algo> <!-- ETSI TS 102 176-1
(Historical) V2.1.1 -->
  <Algo Date="2013" Size="192">PLAIN-ECDSA</Algo> <!-- ETSI TS 102 176-1

```

```

(Historical) V2.1.1 -->
    <Algo Date="2016" Size="224">PLAIN-ECDSA</Algo> <!-- ETSI 119 312 V1.1.1
-->
    <Algo Date="2026" Size="256">PLAIN-ECDSA</Algo> <!-- ETSI 119 312 V1.3.1
-->
    <Algo Date="2026" Size="384">PLAIN-ECDSA</Algo> <!-- ETSI 119 312 V1.3.1
-->
    <Algo Date="2026" Size="512">PLAIN-ECDSA</Algo> <!-- ETSI 119 312 V1.3.1
-->

    <!--          <Algo Date="2026" Size="32">Ed25519</Algo>      Not
referenced in ETSI/SOGIS -->
    <!-- end Encryption algorithms -->
</AlgoExpirationDate>
</Cryptographic>

<Model Value="SHELL" />

<!-- eIDAS REGL 910/EU/2014 -->
<eIDAS>
    <TLFreshness Level="WARN" Unit="HOURS" Value="6" />
    <TLNotExpired Level="WARN" />
    <TLWellSigned Level="WARN" />
    <TLVersion Level="FAIL" value="5" />
</eIDAS>
</ConstraintsParameters>

```

CAdES signature (CMS)

To familiarize yourself with this type of signature it is advisable to read the following document:

- CAdES Specifications (cf. [\[R02\]](#))

To implement this form of signature you can use the XAdES examples. You only need to instantiate the CAdES object service and change the SignatureLevel parameter value. Below is an example of the CAdES-Baseline-B signature:

*Sig*ning a file with CAdES

```
// Preparing parameters for the CAdES signature
CAdESSignatureParameters parameters = new CAdESSignatureParameters();
// We choose the level of the signature (-B, -T, -LT, -LTA).
parameters.setSignatureLevel(SignatureLevel.CAdES_BASELINE_B);
// We choose the type of the signature packaging (ENVELOPING, DETACHED).
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPING);
// We set the digest algorithm to use with the signature algorithm. You must use the
// same parameter when you invoke the method sign on the token. The default value is
// SHA256
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
// Create CAdESService for signature
CAdESService service = new CAdESService(commonCertificateVerifier);

// Get the SignedInfo segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm
DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm,
privateKey);

// We invoke the CAdESService to sign the document with the signature value obtained
// in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);
```

PAdES signature (PDF)

The standard ISO 32000-1 (cf. [\[R06\]](#)) allows defining a file format for portable electronic documents. It is based on PDF 1.7 of Adobe Systems. Concerning the digital signature it supports three operations:

- Adding a digital signature to a document,
- Providing a placeholder field for signatures,
- Checking signatures for validity.

PAdES defines eight different profiles to be used with advanced electronic signature in the meaning of European Union Directive 1999/93/EC (cf. [\[R07\]](#)):

- PAdES Basic - PDF signature as specified in ISO 32000-1 (cf. [\[R06\]](#)). The profile is specified in ETSI EN 319 142 (cf. [\[R03\]](#)).
- PAdES-BES Profile - based upon CAdES-BES as specified in ETSI EN 319 122 (cf. [\[R02\]](#)) with the option of a signature time-stamp (CAdES-T).
- PAdES-EPES profile - based upon CAdES-EPES as specified in ETSI EN 319 122 (cf. [\[R02\]](#)). This profile is the same as the PAdES - BES with the addition of a signature policy identifier and optionally a commitment type indication.
- PAdES-LTV Profile - This profile supports the long term validation of PDF Signatures and can be used in conjunction with the above-mentioned profiles.
- Four other PAdES profiles for XML Content.

To familiarize yourself with this type of signature it is advisable to read the documents referenced above.

Below is an example of code to perform a PAdES-BASELINE-B type signature:

*Sig*ning a PDF file with PAdES

```
// Preparing parameters for the PAdES signature
PAdESSignatureParameters parameters = new PAdESSignatureParameters();
// We choose the level of the signature (-B, -T, -LT, -LTA).
parameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_B);
// We set the digest algorithm to use with the signature algorithm. You must use the
// same parameter when you invoke the method sign on the token. The default value is
// SHA256
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
// Create PAdESService for signature
PAdESService service = new PAdESService(commonCertificateVerifier);

// Get the SignedInfo segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm
DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm,
privateKey);

// Optionally or for debug purpose :
// Validate the signature value against the original dataToSign
assertTrue(service.isValidSignatureValue(dataToSign, signatureValue,
privateKey.getCertificate()));

// We invoke the xadesService to sign the document with the signature value obtained
// in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);
```

In order to add a timestamp to the signature (PAdES-T or LTA), a TSP source must be provided to the service.

To create PAdES-BASELINE-B level with additional options: signature policy identifier and optionally a commitment type indication, please observe the following example in code 5.

All these parameters are optional.

```
// Instantiate a Policy object
Policy signaturePolicy = new Policy();
// The string representation of the OID of the signature policy to use when signing.
signaturePolicy.setId("1.2.3.4.5.6");
// Defines a policy identifier qualifier
signaturePolicy.setQualifier(ObjectIdentifierQualifier.OID_AS_URN);
// Defines a description for a signature policy
signaturePolicy.setDescription("Perfect Signature Policy");
// The hash function used to compute the value of the SignaturePolicyHashValue entry.
// Entries must be represented the same way as in table 257 of ISO 32000-1 (cf.
<<R05>>).
signaturePolicy.setDigestAlgorithm(DigestAlgorithm.SHA1);
// The value of the hash of the signature policy, computed the same way as
// in clause 5.2.9 of CAdES (ETSI EN 319 122 (cf. <<R02>>)).
signaturePolicy.setDigestValue(new byte[] { 'd', 'i', 'g', 'e', 's', 't', 'v', 'a',
'1', 'u', 'e' });
// Defines a URI where the policy can be accessed from
signaturePolicy.setSpuri("http://spuri.test");
parameters.bLevel().setSignaturePolicy(signaturePolicy);
```

The extension of a signature of the level PAdES-BASELINE-B up to PAdES-BASELINE-LTA profile will add the following features:

- Addition of validation data to an existing PDF document which may be used to validate earlier signatures within the document (including PDF signatures and time-stamp signatures).
- Addition of a document time-stamp which protects the existing document and any validation data.
- Further validation data and document time-stamp may be added to a document over time to maintain its authenticity and integrity.

PAdES Visible Signature

The framework also allows creation of PDF files with visible signature as specified in ETSI EN 319 142 (cf. [\[R03\]](#)). In the `SignatureParameters` object, there's a special attribute named `SignatureImageParameters`. This parameter allows you customize the visual signature (with text, with image or with image and text). Below there is an example of code to perform a PAdES-BASELINE-B type signature with a visible signature:

Add a visible signature to a PDF document

```
// Preparing parameters for the PAdES signature
PAdESSignatureParameters parameters = new PAdESSignatureParameters();
// We choose the level of the signature (-B, -T, -LT, -LTA).
parameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_B);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

// Initialize visual signature and configure
SignatureImageParameters imageParameters = new SignatureImageParameters();
// set an image
imageParameters.setImage(new
InMemoryDocument(getClass().getResourceAsStream("/signature-pen.png")));

// initialize signature field parameters
SignatureFieldParameters fieldParameters = new SignatureFieldParameters();
imageParameters.setFieldParameters(fieldParameters);
// the origin is the left and top corner of the page
fieldParameters.setOriginX(200);
fieldParameters.setOriginY(400);
fieldParameters.setWidth(300);
fieldParameters.setHeight(200);
parameters.setImageParameters(imageParameters);

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
// Create PAdESService for signature
PAdESService service = new PAdESService(commonCertificateVerifier);
service.setPdfObjFactory(new PdfBoxNativeObjectFactory());
// Get the SignedInfo segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm
DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm,
privateKey);

// We invoke the xadesService to sign the document with the signature value obtained
// in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);
```

Additionally, DSS also allows you to insert a visible signature to an existing field :

Add a visible signature to an existing field

```
SignatureFieldParameters fieldParameters = new SignatureFieldParameters();
fieldParameters.setFieldId("field-id");
```

In case of placing an image or text to an existing field, the visible signature will fill out the whole available area of the field.

Visible signature parameters (image and text)

This chapter introduces existing parameters for creation of visible signatures with DSS. DSS has three implementations for visible signature drawing:

- **OpenPDF (iText)** - supports separate image and text drawing;
- **PDFBox Default** - supports separate image and text drawing, as well as a joint drawing of image and text together. Transforms text to an image;
- **PDFBox Native** - supports separate image and text drawing, as well as a joint drawing of image and text together. Prints text in a native way, that increases quality of the produced signature.

Positioning

DSS provides a set of functions allowing to place the signature field on a specific place in the PDF page :

Visible signature positioning

```
// Object containing a list of visible signature parameters
SignatureImageParameters signatureImageParameters = new SignatureImageParameters();

// Allows alignment of a signature field horizontally to a page. Allows the following
values:
/* _NONE_ (_DEFAULT value._ None alignment is applied, coordinates are counted from
the left page side);
   _LEFT_ (the signature is aligned to the left side, coordinates are counted from the
left page side);
   _CENTER_ (the signature is aligned to the center of the page, coordinates are
counted automatically);
   _RIGHT_ (the signature is aligned to the right side, coordinates are counted from
the right page side). */
signatureImageParameters.setAlignmentHorizontal(VisualSignatureAlignmentHorizontal.CEN
TER);

// Allows alignment of a signature field vertically to a page. Allows the following
values:
/* _NONE_ (_DEFAULT value._ None alignment is applied, coordinates are counted from
the top side of a page);
   _TOP_ (the signature is aligned to a top side, coordinates are counted from the top
page side);
   _MIDDLE_ (the signature aligned to a middle of a page, coordinates are counted
```

```

automatically);
    _BOTTOM_ (the signature is aligned to a bottom side, coordinates are counted from
the bottom page side). */
signatureImageParameters.setAlignmentVertical(VisualSignatureAlignmentVertical.TOP);

// Rotates the signature field and changes the coordinates' origin respectively to its
values as following:
/* _NONE_ (_DEFAULT value.) No rotation is applied. The origin of coordinates begins
from the top left corner of a page);
    _AUTOMATIC_ (Rotates a signature field respectively to the page's rotation. Rotates
the signature field on the same value as a defined in a PDF page);
    _ROTATE_90_ (Rotates a signature field for a 90°; clockwise. Coordinates'
origin begins from top right page corner);
    _ROTATE_180_ (Rotates a signature field for a 180°; clockwise. Coordinates'
origin begins from the bottom right page corner);
    _ROTATE_270_ (Rotates a signature field for a 270°; clockwise. Coordinates'
origin begins from the bottom left page corner). */
signatureImageParameters.setRotation(VisualSignatureRotation.AUTOMATIC);

// Defines a zoom of the image. The value is applied to width and height of a
signature field.
// The value must be defined in percentage (default value is 100, no zoom is applied).
signatureImageParameters.setZoom(50);

// Specifies a background color for a signature field.
signatureImageParameters.setBackgroundColor(Color.GREEN);

// Defines the image scaling behavior within a signature field with a fixed size
/*
    STRETCH - the default behavior, stretches the image in both directions in order to
fill the signature field box;
    ZOOM_AND_CENTER - zooms the image to fill the signature box to the closest side, and
centers in another dimension;
    CENTER - centers the image in both dimensions.
*/
signatureImageParameters.setImageScaling(ImageScaling.CENTER);

// set the image parameters to signature parameters
padesSignatureParameters.setImageParameters(signatureImageParameters);

```

Dimensions

DSS framework provides a set of functions to manage the signature field size :

Visible signature dimensions

```

// Object containing a list of visible signature parameters
SignatureImageParameters signatureImageParameters = new SignatureImageParameters();

// Allows alignment of a signature field horizontally to a page. Allows the following
values:

```

```

/* _NONE_ (_DEFAULT value._ None alignment is applied, coordinates are counted from
the left page side);
    _LEFT_ (the signature is aligned to the left side, coordinated are counted from the
left page side);
    _CENTER_ (the signature is aligned to the center of the page, coordinates are
counted automatically);
    _RIGHT_ (the signature is aligned to the right side, coordinated are counted from
the right page side). */
signatureImageParameters.setAlignmentHorizontal(VisualSignatureAlignmentHorizontal.CENTER);

// Allows alignment of a signature field vertically to a page. Allows the following
values:
/* _NONE_ (_DEFAULT value._ None alignment is applied, coordinated are counted from
the top side of a page);
    _TOP_ (the signature is aligned to a top side, coordinated are counted from the top
page side);
    _MIDDLE_ (the signature aligned to a middle of a page, coordinated are counted
automatically);
    _BOTTOM_ (the signature is aligned to a bottom side, coordinated are counted from
the bottom page side). */
signatureImageParameters.setAlignmentVertical(VisualSignatureAlignmentVertical.TOP);

// Rotates the signature field and changes the coordinates' origin respectively to its
values as following:
/* _NONE_ (_DEFAULT value._ No rotation is applied. The origin of coordinates begins
from the top left corner of a page);
    _AUTOMATIC_ (Rotates a signature field respectively to the page's rotation. Rotates
the signature field on the same value as a defined in a PDF page);
    _ROTATE_90_ (Rotates a signature field for a 90°; clockwise. Coordinates'
origin begins from top right page corner);
    _ROTATE_180_ (Rotates a signature field for a 180°; clockwise. Coordinates'
origin begins from the bottom right page corner);
    _ROTATE_270_ (Rotates a signature field for a 270°; clockwise. Coordinates'
origin begins from the bottom left page corner). */
signatureImageParameters.setRotation(VisualSignatureRotation.AUTOMATIC);

// Defines a zoom of the image. The value is applied to width and height of a
signature field.
// The value must be defined in percentage (default value is 100, no zoom is applied).
signatureImageParameters.setZoom(50);

// Specifies a background color for a signature field.
signatureImageParameters.setBackgroundColor(Color.GREEN);

// Defines the image scaling behavior within a signature field with a fixed size
/*
    STRETCH - the default behavior, stretches the image in both directions in order to
fill the signature field box;
    ZOOM_AND_CENTER - zooms the image to fill the signature box to the closest side, and
centers in another dimension;

```

```

    CENTER - centers the image in both dimensions.
*/
signatureImageParameters.setImageScaling(ImageScaling.CENTER);

// set the image parameters to signature parameters
padesSignatureParameters.setImageParameters(signatureImageParameters);

```

Text Parameters

The available implementations allow placing of a visible text to a signature field :

List of available visible text parameters

```

// Instantiates a SignatureImageTextParameters object
SignatureImageTextParameters textParameters = new SignatureImageTextParameters();
// Allows you to set a DSSFont object that defines the text style (see more
// information in the section "Fonts usage")
textParameters.setFont(font);
// Defines the text content
textParameters.setText("My visual signature \n #1");
// Defines the color of the characters
textParameters.setTextColor(Color.BLUE);
// Defines the background color for the area filled out by the text
textParameters.setBackgroundColor(Color.YELLOW);
// Defines a padding between the text and a border of its bounding area
textParameters.setPadding(20);
// TextWrapping parameter allows defining the text wrapping behavior within the
// signature field
/*
    FONT_BASED - the default text wrapping, the text is computed based on the given font
    size;
    FILL_BOX - finds optimal font size to wrap the text to a signature field box;
    FILL_BOX_AND_LINEBREAK - breaks the words to multiple lines in order to find the
    biggest possible font size to wrap the text into a signature field box.
*/
textParameters.setTextWrapping(TextWrapping.FONT_BASED);
// Set textParameters to a SignatureImageParameters object
imageParameters.setTextParameters(textParameters);

```

Text and image combination

DSS provides a set of functions to align a text respectively to an image. The parameters must be applied to a 'SignatureImageTextParameters' object :

Combination of text and image parameters

```
// Specifies a text position relatively to an image (Note: applicable only for joint  
image+text visible signatures).  
// Thus with _SignerPosition.LEFT_ value, the text will be placed on the left side,  
// and image will be aligned to the right side inside the signature field  
textParameters.setSignerTextPosition(SignerTextPosition.LEFT);  
// Specifies a horizontal alignment of a text with respect to its area  
textParameters.setSignerTextHorizontalAlignment(SignerTextHorizontalAlignment.RIGHT);  
// Specifies a vertical alignment of a text block with respect to a signature field  
area  
textParameters.setSignerTextVerticalAlignment(SignerTextVerticalAlignment.TOP);
```

The result of applying the foregoing transformations is provided on the image below:



Fonts usage

Since version 5.5, DSS supports two types of fonts. The custom font must be added as an instance of **DSSFont** interface to a **SignatureImageTextParameters** object. **DSSFont** interface has following common implementations:

- **DSSFileFont** for using of physical fonts, which must be embedded to the produced PDF document. To create an instance of the class, you must pass to a **DSSFileFont** constructor an object of **DSSDocument** type or **InputStream** of the font file;
- **DSSJavaFont** for using of logical fonts (default Java fonts). The logical Java fonts allow you to significantly reduce the document size, because these fonts cannot be embedded to the final PDF document. Be aware, because of the fact, using of logical fonts does not allow producing PDF documents satisfying the PDF/A standard. To create an instance of this class, you should pass as an input a **java.awt.Font** object or target font parameters (name, style, size).



Logical fonts may have different implementations depending on a used PAdES Visible signature service or Operating System (OS). Keep this in mind when switching an implementation or system environment.

As well as classes allowing to define native fonts for used implementations (available since DSS 5.7):

- **ITextNativeFont** to be used with **ITextSignatureDrawerFactory**;

- `PdfBoxNativeFont` to be used with `PdfBoxNativeObjectFactory`.

You can create a custom font as following (for a physical font):

Add a custom font as a file

```
// Initialize text to generate for visual signature
DSSFont font = new
DSSfileFont(getClass().getResourceAsStream("/fonts/OpenSansRegular.ttf"));
```

For a logical font:

Java font usage

```
SignatureImageTextParameters textParameters = new SignatureImageTextParameters();
DSSFont font = new DSSJavaFont(Font.SERIF);
font.setSize(16); // Specifies the text size value (the default font size is 12pt)
textParameters.setFont(font);
textParameters.setTextColor(Color.BLUE);
textParameters.setText("My visual signature");
imageParameters.setTextParameters(textParameters);
```

For a native font:

Native font usage

```
textParameters.setFont(new PdfBoxNativeFont(PDType1Font.HELVETICA));
```

By default, DSS uses a Google font : 'PT Serif Regular' (its physical implementation).



'Native PDFBox Drawer' implementation supports only one of the following fonts: SERIF, SANS-SERIF, MONOSPACED, DIALOG and DIALOG_INPUT.

Shadow attack detection

"Shadow attack" is a class of attacks on a signed PDF document that constitutes a change of a visual content of a document after the signature has been made. Due to a structure of PDF document, the signature stays cryptographically valid even after the content's modification has been taken place. There are no algorithms to detect the malicious change with 100% guarantee. For more information, please refer to [the website](#).

Since v5.8, DSS provides a set of own utils to detect the "shadow attack" on a signed PDF document. The following algorithms have been introduced:

- `Page amount difference` - the validation tool compares the number of pages between the obtained PDF and signed revision. If the numbers do not match, the validation fail. The validation level can be configured within the `Validation Policy` with the constraint `<PdfPageDifference>`.

- **Annotations overlap** - DSS checks if any annotation overlaps occurred. The overlapping is potentially dangerous, because some annotations can cover a visual content, e.g. forms and signature fields. The validation level can be configured with the constraint `<PdfAnnotationOverlap>`.
- **Visual difference** - DSS verifies the visual difference between the provided document and signed revision, excluding the newly created annotations (between the validating revisions). The validation level can be configured with the constraint `<PdfVisualDifference>`.

JAdES signature (JWS)

Since v5.8, DSS includes a possibility of creation and validation of JSON Advanced signatures.

JSON format for AdES Signatures (cf. [R05]) represents an extension of JSON Web Signatures (JWS) as specified in [IETF RFC 7515](#).

A typical example of a JAdES signature creation is represented below:

```
// Prepare parameters for the JAdES signature
JAdESSignatureParameters parameters = new JAdESSignatureParameters();
// Choose the level of the signature (-B, -T, -LT, -LTA).
parameters.setSignatureLevel(SignatureLevel.JAdES_BASELINE_B);
// Choose the type of the signature packaging (ENVELOPING, DETACHED).
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPING);
// Choose the form of the signature (COMPACT_SERIALIZATION, JSON_SERIALIZATION,
FLATTENED_JSON_SERIALIZATION)
parameters.setJwsSerializationType(JWS.serializationType.COMPACT_SERIALIZATION);

// Set the digest algorithm
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);
// Set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// Set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
// Create JAdESService for signature
JAdESService service = new JAdESService(commonCertificateVerifier);

// Get the SignedInfo segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm
DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm,
privateKey);

// We invoke the JAdESService to sign the document with the signature value obtained
// in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);
```

The specific parameters for JAdES signature are described in the next sections.

JWS Serialization type

A JWS signature can be represented in different forms which are supported by the JAdES as well:

- **COMPACT_SERIALIZATION** represents a compact, URL-safe serialization. It has no JWS Unprotected Header, therefore only JAdES-BASELINE-B level is possible with the format.
- **JSON_SERIALIZATION** represents a JSON object with a collection of signatures inside the **signatures** header that allows a parallel signing. It allows JAdES-BASELINE-T/-LT/-LTA signature extension

levels.

- **FLATTENED_JSON_SERIALIZATION** represents a JSON object with a single signature container. It allows JAdES-BASELINE-T/-LT/-LTA signature extension levels.

JWS Serialization type usage

```
// Choose the form of the signature (COMPACT_SERIALIZATION, JSON_SERIALIZATION,  
FLATTENED_JSON_SERIALIZATION)  
parameters.setJwsSerializationType(JWSSerializationType.COMPACT_SERIALIZATION);
```

JAdES Signature Packaging

JAdES signatures allow two types of JWS Payload (signed data) inclusion: **ENVELOPING** and **DETACHED**.

Enveloping packaging

With **ENVELOPING** packaging the JWS Payload is enveloped into the JAdES Signature. The type only allows signing one document.

Detached packaging

A simple JWS signature allows a **DETACHED** packaging by omitting the JWS Payload in the created signature. For the validation process the detached content shall be provided and it is treated in the same way as attached.

To create a such signature, the parameter **SigDMechanism.NO_SIG_D** shall be set. The solution allows signing of only one document.

The JAdES standard [R05] provides a possibility for signing of multiple documents withing one signature in a detached way.

The following mechanisms are possible:

- **HTTP_HEADERS** is used to sign an HTTP request. The signature may explicitly sign several HTTP headers (represented by the class **HTTPHeader**), as well as the HTTP message body (see the **HTTPHeaderDigest** class).

Configuration for signing with detached mechanism *HttpHeaders*

```
// Set Detached packaging
parameters.setSignaturePackaging(SignaturePackaging.DETACHED);
// Set Mechanism HttpHeaders for 'sigD' header
parameters.setSigDMechanism(SigDMechanism.HTTP_HEADERS);
// The HttpHeaders mechanism shall be used with unencoded JWS payload ("b64"="false")
parameters.setBase64UrlEncodedPayload(false);
// Create a list of headers to be signed
List<DSSDocument> documentsToSign = new ArrayList<>();
documentsToSign.add(new HTTPHeader("content-type", "application/json"));
documentsToSign.add(new HTTPHeader("x-example", "HTTP Headers Example"));
documentsToSign.add(new HTTPHeader("x-example", "Duplicated Header"));
// Add a document representing the HTTP message body (optional)
// Requires the message body content + digest algorithm to compute the hash to be
signed
documentsToSign.add(new HTTPHeaderDigest(toSignDocument, DigestAlgorithm.SHA1));
```

- **OBJECT_ID_BY_URI** can be used for signing of multiple documents. The signed files are dereferenced by URIs and their content is concatenated for generation of the JWS Payload.
- **OBJECT_ID_BY_URI_HASH** similarly provides a possibility to sign multiple documents, by signing the computed digests of the original documents. The JWS Payload for this format stays empty.

Configuration for signing with detached mechanism *ObjectIdByURIHash*

```
parameters.setSignaturePackaging(SignaturePackaging.DETACHED);
parameters.setSigDMechanism(SigDMechanism.OBJECT_ID_BY_URI_HASH);
// Prepare the documents to be signed
documentsToBeSigned = new ArrayList<>();
documentsToBeSigned.add(new FileDocument("src/main/resources/hello-world.pdf"));
documentsToBeSigned.add(new FileDocument("src/main/resources/xml_example.xml"));
```

Base64Url encoding

The **Base64Url** represents a Base64 encoded format with URI safe alphabet (see [RFC 4648](#)).

JAdES signatures (as well as JWS) force some values to be Base64Url-encoded, while provides a possibility to customize the format for some of them.

DSS provides options to configure encoding for the following elements:

- JWS Payload can be represented as Base64Url encoded octets (by default), as well as can be present in its initial form (with the protected header **b64** set to **false**).

Use unencoded JWS Payload

```
parameters.setBase64UrlEncodedPayload(false);
```

- The components of the unsigned header `etsiu` can occur either as Base64Url encoded strings (by default), or as clear JSON objects.



All components inside the `etsiu` header shall be present in the same form (Base64Url encoded or as clear JSON).



The current version of DSS does not allow JAdES-BASELINE-LTA level creation for `etsiu` components in their clear JSON representation.

Represent Etsiu components as clear JSON instances

```
parameters.setBase64UrlEncodedEtsiuComponents(false);
```

ASiC signature (containers)

When creating a digital signature, the user must choose between different packaging elements, namely enveloping, enveloped or detached. This choice is not obvious, because in one case the signature will alter the signed document and in the other case it is possible to lose the association between the signed document and its signature. That's where the standard ETSI EN 319 162 (cf. [R04]) offers a standardized use of container forms to establish a common way for associating data objects with advanced signatures or time-stamp tokens.

A number of application environments use ZIP based container formats to package sets of files together with meta-information. ASiC technical specification is designed to operate with a range of such ZIP based application environments. Rather than enforcing a single packaging structure, ASiC describes how these package formats can be used to associate advanced electronic signatures with any data objects.

The standard defines two types of containers; the first (ASiC-S) allows you to associate one or more signatures with a single data element. In this case the structure of the signature can be based (in a general way) on a single CAdES signature or on multiple XAdES signatures or finally on a single TST; the second is an extended container (ASiC-E) that includes multiple data objects. Each data object may be signed by one or more signatures which structure is similar to ASiC-S. This second type of container is compatible with OCF, UCF and ODF formats.

For the moment the DSS framework has some restrictions on the containers you can generate, depending on the input file. If the input file is already an ASiC container, the output container must be the same type of container based on the same type of signature. If the input is any other file, the output does not have any restriction.

Table 6. ASiC containers

Input	Output
ASiC-S CAdES	ASiC-S CAdES
ASiC-S XAdES	ASiC-S XAdES
ASiC-E CAdES	ASiC-E CAdES

Input	Output
ASiC-E XAdES	ASiC-E XAdES
Binary	ASiC-S CAdES, ASiC-S XAdES, ASiC-E CAdES, ASiC-E XAdES

This is an example of the source code for signing a document using ASiCS-S based on XAdES-B:

Sign a file within an ASiC-S container

```

// Preparing parameters for the AsicS signature
ASiCWithXAdESSignatureParameters parameters = new ASiCWithXAdESSignatureParameters();
// We choose the level of the signature (-B, -T, -LT, LTA).
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_B);
// We choose the container type (ASiC-S or ASiC-E)
parameters.aSiC().setContainerType(ASiCContainerType.ASiC_S);

// We set the digest algorithm to use with the signature algorithm. You must use the
// same parameter when you invoke the method sign on the token. The default value is
// SHA256
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
// Create ASiC service for signature
ASiCWithXAdESService service = new ASiCWithXAdESService(commonCertificateVerifier);

// Get the SignedInfo segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm
DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm,
privateKey);

// We invoke the xadesService to sign the document with the signature value obtained
// in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);

```

This is another example of the source code for signing multiple documents using ASiCS-E based on CAdES:

```
// Preparing the documents to be embedded in the container and signed
List<DSSDocument> documentsToBeSigned = new ArrayList<>();
documentsToBeSigned.add(new FileDocument("src/main/resources/hello-world.pdf"));
documentsToBeSigned.add(new FileDocument("src/main/resources/xml_example.xml"));

// Preparing parameters for the ASiC-E signature
ASiCWithCAdESSignatureParameters parameters = new ASiCWithCAdESSignatureParameters();

// We choose the level of the signature (-B, -T, -LT or -LTA).
parameters.setSignatureLevel(SignatureLevel.CAdES_BASELINE_B);
// We choose the container type (ASiC-S or ASiC-E)
parameters.aSiC().setContainerType(ASiCContainerType.ASiC_E);

// We set the digest algorithm to use with the signature algorithm. You
// must use the
// same parameter when you invoke the method sign on the token. The
// default value is
// SHA256
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
// Create ASiC service for signature
ASiCWithCAdESService service = new ASiCWithCAdESService(commonCertificateVerifier);

// Get the SignedInfo segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(documentsToBeSigned, parameters);

// This function obtains the signature value for signed information
// using the
// private key and specified algorithm
DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm,
privateKey);

// We invoke the xadesService to sign the document with the signature
// value obtained in
// the previous step.
DSSDocument signedDocument = service.signDocument(documentsToBeSigned, parameters,
signatureValue);
```

Please note that you need to pass only few parameters to the service. Other parameters, although are positioned, will be overwritten by the internal implementation of the service. Therefore, the

obtained signature is always based on CAdES and of DETACHED packaging.

It is also possible with the framework DSS to make an extension of an ASiC container to the level XAdES-BASELINE-T or -LT.

Counter signatures

Since v5.8 DSS allows producing of counter signatures according to the corresponding AdES formats.



Counter signature does not provide a Proof Of Existence for a signed signature!
Use signature extension / timestamping for this purpose.

The following formats are supported for the counter signature creation:

- **XAdES** - multiple, nested and extended counter signatures (up to LTA level) are allowed;
- **CAdES** - B-level counter signatures are allowed, as well as multiple counter signatures;
- **JAdES** - multiple, nested and extended signatures (up to LTA level) are allowed;
- **ASiC** - counter signatures are allowed according to the used format (XAdES or CAdES).

In order to create a counter signature, the DSS Identifier (or XML Id for XAdES) of the target signature you want to sign shall be provided within the parameters. The example below represents a counter signature creation:

Counter signature creation

```
// Initialize counter signature parameters
XAdESCounterSignatureParameters counterSignatureParameters = new
XAdESCounterSignatureParameters();
// Set signing certificate parameters
counterSignatureParameters.setSigningCertificate(privateKey.getCertificate());
counterSignatureParameters.setCertificateChain(privateKey.getCertificateChain());
// Set target level of the counter signature
counterSignatureParameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_B);

// Next step is to extract and set the Id of a signature to be counter signed

// Initialize a validator over the signedDocument in order to extract the master
signature Id
DocumentValidator validator = SignedDocumentValidator.fromDocument(signedDocument);
// Get list of signatures
List<AdvancedSignature> signatures = validator.getSignatures();
// Get Id of the target signature
AdvancedSignature signature = signatures.iterator().next();
String signatureId = signature.getId();
// For XAdES, the XML Id can be used
signatureId = signature.getDAIdentifier();
// Set the Id to parameters
counterSignatureParameters.setSignatureIdToCounterSign(signatureId);

// Initialize a new service for the counter signature creation
// The counter signature will be created in three steps, similarly as a normal
signature
XAdESService service = new XAdESService(commonCertificateVerifier);
// First step is to get toBeSigned, which represents a SignatureValue of the master
signature
ToBeSigned dataToBeCounterSigned = service.getDataToBeCounterSigned(signedDocument,
counterSignatureParameters);
// Second step is to compute the signatureValue on the dataToBeCounterSigned
SignatureValue signatureValue = signingToken.sign(dataToBeCounterSigned,
counterSignatureParameters.getDigestAlgorithm(), privateKey);
// Third step is to create the counter signed signature document
DSSDocument counterSignedSignature = service.counterSignSignature(signedDocument,
counterSignatureParameters, signatureValue);
```

Various parameters

Signature policy

With the new standards the policy handling is linked to -B level. The old -EPES level is not used anymore by the framework. This does not alter the structure of the old signature but only modifies how to control the process of its creation.

The DSS framework allows you to reference a signature policy, which is a set of rules for the creation and validation of an electronic signature. It includes two kinds of text:

- In a human readable form: It can be assessed to meet the requirements of the legal and contractual context in which it is being applied.
- In a machine processable form: To facilitate its automatic processing using the electronic rules.

If no signature policy is identified then the signature may be assumed to have been generated or verified without any policy constraints, and hence may be given no specific legal or contractual significance through the context of a signature policy.

The signer may reference the policy either implicitly or explicitly. An implied policy means the signer follows the rules of the policy but the signature does not indicate which policy. It is assumed the choice of policy is clear from the context in which the signature is used and `SignaturePolicyIdentifier` element will be empty. When the policy is not implied, the signature contains an `ObjectIdentifier` that uniquely identifies the version of the policy in use. The signature also contains a hash of the policy document to make sure that the signer and verifier agree on the contents of the policy document.

This example demonstrates an implicit policy identifier. To implement this alternative you must set `SignaturePolicyId` to empty string.

XAdES with implicit policy

```
XAdESSignatureParameters parameters = new XAdESSignatureParameters();
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_B);
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPED);
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

BLevelParameters bLevelParameters = parameters.bLevel();

Policy policy = new Policy();
policy.setId("");

bLevelParameters.setSignaturePolicy(policy);

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
// Create xadesService for signature
XAdESService service = new XAdESService(commonCertificateVerifier);

// Get the SignedInfo segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm
DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm,
privateKey);

// We invoke the xadesService to sign the document with the signature value obtained
// in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);
```

An XML segment will be added to the signature's qualified and signed properties:

```
<xades:SignaturePolicyIdentifier>
  <xades:SignaturePolicyImplied/>
</xades:SignaturePolicyIdentifier>
```

The next example demonstrates an explicit policy identifier. This is obtained by setting -B profile signature policy and assigning values to the policy parameters. The Signature Policy Identifier is a URI or OID that uniquely identifies the version of the policy document. The signature will contain the identifier of the hash algorithm and the hash value of the policy document. The DSS framework

does not automatically calculate the hash value; it is to the developer to proceed with the calculation using for example `java.security.MessageDigest` class (rt.jar). It is important to keep the policy file intact in order to keep the hash constant. It would be wise to make the policy file read-only. See also chapter 7 for further information.

```
XAdESSignatureParameters parameters = new XAdESSignatureParameters();
parameters.setSignatureLevel(SignatureLevel.XAdES_BASELINE_B);
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPED);
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// We set the signing certificate
parameters.setSigningCertificate(privateKey.getCertificate());
// We set the certificate chain
parameters.setCertificateChain(privateKey.getCertificateChain());

BLevelParameters bLevelParameters = parameters.bLevel();

// Get and use the explicit policy
String signaturePolicyId = "http://www.example.com/policy.txt";
DigestAlgorithm signaturePolicyHashAlgo = DigestAlgorithm.SHA256;
DSSDocument policyContent = new InMemoryDocument("Policy text to digest".getBytes());
byte[] digestedBytes = DSSUtils.digest(signaturePolicyHashAlgo, policyContent);

Policy policy = new Policy();
policy.setId(signaturePolicyId);
policy.setDigestAlgorithm(signaturePolicyHashAlgo);
policy.setDigestValue(digestedBytes);

bLevelParameters.setSignaturePolicy(policy);

// Create common certificate verifier
CommonCertificateVerifier commonCertificateVerifier = new CommonCertificateVerifier();
// Create xadesService for signature
XAdESService service = new XAdESService(commonCertificateVerifier);

// Get the SignedInfo segment that need to be signed.
ToBeSigned dataToSign = service.getDataToSign(toSignDocument, parameters);

// This function obtains the signature value for signed information using the
// private key and specified algorithm
DigestAlgorithm digestAlgorithm = parameters.getDigestAlgorithm();
SignatureValue signatureValue = signingToken.sign(dataToSign, digestAlgorithm,
privateKey);

// We invoke the xadesService to sign the document with the signature value obtained
// in
// the previous step.
DSSDocument signedDocument = service.signDocument(toSignDocument, parameters,
signatureValue);
```

The following XML segment will be added to the signature qualified & signed properties (<QualifyingProperties><SignedProperties>):

```
<xades:SignaturePolicyIdentifier>
  <xades:SignaturePolicyId>
    <xades:SigPolicyId>
      <xades:Identifier>http://www.example.com/policy.txt</xades:Identifier>
    </xades:SigPolicyId>
    <xades:SigPolicyHash>
      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
      <ds:DigestValue>Uw3PxkrX4SpF03jDvkSu6Zqm9UXDxs56FFXeg7MWy0c=</ds:DigestValue>
    </xades:SigPolicyHash>
  </xades:SignaturePolicyId>
</xades:SignaturePolicyIdentifier>
```

Signature Policy Store

Since v5.8 DSS provides a possibility of incorporation of a Signature Policy Store element as an unsigned property to the existing signature file.

The following signature formats support the Signature Policy Store addition:

- XAdES (as well as ASiC with XAdES);
- CAdES (as well as ASiC with CAdES);
- JAdES.



Being an unsigned component the Signature Policy Store is not protected by a digital signature, unlike a Signature Policy Identifier incorporated into the signed properties.

Before incorporating of a Signature Policy Store, you need to ensure the target signature contains the matching Signature Policy Identifier element (see ch. [Signature policy](#)).

An example of a Signature Policy Store creation is available below:

```
// Create the SignaturePolicyStore object
SignaturePolicyStore signaturePolicyStore = new SignaturePolicyStore();
// Provide the policy content referenced within Signature Policy Identifier
signaturePolicyStore.setSignaturePolicyContent(policyContent);
// Define Id of the policy
SpDocSpecification spDocSpec = new SpDocSpecification();
spDocSpec.setId(signaturePolicyId);
signaturePolicyStore.setSpDocSpecification(spDocSpec);

// add the SignaturePolicyStore
XAdESService xadesService = new XAdESService(commonCertificateVerifier);
DSSDocument signedDocumentWithSignaturePolicyStore =
xadesService.addSignaturePolicyStore(signedDocument, signaturePolicyStore);
```

Trust anchor inclusion policy

It is possible to indicate to the framework if the certificate related to the trust anchor should be included to the signature or not. The setter `#setTrustAnchorBPPolicy` of the `BLevelParameters` class should be used for this purpose.

This rule applies as follows: when -B level is constructed the trust anchor is not included, when -LT level is constructed the trust anchor is included.



When trust anchor baseline profile policy is defined only the certificates previous to the trust anchor are included when -B level is constructed.

Timestamps

Timestamp creation

Since DSS 5.6 the framework allows an independent document timestamping (without a signature). The following Document Signature Services support the timestamping :

- **PAdESService** - adds a timestamp to a PDF document;
- **ASiCWithCAdESService** - creates a timestamped ASiC container with provided documents.

PDF timestamping

```
// Loads a document to be timestamped
DSSDocument documentToTimestamp = new FileDocument(new File("src/main/resources/hello-world.pdf"));

// Configure a PAdES service for PDF timestamping
PAdESService service = new PAdESService(getCompleteCertificateVerifier());
service.setTspSource(getGoodTsa());

// Execute the timestamp method
DSSDocument timestampedDoc = service.timestamp(documentToTimestamp, new
PAdESTimestampParameters());
```

Timestamp validation

As well as a single timestamp creation, DSS provides a validation service for timestamped documents. The timestamp validation process represents "5.4 Time-stamp validation building block" (cf. [\[R09\]](#)). The validation process is identical to [Signature Validation](#) process. An appropriate validator will be selected automatically. In total, DSS supports timestamp-alone validation for the following file formats:

- Detached CMS timestamp ([DetachedTimestampValidator](#)) - a detached signed content must be provided (or its digest);
- PDF document ([PDFDocumentValidator](#));
- ASiC CAdES container with a timestamp ([ASiCWithCAdESTimestampValidator](#)).

The validation process can be run with the following inputs :

Timestamped document validation

```
// Load a document validator. The appropriate validator class will be determined
// automatically.
SignedDocumentValidator validator =
SignedDocumentValidator.fromDocument(timestampedDoc);
// Configure the validator. Provide a certificate verifier.
validator.setCertificateVerifier(getCompleteCertificateVerifier());
// Validate the document
Reports reports = validator.validateDocument();
```

The produced reports use the same structure as for [Signature Validation](#).

Timestamp qualification

DSS is also able to determine a qualification level of a timestamp, if a relative information about TrustServiceProviders is provided to a certificate verifier (loaded automatically to a trusted certificate source with [TLValidationJob](#)) (cf. [\[R14\]](#)).

Three qualification levels are supported by DSS and can be obtained :

- **QTSA** (issued from a granted trust service with TSA/QTST type at the timestamp production time);
- **TSA** any other from a known trust anchor;
- **N/A** for others.

An example of a produced Detailed Report you can see below:

Timestamp Detailed Report

```
<DetailedReport ValidationTime="2021-07-09T10:10:12Z"
  xmlns="http://dss.esig.europa.eu/validation/detailed-report">
  <Timestamp Id="T-
  DBCE6DC6BFDD5713D9281FE79CAB93927961746BC0D2F13B3F4DE252DF494684">
    <ValidationProcessTimestamp Type="CONTENT_TIMESTAMP" ProductionTime="2021-07-
  09T10:10:11Z" Title="Validation Process for Time-stamps">
      <Constraint Id="T-
  DBCE6DC6BFDD5713D9281FE79CAB93927961746BC0D2F13B3F4DE252DF494684-ISC">
        <Name Key="BSV_IISCRC">Is the result of the 'Identification of Signing
  Certificate' building block conclusive?</Name>
        <Status>OK</Status>
      </Constraint>
      <Constraint Id="T-
  DBCE6DC6BFDD5713D9281FE79CAB93927961746BC0D2F13B3F4DE252DF494684-XCV">
        <Name Key="BSV_IIXCVR">Is the result of the 'X.509 Certificate
  Validation' building block conclusive?</Name>
        <Status>WARNING</Status>
        <Warning Key="BSV_IIXCVR_ANTS">The result of the 'X.509 Certificate
  Validation' building block is not conclusive!</Warning>
      </Constraint>
      <Constraint Id="T-
  DBCE6DC6BFDD5713D9281FE79CAB93927961746BC0D2F13B3F4DE252DF494684-XCV">
        <Name Key="BSV_ISCRREVTC">Is the signing certificate not revoked at
  validation time?</Name>
        <Status>OK</Status>
      </Constraint>
      <Constraint Id="T-
  DBCE6DC6BFDD5713D9281FE79CAB93927961746BC0D2F13B3F4DE252DF494684-XCV">
        <Name Key="BSV_IVTAVRSC">Is the validation time in the validity range
  of the signing certificate?</Name>
        <Status>OK</Status>
      </Constraint>
      <Constraint Id="T-
  DBCE6DC6BFDD5713D9281FE79CAB93927961746BC0D2F13B3F4DE252DF494684-CV">
        <Name Key="BSV_ICVRC">Is the result of the 'Cryptographic
  Verification' building block conclusive?</Name>
        <Status>OK</Status>
      </Constraint>
      <Constraint Id="T-
  DBCE6DC6BFDD5713D9281FE79CAB93927961746BC0D2F13B3F4DE252DF494684">
```

```

        <Name Key="ADEST_ROBVPIIC">Is the result of the Basic Validation
Process conclusive?</Name>
        <Status>NOT OK</Status>
        <Error Key="ADEST_ROBVPIIC_ANS">The result of the Basic validation
process is not conclusive!</Error>
        <AdditionalInfo>Basic Signature Validation process failed with
INDETERMINATE/TRY_LATER indication</AdditionalInfo>
        </Constraint>
        <Conclusion>
            <Indication>INDETERMINATE</Indication>
            <SubIndication>TRY_LATER</SubIndication>
            <Errors Key="BBB_XCV_SUB_ANS">The certificate validation is not
conclusive!</Errors>
            <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
        </Conclusion>
    </ValidationProcessTimestamp>
    <ValidationTimestampQualification TimestampQualification="N/A" Title="Time-
stamp Qualification">
        <Constraint>
            <Name Key="QUAL_CERT_TRUSTED_LIST_REACHED">Has a trusted list been
reached for the certificate chain?</Name>
            <Status>NOT OK</Status>
            <Error Key="QUAL_CERT_TRUSTED_LIST_REACHED_ANS">Unable to build a
certificate chain up to a trusted list!</Error>
        </Constraint>
        <Conclusion>
            <Indication>FAILED</Indication>
            <Errors Key="QUAL_CERT_TRUSTED_LIST_REACHED_ANS">Unable to build a
certificate chain up to a trusted list!</Errors>
        </Conclusion>
    </ValidationTimestampQualification>
</Timestamp>
<BasicBuildingBlocks Id="T-
DBCE6DC6BFDD5713D9281FE79CAB93927961746BC0D2F13B3F4DE252DF494684" Type="TIMESTAMP">
    <ISC Title="Identification of the Signing Certificate">
        <Constraint>
            <Name Key="BBB_ICS_ISCI">Is there an identified candidate for the
signing certificate?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name Key="BBB_ICS_ISACDP">Is the signed attribute: 'cert-digest' of
the certificate present?</Name>
            <Status>OK</Status>
        </Constraint>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
        <CertificateChain>
            <ChainItem Id="C-

```

```

C28D73354E868E7E672D13A872658539C579E820059446AC04EB002DCCCB02A7">
    <Source>TIMESTAMP</Source>
    </ChainItem>
    <ChainItem Id="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F">
    <Source>TRUSTED_STORE</Source>
    </ChainItem>
    </CertificateChain>
</ISC>
<XCV Title="X509 Certificate Validation">
    <Constraint>
        <Name Key="BBB_XCV_CCCBB">Can the certificate chain be built till a
trust anchor?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint Id="C-
C28D73354E868E7E672D13A872658539C579E820059446AC04EB002DCCCB02A7" BlockType="SUB_XCV">
        <Name Key="BBB_XCV_SUB">Is the certificate validation
conclusive?</Name>
        <Status>NOT OK</Status>
        <Error Key="BBB_XCV_SUB_ANS">The certificate validation is not
conclusive!</Error>
    </Constraint>
    <Conclusion>
        <Indication>INDETERMINATE</Indication>
        <SubIndication>TRY_LATER</SubIndication>
        <Errors Key="BBB_XCV_SUB_ANS">The certificate validation is not
conclusive!</Errors>
        <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
    </Conclusion>
    <SubXCV Id="C-
C28D73354E868E7E672D13A872658539C579E820059446AC04EB002DCCCB02A7" TrustAnchor="false"
SelfSigned="false" Title="Certificate">
        <Constraint>
            <Name Key="BBB_XCV_ISNSSC">Is certificate not self-signed?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name Key="BBB_XCV_ICSI">Is the certificate signature
intact?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name Key="BBB_XCV_ISCGEKGU">Does the signer's certificate have an
expected 'extended key-usage(s)'?</Name>
            <Status>OK</Status>
            <AdditionalInfo>Extended key usage :
[timeStamping]</AdditionalInfo>
            <Status>OK</Status>
        </Constraint>
        <Constraint>

```

```

<Name Key="BBB_XCV_IRDPFC">Is the revocation data present for the
certificate?</Name>
    <Status>NOT OK</Status>
    <Error Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Error>
    </Constraint>
    <Conclusion>
        <Indication>INDETERMINATE</Indication>
        <SubIndication>TRY_LATER</SubIndication>
        <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
    </Conclusion>
    <RFC Title="Revocation Freshness Checker">
        <Constraint>
            <Name Key="BBB_XCV_IARDPFC">Is an acceptable revocation data
present for the certificate?</Name>
            <Status>NOT OK</Status>
            <Error Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation data
for the certificate!</Error>
            </Constraint>
            <Conclusion>
                <Indication>INDETERMINATE</Indication>
                <SubIndication>TRY_LATER</SubIndication>
                <Errors Key="BBB_XCV_IARDPFC_ANS">No acceptable revocation
data for the certificate!</Errors>
            </Conclusion>
        </RFC>
    </SubXCV>
    <SubXCV Id="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F" TrustAnchor="true"
SelfSigned="true" Title="Certificate">
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
    </SubXCV>
</XCV>
<CV Title="Cryptographic Verification">
    <Constraint>
        <Name Key="BBB_CV_TSP_IRDOF">Has the message imprint data been
found?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint>
        <Name Key="BBB_CV_TSP_IRDOI">Is the message imprint data
intact?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint>
        <Name Key="BBB_CV_ISIT">Is time-stamp's signature intact?</Name>
        <Status>OK</Status>
    </Constraint>

```

```

<Conclusion>
    <Indication>PASSED</Indication>
</Conclusion>
</CV>
<SAV Title="Signature Acceptance Validation">
    <Constraint>
        <Name Key="BBB_ICS_ISASCP">Is the signed attribute: 'signing-
certificate' present?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint>
        <Name Key="BBB_SAV_DSCACRCC">Does the 'Signing Certificate' attribute
contain references only to the certificate chain?</Name>
        <Status>OK</Status>
    </Constraint>
    <Constraint>
        <Name Key="ACCM">Are cryptographic constraints met for the time-stamp
signature?</Name>
        <Status>OK</Status>
        <AdditionalInfo>Signature algorithm RSA with SHA256 with key size 2048
at validation time : 2021-07-09 10:10</AdditionalInfo>
    </Constraint>
    <Constraint>
        <Name Key="ACCM">Are cryptographic constraints met for the message
imprint?</Name>
        <Status>OK</Status>
        <AdditionalInfo>Digest algorithm SHA256 at validation time : 2021-07-
09 10:10 for message imprint</AdditionalInfo>
    </Constraint>
    <Conclusion>
        <Indication>PASSED</Indication>
    </Conclusion>
    <CryptographicValidation secure="true">
        <Algorithm>
            <Name>RSA with SHA256</Name>
            <Uri>http://www.w3.org/2001/04/xmldsig-more#rsa-sha256</Uri>
            <KeyLength>2048</KeyLength>
        </Algorithm>
        <NotAfter>2022-12-31T23:00:00Z</NotAfter>
        <ValidationTime>2021-07-09T10:10:12Z</ValidationTime>
        <ConcernedMaterial>T-
DBCE6DC6BFDD5713D9281FE79CAB93927961746BC0D2F13B3F4DE252DF494684</ConcernedMaterial>
        </CryptographicValidation>
    </SAV>
    <CertificateChain>
        <ChainItem Id="C-
C28D73354E868E7E672D13A872658539C579E820059446AC04EB002DCCCB02A7">
            <Source>TIMESTAMP</Source>
        </ChainItem>
        <ChainItem Id="C-
DE4E0E6E69ED670AC0AE41C815735B21F9394EA8A202F2497510EAAF5A4F998F">

```

```

<Source>TRUSTED_STORE</Source>
</ChainItem>
</CertificateChain>
<Conclusion>
  <Indication>INDETERMINATE</Indication>
  <SubIndication>TRY_LATER</SubIndication>
  <Errors Key="BBB_XCV_SUB_ANS">The certificate validation is not
conclusive!</Errors>
  <Errors Key="BBB_XCV_IRDPFC_ANS">No revocation data found for the
certificate!</Errors>
</Conclusion>
</BasicBuildingBlocks>
</DetailedReport>

```

Available implementations of DSSDocument

DSS allows creation of different kinds of DSSDocument :

- **InMemoryDocument** : fully loads in memory. This type of DSSDocument can be instantiated with an array of bytes, an InputStream,...
- **FileDocument** : refers an existing File
- **DigestDocument** : only contains pre-computed digest values for a given document. That allows a user to avoid sending the full document (detached signatures).

DigestDocument

```

// Firstly, we load a basic DSSDocument (FileDocument or InMemoryDocument)
DSSDocument fileDocument = new FileDocument("src/main/resources/xml_example.xml");

// After that, we create a DigestDocument
DigestDocument digestDocument = new DigestDocument(DigestAlgorithm.SHA1,
fileDocument.getDigest(DigestAlgorithm.SHA1));
digestDocument.setName(fileDocument.getName());

// We can add an additional needed digest value(s). Eg : for a SHA-256 based signature
digestDocument.addDigest(DigestAlgorithm.SHA256,
fileDocument.getDigest(DigestAlgorithm.SHA256));

```

Management of signature tokens

The DSS framework is able to create signatures from PKCS#11, PKCS#12 and MS CAPI. Java 6 is inherently capable of communicating with these kinds of KeyStores. To be independent of the signing media, DSS framework uses an interface named SignatureTokenConnection to manage different implementations of the signing process. The base implementation is able to sign a stream of the data in one step. That means that all the data to be signed needs to be sent to the SSCD. This is the case for MS CAPI. As to the PKCS#11 and PKCS#12, which give to the developer a finer control in

the signature operation, the DSS framework implements the AsyncSignatureTokenConnection abstract class that permits to execute the digest operation and signature operation in two different threads or even two different hardwares.

This design permits also other card providers/adopters to create own implementations. For example, this can be used for a direct connection to the Smartcard through Java 6 PC/SC.

PKCS#11

PKCS#11 is widely used to access smart cards and HSMs. Most commercial software uses PKCS#11 to access the signature key of the CA or to enrol user certificates. In the DSS framework, this standard is encapsulated in the class Pkcs11SignatureToken.

Pkcs11SignatureToken usage

```
try (Pkcs11SignatureToken token = new
Pkcs11SignatureToken("C:\\Windows\\System32\\beidpkcs11.dll")) {

    List<DSSPrivateKeyEntry> keys = token.getKeys();
    for (DSSPrivateKeyEntry entry : keys) {
        System.out.println(entry.getCertificate().getCertificate());
    }

    ToBeSigned toBeSigned = new ToBeSigned("Hello world".getBytes());
    SignatureValue signatureValue = token.sign(toBeSigned, DigestAlgorithm.SHA256,
keys.get(0));

    System.out.println("Signature value : " +
Utils.toBase64(signatureValue.getValue()));
}
```

PKCS#12

This standard defines a file format commonly used to store the private key and corresponding public key certificate protecting them by password.

In order to use this format with the DSS framework you have to go through the class Pkcs12SignatureToken.

Pkcs12SignatureToken usage

```
try (Pkcs12SignatureToken token = new
Pkcs12SignatureToken("src/main/resources/user_a_rsa.p12", new
PasswordProtection("password".toCharArray())) {

    List<DSSPrivateKeyEntry> keys = token.getKeys();
    for (DSSPrivateKeyEntry entry : keys) {
        System.out.println(entry.getCertificate().getCertificate());
    }

    ToBeSigned toBeSigned = new ToBeSigned("Hello world".getBytes());
    SignatureValue signatureValue = token.sign(toBeSigned, DigestAlgorithm.SHA256,
keys.get(0));

    System.out.println("Signature value : " +
Utils.toBase64(signatureValue.getValue()));
}
```

MS CAPI

If the middleware for communicating with an SSDC provides a CSP based on MS CAPI specification, then to sign the documents you can use `MSCAPISignatureToken` class.

MSCAPISignatureToken usage

```
try (MSCAPISignatureToken token = new MSCAPISignatureToken()) {

    List<DSSPrivateKeyEntry> keys = token.getKeys();
    for (DSSPrivateKeyEntry entry : keys) {
        System.out.println(entry.getCertificate().getCertificate());
    }

    ToBeSigned toBeSigned = new ToBeSigned("Hello world".getBytes());
    SignatureValue signatureValue = token.sign(toBeSigned, DigestAlgorithm.SHA256,
keys.get(0));

    System.out.println("Signature value : " +
Utils.toBase64(signatureValue.getValue()));
}
```

Other Implementations

As you can see, it is easy to add another implementation of the `SignatureTokenConnection`, thus enabling the framework to use other API than the provided three (PKCS#11, PKCS#12 and MS CAPI). For example, it is likely that in the future PC/SC will be the preferred way of accessing a Smartcard. Although PKCS#11 is currently the most used API, DSS framework is extensible and can use PC/SC. For our design example we propose to use PC/SC to communicate with the Smartcard.

Management of certificates sources

The validation of a certificate requires the access to some other certificates from multiple sources like trusted lists, trust store, the signature itself: certificates can be contained inside or any other source. Within the framework, an X509 certificate is wrapped through the class:

- `eu.europa.esig.dss.model.x509.CertificateToken`

This encapsulation helps make certificate handling more suited to the needs of the validation in the context of trust. The framework associates two internal identifiers to the certificate : the DSS Id based on the certificate binary (unique for each certificate) and the Entity Id based on its public key (common to cross-signed certificates).

Certificate tokens are grouped into sources. A certificate token can be declared in several sources. The class that models a source is called:

- `eu.europa.esig.dss.spi.x509.CertificateSource`

This class stores all extracted/injected certificates for a specific source (Signature, OCSP Response, Trust store, Trusted-list,...). All source types are specified in the enumeration :

- `eu.europa.esig.dss.enumerations.CertificateSourceType`

This information is used, for example, to distinguish between the certificate from a trusted source and the others. A source has one and only one type, but a certificate token can be found in multiple sources. The DSS framework supplies some standard implementations, but also gives the possibility to implement owner solutions. Among the standard solutions you can find:

- `eu.europa.esig.dss.spi.x509.CommonCertificateSource`

This is the superclass of almost of the certificate sources. It stores the extracted certificates and implements the common methods from the `CertificateSource` to retrieve certificate(s) by subject, public key, subject key identifier (ski),...

It also exposes the method `CommonCertificateSource#addCertificate` which gives the possibility to add manually any `CertificateToken` as a part of this source.

- `eu.europa.esig.dss.spi.x509.CommonTrustedCertificateSource`

The `CommonTrustedCertificateSource` is a certificate source for trusted certificates. All added certificates are marked as trust anchors and no revocation data are required for these certificates.

- `eu.europa.esig.dss.validation.SignatureCertificateSource`

This class and its sub-classes are used to extract and collect certificates from signatures / timestamps. It also has methods to retrieve certificates / certificate references by their origin (eg : `SigningCertificate` attribute, DSS Dictionary,...).

- `eu.europa.esig.dss.spi.tsl.TrustedListsCertificateSource`

Certificates coming from the list of Trusted Lists. This class inherits of

[CommonTrustedCertificateSource](#) and gives the mechanism to define the set of trusted certificates (trust anchors). They are used in the validation process to decide if the prospective certificate chain has a trust anchor. See chapter [TLValidationJob](#) to get more information about trusted lists loading (e.g. EU Trusted List).

- [eu.europa.esig.dss.spi.x509.ListCertificateSource](#)

This class follows the composite design pattern with a list of CertificateSources. That's used in the validation to retrieve all sources from the signatures / timestamps / revocation data / trusted lists /... It contains some methods which check over all sources to retrieve certificates or verify if a certificate is trusted.

Management of CRL and OCSP sources

A CRL is a time-stamped list identifying revoked certificates. It is signed by a Certificate Authority (CA) and made freely available in a public repository. Each revoked certificate is identified in a CRL by its certificate serial number.

The Online Certificate Status Protocol (OCSP) is an Internet protocol used for obtaining the revocation status of an unique X.509 digital certificate.

For every certificate, the validity has to be checked via CRL or OCSP responses. The information may originate from different CRLSources or OCSPSources: For easing the usage of such sources, DSS implements a CRLSource and OCSPSource interfaces (which inherit from RevocationSource), which offer a generic, uniform way of accessing CRL and OCSP sources. Furthermore, a caching mechanism can be easily attached to those sources, optimizing the access time to revocation information by reducing network connections to online servers.

The interface CRLSource defines the method which returns CRLToken for the given certificate/issuer certificate couple:

CRLSource usage

```
CRLToken crlToken = crlSource.getRevocationToken(certificateToken,  
issuerCertificateToken);
```

The interface OCSPSource defines the method which returns OCSPToken for the given certificate/issuer certificate couple:

OCSPSource usage

```
OCSPToken ocspToken = ocspSource.getRevocationToken(certificateToken,  
issuerCertificateToken);
```

We use these classes during the certificate validation process through "validationContext" object (based on ValidationContext class) which is a "cache" for one validation request that contains every object retrieved so far. This object in turn instantiates a "verifier" based on [RevocationDataLoadingStrategy](#) class whose role is to fetch revocation data by querying an OCSP or

CRL source in the defined order and return the succeeded result. In general, we can distinguish three main sources:

- Offline sources ([OfflineRevocationSource](#));
- Online sources ([OnlineRevocationSource](#));
- Sources with the cache mechanism;
- List of sources ([ListRevocationSource](#)) with a collection of several sources.

Repository Revocation Source

The above-mentioned class allows caching of CRL and OCSP responses to a user-chosen source. By default DSS provides a JDBC based implementation for this class, but other implementations also can be created. The class contains a complete set of functions to save revocation data to a database, extract, update and remove it.

Furthermore, the [RepositoryRevocationSource](#) allows the implementer to define a backup revocation source, for the case if the database does not contains the certificate's revocation data yet.

List of cached Revocation sources implemented in DSS:

- [JdbcRevocationSource](#)
 - [JdbcCacheCRLSource](#)
 - [JdbcCacheOCSPSource](#)

The classes allow the following configuration :

JdbcCacheCRLSource usage

```
// Creates an instance of JdbcCacheCRLSource
JdbcCacheCRLSource cacheCRLSource = new JdbcCacheCRLSource();

// Initialize the JdbcCacheConnector
JdbcCacheConnector jdbcCacheConnector = new JdbcCacheConnector(dataSource);

// Set the JdbcCacheConnector
cacheCRLSource.setJdbcCacheConnector(jdbcCacheConnector);

// Allows definition of an alternative dataLoadet to be used to access a revocation
// from online sources if a requested revocation is not present in the repository or
// has been expired (see below).
cacheCRLSource.setProxySource(onlineCRLSource);

// All setters accept values in seconds
Long oneWeek = (long) (60 * 60 * 24 * 7); // seconds * minutes * hours * days

// If "nextUpdate" field is not defined for a revocation token, the value of
// "defaultNextUpdateDelay"
// will be used in order to determine when a new revocation data should be requested.
// If the current time is not beyond the "thisUpdate" time + "defaultNextUpdateDelay",
// then a revocation data will be retrieved from the repository source, otherwise a
// new revocation data
// will be requested from a proxiedSource.
// Default : null (a new revocation data will be requested of "nestUpdate" field is
// not defined).
cacheCRLSource.setDefaultNextUpdateDelay(oneWeek);

// Defines a custom maximum possible nextUpdate delay. Allows limiting of a time
// interval
// from "thisUpdate" to "nextUpdate" defined in a revocation data.
// Default : null (not specified, the "nextUpdate" value provided in a revocation is
// used).
cacheCRLSource.setMaxNextUpdateDelay(oneWeek); // force refresh every week (eg : ARL)

// Defines if a revocation should be removed on its expiration.
// Default : true (removes revocation from a repository if expired).
cacheCRLSource.setRemoveExpired(true);

// Creates an SQL table
cacheCRLSource.initTable();

// Extract CRL for a certificate
CRLToken crlRevocationToken = cacheCRLSource.getRevocationToken(certificateToken,
issuerCertificateToken);
```

And an example for JdbcCacheOCSPSource :

JdbcCacheOCSPSource usage

```
JdbcCacheOCSPSource cacheOCSPSource = new JdbcCacheOCSPSource();
JdbcCacheConnector jdbcCacheConnector = new JdbcCacheConnector(dataSource);
cacheOCSPSource.setJdbcCacheConnector(jdbcCacheConnector);
cacheOCSPSource.setProxySource(onlineOCSPSource);
Long threeMinutes = (long) (60 * 3);
cacheOCSPSource.setDefaultNextUpdateDelay(threeMinutes); // default nextUpdateDelay
(if not defined in the revocation data)
cacheOCSPSource.initTable();
OCSPToken ocspRevocationToken = cacheOCSPSource.getRevocationToken(certificateToken,
certificateToken);
```

Be aware that you have to initialize a table before start of working with the cached revocation repository.

Other implementations of CRL and OCSP Sources

Such sources find the status of a certificate either from a list stored locally or using the information contained in the advanced signature or online way. Here is the list of sources already implemented in the DSS framework:

- CRL sources
 - **OfflineCRLSource** : This class implements the **OfflineRevocationSource** and retrieves the revocation data from extracted information. The code is common for all signature formats and CRL contents are injected by its sub-classes :
 - **CMSCLSource** : Extracts CRLs and CRL references from a CMS Signed Data :
 - **CAdESCRLSource** : Sub-class of **CMSCLSource** for a CAdES Signature;
 - **TimestampCRLSource** : Sub-class of **CMSCLSource** for a Timestamp token (RFC 3161);
 - **PAdESCRLSource** : Extracts CRLs and CRL references from a PAdES signature.
 - **XAdESCRLSource** : Extracts CRLs and CRL references from a XAdES signature.
 - **ExternalResourcesCRLSource** : A class that can instantiate a list of certificate revocation lists from a directory where should be the individual lists.
 - **OnlineCRLSource** : Retrieves CRL files from online sources with the CRL Distribution Points information from the certificate.
 - **JdbcCacheCrlSource** : Implementation of the **JdbcRevocationSource**. This implementation allows storage of valid CRL entries to a defined **DataSource** and retrieve them locally.
- OCSP sources
 - **OfflineOCSPSource** : This class implements the **OfflineRevocationSource** and retrieves the revocation data from extracted information. The code is common for all signature formats and OCSP responses are injected by its sub-classes :
 - **CMSOCSPSource** : Extracts OCSP responses and OCSP references from a CMS Signed Data :
 - **CAdESOCSPSource** : Sub-class of **CMSOCSPSource** for a CAdES Signature;

- **TimestampOCSPSource**: Sub-class of **CMSOCSPSource** for a Timestamp token (RFC 3161);
- **PAdSOCSPSource** : Extracts OCSP responses and OCSP references from a PAdES signature.
- **XAdSOCSPSource** : Extracts OCSP responses and OCSP references from a XAdES signature.
 - **ExternalResourcesOCSPSource** : A class that can instantiate a list of OCSPToken from a directory where should be the individual DER Encoded X509 certificates files.
- **OnlineOCSPSource** : Retrieves OCSP responses from online source.
- **JdbcCacheOcspSource** : Implementation of the **JdbcRevocationSource**. This implementation allows storage of valid OCSP entries to a defined **DataSource** and retrieve them locally.

Online CRL Source

This is a representation of an Online CRL repository. This implementation will contact using HTTP protocol the CRL Responder to download the CRLs from the given URI. Note that certificate's Authority Information Access (AIA) extension is used to find issuer's resources location like CRL file and/or Online Certificate Status Protocol (OCSP). The URIs of CRL server will be extracted from this property (OID value: 1.3.6.1.5.5.7.48.1.3).

It allows the following configuration :

OnlineCRLSource usage

```
// Instantiates a new OnlineCRLSource
OnlineCRLSource onlineCRLSource = new OnlineCRLSource();

// Allows setting an implementation of 'DataLoader' interface,
// processing a querying of a remote revocation server.
// 'CommonsDataLoader' instance is used by default.
onlineCRLSource.setDataLoader(new CommonsDataLoader());

// Sets a preferred protocol that will be used for obtaining a CRL.
// E.g. for a list of urls with protocols HTTP, LDAP and FTP, with a defined preferred
// protocol as FTP,
// the FTP url will be called first, and in case of an unsuccessful result other url
// calls will follow.
// Default : null (urls will be called in a provided order).
onlineCRLSource.setPreferredProtocol(Protocol.FTP);
```

Online OCSP Source

This is a representation of an Online OCSP repository. This implementation will contact using HTTP protocol the OCSP Responder to retrieve the OCSP response. Note that certificate's Authority Information Access (AIA) extension is used to find issuer's resources location like CRT file and/or Online Certificate Status Protocol (OCSP). The URIs of OCSP server will be extracted from this property (OID value: 1.3.6.1.5.5.7.48.1).

It allows the following configuration :

OnlineOCSPSource usage

```
// Instantiates a new OnlineOCSPSource object
OnlineOCSPSource onlineOCSPSource = new OnlineOCSPSource();

// Allows setting an implementation of 'DataLoader' interface,
// processing a querying of a remote revocation server.
// 'CommonsDataLoader' instance is used by default.
onlineOCSPSource.setDataLoader(new OCSPDataLoader());

// Defines an arbitrary integer used in OCSP source querying in order to prevent a
// replay attack.
// Default : null (not used by default).
onlineOCSPSource.setNonceSource(new SecureRandomNonceSource());

// Defines a DigestAlgorithm being used to generate a CertificateID in order to
// complete an OCSP request.
// OCSP servers supporting multiple hash functions may produce a revocation response
// with a digest algorithm depending on the provided CertificateID's algorithm.
// Default : SHA1 (as a mandatory requirement to be implemented by OCSP servers. See
// RFC 5019).
onlineOCSPSource.setCertIDDigestAlgorithm(DigestAlgorithm.SHA1);
```

CertificateVerifier configuration

The CertificateVerifier and its implementation CommonCertificateVerifier determines how DSS accesses the external resources and how it should react in some occasions. This configuration is used in both extension and validation mode.

CertificateVerifier usage

```
CertificateVerifier cv = new CommonCertificateVerifier();

// The AIA source is used to collect certificates from external resources (AIA)
cv.setAIASource(aiaSource);

// The adjunct certificate source is used to provide missing intermediate certificates
// (not trusted certificates)
cv.setAdjunctCertSources(adjunctCertSource);

// The trusted certificate source is used to provide trusted certificates
// (the trust anchors where the certificate chain building should stop)
cv.setTrustedCertSources(trustedCertSource);

// The CRL Source to be used for external accesses (can be configured with a
// cache,...)
cv.setCrlSource(crlSource);

// The OCSP Source to be used for external accesses (can be configured with a
```

```

// cache,...)
cv.setOcspSource(ocspSource);

// Sets the default digest algorithm that will be used for digest calculation
// of tokens used during the validation process.
// The values will be used in validation reports.
// Default : DigestAlgorithm.SHA256
cv.setDefaultDigestAlgorithm(DigestAlgorithm.SHA512);

// Define the behavior to be followed by DSS in case of revocation checking for
// certificates issued from an unsure source (DSS v5.4+)
// Default : revocation check is disabled for unsure sources (security reasons)
cv.setCheckRevocationForUntrustedChains(false);

// DSS v5.4+ : The 3 below configurations concern the extension mode (LT/LTA
// extension)

// Defines a behavior in case of missing revocation data
// Default : ExceptionOnStatusAlert -> interrupt the process
cv.setAlertOnMissingRevocationData(new ExceptionOnStatusAlert());

// Defines a behavior if a TSU certificate chain is not covered with a
// revocation data (timestamp generation time > CRL/OCSP production time).
// Default : LogOnStatusAlert -> a WARN log
cv.setAlertOnUncoveredPOE(new LogOnStatusAlert(Level.WARN));

// Defines a behavior if a revoked certificate is present
// Default : ExceptionOnStatusAlert -> interrupt the process
cv.setAlertOnRevokedCertificate(new ExceptionOnStatusAlert());

// Defines a behavior if an invalid timestamp is found
// Default : ExceptionOnStatusAlert -> interrupt the process
cv.setAlertOnInvalidTimestamp(new ExceptionOnStatusAlert());

// DSS v5.5+ : defines a behavior in case if there is no valid revocation
// data with thisUpdate time after the best signature time
// Example: if a signature was extended to T level then the obtained revocation
// must have thisUpdate time after production time of the signature timestamp.
// Default : LogOnStatusAlert -> a WARN log
cv.setAlertOnNoRevocationAfterBestSignatureTime(new LogOnStatusAlert(Level.ERROR));

// DSS 5.9+ :
// Defines behavior in case if the signing certificate or its related POE(s) have been
// expired
// Default : ExceptionOnStatusAlert -> interrupt the process
cv.setAlertOnExpiredSignature(new ExceptionOnStatusAlert());

// DSS 5.9+ :
// RevocationDataLoadingStrategy defines logic for loading OCSP or CRL data
// Default : OCSPFirstRevocationDataLoadingStrategy -> loads OCSP first,
//           if not available or the response is invalid, then tries to load CRL

```

```
cv.setRevocationDataLoadingStrategy(new OCSPFirstRevocationDataLoadingStrategy());
```

Trust Anchor(s) configuration

Trust anchors represent an important part in the signature creation / validation. That defines which are the trusted entities, which signatures can be trusted,... Do I trust certificates/signatures from another company / country / ... ?

Since the version 5.6, DSS allows to configure one or more trusted certificate source(s). These sources can be configured from a TrustStore (kind of keystore which only contains certificates), a trusted list and/or a list of trusted lists.

Multiple trusted certificate sources usage

```
CertificateVerifier cv = new CommonCertificateVerifier();
cv.setTrustedCertSources(trustStoreSource(), trustedListSource());
```

Trust store initialization

If you have a collection of certificates to trust, the easier way to provide them to DSS is to use a KeyStore / TrustStore.

```
public CertificateSource trustStoreSource() throws IOException {
    KeyStoreCertificateSource keystore = new KeyStoreCertificateSource(new
File("src/main/resources/keystore.p12"), "PKCS12", getPassword());

    CommonTrustedCertificateSource trustedCertificateSource = new
CommonTrustedCertificateSource();
    trustedCertificateSource.importAsTrusted(keystore);

    // Optionally, certificates can also be directly added

    trustedCertificateSource.addCertificate(DSSUtils.loadCertificateFromBase64EncodedString(
        "MIIC9TCCAd2gAwIBAgIBAjANBgkqhkiG9w0BAQUFADArMQswCQYDVQQGEwJBQTEMAoGA1UEChMDRFNTMQ4wD
        AYDVQQDEwVJQ0EgQTAeFw0xMzEyMDIxNzMzMTBaFw0xNTEyMDIxNzMzMTBaMDAxCzAJBgNVBAYTakFBM
        QwwCgYDVQQKEwNEU1MxEzARBgNVBAMTCnVzZXIgQSBSU0EwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJ
        AoGBAJUHHAphmSDdQ1t62tppK+dLTANsE2nAj+HCpasS3oh1BsrhteRsvTAbryDyIzCmTYWu/nVI4T
        GvbzBESwV/Qit1koMLpYFw32MIBf2DLmEcJGJ3vm5haw6u8S9quR1h8Vu7QWd+5KMabZuR+j91Ri
        SuoY0xS2ZQxJw1vhvW9hRYjAgMBAAGjgAIwgZ8wCQYDVR0TBAIwADAdBgNVHQ4EFgQU9ESnTWfwg
        13c3LQZzqqwibY5WVYwUwYDVR0jBEwwSoAUI01CDsBSUcEoFZxKaWf1PAL1U+uhL6QtMCsxDDAKB
        gNVBAoTA0RTUzELMAkGA1UEBhMCQUExDjAMBgNVBAMTBVJDQSBBg
        gEBMAsGA1UdDwQEAvIHgDARBgNVHSAEcjAIMAYGBFUdIAAwDQYJKoZIhvcNAQEFBQADggEBAGnhhno
        yVUhDnr/BSbZ/uWfSuwzFPG+2V9K6WxdIaaX00RFGIdFwG1AwA/Qzp9snfBxuTkAykxq0uEDhHTj0qX
        xWRjQ+Dop/Drmc
        coF/zDvgGusyY1YXaABd/kc3IYt7ns7z3tpiqIz4A7a/UHplBRXfqjyaZurZuJQRaSdxh6CNh
        dEUiUBxkbb1SdMju0gjzSDjcDjcegjvDquMKdDetvtu2Qh4ConBBo3fUImwifRWnbudS5H2HE18ikC7gY
        /QIuNr7USf1PNyUgcG2g31cMtemj7UTBHZ2V/jPf7ZXqwfVSAyK
        NvM3weAI6R3PI0STjdxN6a9qjt9xld40YEdw="));
    return trustedCertificateSource;
}
```

To generate the trust store, there's an utility class `CreateKeyStoreApp` in the `dss-cookbook` module.

Trusted List Certificate Source

In several countries, a list of Trust Service Providers (TSP) is published. This list is usually published in a machine processable format (XML) and sometimes in a human-readable format (PDF). A standard (ETSI TS 119 612) exists with the specifications for the XML format.

DSS contains all needed resources to download, parse, validate and interpret the trusted list contents. Since DSS 5.6, that's possible to configure one or more independent trusted list(s) (aka not linked to a list of trusted lists) and/or one or more list of trusted lists.

If you want to collect your trusted certificates from trusted list(s), the `TrustedListsCertificateSource` is required. The trusted list(s) loading can require some times (connection time-out, xml parsing, xml validation,...). This process is usually executed in background. An instance of `TrustedListsCertificateSource` needs to be created. That will be synchronized with the `TLValidationJob`.

```
public CertificateSource trustedListSource() {  
    return new TrustedListsCertificateSource();  
}
```

TLValidationJob

The TLValidationJob allows to download, parse, validate the Trusted List(s) and Lists Of Trusted Lists (LOTL). Once the task is done, its result is stored in the TrustedListsCertificateSource. The job uses 3 different caches (download, parsing and validation) and a state-machine to be efficient.

Trusted lists are stored on the file system. That offers the possibility to run in offline mode with the stored trusted lists. Trusted Lists can be loaded from the file system and/or from Internet.

In the next sections the different configurations will be covered.

TLSource and LOTLSource

Several TLSources and several LOTLSources can be injected in a TLValidationJob. The only constraint is the uniqueness of the Trusted List URLs.

Multiple TLSources and multiple LOTLSources configuration

```
TLValidationJob validationJob = new TLValidationJob();  
// Specify where is the TL/LOTL is hosted and which are the signing certificate(s) for  
// these TL/LOTL.  
validationJob.setTrustedListSources(boliviaTLSource(), costaRicaTLSource());  
validationJob.setListOfTrustedListSources(europeanLOTLSource(),  
unitedStatesLOTLSource());
```

Trusted List Source (TLSource)

A TLSource allows to quickly setup a trusted list configuration. The URL and the signing certificates for this TL are mandatory. Optionally, predicates / filters can be configured to retrieve only a part of the trust service providers or trust services.

TLSource configuration

```
TLSource tlSource = new TLSource();

// Mandatory : The url where the TL needs to be downloaded
tlSource.setUrl("http://www.ssi.gouv.fr/eidas/TL-FR.xml");

// A certificate source which contains the signing certificate(s) for the
// current trusted list
tlSource.setCertificateSource(getSigningCertificatesForFrenchTL());

// Optional : predicate to filter trust services which are/were granted or
// equivalent (pre/post eIDAS).
// Input : implementation of TrustServicePredicate interface.
// Default : none (select all)
tlSource.setTrustServicePredicate(new GrantedTrustService());

// Optional : predicate to filter the trust service providers
// Input : implementation of TrustServiceProviderPredicate interface.
// Default : none (select all)
tlSource.setTrustServiceProviderPredicate(new CryptologOnlyTrustServiceProvider());

//instance of CertificateSource where all trusted certificates and their properties
//(service type,...) are stored.
tlValidationJob.setTrustedListSources(tlSource);
```

List Of Trusted Lists Source (LOTLSouce)

A similar configuration is possible for Lists Of Trusted Lists (LOTL). That requires an URL and the possible LOTL signers. Some other parameters are possible. By default, all listed trusted lists are loaded.

LOTLSouce configuration

```
LOTLSouce lotlSource = new LOTLSouce();

// Mandatory : The url where the LOTL needs to be downloaded
lotlSource.setUrl("https://ec.europa.eu/tools/lotl/eu-lotl.xml");

// A certificate source which contains the signing certificate(s) for the
// current list of trusted lists
lotlSource.setCertificateSource(getSigningCertificatesForEuropeanLOTL());

// true or false for the pivot support. Default = false
// More information :
// https://ec.europa.eu/tools/lotl/pivot-lotl-explanation.html
lotlSource.setPivotSupport(true);

// Optional : the predicate which allows to find the LOTL definition in the LOTL
// Input : implementation of Predicate<OtherTSLPointerType> interface (e.g.
```

```

OtherTSLPointerPredicate)
// Default : European configuration
lotlSource.setLotlPredicate(new EULOTLOtherTSLPointer().and(new
XMLOtherTSLPointer()));

// Optional : the predicate which allows to find and/or filter the TL
// definitions in the LOTL
// Input : implementation of Predicate<OtherTSLPointerType> interface (e.g.
OtherTSLPointerPredicate)
// Default : all found trusted lists in the European LOTL
lotlSource.setTlPredicate(new EUTLOtherTSLPointer().and(new XMLOtherTSLPointer()));

// Optional : a predicate which allows to find back the signing certificates for
// the current LOTL
// Input : implementation of LOTLSigningCertificatesAnnouncementSchemeInformationURI
interface.
// Default : not defined
//
// OfficialJournalSchemeInformationURI allows to specify the Official Journal
// URL where are published the signing certificates
lotlSource.setSigningCertificatesAnnouncementPredicate(
    new OfficialJournalSchemeInformationURI("https://eur-lex.europa.eu/legal-
content/EN/TXT/?uri=uriserv:OJ.C_.2019.276.01.0001.01.ENG"));

// Optional : predicate to filter trust services which are/were granted or
// equivalent (pre/post eIDAS). This parameter is applied on the related trusted
// lists
// Input : implementation of TrustServicePredicate interface.
// Default : none (select all)
lotlSource.setTrustServicePredicate(new GrantedTrustService());

// Optional : predicate to filter the trust service providers. This parameter is
// applied on the related trusted lists
// Input : implementation of TrustServiceProviderPredicate interface.
// Default : none (select all)
lotlSource.setTrustServiceProviderPredicate(new CryptologOnlyTrustServiceProvider());

tlValidationJob.setListOfTrustedListSources(lotlSource);

```

DSSFileLoader

The FileCacheDataLoader is used to download the trusted list contents on the file-system. Two different configurations are needed. Both of them share the same folder :

- offline refresh : disabled download from Internet and unlimited cache expiration
- online refresh : enabled download from Internet and limited cache expiration

Offline and Online refresh configuration

```
public DSSFileLoader offlineLoader() {
    FileCacheDataLoader offlineFileLoader = new FileCacheDataLoader();
    offlineFileLoader.setCacheExpirationTime(Long.MAX_VALUE);
    offlineFileLoader.setDataLoader(new IgnoreDataLoader()); // do not download from
Internet
    offlineFileLoader.setFileCacheDirectory(tlCacheDirectory());
    return offlineFileLoader;
}

public DSSFileLoader onlineLoader() {
    FileCacheDataLoader onlineFileLoader = new FileCacheDataLoader();
    onlineFileLoader.setCacheExpirationTime(0);
    onlineFileLoader.setDataLoader(dataLoader()); // instance of DataLoader which can
access to Internet (proxy,...)
    onlineFileLoader.setFileCacheDirectory(tlCacheDirectory());
    return onlineFileLoader;
}
```

The SynchronizationStrategy

The SynchronizationStrategy defines which are the trusted lists or list of trusted lists to be synchronized. By default, DSS synchronizes all of them. DSS don't reject any expired / invalid /... trusted lists. The content is trusted and a warning is added in a signature / certificate validation.

The strategy is configurable via the interface SynchronizationStrategy :

Example of a custom SynchronizationStrategy

```
public SynchronizationStrategy allValidTrustedListsStrategy() {

    return new SynchronizationStrategy() {

        @Override
        public boolean canBeSynchronized(TLInfo trustedList) {
            return trustedList.getValidationCacheInfo().isValid();
        }

        @Override
        public boolean canBeSynchronized(LOTLInfo listOfTrustedList) {
            return listOfTrustedList.getValidationCacheInfo().isValid();
        }

    };
}
```

DSS provides two implementations : `ExpirationAndSignatureCheckStrategy` and `AcceptAllStrategy`

(default).

The CacheCleaner

The CacheCleaner specifies how DSS clear the cache in case of expired URL,... 2 options are available : memory and file-system.

CacheCleaner Configuration

```
public CacheCleaner cacheCleaner() {  
    CacheCleaner cacheCleaner = new CacheCleaner();  
  
    cacheCleaner.setCleanMemory(true); // free the space in memory  
  
    cacheCleaner.setCleanFileSystem(true); // remove the stored file(s) on the file-  
    // system  
  
    // if the file-system cleaner is enabled, inject the configured loader from the  
    // online or offline refresh data loader.  
    cacheCleaner.setDSSFileLoader(offlineLoader());  
  
    return cacheCleaner;  
}
```

Alerting from TL Loading

DSS allows running of custom alerts in some situations (eg : invalid TL signature, LOTL location change,...). Alert works with two concepts : detection and alert handler. After the download/parsing/validation and before the synchronization, the results are tested to detect events and launch alert(s).

Examples of Alerting

```
TLValidationJob job = new TLValidationJob();  
// ...  
  
// Add a log message in case of invalid signatures  
TLAlert tlBrokenSignatureAlert = new TLAlert(new TLSignatureErrorDetection(), new  
LogTLSignatureErrorAlertHandler());  
  
// Send an email in case of new Official Journal detected  
AlertHandler<LOTLInfo> mailSender = new AlertHandler<LOTLInfo>() {  
  
    @Override  
    public void process(LOTLInfo currentInfo) {  
        String newOJUrl =  
currentInfo.getParsingCacheInfo().getSigningCertificateAnnouncementUrl();  
        // code to send an email  
        SampleUtils.sendEmail(newOJUrl);  
    }  
}
```

```

};

// The europeanLOTLSouce is configured with an
// OfficialJournalSchemeInformationURI
LOTLAlert officialJournalDesynchronizationAlert = new LOTLAlert(new
OJUrlChangeDetection(europeanLOTLSouce()), mailSender);

// Update a database in case of LOTL location change
AlertHandler<LOTLInfo> databaseUpgrader = new AlertHandler<LOTLInfo>() {

    @Override
    public void process(LOTLInfo currentInfo) {
        String newLOTLUrl = null;

        String currentLOTLUrl = currentInfo.getUrl();
        List<PivotInfo> pivots = currentInfo.getPivotInfos();
        for (PivotInfo pivot : pivots) {
            if (!Utils.areStringsEqual(currentLOTLUrl, pivot.getLOTLLocation())) {
                newLOTLUrl = pivot.getLOTLLocation();
                break;
            }
        }
    }

    // code to update a database
    SampleUtils.updateDatabase(newLOTLUrl);
}

};

LOTLAlert lotlLocationChangeAlert = new LOTLAlert(new
LOTLLocationChangeDetection(europeanLOTLSouce()), databaseUpgrader);

// add all alerts on the job
job.setLOTLAlerts(Arrays.asList(officialJournalDesynchronizationAlert,
lotlLocationChangeAlert));
job.setTLAlerts(Arrays.asList(tlBrokenSignatureAlert));

```

LOTL/TL filter predicates

TSL predicates provide an option to filter the extracted TSL Pointers from LOTL or TL sources, allowing a customization of a trusted certificates and trusted services loading.

The following predicates are provided within the framework:

- **EULOTLOtherTSLPointer** - filters the EU LOTL pointer;
- **EUTLOtherTSLPointer** - filters the EU TL pointers;
- **MimetypeOtherTSLPointer** - filters TL pointers by a MimeType (e.g. XML only);

- **XMLOtherTSLPointer** - filters XML TL pointers;
- **PDFOtherTSLPointer** - filters PDF TL pointers;
- **SchemeTerritoryOtherTSLPointer** - filters TL pointers with a specific scheme territory (i.e. filter by country).

Examples of TSL Loading Predicates configuration

```
// the predicates filter TSL pointers to XML documents with
// "http://uri.etsi.org/TrstSvc/TrustedList/TSLType/EUlistofthelists" type
lotlSource.setLotlPredicate(new EULOTLOtherTSLPointer().and(new
XMLOtherTSLPointer()));

// the predicates filter only TSL pointers with scheme territories "DE" (Germany) and
// "RO" (Romania)
// to XML documents with "http://uri.etsi.org/TrstSvc/TrustedList/TSLType/EUgeneric"
// type
lotlSource.setTlPredicate(new SchemeTerritoryOtherTSLPointer(Arrays.asList("DE", "RO"))
.and(new EULOTLOtherTSLPointer()).and(new XMLOtherTSLPointer()));
```

Executor Service

An Executor Service parameter allows you to customize a way of the program execution on your Java machine, by configuring a number of possible threads to be running, await time and so on.

Executor Service

```
// Allows configuration of the execution process
// Default : Executors.newCachedThreadPool() is used
tlValidationJob.setExecutorService(Executors.newSingleThreadExecutor());
```

Complete configuration for the European LOTL

Below, you can find a complete configuration for the European List Of Trusted Lists. The URLs need to be externalized.

European LOTL Configuration

```
// Should be externalized
private static final String LOTL_URL = "https://ec.europa.eu/tools/lotl/eu-
lotl.xml";
private static final String OJ_URL = "https://eur-lex.europa.eu/legal-
content/EN/TXT/?uri=uriserv:OJ.C_.2019.276.01.0001.01.ENG";

@Test
public void test() {
    CommonCertificateVerifier commonCertificateVerifier = new
CommonCertificateVerifier();
    TLValidationJob job = job();
```

```

        TrustedListsCertificateSource trustedListsCertificateSource = new
TrustedListsCertificateSource();
        job.setTrustedListCertificateSource(trustedListsCertificateSource);
        job.onlineRefresh();

commonCertificateVerifier.setTrustedCertSources(trustedListsCertificateSource);
        commonCertificateVerifier.setCrlSource(new OnlineCRLSource());
        commonCertificateVerifier.setOcspSource(new OnlineOCSPSource());
        commonCertificateVerifier.setAIAsource(new DefaultAIAsource());

        SignedDocumentValidator validator = SignedDocumentValidator.fromDocument(
                new FileDocument("src/test/resources/signature-
pool/signedXmlXadesB.xml"));
        validator.setCertificateVerifier(commonCertificateVerifier);

        validator.validateDocument();
    }

public TLValidationJob job() {
    TLValidationJob job = new TLValidationJob();
    job.setOfflineDataLoader(offlineLoader());
    job.setOnlineDataLoader(onlineLoader());
    job.setTrustedListCertificateSource(trustedCertificateSource());
    job.setSynchronizationStrategy(new AcceptAllStrategy());
    job.setCacheCleaner(cacheCleaner());

    LOTLSource europeanLOTL = europeanLOTL();
    job.setListOfTrustedListSources(europeanLOTL);

    job.setLOTLAlerts(Arrays.asList(ojUrlAlert(europeanLOTL),
lotlLocationAlert(europeanLOTL)));
    job.setTlAlerts(Arrays.asList(tlSigningAlert(), tlExpirationDetection()));

    return job;
}

public TrustedListsCertificateSource trustedCertificateSource() {
    return new TrustedListsCertificateSource();
}

public LOTLSource europeanLOTL() {
    LOTLSource lotlSource = new LOTLSource();
    lotlSource.setUrl(LOTL_URL);
//    lotlSource.setCertificateSource(officialJournalContentKeyStore());
    lotlSource.setCertificateSource(new CommonCertificateSource());
    lotlSource.setSigningCertificatesAnnouncementPredicate(new
OfficialJournalSchemeInformationURI(OJ_URL));
    lotlSource.setPivotSupport(true);
    return lotlSource;
}

```

```

public CertificateSource officialJournalContentKeyStore() {
    try {
        return new KeyStoreCertificateSource(new
File("src/main/resources/keystore.p12"), "PKCS12", "dss-password");
    } catch (IOException e) {
        throw new DSSEException("Unable to load the keystore", e);
    }
}

public DSSFileLoader offlineLoader() {
    FileCacheDataLoader offlineFileLoader = new FileCacheDataLoader();
    offlineFileLoader.setCacheExpirationTime(Long.MAX_VALUE);
    offlineFileLoader.setDataLoader(new IgnoreDataLoader());
    offlineFileLoader.setFileCacheDirectory(tlCacheDirectory());
    return offlineFileLoader;
}

public DSSFileLoader onlineLoader() {
    FileCacheDataLoader onlineFileLoader = new FileCacheDataLoader();
    onlineFileLoader.setCacheExpirationTime(0);
    onlineFileLoader.setDataLoader(dataLoader());
    onlineFileLoader.setFileCacheDirectory(tlCacheDirectory());
    return onlineFileLoader;
}

public File tlCacheDirectory() {
    File rootFolder = new File(System.getProperty("java.io.tmpdir"));
    File tslCache = new File(rootFolder, "dss-tsl-loader");
    if (tslCache.mkdirs()) {
        LOG.info("TL Cache folder : {}", tslCache.getAbsolutePath());
    }
    return tslCache;
}

public CommonsDataLoader dataLoader() {
    return new CommonsDataLoader();
}

public CacheCleaner cacheCleaner() {
    CacheCleaner cacheCleaner = new CacheCleaner();
    cacheCleaner.setCleanMemory(true);
    cacheCleaner.setCleanFileSystem(true);
    cacheCleaner.setDSSFileLoader(offlineLoader());
    return cacheCleaner;
}

// Optionally : alerting.
// Recommended detections : OJUrlChangeDetection + LOTLLocationChangeDetection

public TLAlert tlSigningAlert() {
    TLSignatureErrorDetection signingDetection = new TLSignatureErrorDetection();

```

```

    LogTLSignatureErrorHandler handler = new
    LogTLSignatureErrorHandler();
        return new TLAlert(signingDetection, handler);
    }

    public TLAlert tlExpirationDetection() {
        TLExpirationDetection expirationDetection = new TLExpirationDetection();
        LogTLExpirationErrorHandler handler = new LogTLExpirationErrorHandler();
        return new TLAlert(expirationDetection, handler);
    }

    public LOTLAlert ojUrlAlert(LOTLSource source) {
        OJUrlChangeDetection ojUrlDetection = new OJUrlChangeDetection(source);
        LogOJUrlChangeAlertHandler handler = new LogOJUrlChangeAlertHandler();
        return new LOTLAlert(ojUrlDetection, handler);
    }

    public LOTLAlert lotlLocationAlert(LOTLSource source) {
        LOTLLocationChangeDetection lotlLocationDetection = new
        LOTLLocationChangeDetection(source);
        LogLOTLLocationChangeAlertHandler handler = new
        LogLOTLLocationChangeAlertHandler();
        return new LOTLAlert(lotlLocationDetection, handler);
    }
}

```

The TL / LOTL refresh

The TL / LOTL loading in DSS works as below :

- Download / parse / validate all LOTLSources from the configuration with/without pivot support (multi-threaded)
- Analyze introduced changes and expire cache entries (new TL URLs, new signing certificates for a TL,...)
- Create TLSources from the retrieved LOTLs
- Combine these TLSources with independent TLSources (from the configuration)
- Download / parse / validate all TLs (multi-threaded)
- If alerts are configured, test if an alert needs to be launched
- If the debug is enabled, print in the log the cache status
- Synchronize the TrustedListCertificateSource
- If the cache cleaner is configured, execute it
- If the debug is enabled, print in the log the cache status

The refresh can be called with the offline or the online loader and run exactly the same code

How to refresh the Trusted List(s) and Lists of Trusted Lists

```
TLValidationJob validationJob = new TLValidationJob();

// call with the Offline Loader (application initialization)
validationJob.offlineRefresh();

// call with the Online Loader (callable every day/hour in a cron)
validationJob.onlineRefresh();
```

Java Keystore Management

Generally (like in case of European LOTL) DSS downloads Trusted Lists by using the SSL protocol (for resources using HTTPS extension), that requires to have a certificate of a remote source in the Java trust store. The certificates have their own validity period and can expire. If a certificated is expired, it will be replaced on a server by a new one in order to support a secure SSL connection. The easiest way to know if your Java trust store is outdated and new certificates need to be added is to check your logs during a TLValidationJob update :

```
ERROR 14052 --- [pool-2-thread-30] e.e.e.dss.tsl.runnable.AbstractAnalysis : Unable
to process GET call for url [https://sr.riik.ee/tsl/estonian-tsl.xml]. Reason : [PKIX
path building failed: sun.security.provider.certpath.SunCertPathBuilderException:
unable to find valid certification path to requested target]
```

The **SunCertPathBuilderException** means that the certificate established the secure connection is not trusted by your Java Virtual Machine. In order to add the certificate to the trust store, you need to do the following steps (the example is based on Windows OS and Google Chrome browser):

1. Open the failed URL in your browser. In our case it will be 'https://sr.riik.ee/tsl/estonian-tsl.xml' obtained from the logs.
2. Click on a lock icon next to the URL in the tab you just opened. It will open a window about the current connection status.
3. Click on 'Certificate' button to open the Certificate window.
4. Go to 'Details' tab and choose 'Copy to File...'.
5. Process the 'Certificate Export Wizard', by saving the certificate in one of '.CER' formats. Store the file in your file system. For us it will create a file 'ee.cer'.
6. Run 'Command Prompt' with administrator permissions (right click → 'Run As Administrator').
7. Execute the following line (ensure that 'keytool' is installed) :

Certificate import

```
keytool -import -alias newCert -file pathToCert\ee.cer -keystore
pathToJavaDirectory\lib\security\cacerts -storepass changeit
```

The default password for a Java keystore is "changeit". Ensure that you have a default

configuration, or use another password you have configured.



In order to apply changes, the application using Java must be rebooted.

After these steps the `TLValidationJob` will successfully download the target Trusted List (i.e. Estonian in our example).



This described algorithm is not only one available solution, if you have difficulties with this, you can search in the Internet for another working for you solution.

TLValidationJobSummary

The class `TLValidationJobSummary` contains all processed data about the download (time, error,...), the parsing (extracted information, parsing error,...) and the signature validation (signing certificate, signing time,...).

How to retrieve the information about the `TLValidationJob` process

```
TrustedListsCertificateSource trustedListCertificateSource = new
TrustedListsCertificateSource();

TLValidationJob job = new TLValidationJob();
job.setTrustedListCertificateSource(trustedListCertificateSource);

// ... config & refresh ...

// A cache content summary can be computed on request
TLValidationJobSummary summary = job.getSummary();

// All information about processed LOTLSources
List<LOTLInfo> lotlInfos = summary.getLOTLInfos();
LOTLInfo lotlInfo = lotlInfos.get(0);
// All data about the download (last occurrence, cache status, error,...)
DownloadInfoRecord downloadCacheInfo = lotlInfo.getDownloadCacheInfo();

// All data about the parsing (date, extracted data, cache status,...)
ParsingInfoRecord parsingCacheInfo = lotlInfo.getParsingCacheInfo();

// All data about the signature validation (signing certificate, validation
// result, cache status,...)
ValidationInfoRecord validationCacheInfo = lotlInfo.getValidationCacheInfo();

// All information about processed TLSources (which are not linked to a
// LOTLSource)
List<TLInfo> otherTLInfos = summary.getOtherTLInfos();

// or the last update can be collected from the TrustedListsCertificateSource
TLValidationJobSummary lastSynchronizedSummary =
trustedListCertificateSource.getSummary();
```

TSP Sources

The Time Stamp Authority by creating time-stamp tokens provides independent and irrefutable proof of time for business transactions, e-documents and digital signatures. The TSA must comply with the IETF RFC 3161 specifications (cf. [\[R08\]](#)). A time-stamp is obtained by sending the digest value of the given data and digest algorithm to the Time Stamp Authority. The returned time-stamp is a signed data that contains the digest value, the identity of the TSA, and the time of stamping. This proves that the given data existed before the time of stamping. The DSS framework proposes TSPSource interface to implement the communication with TSA. The class OnlineTSPSource is the default implementation of TSP using HTTP(S) communication layer. The following bit of Java code illustrates how you might use this class:

OnlineTSPSource usage

```
final String tspServer = "http://dss.nowina.lu/pki-factory/tsa/good-tsa";
OnlineTSPSource tspSource = new OnlineTSPSource(tspServer);
tspSource.setDataLoader(new TimestampDataLoader()); // uses the specific content-type

final DigestAlgorithm digestAlgorithm = DigestAlgorithm.SHA256;
final byte[] toDigest = "Hello world".getBytes("UTF-8");
final byte[] digestValue = DSSUtils.digest(digestAlgorithm, toDigest);
final TimestampBinary tsBinary = tspSource.getTimeStampResponse(digestAlgorithm,
digestValue);

LOG.info(DSSUtils.toHex(tsBinary.getBytes()));
```

Time-stamp policy

A time-stamp policy is a "named set of rules that indicates the applicability of a time-stamp token to a particular community and/or class of application with common security requirements". A TSA may define its own policy which enhances the policy defined in RFC 3628. Such a policy shall incorporate or further constrain the requirements identified in RFC 3628. A time-stamp policy may be defined by the user of times-stamp services.

Composite TSP Source

Sometimes, timestamping servers may encounter interruptions (restart,...). To avoid failing signature extension, DSS allows a user to configure several TSP Sources. DSS will try source by source until getting an usable timestamp token.

```
// Create a map with several TSPSources
TimestampDataLoader timestampDataLoader = new TimestampDataLoader(); // uses the
specific content-type

OnlineTSPSource tsa1 = new OnlineTSPSource("http://dss.nowina.lu/pki-factory/tsa/ee-
good-tsa");
tsa1.setDataLoader(timestampDataLoader);
OnlineTSPSource tsa2 = new OnlineTSPSource("http://dss.nowina.lu/pki-factory/tsa/good-
tsa");
tsa2.setDataLoader(timestampDataLoader);

Map<String, TSPSource> tspSources = new HashMap<>();
tspSources.put("TSA1", tsa1);
tspSources.put("TSA2", tsa2);

// Instantiate a new CompositeTSPSource and set the different sources
CompositeTSPSource tspSource = new CompositeTSPSource();
tspSource.setTspSources(tspSources);

final DigestAlgorithm digestAlgorithm = DigestAlgorithm.SHA256;
final byte[] toDigest = "Hello world".getBytes("UTF-8");
final byte[] digestValue = DSSUtils.digest(digestAlgorithm, toDigest);

// DSS will request the tsp sources (one by one) until getting a valid token.
// If none of them succeed, a DSSEException is thrown.
final TimestampBinary tsBinary = tspSource.getTimeStampResponse(digestAlgorithm,
digestValue);

LOG.info(DSSUtils.toHex(tsBinary.getBytes()));
```

Supported algorithms

DSS supports several signature algorithms (combination of an encryption algorithm and a digest algorithm). Below, you can find the supported combinations. The support of the algorithms depends on the registered OID (ASN1) or URI (XML).

In the next table, XAdES also applies to ASiC with embedded XAdES signatures and CAdES also concerns PAdES and ASiC with embedded CAdES signatures.



SmartCards/HSMs don't allow signing with all digest algorithms. Please refer to your SmartCard/HSM provider.

Table 7. Supported algorithms

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512	SHA3-224	SHA3-256	SHA3-384	SHA3-512	MD2	MD5	RIPE MD160
RSA												
XAdES	✓	✓	✓	✓	✓						✓	✓
CAdES	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
JAdES			✓	✓	✓							
RSA-PSS												
XAdES	✓	✓	✓	✓	✓	✓	✓	✓	✓			
CAdES	✓	✓	✓	✓	✓	✓	✓	✓	✓			
JAdES			✓	✓	✓							
ECDSA												
XAdES	✓	✓	✓	✓	✓							✓
CAdES	✓	✓	✓	✓	✓	✓	✓	✓	✓			
JAdES			✓	✓	✓							
Ed25519												
XAdES												
CAdES					✓							
DSA												
XAdES	✓		✓									
CAdES	✓	✓	✓	✓	✓	✓	✓	✓	✓			
JAdES												
HMAC												
XAdES	✓	✓	✓	✓	✓							✓
CAdES	✓	✓	✓	✓	✓	✓	✓	✓	✓			
JAdES			✓	✓	✓							

Implementation management with ServiceLoader

DSS incorporates modules that are loaded in the run time based on the chosen configuration and the input data via a [ServiceLoader](#). This provides a flexibility for an end-user to work only with selected modules and a possibility to expand DSS with custom implementations.

In order to provide a chosen implementation(s) to ServiceLoader, a file listing all the desired implementations should be created in the resource directory [META-INF/services](#) with a name matching the implemented interface. When merging sources (e.g. creating a Fat JAR module), the

files can be lost/overwritten, and should be configured manually (all the required implementations shall be listed).



If a DSS module(s) implementing a required interface(s) is added to your project's dependency list, the implementation shall be loaded automatically.

The following modules are provided with independent implementations:

- [DSS Utils](#);
- [DSS CRL Parser](#);
- [DSS PAdES](#).

Additionally, DSS is able to choose the required implementation for the following interfaces:

- [DocumentValidationFactory](#) - checks a provided signed file's format and loads a relevant validator;
- [SignaturePolicyValidator](#) - checks a signature policy file and loads a relevant validator to be able to process the detected format.



If no appropriate available implementation is found, an exception will be thrown.

Document Validation Factory

This factory is used to create a required instance of a [DocumentValidator](#) based on the provided file's format (signature or timestamp). An implementation shall process a file format check and load the related [SignedDocumentValidator](#) implementation to be used for the file's validation.

The following implementations are present in DSS:

- CMSDocumentValidatorFactory : loads CMSDocumentValidator, used for a CAdES validation (delivered in dss-cades module);
- XMLDocumentValidatorFactory : loads XMLDocumentValidator, used for a XAdES validation (delivered in dss-xades module);
- PDFDocumentValidatorFactory : loads PDFDocumentValidator, used for a PAdES validation (delivered in dss-pades module);
- JAdESDocumentValidatorFactory : loads JWSCompactDocumentValidator or JWSSerializationDocumentValidator, depending on provided JSON signature type (delivered in dss-jades module);
- ASiCContainerWithCAdESValidatorFactory : loads ASiCContainerWithCAdESValidator (delivered in dss-asic-cades module);
- ASiCContainerWithXAdESValidatorFactory : loads ASiCContainerWithXAdESValidator (delivered in dss-asic-xades module);
- DetachedTimestampValidatorFactory : loads DetachedTimestampValidator, for an independent timestamp validation (delivered in dss-document module).

Signature Policy Validator

This interface is used to validate a signature policy reference extracted from a signature. The following implementations are provided:

- `BasicASNSignaturePolicyValidator` : validates policy files, which are based on ETSI TR 102 272;
- `XMLSignaturePolicyValidator` : validates XML signature policies supporting transformations;
- `NonASN1SignaturePolicyValidator` : validates a policy by digest computed on an original file's content;
- `ZeroHashSignaturePolicyValidator` : validates a policy if "zero hash" value is defined in a signature (see [\[R02\]](#));
- `EmptySignaturePolicyValidator` : is proceeded if a policy file is not found or not accessible.

Multi-threading

DSS can be used in multi-threaded environments but some points need to be considered like resources sharing and caching. All operations are stateless and this fact requires to be maintained. Some resources can be shared, others are proper to an operation.

For each provided operation, DSS requires a `CertificateVerifier` object. This object is responsible to provide certificates and accesses to external resources (AIA, CRL, OCSP,...). At the beginning of all operation, `CertificateSources` and `RevocationSources` are created for each signature / timestamp / revocation data. Extracted information are combined with the configured sources in the `CertificateVerifier`. For these reasons, integrators need to be careful about the `CertificateVerifier` configuration.

Resource sharing

The trusted certificates can be shared between multiple threads because these certificates are static. This means they don't require more analysis. Their status won't evolve. For these certificates, DSS doesn't need to collect issuer certificate and/or their revocation data.

In opposition, the adjunct certificates cannot be shared. These certificates concern a specific signature/validation operation. This parameter is used to provide missing certificate(s). When DSS is unable to build the complete certificate path with the provided certificates (as signature parameters or embedded within a signature), it is possible to inject not present certificates. These certificates are not necessarily trusted and may require future "modifications" like revocation data collection,...

Caching

In case of multi-threading usage, we strongly recommend caching of external resources. All external resources can be cached (AIA, CRL, OCSP) to improve performances and to avoid requesting too much time the same resources. `FileCacheDataLoader` and `JdbcCacheCRLSource` can help you in this way.

XML Securities

Since DSS 5.7, the framework allows custom configuration of XML-related modules for enabling/disabling of XML securities (e.g. in order to use Xalan or Xerces).



We strongly do not recommend disabling of security features and usage of deprecated dependencies. Be aware: the feature is designed only for experienced users, and all changes made in the module are at your own risk.

The configuration is available for the following classes:

- `javax.xml.parsers.DocumentBuilderFactory` with a `DocumentBuilderFactoryBuilder` - builds a DOM document object from the obtained XML file and creates a new Document;
- `javax.xml.transform.TransformerFactory` with a `TransformerFactoryBuilder` - loads XML templates and builds DOM objects;
- `javax.xml.validation.SchemaFactory` with a `SchemaFactoryBuilder` - loads XML Schema;
- `javax.xml.validation.Validator` with a `ValidatorConfigurator` - configures a validator to validate an XML document against an XML Schema.

All the classes can be configured with the following methods (example for `TransformerFactory`):

XMLSecurities configuration

```
// Obtain a singleton instance of {@link XmlDefinerUtils}
XmlDefinerUtils xmlDefinerUtils = XmlDefinerUtils.getInstance();

// returns a predefined {@link TransformerFactoryBuilder} with all securities in place
TransformerFactoryBuilder transformerBuilder =
TransformerFactoryBuilder.getSecureTransformerBuilder();

// sets an alert in case of exception on feature/attribute setting
transformerBuilder.setSecurityExceptionAlert(new LogOnStatusAlert(Level.WARN));

// allows to enable a feature
transformerBuilder.enableFeature(XMLConstants.FEATURE_SECURE_PROCESSING);

// allows to disable a feature
transformerBuilder.disableFeature("FEATURE_TO_DISABLE");

// allows to set an attribute with a value
transformerBuilder.setAttribute(XMLConstants.ACCESS_EXTERNAL_DTD, "");

// sets the transformer (will be applied for all calls)
xmlDefinerUtils.setTransformerFactoryBuilder(transformerBuilder);
```

The `javax.xml.parsers.DocumentBuilderFactory`, that allows XML files parsing and creation of DOM `Document` object, can be configured with the following methods:



Since DSS 5.9 the configuration of `javax.xml.parsers.DocumentBuilderFactory` has been moved from `DomUtils` to a new singleton class `DocumentBuilderFactoryBuilder`.

DocumentBuilderFactory configuration

```
// returns a configured secure instance of {@link DocumentBuilderFactory}
DocumentBuilderFactoryBuilder documentBuilderFactoryBuilder =
DocumentBuilderFactoryBuilder.getSecureDocumentBuilderFactoryBuilder();

// allows enabling of a feature
documentBuilderFactoryBuilder.enableFeature("http://xml.org/sax/features/external-
general-entities");

// allows disabling of a feature
documentBuilderFactoryBuilder.disableFeature("http://apache.org/xml/features/nonvalida-
ting/load-external-dtd");

// allows to set an attribute
documentBuilderFactoryBuilder.setAttribute(XMLConstants.ACCESS_EXTERNAL_DTD, "");

// sets the DocumentBuilderFactoryBuilder (will be applied for all calls)
xmlDefinerUtils.setDocumentBuilderFactoryBuilder(documentBuilderFactoryBuilder);
```

The class `XmlDefinerUtils` is a singleton, therefore all changes performed on the instance will have an impact to all calls of the related methods.

JAXB modules

Since the version 5.5, DSS provides the following JAXB modules with a harmonized structure :

- `dss-policy-jaxb` - defines validation policy JAXB model
- `dss-diagnostic-jaxb` - defines Diagnostic Data JAXB model
- `dss-detailed-report-jaxb` - defines Detailed Report JAXB model
- `dss-simple-report-jaxb` - defines Simple Report JAXB model
- `dss-simple-certificate-report-jaxb` - defines Certificate Simple Report JAXB model

All modules share the same logic and have the following structure (where *** is a model name):

```
dss-***-jaxb
  src/main/java
    eu.europa.esig.dss.***
      • ***.java - wrapper(s) which eases the JAXB manipulation
      • ...
      • ***Facade.java - class which allows marshalling/unmarshalling of jaxb objects,
        generation of HTML/PDF content, etc.
```

- *****XmlDefiner.java** - class which contains the model definition (XSD, XSLT references, ObjectFactory)
- **jaxb** - generated on compile time
 - **Xml***.java** - JAXB model
 - ...

src/main/resources

xsd

- *****.xsd** - XML Schema (XSD) for the Detailed Report model
- **binding.xml** - XJC instructions to generate the JAXB model from the XSD

xslt

- **html**
 - *****.xslt** - XML Stylesheet for the HTML generation
- **pdf**
 - *****.xslt** - XML Stylesheet for the PDF generation

In the main classes, a **Facade** is present to quickly operate with the JAXB objects (eg: marshall, unmarshall, generate the HTML/PDF, validate the XML structure,...).

DetailedReportFacade usage

```
Reports completeReports = documentValidator.validateDocument();

DetailedReportFacade detailedReportFacade = DetailedReportFacade.newFacade();

// Transforms the JAXB object to String (xml content)
String marshalledDetailedReport =
detailedReportFacade.marshall(completeReports.getDetailedReportJaxb());

// Transforms the String (xml content) to a JAXB Object
XmlDetailedReport xmlDetailedReport =
detailedReportFacade.unmarshall(marshalledDetailedReport);

// Generates the HTML content for the given Detailed Report (compatible with
// BootStrap)
// Similar method is available for PDF generation (requires Apache FOP)
String htmlDetailedReport =
detailedReportFacade.generateHtmlReport(completeReports.getDetailedReportJaxb());
```

A **XmlDefiner** is also available with the access to the embedded XML Schemas (XSD), the XML Stylesheets (XSLT) to be able to generate the HTML or the PDF content (for DSS specific JAXB) and the JAXB Object Factory.

DetailedReportXmlDefiner usage

```
// The JAXB Object Factory
ObjectFactory objectFactory = DetailedReportXmlDefiner.OBJECT_FACTORY;

// The JAXBContext (cached)
JAXBContext jaxbContext = DetailedReportXmlDefiner.getJAXBContext();

// The XML Schema to validate a XML content (cached)
Schema schema = DetailedReportXmlDefiner.getSchema();

// The Templates object with the loaded XML Stylesheet to generate the HTML
// content from the JAXB Object (cached)
Templates bootstrap4Templates = DetailedReportXmlDefiner.getHtmlBootstrap4Templates();

// The Templates object with the loaded XML Stylesheet to generate the PDF
// content from the JAXB Object (cached)
Templates pdfTemplates = DetailedReportXmlDefiner.getPdfTemplates();
```

Report stylesheets

The report modules (namely: [dss-simple-report-jaxb](#), [dss-simple-certificate-report-jaxb](#) and [dss-detailed-report-jaxb](#)) contain two XSLT style sheets each for final reports generation:

- Bootstrap 4 XSLT for HTML report;
- PDF XSLT for PDF report.



Since DSS 5.9 only Bootstrap 4 XSLT is provided within the framework for HTML report generation.

In order to generate a report with a selected style sheet you need to call a relevant method in a Facade class (see classes definition above):

HTML report generation

```
String bootstrap4Report =
SimpleReportFacade.newFacade().generateHtmlReport(xmlSimpleReport);
```

Otherwise, in case if you need to customize the transformer, you can create a report by using an XmlDefiner:

```
try (Writer writer = new StringWriter()) {
    Transformer transformer =
SimpleCertificateReportXmlDefiner.getHtmlBootstrap4Templates().newTransformer();
    // specify custom parameters if needed
    transformer.transform(new StreamSource(new StringReader(simpleReport)), new
StreamResult(writer));
    String bootstrap4Report = writer.toString();
}
```

Alerts

Since DSS 5.7 the framework includes an extended possibility to execute custom processes in case of arbitrary defined events.

The **Alert** is a basic interface used to trigger a process on a passed object. DSS provides an **AbstractAlert** implementation of the interface with a clearly defined structure. The class must be instantiated with two attributes:

- **AlertDetector** - used to detect an event/state of the object and trigger a process;
- **AlertHandler** - defines a process to be executed on an object.

In its basic module, framework provides a few alerts based on a **Status**:

- **ExceptionOnStatusAlert** - throws an **AlertException** (RuntimeException) when the status reports an issue;
- **LogOnStatusAlert** - logs a message with the defined log level;
- **SilentOnStatusAlert** - ignores the reported issue and does nothing.

The usage of alerts is available in the following classes:

- **XML Securities** configurators from **dss-jaxb-parsers** module : **TransformerFactoryBuilder**, **SchemaFactoryBuilder**, **ValidatorConfigurator**;
- **CertificateVerifier configuration** - to handle the unexpected situation(s) in a custom way (introduced **AlertException** to re-throw exceptions);
- **TLValidationJob** - to process custom actions on change/state on loading of LOTL/TLS (see **LOTLAlert** and **TLAlert**).

Exceptions

This section provides an overview of runtime Exceptions which are being thrown by various modules of DSS framework.

The following Exceptions can be obtained by the upper level:

- **NullPointerException** is thrown when a mandatory parameter has not been provided by the end-user to the method/process, requiring the property;
- **IllegalArgumentException** is thrown when a parameters configuration is not valid for the called method or some parameters cannot be used together (e.g. on a signature creation);
- **IllegalInputException** is thrown when a provided input document is not valid for the requested process and/or the configuration of parameters is not applicable for the given document;
- **UnsupportedOperationException** is thrown when a method is not implemented or its usage with the requested parameters is not (yet) supported;
- **IllegalStateException** is thrown when the requested method cannot be performed at the current method (e.g. another method shall be executed before);
- **DSSException** is thrown in case of an error obtained during the internal DSS process (e.g. data conversion, CRL/OCSP parsing, etc.);
- **DSSExternalResourceException** is thrown if an error occurs during a remote source request (AIA, CRL, OCPS requests, etc.);
- **DSSRemoteServiceException** is thrown in case of a request/response error within **REST and SOAP Services**.

I18N (Internationalization)

Since DSS 5.6 a new module has been introduced allowing changing of a language for reports generated by DSS. The current version of the framework allows customization of text values only for a **DetailedReport**.

A target language of the report can be set with the following code:

Language customization

```
SignedDocumentValidator validator =
SignedDocumentValidator.fromDocument(signedDocument);
// A target Locale must be defined for the validator
validator.setLocale(Locale.FRENCH); // for French language
```

In case if no language is specified, the framework will use a default Locale obtained from OS on a running machine. If a requested language is not found, a default translation will be used.

As a default configuration DSS provides English translation.

In order to provide a custom translation, a new file must be created inside **src\main\resources** directory of your project with a name followed by one of the patterns:

dss-messages_XX.properties or **dss-messages_XX_YY.properties**, where:

- XX - an abbreviation of a target language;
- YY - a country code.

For example, for a French language a file with a name `dss-messages_fr.properties` need to be created, or `dss-messages_fr_FR.properties` to use it only in France local.

Additional features

Certificate validation

DSS offers the possibility to validate a certificate. For a given certificate, the framework builds a certificate path until a known trust anchor (trusted list, keystore,...), validates each found certificate (OCSP / CRL) and determines its European "qualification".

To determine the certificate qualification, DSS follows the draft standard ETSI TS 119 172-4 ([\[R10\]](#)). It analyses the certificate properties (QCStatements, Certificate Policies,...) and applies possible overrules from the related trusted list ("caught" qualifiers from a trust service). More information about qualifiers can be found in the standard ETSI TS 119 612 ([\[R11\]](#)).

DSS always computes the status at 2 different times : certificate issuance and signing/validation time. The certificate qualification can evolve in the time, its status is not immutable (eg: a trust service provider lost its granted status). The eIDAS regulation ([\[R12\]](#)) clearly defines these different times in the Article 32 and related Annex I.

```
// Firstly, we load the certificate to be validated
CertificateToken token = DSSUtils.loadCertificate(new
File("src/main/resources/keystore/ec.europa.eu.1.cer"));

// We need a certificate verifier and configure it (see specific chapter about the
// CertificateVerifier configuration)
CertificateVerifier cv = new CommonCertificateVerifier();

// We create an instance of the CertificateValidator with the certificate
CertificateValidator validator = CertificateValidator.fromCertificate(token);
validator.setCertificateVerifier(cv);

// Allows specifying which tokens need to be extracted in the diagnostic data
// (Base64).
// Default : NONE)
validator.setTokenExtractionStrategy(TokenExtractionStrategy.EXTRACT_CERTIFICATES_AND_
REVOCATION_DATA);

// We execute the validation
CertificateReports certificateReports = validator.validate();

// We have 3 reports
// The diagnostic data which contains all used and static data
DiagnosticData diagnosticData = certificateReports.getDiagnosticData();

// The detailed report which is the result of the process of the diagnostic data and
// the validation policy
DetailedReport detailedReport = certificateReports.getDetailedReport();

// The simple report is a summary of the detailed report or diagnostic data (more
// user-friendly)
SimpleCertificateReport simpleReport = certificateReports.getSimpleReport();
```

SSL Certificate validation (QWAC)

With DSS, that's also possible to validate SSL certificate against the EUMS TL and the ETSI TS 119 615 to determine if it is a Qualified certificate for WebSite Authentication (QWAC).

Validate a SSL certificate and retrieve its qualification level

```
// We firstly need an Internet Access. Additional configuration may be required
// (proxy,...)
CommonsDataLoader dataLoader = new CommonsDataLoader();

// We set an instance of TrustAllStrategy to rely on the Trusted Lists content
// instead of the JVM trust store.
dataLoader.setTrustStrategy(TrustAllStrategy.INSTANCE);

// Secondly, we create an instance of SSLCertificateLoader which is responsible
// of the SSL certificate(s) download.
SSLCertificateLoader sslCertificateLoader = new SSLCertificateLoader();
// We set the configured dataLoader
sslCertificateLoader.setCommonsDataLoader(dataLoader);

// Thirdly, we need to configure the CertificateVerifier
CertificateVerifier cv = new CommonCertificateVerifier();
cv.setTrustedCertSources(trustedListsCertificateSource); // configured trusted list
certificate source
cv.setAIAsource(aiaSource); // configured AIA Access
cv.setOcspSource(ocspSource); // configured OCSP Access
cv.setCrlSource(crlSource); // configured CRL Access

// We retrieve the SSL certificates for the given URL
List<CertificateToken> certificates =
sslCertificateLoader.getCertificates("https://www.microsec.hu");

CertificateToken sslCertificate = certificates.get(0);

// Add intermediate certificates as non trusted certificates (adjunct)
CertificateSource adjunctCertSource = new CommonCertificateSource();
for (CertificateToken certificateToken : certificates) {
    adjunctCertSource.addCertificate(certificateToken);
}
cv.setAdjunctCertSources(adjunctCertSource);

// Create an instance of CertificateValidator for the SSL Certificate with the
// CertificateVerifier
CertificateValidator validator = CertificateValidator.fromCertificate(sslCertificate);
validator.setCertificateVerifier(cv);

CertificateReports reports = validator.validate();
SimpleCertificateReport simpleReport = reports.getSimpleReport();
DetailedReport detailedReport = reports.getDetailedReport();
DiagnosticData diagnosticData = reports.getDiagnosticData();
```

Extract the signed data from a signature

DSS is able to retrieve the original data from a valid signature.

Retrieve original data from a signed document

```
// We have our signed document, we want to retrieve the original/signed data
DSSDocument signedDocument = new FileDocument("src/test/resources/signature-
pool/signedXmlXadesB.xml");

// We create an instance of DocumentValidator. DSS automatically selects the validator
depending of the
// signature file
SignedDocumentValidator documentValidator =
SignedDocumentValidator.fromDocument(signedDocument);

// We set a certificate verifier. It handles the certificate pool, allows to check the
certificate status,...
documentValidator.setCertificateVerifier(new CommonCertificateVerifier());

// We retrieve the found signatures
List<AdvancedSignature> signatures = documentValidator.getSignatures();

// We select the wanted signature (the first one in our current case)
AdvancedSignature advancedSignature = signatures.get(0);

// We call get original document with the related signature id (DSS unique ID)
List<DSSDocument> originalDocuments =
documentValidator.getOriginalDocuments(advancedSignature.getId());

// We can have one or more original documents depending of the signature (ASiC,
PDF,...)
DSSDocument original = originalDocuments.get(0);

// Save the extracted original document if needed
original.save(targetPath);
```

REST and SOAP Services

DSS offers REST and SOAP web services. Additionally, we also provide a SOAP-UI project and Postman samples in the [dss-cookbook](#) module.

The different webservices are :

- Signature webservices ([dss-signature-soap](#) / [dss-signature-rest](#)) and their clients : they expose methods to allow signing and extending or counter-signing a signature from a client.
- Server-signing webservice ([dss-server-signing-soap](#) / [dss-server-signing-rest](#)) and their clients : they expose method to retrieve keys from a server (PKCS#11, PKCS#12, HSM,...) and to sign the

digest on the server side.

- Signature validation webservices ([dss-validation-soap](#) / [dss-validation-rest](#)) and their clients : they expose methods to allow signature validation, with an optional detached file and an optional validation policy.
- Certificate validation webservices ([dss-certificate-validation-soap](#) / [dss-certificate-validation-rest](#)) and their clients : they expose methods to allow certificate validation, with an optional provided certificate chain and custom validation time.
- Timestamp webservices ([dss-timestamp-remote-soap](#) / [dss-timestamp-remote-rest](#)) and their clients : they expose methos to allow remote timestamp creation, by providing digest value to be timestamped and a digest algorithm, used for the digets calculation.

The data structure in webservices is similar in both REST and SOAP modules.

The documentation will covers the REST calls. All the REST services present in DSS are compliant with [OpenAPI Specification](#).

REST signature service

This service exposes 4 methods for one or more document(s) :

Rest signature service

```
// Initializes the rest client
RestDocumentSignatureService restClient = new RestDocumentSignatureServiceImpl();

// Defines RemoteSignatureParameters
RemoteSignatureParameters parameters = new RemoteSignatureParameters();
parameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_B);
parameters.setSigningCertificate(new
RemoteCertificate(privateKey.getCertificate().getEncoded()));
parameters.setSignaturePackaging(SignaturePackaging.ENVELOPING);
parameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// Initialize a RemoteDocument object to be signed
FileDocument fileToSign = new FileDocument(new File("src/test/resources/sample.pdf"));
RemoteDocument toSignDocument = new
RemoteDocument(Utils.toByteArray(fileToSign.openStream()), fileToSign.getName());

// computes the digest to be signed
ToBeSignedDTO dataToSign = restClient.getDataToSign(new
DataToSignOneDocumentDTO(toSignDocument, parameters));

// Creates a SignOneDocumentDTO
SignatureValue signatureValue =
signingToken.sign(DTOConverter.toToBeSigned(dataToSign), DigestAlgorithm.SHA256,
privateKey);
SignOneDocumentDTO signDocument = new SignOneDocumentDTO(toSignDocument, parameters,
new SignatureValueDTO(signatureValue.getAlgorithm(),
signatureValue.getValue()));
```

```

// Adds the signature value to the document
RemoteDocument signedDocument = restClient.signDocument(signDocument);

// Define the extension parameters
RemoteSignatureParameters extendParameters = new RemoteSignatureParameters();
extendParameters.setSignatureLevel(SignatureLevel.PAdES_BASELINE_T);

// Extends the existing signature
RemoteDocument extendedDocument = restClient.extendDocument(new
ExtendDocumentDTO(signedDocument, extendParameters));

// Defines timestamp parameters
RemoteTimestampParameters remoteTimestampParameters = new RemoteTimestampParameters();
remoteTimestampParameters.setDigestAlgorithm(DigestAlgorithm.SHA256);

// Defines a Timestamp document DTO
TimestampOneDocumentDTO timestampOneDocumentDTO = new
TimestampOneDocumentDTO(extendedDocument, remoteTimestampParameters);

// Timestamps a provided document (available for PDF, ASiC-E and ASiC-S container
formats)
RemoteDocument timestampedDocument =
restClient.timestampDocument(timestampOneDocumentDTO);

```

Get data to sign

The method allows retrieving the data to be signed. The user sends the document to be signed, the parameters (signature level,...) and the certificate chain.



The parameters in `getDataToSign` and `signDocument` MUST be the same (especially the signing date).

Request

```

POST /services/rest/signature/one-document/getDataToSign HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 3080

{
  "parameters" : {
    "signingCertificate" : {
      "encodedCertificate" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMME1Jvb3RTZWxmU21nbmVkB
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTCwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtLMREWdWYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZH9ax8F1fE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt

```

```

sStkYXdULqp wz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIB3DQEBCwUA4IBAQCK6LGA01TR+rmU8p6yhAi40kDN2b1
dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUi x00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1L0He3WKBNB6GZALT1ewjh7hSbjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6ky0I0Bvw14PWtsWG10Qd0SRvIBBrP4adCnGT
9jgjk9LTc08B8FKrr+81HGuc0bp4lIUToiUkGILXsiEeEg9WAqm+Xq0"
},
"certificateChain" : [ ],
"detachedContents" : null,
"asicContainerType" : null,
"signatureLevel" : "XAdES_BASELINE_B",
"signaturePackaging" : "ENVELOPING",
"jwsSerializationType" : null,
"sigDMechanism" : null,
"signatureAlgorithm" : "RSA_SHA256",
"digestAlgorithm" : "SHA256",
"encryptionAlgorithm" : "RSA",
"referenceDigestAlgorithm" : null,
"maskGenerationFunction" : null,
"contentTimestamps" : null,
"contentTimestampParameters" : {
  "digestAlgorithm" : "SHA256",
  "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
  "timestampContainerForm" : null
},
"signatureTimestampParameters" : {
  "digestAlgorithm" : "SHA256",
  "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
  "timestampContainerForm" : null
},
"archiveTimestampParameters" : {
  "digestAlgorithm" : "SHA256",
  "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
  "timestampContainerForm" : null
},
"signWithExpiredCertificate" : false,
"generateTBSWithoutCertificate" : false,
"imageParameters" : null,
"signatureIdToCounterSign" : null,
"blevelParams" : {
  "trustAnchorBPPolicy" : true,
  "signingDate" : 1625817756459,
  "claimedSignerRoles" : null,
  "policyId" : null,
  "policyQualifier" : null,
  "policyDescription" : null,
  "policyDigestAlgorithm" : null,
  "policyDigestValue" : null,
  "policySpuri" : null,
  "commitmentTypeIndications" : null,
}

```

```
    "signerLocationPostalAddress" : [ ],
    "signerLocationPostalCode" : null,
    "signerLocationLocality" : null,
    "signerLocationStateOrProvince" : null,
    "signerLocationCountry" : null,
    "signerLocationStreet" : null
  },
},
"toSignDocument" : {
  "bytes" : "SGVsbG8=",
  "digestAlgorithm" : null,
  "name" : "RemoteDocument"
}
}
```

Response

```
HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:36 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 1536

{
  "bytes" :
"PGRz01NpZ251ZE1uZm8geG1sbnM6ZHM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyMiPjxkc
zpDYW5vbmljYWxpeM0aW9uTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWw
tZXhjLWMxNG4jIj48L2Rz0kNhbm9uaWNhbG16YXRpb25NZXR0b2Q+PGRz01NpZ25hdHVyZU1ldGhvZCBBbGdvc
m10aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiPjwvZHM6U21
nbmF0dXJ1TWV0aG9kPjxkzpSZW1cmVuY2UgSWQ9InItaWQtY2Y4YTFiMTR1NGIxYzg3ZjdmOGJiZTExZWZiY
2I1zjQtMSIgVHlwZT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc21nI09iamVjdCIgVVJJPSIjby1
pZC1jZjhMWIxNGU0YjFjODdmN2Y4YmJ1MTF1ZmJjYjVmNC0xIj48ZHM6VHJhbNmb3Jtcz48ZHM6VHJhbNmb
3JtIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc21nI2Jhc2U2NCI+PC9kczpUcmF
uc2Zvcm0+PC9kczpUcmFuc2Zvcm1zPjxkzpEaWd1c3RNZXRob2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczL
m9yZy8yMDAxLzA0L3htbGVuYyNzaGEyNTYiPjwvZHM6RG1nZXN0TWV0aG9kPjxkzpEaWd1c3RWYWx1ZT5HRit
0c31KeC9pWDFZYWI4azRzdUprTUc3REJPMmxHQUI5RjJTQ1k0R1drPTwvZHM6RG1nZXN0VmFsduWU+PC9kczpSZ
WZ1cmVuY2U+PGRz01J1ZmVyZW5jZSBUeXB1PSJodHRw0i8vdXJpLmV0c2kub3JnLzAxOTAzI1NpZ25lZFByb3B
1cnRpZXMiIFVSS0iI3hhZGVzLW1kLWNm0GExYjE0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0Ij48ZHM6VHJhb
nNmb3Jtcz48ZHM6VHJhbNmb3JtIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWtZXh
jLWMxNG4jIj48L2Rz01RyYW5zZm9ybT48L2Rz01RyYW5zZm9ybXM+PGRz0kRpZ2VzdE1ldGhvZCBBbGdvcml0a
G09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiI+PC9kczpEaWd1c3RNZXRob2Q+PGR
z0kRpZ2VzdFZhHV1PnpPa0VRNVRFSytybWx4Vkftd1JrWWk0ZGxCbWtwWHDPMUx1V1NnYkUrd3M9PC9kczpEa
Wd1c3RWYWx1ZT48L2Rz01J1ZmVyZW5jZT48L2Rz01NpZ251ZE1uZm8+"
}
```

Sign document

The method allows generation of the signed document with the received signature value.



The parameters in `getDataToSign` and `signDocument` MUST be the same (especially the signing date).

Request

```
POST /services/rest/signature/one-document/signDocument HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
```

Content-Type: application/json; charset=UTF-8

Host: localhost:8080

Content-Length: 3505

```
{  
  "parameters" : {  
    "signingCertificate" : {  
      "encodedCertificate" : "  
MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMME1Jvb3RTZWxmU2lnbmVkr  
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ  
DDApTaWduZXJGYWt1MREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA  
MI3kZhtnipn+iIZHZ9ax8FlfE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH  
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt  
sStkYXdxULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC71Zg7PUZUTrdegABTUzYCR  
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA  
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi40kDN2b1  
dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC  
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz  
D7UT93Nuw3xcV8VIfIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyO10Bvw14PWtsWG10Qd0SRvIBBrP4adCnGT  
gjgjk9LTc08B8FKrr+81HGuc0bp4lIUToiUkGILXsiEeEg9WAqm+Xq0"  
    },  
    "certificateChain" : [ ],  
    "detachedContents" : null,  
    "asicContainerType" : null,  
    "signatureLevel" : "XAdES_BASELINE_B",  
    "signaturePackaging" : "ENVELOPING",  
    "jwsSerializationType" : null,  
    "sigDMechanism" : null,  
    "signatureAlgorithm" : "RSA_SHA256",  
    "digestAlgorithm" : "SHA256",  
    "encryptionAlgorithm" : "RSA",  
    "referenceDigestAlgorithm" : null,  
    "maskGenerationFunction" : null,  
    "contentTimestamps" : null,  
    "contentTimestampParameters" : {  
      "digestAlgorithm" : "SHA256",  
      "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",  
      "timestampContainerForm" : null  
    },  
    "signatureTimestampParameters" : {  
      "digestAlgorithm" : "SHA256",  
      "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",  
      "timestampContainerForm" : null  
    },  
    "archiveTimestampParameters" : {  
      "digestAlgorithm" : "SHA256",  
      "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",  
      "timestampContainerForm" : null  
    },  
    "signWithExpiredCertificate" : false,  
    "generateTBSWithoutCertificate" : false,  
  },  
  "signingAlgorithm" : "RSASSA-PSS",  
  "signingKey" : {  
    "key" : "MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMME1Jvb3RTZWxmU2lnbmVkr  
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ  
DDApTaWduZXJGYWt1MREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA  
MI3kZhtnipn+iIZHZ9ax8FlfE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH  
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt  
sStkYXdxULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC71Zg7PUZUTrdegABTUzYCR  
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA  
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi40kDN2b1  
dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC  
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz  
D7UT93Nuw3xcV8VIfIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyO10Bvw14PWtsWG10Qd0SRvIBBrP4adCnGT  
gjgjk9LTc08B8FKrr+81HGuc0bp4lIUToiUkGILXsiEeEg9WAqm+Xq0"  
  },  
  "signingKeyUsage" : "keyAgreement",  
  "signingAlgorithmUsage" : "sign",  
  "signingKeyUsageFlags" : 16384  
}
```

```

"imageParameters" : null,
"signatureIdToCounterSign" : null,
"blevelParams" : {
    "trustAnchorBPPolicy" : true,
    "signingDate" : 1625817756459,
    "claimedSignerRoles" : null,
    "policyId" : null,
    "policyQualifier" : null,
    "policyDescription" : null,
    "policyDigestAlgorithm" : null,
    "policyDigestValue" : null,
    "policySpuri" : null,
    "commitmentTypeIndications" : null,
    "signerLocationPostalAddress" : [ ],
    "signerLocationPostalCode" : null,
    "signerLocationLocality" : null,
    "signerLocationStateOrProvince" : null,
    "signerLocationCountry" : null,
    "signerLocationStreet" : null
},
"signatureValue" : {
    "algorithm" : "RSA_SHA256",
    "value" :
"V3ARz01hBrc943rWL1vyPzW0Ee3lsqYifpqYEEkPYHC4zVSAaCLUPTI8TYeroQKry6urZiQW9IqWyPMeqFgK
HCGY1EVUgYkj5Mj6QcDtIH/echUd+lQIJK7xdPq9sPA2znE9zgXI0EjByMU6u+xtH8S0B70VbjnAN/TN+4gCXu
j6aDpCz0ZStGei20bQLbh5DXUGuKNrLr51ROSz1iIqV4zqfRfaAfa+nF2RYQZDVgxnnndscGDZZG9/Fi0XhSqJ
6sXzc4fSQ+nDPfDmFOTY0fp8nbw0fRc4DFhIiG/Yqh10F090La0qLzfGRpsw7f43x9TzKrMFYQeLZPPeZFkRQ=
",
},
"toSignDocument" : {
    "bytes" : "SGVsbG8=",
    "digestAlgorithm" : null,
    "name" : "RemoteDocument"
}
}

```

Response

```

HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:36 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20

```

Connection: keep-alive

Content-Length: 5207

{

 "bytes" :

"PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGlubZ0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+PGRz01NpZ25hdHVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc21nIyIgSWQ9Im1kLWNm0GEExYjE0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0Ij48ZHM6U21nbmVksW5mbz48ZHM6Q2Fub25pY2FsaXphdG1vbk1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxkczpTaWduYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yYz8yMDAxLzA0L3htbGRzaWctbW9yZSNyc2Etc2hhMjU2Ii8+PGRz01J1ZmVzW5jZSBjZD0ic1pZC1jZjhMWIxNGU0YjFj0DdmN2Y4YmJ1MTF1ZmJjYjVmNC0xIiBUeXB1PSJodHRw0i8vd3d3LnczLm9yYz8yMDAwLzA5L3htbGRzaWcjT2JqZWN0IiBVUkk9IiNvLW1kLWNm0GEExYjE0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0LTEiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yYz8yMDAwLzA5L3htbGRzaWcjYmFzTY0Ii8+PC9kczpUcmFuc2Zvcm1zPjxkczpEaWdlc3RNZXRob2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yYz8yMDAxLzA0L3htbGVuYyNzaGEyNTYiLz48ZHM6RG1nZXN0VmFsdWU+R0YrTnN5SngvaVgxWWFi0G6s0c3VKA01HN0RCTzJsR0FCOUYyU0NZNEdXaz08L2Rz0kRpZ2VzdFZhHV1PjwvZHM6UmVmZXJ1bmNLPjxkczpS2WZ1cmVuY2UgVHlwZT0iaHR0cDoVl3VyaS51dHnpLm9yYz8wMTkwMyNTaWduZWRQcm9wZXJ0aWVzIiBVUkk9IiN4YWR1cy1pZC1jZjhMWIxNGU0YjFj0DdmN2Y4YmJ1MTF1ZmJjYjVmNCI+PGRz01RyYW5zZm9ybXM+PGRz01RyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjwvZHM6VHjhbnNmb3Jtcz48ZHM6RG1nZXN0TW0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMj2hhMjU2Ii8+PGRz0kRpZ2VzdFZhHV1PnpPa0VRNVRFSytybWx4Vkf1JrWw0ZGxCbWtwWHdPMUx1V1NnYkUrd3M9PC9kczpEaWdlc3RWYw1ZT48L2Rz01J1ZmVzW5jZT48L2Rz01NpZ251ZE1uZm8+PGRz01NpZ25hdHVyZVZhHV1IE1kPSJ2YWx1ZS1pZC1jZjhMWIxNGU0YjFj0DdmN2Y4YmJ1MTF1ZmJjYjVmNCI+VjNBUnpPMWhCcmM5NDNyV0wdxnlQe1dPRWUzbHNxWw1mcHFpWUV1s1BZSEM0e1ZTQWFDFTVQVEk4Vf1lcm9RS3J5NnVw1Rvz1JcVd5UE1lUZnS0hDR1lsvRVZVZ1lrajVNajZRY0R0aUgvZwNoVWQrbFFJSks3eGRQcT1zUEEyem5FOXpnWEkwRWpCeU1VNnUreHRI0FMwQjcwVmJqbkF0L1R0KzRnQ1h1ajZhRHBDejBaU3RHZwkyMGJRVUx1aDVEWFVhdU0ckxyNwXST1N6Mw1JcVY0enFmUmZhQWZhK25GM1JZUvPevmd4bm5kc0NHRFpaRzkvRmkwWGHtCuo2c1h6YzRmU1ErbkRQZkRtRk9UWTBmcDhuYndPZ1jNERGaElpRy9ZcWgxT0ZPOTBMYU9xTHpmR1Jwc3c3ZjQzeD1UektyTUZZUWVMw1BQZVpGa1JRPT08L2Rz01NpZ25hdHVyZVZhHV1PjxkczpLZX1JbmZvPjxkczpYNTA5RGF0Yt48ZHM6WDUw0UN1cnRpZm1jYXR1P1J1SUM2akNDQWRLZ0F3SUJBZ01HTHRZVTE3dFhNQTBHQ1NxR1NjYjNEUUVQ3dVQU1EQXhHeKFaQmd0VkJBTU1FbEp2YjNSVFpXeG1VMmxuYm1Wa1JtRnJaVEVSTUE4R0ExVUVdz3dJukz0VExYUmxjM1F3SGhjTk1UY3d0akE0TVRFeU5qQXhXaGNOTkRjd056QTBNRMxTnpJMFdqQW9NUk13RvFZRFZRUUREQXBUVvdKdVpYSkdZV3RsTVJFd0R3WURWUVFLREFoRVUxTXRkr1Z6ZERDQ0FTSxdEUV1LKS29aSWh2Y05BUUVQ1FBRGdnRVBBRENDQVFvQ2dnRUJBTUkza1podG5pcG4raWlaSFo5YXg4RmxmRTVpdy9jRndCVGZBRW1zUjFaUVVwNi9CUW5CdDpBzBKv0j0Yz1xa3Y3S1VEZGNCS1hQVjVRV1M1QX1NUehwcVE3NUhpdpGpzcS9GenU4ZUh0a0twRm16Y3hHT1CwMrrUWpoNHJTCnRPMUtzqzBSZDVEUXRXU2drZVZDQ04w0Wt0MFpzWjBFT1krSXA4UXhTbX16dHNTdGtZWGRVTHFwd3o0SkVYVz12ejY0ZVRiZGU0d1FKNnBqSEdhckpmMwdRTkVjM1h6aG1JL3ByWEEx5c1d0cUM3bFpnN1BVW1VUcmR1Z0FCVFV6WUNSSjFrV0JSUG00cW8wTE40MDVj0TRRUWQ0NWE1a1Rnb3dIekVnTG5BUUkyOHgwTTNBNT1LUS0MraWVOYzZWRjFQc1RMcFV3N1BOSTJWc3RYNwpBdWFzQ0F3RUFByU1TTUJBD0RnWURWUjBQQVFI0JBUURBZ0VHTUEwR0NTcUdTSWIzRFFFQkN3VUFN1LCQVFDSzZMR0EwMVRsk3JtVThwNn1oQWk0T2tETj1iMWRiSuw4bDhpQ01zB3BMQ3h40HhxcTN1Y1pDT3hxaDFYMMo2cGdXemFyYjBiL01VaXgwME1vVXZ0YkZPeEFXN1BCWk1LRExubTZMc2NrUnhzMVUzMnNDOWQxTE9IZTNX0J0QjZHWkFMVDF1d2poN2hTY1ldqZnRsbWVndnErNmVWR0E1Y3ZmMnUvMitUa0treUhWl05Smzk0b1hyZHNkcHZ5Z3d5cEV0WGP1dHpEN1VUOTN0dXczeGNWOFZJZnRJdkhmOUxqVTdoK1VqR21LWEc5YzE1ZV1yM1N6VW12Nmt5T0kwQnZ3MTRQV3RzV0dsMFFkT1NSdk1CQnJQNGFkQ25HVGdqZ2pr0UxUY084QjhGS3JyKzhsSEd1YzBicDRsSVVUb21Va0dJTFhzaUV1RWc5V0FxbStYcU88L2Rz01g1MD1DZXJ0aWZpY2F0ZT48L2Rz01g1MD1EYXRhPjwvZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWR1czpRdWFsaWZ5aW5nUHJvcGVydG11cyB4bWxuczp4YWR1cz0iaHR0cDovL3VyaS51dHnpLm9yYz8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9IiNpZC1jZjhMWIxNGU0YjFj0DdmN2Y4YmJ1MTF1ZmJjYjVmNCI+PHhhZGVz01NpZ251ZFBByb3B1cnRpZXMgSWQ9InhhZGVzW1kLWNm0GEExYjE0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0Ij48eGFkZXM6U21nbmVku21nbmF0dXJ1UHJvcGVydG11cz48eGFkZXM6U21nbmluZ1RpbWU+MjAyMS0wNy0w0VQw0DowMj0zN1o8L3hhZGVz01NpZ25pbmduAaW1lPjx4YWR1czpTaWd

```

uaW5nQ2VydGlmaWNhdGVWMj48eGFkZXm6Q2VydD48eGFkZXm6Q2VydERpZ2VzdD48ZHM6RG1nZXN0TWV0aG9kI
EFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hhNTEyIi8+PGRz0kRpZ2VzdFZ
hbHV1PjE0d01qREZ6ZnNxa2RaVXNuUEgwL2grWm84cno4REV3aU1xM11PMX0VGZwbEwzd2tRN0UwbDJ5W1BZZ
GVxR0s5U3hDVGx6cDExUk5FVWVMS01zQ2VRPT08L2Rz0kRpZ2VzdFZhbHV1PjwveGFkZXm6Q2VydERpZ2VzdD4
8eGFkZXm6SXNzdWVvU2VyaWFsVjI+TUQ0d05LUX1NREF4R3pBWKJnT1ZCQU1NRWxKdmIzU1RaV3htVTJsbmJtV
mtSbUZyW1RFUK1BOEdBMVVFQ2d3SVJGT1RMWFJsYzNRQ0JpN1dGTmU3Vnc9PTwveGFkZXm6SXNzdWVvU2VyaWF
sVjI+PC94YWRLczpDZXJ0PjwveGFkZXm6U21nbm1uZ0N1cnRpZm1jYXR1VjI+PC94YWRLczpTaWduZWRTaWduY
XR1cmVQcm9wZXJ0aWVzPjx4YWRLczpTaWduZWREYXRhT2JqZWN0UHJvcGvydGllcz48eGFkZXm6RGF0YU9iamV
jdEZvcm1hdCBPYmp1Y3RSZWZ1cmVuY2U9IiNyLW1kLWNm0GEExYjE0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0L
TEiPjx4YWRLczpNaW1lVH1wZT5hcHBsaWNhdG1vb19vY3R1dC1zdHJ1YW08L3hhZGVz0k1pbWVUeXB1PjwveGF
kZXm6RGF0YU9iamVjdEZvcm1hdD48L3hhZGVz01NpZ251ZERhdGFPYmp1Y3RQcm9wZXJ0aWVzPjwveGFkZXm6U
21nbmVkUHJvcGvydGllcz48L3hhZGVz01F1YWxpZn1pbmdQcm9wZXJ0aWVzPjwvZHM6T2JqZWN0PjxkczpPYmp
1Y3QgSWQ9Im8taWQtY2Y4YTFiMTR1NGIxYzg3Zjdm0GJiZTExZWz1Y2I1ZjQtMSI+U0dWc2JHOD08L2Rz0k9ia
mVjdD48L2Rz01NpZ25hdHVyZT4=",
  "digestAlgorithm" : null,
  "name" : "RemoteDocument-signed-xades-baseline-b.xml"
}

```

Extend document

The method allows extension of an existing signature to a stronger level.

Request

```

POST /services/rest/signature/one-document/extendDocument HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 7157

{
  "toExtendDocument" : {
    "bytes" :
"PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmu9Im5vIj8+PGRz01NpZ25hd
HVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc21nIyIgSWQ9Im1kLWNm0GEExYjE
0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0Ij48ZHM6U21nbmVkSW5mbz48ZHM6Q2Fub25pY2FsaXphdG1vbk1ld
GhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxkczpTaWd
uYXR1cmVNZXR0b2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNyc
2Etc2hhMjU2Ii8+PGRz01J1ZmVyZw5jZSBjZD0ici1pZC1jZjhMWIxNGU0YjFjODdmN2Y4YmJ1MTF1ZmJjYjV
mNC0xIiBUeXB1PSJodHRw0i8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjT2JqZWN0IiBVUkk9IiNvLW1kL
WNm0GEExYjE0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0LTEiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0
gQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjYmFzZTY0Ii8+PC9kczpUcmFuc
2Zvcm1zPjxkczpEaWd1c3RNZXRob2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGV
uYyNzaGEyNTYiLz48ZHM6RG1nZXN0VmFsdWU+R0YrTnN5SngvaVgxWWFi0G50c3VKA01HN0RCTzJ5R0FCOUYyU
0NZNEdXaz08L2Rz0kRpZ2VzdFZhbHV1PjwvZHM6UmVmZJ1bmN1PjxkczpSZWZ1cmVuY2UgVH1wZT0iaHR0cDo
vL3VyaS51dHNpLm9yZy8wMTkwMyNTaWduZWRQcm9wZXJ0aWVzIiBVUkk9IiN4YWR1cy1pZC1jZjhMWIxNGU0Y
jFjODdmN2Y4YmJ1MTF1ZmJjYjVmNC1+PGRz01RyYW5zZm9ybXM+PGRz01RyYW5zZm9ybSBBbGdvcm10aG09Imh
0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjwvZHM6VHJhbnNmb3Jtcz48ZHM6RG1nZ
XN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hhMjU2Ii8+PGR
z0kRpZ2VzdFZhbHV1PnpPa0VRNVRFSytybWx4Vkftd1JrWWk0ZGxCbWtwWhdPMUx1V1NnYkUrd3M9PC9kczpEa

```

```

Wd1c3RWYWx1ZT48L2Rz01J1ZmVyZw5jZT48L2Rz01NpZ251ZE1uZm8+PGRz01NpZ25hdHVyZVzhbHV1IE1kPSJ
2YWx1ZS1pZC1jZjhMWIxNGU0YjFj0DdmN2Y4YmJ1MTF1ZmJjYjVmNCI+VjNBUnpPMWhCcmM5NDNyV0wdxnlQe
1dPRWUzbHNxWwlmcHFpWUV1s1BZSEM0e1ZTQWFDTFVQVEk4VFl1cm9RS3J5NnVyWm1RVz1JcVd5UE11cUZnS0h
DR11sRVZVZ11rajVNajZRY0R0aUgvZWNoVWQrbFFJSks3eGRQcT1zUEEyem5FOXpnWEkwRWpCeU1VNnUreHRI0
FMwQjcwVmJqbkFOL1R0KzRnQ1h1ajZhRHBDejBaU3RHZwkyMGJRVUxiaDVEWFVHdUt0ckxyNWxST1N6MW1JcVY
0enFmUmZhQWZhK25GM1JZUVpEVmd4bm5kc0NHRFpaRzkvRmkwWGHtCuo2c1h6YzRmU1ErbkRQZkRtRk9UWTBmc
DhuYndPZ1JjNERGaElpRy9ZcWgxT0ZPOTBMU9xTHpmR1Jwc3c3ZjQzeDIUektyTUZZUWVMW1BQZVpGa1JRPT0
8L2Rz01NpZ25hdHVyZVzhbHV1PjxkczpLZX1JbmZvPjxkczpYNTA5RGF0YT48ZHM6WDUwOUN1cnRpZm1jYXR1P
k1JSUM2akNDQWRLZ0F3SUJBZ01HTHRZVTE3dFhNQTBHQ1NxR1NjYjNEUUVCQ3dVQU1EQXhHekFaQmd0VkJBTU1
FbEp2YjNSVFpXeG1VMmxuYm1Wa1JtRnJaVEVSTUE4R0ExVUVdz3dJukZ0VExYUmxjM1F3SGhjTk1UY3d0akE0T
VRFeU5qQXhXaGNOTkRjd056QTBNRGMxTnpJMFdqQW9NUk13RVFZRFZRUUREQXBUYVdkdVpYSkdZV3RsTVJFd0R
3WURWUVFLREFoRVUxTXRkR1Z6ZERDQ0FTSxdEUV1ks29aSWh2Y05BUUVCQ1FBRGdnRVBBRENDQVFvQ2dnRUJBT
Ukza1podG5pcG4raWlaSFo5YXg4RmxmRTVpdy9jRndCVGZBRW1zUjFaUVVwNi9CUW5CdDdPbzBKv0J0Yz1xa3Y
3S1VEZGNCS1hQVjVRV1M1QX1NUehwcVE3NUhpdgPzcs9GenU4ZUh0a0twRml6Y3hHYT1CWmRrUWpoNHJTcnRPM
UtqczBSZDVEUXRXU2drZVZDQ04w0Wt0MFpzjBFT1krSXA4UxhTbX16dHNTdGtZWGRVTHFwd3o0SkVYVz12ejY
0ZVRiZGU0d1FKNnBqSEdhckpmMWdRTkVjM1h6aG1JL3ByWEx5c1d0cUM3bFpn1BVW1VUcmR1Z0FCVFV6WUNSS
jFrV0JSUG00cW8wTE40MDVj0TRRUWQ0NWE1a1Rnb3dIekVnTG5BUUky0HgwTTNBNT1US0MraWVOYzZWRjFQc1R
McFV3N1BOSTJWc3RYNwpBdWFzQ0F3RUFByU1TTUJBd0RnWURWUjBQQVFI0JBURBZ0VHTUEwR0NTcUdTSWIzR
FFFQkN3VUFBNE1CQVFDSzZMR0EwMVRsk3jtVThwNn1oQWk0T2tETjJiMWRiSuw4bDhpQ01Zb3BMQ3h40HhxcTN
1Y1pDT3hxaDFYMMo2cGdXemFyYjBi1VaXgwME1vVXZ0YkZPeEFXN1BCWk1LRExubTZMc2NrUnhzMVUzMnNDO
WQxTE91ZTNXS0J0QjZHWkFMVDF1d2poN2hTYldqZnRsbWNvdnErNmVWR0E1Y3ZmMnUvMitUa0treUhWL05Smzk
0b1hyZHNkcHZ5Z3d5cEV0WGP1dHpEN1VUOTN0dXceGNWOFZJZnRJdkhmOUxqVTdoK1VqR21LWEc5YzE1ZV1yM
1N6VW12Nmt5T0kwQnZ3MTRQV3RzV0dsMFFkT1NSdk1CQnJQNGFkQ25HVGdqZ2pr0UxUY084QjhGS3jYkzsSEd
1YzBicDRsSVVUb21Va0dJTFhzaUV1RWc5V0FxbStYcU88L2Rz01g1MD1DZXJ0aWZpY2F0ZT48L2Rz01g1MD1EY
XRhPjwvZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWR1czpRdWFsaWZ5aW5nUHJvcGVydG11cyB4bWxuczp4YWR
1cz0iaHR0cDovL3VyaS51dHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9IiNpZC1jZjhMWIxNGU0YjFj0
DdmN2Y4YmJ1MTF1ZmJjYjVmNCI+PHhhZGVz01NpZ251ZFByb3B1cnRpZXMgSWQ9InhhZGVzLWlkLWNm0GExYjE
0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0Ij48eGFkZXM6U21nbmVku21nbmF0dXJ1UHJvcGVydG11cz48eGFkZ
XM6U21nbmluZ1RpbWU+MjAyMS0wNy0wOVQw0DowMj0zN1o8L3hhZGVz01NpZ25pbmduaW11Pjx4YWR1czpTaWd
uaW5nQ2VydG1maWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlnZXN0TW0aG9kI
EFsZ29yaXRobt0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hhNTeyIi8+PGRz0kRpZ2VzdFZ
hbHV1PjE0d01qREZ6ZnNxa2RaVXNuUEgwL2grWm84cno4REV3aU1xM11PMXd0VGZwbEwzd2tRN0UwbDJ5W1BZZ
GVxR0s5U3hDVGx6cDExUk5FVWVMS01zQ2VRPT08L2Rz0kRpZ2VzdFZhbHV1PjwveGFkZXM6Q2VydERpZ2VzdD4
8eGFkZXM6SXNzdWVYU2VyaWFsVjI+TUQ0d05LUX1NREF4R3pBWkJnT1ZCQU1NRWxKdmIzU1RaV3htVTJsbmJtV
mtSbUZyW1Rfuk1BOEdBMVVFQ2d3SVJGT1RMWFjsYzNRQ0jPn1dGTmU3Vnc9PTwveGFkZXM6SXNzdWVYU2VyaWF
sVjI+PC94YWR1czpDZXJ0PjwveGFkZXM6U21nbmluZ0N1cnRpZm1jYXR1VjI+PC94YWR1czpTaWduZWRtaWduY
XR1cmVQcm9wZXJ0aWVzPjx4YWR1czpTaWduZWRREYXrhT2JqZWN0UHJvcGVydG11cz48eGFkZXM6RGF0YU9iamV
jdEZvcm1hdCBPYmp1Y3RSZwZ1cmVuY2U9IiNyLWlkLWNm0GExYjE0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0L
TEiPjx4YWR1czpNaW11VH1wZT5hcHBsaWNhdG1vb19vY3R1dC1zdHJ1Yw08L3hhZGVz0k1pbWVUeXB1PjwveGF
kZXM6RGF0YU9iamVjdEZvcm1hdD48L3hhZGVz01NpZ251ZERhdGFpYmp1Y3RQcm9wZXJ0aWVzPjwveGFkZXM6U
21nbmVkuUHJvcGVydG11cz48L3hhZGVz01F1YwpxZn1pbmdQcm9wZXJ0aWVzPjwvZHM6T2JqZWN0PjxkczpPYmp
1Y3QgSWQ9Im8taWQtY2Y4YTFiMTR1NGIxYzg3Zjdm0Gj1ZExZWZiY2I1ZjQtMSI+U0dWc2JHOD08L2Rz0k9ia
mVjdD48L2Rz01NpZ25hdHVyZT4=",
  "digestAlgorithm" : null,
  "name" : "RemoteDocument-signed-xades-baseline-b.xml"
},
"parameters" : {
  "signingCertificate" : null,
  "certificateChain" : [ ],
  "detachedContents" : null,
  "asicContainerType" : null,
}

```

```

"signatureLevel" : "XAdES_BASELINE_T",
"signaturePackaging" : null,
"jwsSerializationType" : null,
"sigDMechanism" : null,
"signatureAlgorithm" : "RSA_SHA256",
"digestAlgorithm" : "SHA256",
"encryptionAlgorithm" : "RSA",
"referenceDigestAlgorithm" : null,
"maskGenerationFunction" : null,
"contentTimestamps" : null,
"contentTimestampParameters" : {
  "digestAlgorithm" : "SHA256",
  "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
  "timestampContainerForm" : null
},
"signatureTimestampParameters" : {
  "digestAlgorithm" : "SHA256",
  "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
  "timestampContainerForm" : null
},
"archiveTimestampParameters" : {
  "digestAlgorithm" : "SHA256",
  "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
  "timestampContainerForm" : null
},
"signWithExpiredCertificate" : false,
"generateTBSWithoutCertificate" : false,
"imageParameters" : null,
"signatureIdToCounterSign" : null,
"blevelParams" : {
  "trustAnchorBPPolicy" : true,
  "signingDate" : 1625817756630,
  "claimedSignerRoles" : null,
  "policyId" : null,
  "policyQualifier" : null,
  "policyDescription" : null,
  "policyDigestAlgorithm" : null,
  "policyDigestValue" : null,
  "policySpuri" : null,
  "commitmentTypeIndications" : null,
  "signerLocationPostalAddress" : [ ],
  "signerLocationPostalCode" : null,
  "signerLocationLocality" : null,
  "signerLocationStateOrProvince" : null,
  "signerLocationCountry" : null,
  "signerLocationStreet" : null
}
}
}

```

Response

HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:37 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 10129

{
 "bytes" :
 "PD94bWwgdmlvc21vbj0iMS4wIiB1bmNvZGlubz0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+PGRz01NpZ25hd
 HVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyIgSWQ9ImlkLWNm0GExYjE
 0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0Ij48ZHM6U2lnbmVkSW5mbz48ZHM6Q2Fub25pY2FsaXphdG1vbk1ld
 GhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxkczpTaWd
 uYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNyc
 2Etc2hhMjU2Ii8+PGRz01J1ZmVzW5jZSBjZD0ici1pZC1jZjhMWIxNGU0YjFjODdmN2Y4YmJlMTF1ZmJjYjV
 mNC0xIiBUeXB1PSJodHRw0i8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjT2JqZWN0IiBVUkk9IiNvLW1kL
 WNm0GExYjE0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0LTEiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0
 gQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjYmFzTY0Ii8+PC9kczpUcmFuc
 2Zvcm1zPjxkczpEaWd1c3RNZXRob2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGV
 uYyNzaGEyNTYiLz48ZHM6RG1nZxN0VmFsdWU+R0YrTnN5SngvaVgxWWFi06s0c3VKA01HN0RCTzJsR0FCOUYyU
 0NZNEdXaz08L2Rz0kRpZ2VzdFZhbHV1PjwvZHM6UmVmZxJ1bmN1PjxkczpSZWZ1cmVuY2UgVHlwZT0iaHR0cDo
 vL3VyaS51dHNpLm9yZy8wMTkwMyNTaWduZWRQcm9wZXJ0aWVzIiBVUkk9IiN4YWR1cy1pZC1jZjhMWIxNGU0Y
 jFjODdmN2Y4YmJlMTF1ZmJjYjVmNCI+PGRz01RyYW5zZm9ybXM+PGRz01RyYW5zZm9ybSBBbGdvcml0aG09Imh
 0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjwvZHM6VHJhbnNmb3Jtcz48ZHM6RG1nZ
 XN0TW0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hhMjU2Ii8+PGR
 z0kRpZ2VzdFZhbHV1PnpPa0VRNVRFSytybWx4Vkf1d1JrWWk0ZGxCbWtwWhdPMUx1V1NnYkUrd3M9PC9kczpEa
 Wd1c3RWYw1ZT48L2Rz01J1ZmVzW5jZT48L2Rz01NpZ251Z1uZm8+PGRz01NpZ25hdHVyZVZhbHV1IE1kPSJ
 2YWx1ZS1pZC1jZjhMWIxNGU0YjFjODdmN2Y4YmJlMTF1ZmJjYjVmNCI+VjNBUnpPMWhCcmM5NDNyV0wdxnlQe
 1dPRWUzbHNxWWlmcHFpWUV1S1BZSEM0e1ZTQWFDTFVQVEk4VFl1cm9RS3J5NnVyWmlRVz1JcVd5UE1lCzN0h
 DR11sRVZVZ11rajVNajZRY0R0aUgvZWN0VWQrbFFJSks3eGRQct1zUEEyem5FOXpnWEkwRWpCeU1VNnUreHRI0
 FMwQjcwVmJqbkF0L1R0KzRnQ1h1ajZhRHBDejBaU3RHZwkyMGJRVUxiaDVEWFVhdUt0ckxyNWxST1N6MW1JcVY
 0enFmUmZhQWZhK25GM1JZUVpEVmd4bm5kc0NHRFpaRzkvRmkwWGHtCuo2c1h6YzRmU1ErbkRQZkRtRk9UWTBmc
 DhuYndPZ1JjNERGaElpRy9ZcWgxT0ZPOTBMYU9xTHpmR1Jwc3c3ZjQzeD1UektyTUZZUWVMW1BQZVpGa1JRPT0
 8L2Rz01NpZ25hdHVyZVZhbHV1PjxkczpLZX1JbmZvPjxkczpYNTA5RGF0YT48ZHM6WDUwOUN1cnRpZm1jYXR1P
 k1JSUM2akNDQWRLZ0F3SUJBZ01HTHZVTE3dFhNQTBHQ1NxR1NjYjNEUUVQ3dVQU1EQXhHekFaQmd0VkJBTU1
 FbEp2YjNSVFpXeG1VmMxuYm1Wa1JtRnJaVEVSTUE4R0ExVUVDZ3dJukz0VExYUmxjM1F3SGhjTk1UY3d0akE0T
 VRFeU5qQXhXaGNOTkRjd056QTBNRGmxTnpJMFdqQW9NUk13RVFZRFZRUUREQXBUYVdkdVpYSkdZV3RsTVJFd0R
 3WURWUVFLREFoRVUxTXRkR1Z6ZERDQ0FTSxdEUV1LKS29aSWh2Y05BUUVQ1FBRGdnRVBBRENDQFvQ2dnRUJBT
 Ukza1podG5pcG4raWlaSFo5YXg4RmxmRTVpdy9jRndCVGZBRW1zUjFaUVVwNi9CUW5CdDpBzBKV0J0Yz1xa3Y
 3S1VEZGNCS1hQVjVRV1M1QX1NUehwcVE3NUhpdgPzcs9GenU4ZUh0a0twRml6Y3hHYT1CWmRrUWpoNHJTcnRPM
 UtqczBSZDVEUXRXU2drZVZDQ04w0Wt0MFpzWjBFT1krSXA4UXhTbX16dHNTdGtZWGRVTHFwd3o0SkVYVz12ejY
 0ZVRiZGU0d1FKNnBqSEdhckpmMWdRTkVjM1h6aG1JL3ByWEx5c1d0cUM3bFpnN1BVW1VUcmR1Z0FCVFV6WUNSS

jFrV0JSUG00cW8wTE40MDVj0TRRUWQ0NWE1a1Rnb3dIekVnTG5BUUkyOHgwTTNBNT1US0MraWVOYzzWRjFQc1R
McFV3N1BOSTJWc3RYNWPBdWFzQ0F3RUFBYU1TTUJBd0RnWURWUjBQQVFILOJBURBZ0VHTUEwR0NTcUdTSWIzR
FFFQkN3VUFBNE1CQVFDszZMR0EwMVRSK3jtVThwNnloQWk0T2tETjJiMWRiSuw4bDhpQ01Zb3BMQ3h40HhxTN
1YlpDT3hxaDFYMmo2cGdXemFyYjBiL01VaXgwME1vVXZOYkZPeEFXN1BCWk1LRExubTZMc2NrUhzMVUzMnND0
WQxTE9IZTNXS0J0QjZHWkFMVDFld2poN2hTY1dqZnRsbWNvdnErNmVWR0E1Y3zmnUvMitUa0treUhWL05SmzK
0b1hyZHNkcHZ5Z3d5cEV0WGP1dHpEN1VUOTN0dXczeGNWOFZJZnRJdkhmOUxqVTdoK1VqR21LWEc5YzE1ZV1yM
1N6VW12Nmt5T0kwQnZ3MTRQV3RzV0dsMFFkt1NSdk1CQnJQNGFkQ25HVGdqZ2prOUxUY084QjhGS3JyKzhsSEd
1YzBicDRsSVVUb21Va0dJTFhzauV1RWc5V0FxbStYcU88L2Rz01g1MD1DZXJ0aWZpY2F0ZT48L2Rz01g1MD1EY
XRhPjwvZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWR1czpRdWFsaWZ5aW5nUHJvcGVydG11cyB4bWxuczp4YWR
1cz0iaHR0cDovL3VyaS51dHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZQ9IiNpZC1jZjhMWIxNGU0YjFj0
DdmN2Y4YmJ1MTF1ZmJjYjVmNCI+PHhhZGVz01NpZ251ZFByb3B1cnRpZXMgSWQ9InhhZGVzLWlkLWNm0GExYjE
0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0Ij48eGFkZXM6U21nbmVku21nbmF0dXJ1UHJvcGVydG11cz48eGFkZ
XM6U21nbm1uZ1RpbWU+MjAyMS0wNy0wOVQw0DowMj0zN1o8L3hhZGVz01NpZ25pbmdUaW11Pjx4YWR1czpTaWd
uaW5nQ2VydG1maWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZVzdD48ZHM6RG1nZXN0TWV0aG9kI
EFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hhNTEyIi8+PGRz0kRpZ2VzdFZ
hbHV1PjE0d01qREZ6ZnNxa2RaVXNuUEgwL2grWm84cno4REV3aU1xM11PMXd0VGZwbEwzd2tRN0UwbDJ5W1BZZ
GVxR0s5U3hDVGx6cDExUk5FVWVMS01zQ2VRPT08L2Rz0kRpZ2VzdFZhbHV1PjwveGFkZXM6Q2VydERpZ2VzdD4
8eGFkZXM6SXNzdWVUy2VyaWFsVjI+TUQ0d05LUX1NREF4R3pBwkJnT1ZCQU1NRWxKdmIzU1RaV3htVTJsbmJtV
mtSbUZyW1RFUk1B0EdBMVVFQ2d3SVJGT1RMWFJsYzNRQ0JpN1dGTmU3Vnc9PTwveGFkZXM6SXNzdWVUy2VyaWF
sVjI+PC94YWR1czpDZJ0PjwveGFkZXM6U21nbm1uZ0N1cnRpZm1jYXR1VjI+PC94YWR1czpTaWduZWRaWduY
XR1cmVQcm9wZJ0aWVzPjx4YWR1czpTaWduZREYXrhT2JqZWN0UHJvcGVydG11cz48eGFkZXM6RGF0YU9iamV
jdEZvcm1hdCBPYmp1Y3RSZW1cmVuY2U9IiNyLW1kLWNm0GExYjE0ZTRiMWM4N2Y3ZjhiYmUxMWVmYmNiNWY0L
TEiPjx4YWR1czpNaW11VH1wZT5hchBsaWNhdG1vb19vY3R1dC1zdHJ1Yw08L3hhZGVz0k1pbWVUeXB1PjwveGF
kZXM6RGF0YU9iamVjdEZvcm1hdD48L3hhZGVz01NpZ251ZERhdGFPYmp1Y3RQcm9wZJ0aWVzPjx4YWR1czpVbnNpZ251ZFNpZ25hdHV
yZVByb3B1cnRpZXM+PHhhZGVz01NpZ25hdHVyZVRpbWVTdGFTcCBJZD01VFMtNWQ30Tk1ZmYtNWU3ZS00NTc1L
WFkMDktY2FjNjYxZDY30WU3Ij48ZHM6Q2Fub25pY2FsaXphdG1vbk11dGhvZCBBbGdvcm10aG09Imh0dHA6Ly9
3d3cudzMu3JnLzIwMDEvMTAvE1sLWV4Yy1jMTRuIyIvPjx4YWR1czpFbmNhchN1bGF0ZWRUaW11U3RhbXAgS
WQ9IkVUUy01ZDc50TVmZi01ZTd1LTQ1NzUtYWQwOS1jYWM2NjFkNjc5ZTciPk1JSUppZ11KS29aSWh2Y05BUWN
Db01JSmV6Q0NDWGNQVFNeERUQuxCZ2xnaGtnQ1pRTUVBZ0V3Y0FZTEtvWk1odmN0QVFrUUFRU2dZUVJmTUYwQ
0FRRUdBeW9EQkRBdk1Bc0dDV0NHU0FGbEF3UUNBUVFneUJVUUZkQWRVTmxseGQ1MGxrMG1MWU4yUkVJcmJjK2R
Ub2Evbk5iMHpOTUNFUUQwSGszZ2Yreld1TDdsZEw1dVEzaytHQTh5TURJeE1EY3dPVEE0TURJek5scWdnZ1dGT
U1JRmdUQ0NBm1nQxdJQkFnSVVHVGnpd1h1eVFaMXzXyZc3Ukg5KzNMV0Q3b293RFFZSkvWk1odmN0QVFFTEJ
RQXdVVEVMTUFrR0ExVUVCaE1DUwtVeEZUQVRCZ05WQkFvTURFU1RVeUJVU1Z0VU1GQkxTVEVSTUE4R0ExVUVd
3dJVKvWVFZDQ1FTMgt4R0RBV0JnT1ZCQU1NRD0bGJHWXRjMmxuYm1Wa0xYUnpZVEF1RncweU1ERX1NVF13Tnp
Bek16WmFGdzB6TURFeU1UUxd0ekF6TXpaYU1GRXhDekFKQmd0VkjBwVRBa0pGTVJv0d0V3WURWUVFLREF4RVuXt
WdWRVZUVkNCUVMwa3hFVEFQmd0VkjBc01DR1JGVTFRZ1VFdEpNUmd3RmdZRFZRUUREQT16W1d4bUxYTnBaMjV
sWkMxMGMyRXdnZ01pTUEwR0NTcUdTSWIzRFFFQkFRVUFBN1DRhdBd2dnSUtBb01DQVFERzAxUX1PekNYYjd1L
zg0S0FyUUpJNFJwcVByV1NORm4wUWJLdkZmdWZGak1oZXnX22JWcmtZR1VtUVp0aE1FS3ZYV9CUFVktUxNZFh
nUXNFBFYxb1FyTudn0Dg3NUdGQkQ4SEN1aStiQUFjCdySG4z0TBVbUQyWjdLR1Y5KzNNRxBMcVR1SndhS3dRT
GNnY2kwZ1VqaXpiemtPa25QSS9vUzY30GRHUjR2TzJjR0RXZitoMnVYRTJ2TmR1bEnNQzdLN1pWb2VVSjRGa3p
OVXJoaHZXUVRQR1NybhJtRjJrRG16TUZNSm5pS2FMcKfjSGZGcGhzT1VwV0txR3hQujI0c3ZDRGJ1aWZYeGd1N
GU0bmRFNTcweWwrcWFLT0ROY1UyYWZPTE1BL1VBa3pzYkJSOW90d0dUVHzuSX1talpQaWtodm1tQ1NBb3NYa1U
2aVkvM1A0K2J1d0NRS2dBmBwLzh0VkvVHM3UveitVNzdwVI0UW9UKzB6RWgrYUZhS0pvVmJ5M1E1QmIwTmxmb
ELLcFY3c2tmMzBXU2UvN2FLR1IxZEMyWVvJzW5i1Jvb1EyeS9uRktUd20xTkzrQvhIeThMdDhXeitxME5pQ1V
yelc3SkpRTVovaG1FU011aj15WitzdG1MS1V2Y28wZ1BBMVNINHN1cEUye1MzSEpRRW10eHpCSFBpYURySWIyY
WJvZW1LQ1QxSjBNUUwrZWovUmvCc29iSm14dXFzWjRyc1RIcHRjU0ZMd1hGK3RqY1loKzVqazh1TkZLY1pQVjV
mbDk5anczcE13RTE5Q0UwaF1RS1FZWi9MeExhUVNGeCs10G4zR3AyekhSdXJmeis3STBKWX10TGRkN0NuemcrU
mJUe1RkU2dwaW15aHd1aUJ4MU80U0FVTVFJREFRQUJvMUV3VHPBSkJnT1ZIuk1FQWpBQU1Bc0dBMVVkRHdRRUF
3SUhnREFXQmd0VkhTVUJBZjhFRERBS0JnZ3JCZ0VGQ1fjRENEQWRCZ05WSFE0RUZnUVV4L0JjNngrSTN5SEpuS
01UdW56NjJDMVRsb2d3RFFZSkvWk1odmN0QVFFTEJQRURnZ01CQUhMdFVnb0JkRjBWTm1nWjRwNkZ3bXhMaEN

```

3WlovUktaNVdyd25JUU10L0FtUHI0VjZsdmZ0eVhTZ1JXQy9VZ1dockZuUW5ySmFr dWJsYnAxc3RwY3RyQTl2N
DFxdGFKRGJGV0Qxbn1YTC9qWXBiYzdUUGRNVFV1aCtTNFltcmRsVVN5N25jVGdLYVJ3QVZhdjh0MGxyMEFFTFN
McEJkM2xBTkJrMFNjUEI0T3Z6WWQwZzdyVUsrVHNq0EJPWF1SMzhnTEFvanJ1QmVSUkJJaHVY a2dTCTpL09PW
1ZsMGROTWtxZ1l0Y1AzY1BMWVVtNVJraytMWC9pV0VUQ3vkWGNCTEhuY040aGQyczJ5Qm1NVUxURzRsWWFERnd
ML1ZXT0JyTFNvYTN0cU13cGw4UkZLQmpNVT h6MVYxMFVhK1l0Vz1ETWU2eE55WnJ2SmdDUjNaUzVrYjFYTGtsR
nJ0c09MTW5MaGg3cXExbndPTnBVQ1ZFTzBaaXFDbDNNMnFNRHgxNGpXZ3RaRTV4U1VTR21vK0gxc2pRajlpbUN
XTHVoY1JuVHdFcF1FaHppWWhzaHJ2QzN2N1FHZwpEbDNSanhjV1pHL1JTbzA5TzVETWpXb0JGVkxTWkxGa1NoT
nQ0end1V0tIMGd0UUUV1VzNZYThXZGdCNV1scmcxY0d6eExLV3hKWTN10GM1Nk1vZ2tsUzZzM1UxMFBCbnNQVkY
3Rn15U0Y1MkJn0U9oT21TaWVyOW15ZXJuZ1V6Mk90eFdSanZPbDM4QUFoOWU5S281NUxBdFgxNWF rMmpQbmZZV
Xh5eGtHv mRMejBFQ1da0GpDN0Zrb1NLeHBaNGh3U0NsTU9mKzUrSThKQ2hvR69HQs9rYUFXSGgvVHNQMURRd2p
1S1RkUXJ0WFV0QVv nTV1JRFpqQ0NBmk1DQVFFd2FUQ1JNUXN3Q1FZRFZRUUdFd0pDU1RFVk1CTUdBMVVFQ2d3T
VJGT1RJR1JGVTFRZ1VFdEpNUkV3RhdZRFZRUUxEQWhVU1ZOVU1GQkxTVEVZTUJZROExVUVBd3dQYzJWc1ppMXp
hV2R1W1dRdGRITmhBaFFaTn1L0WQ3SkJuVytwenZ0RWYzN2N0WB1aWpBTEJnbGdoa2dCW1FNRFUf nR2dnZEV3R
2dZSkvWk1odmNOQVFrRE1RMEdDeXFHU01iM0RRRU pFQUVFTUJ3R0NTcUdTSW1zRFFSkJURVBGdzB5TVRB M01
Ea3dPREF5TXpaYU1Dc0dDU3FHU01iM0RRRU pOREV1TUJ3d0N3WUpZSVpJQVdVREJBSUJvUTBHQ1NxR1NJYjNEU
UVCQ3dVQU1D0EdDU3FHU01iM0RRRU pCREVpQkNCTDFDck56VZiNWhjSmU4VnAyd1AyK01Y0EpUdzVydU5TTWI
3ajdTbENFREEzQmdzcWhraUc5dzBCQ1JBQ0x6RW9NQ113SkRBaUJDQTJqUyt2WVE5ckZRVGV1V2QwYTRUUzg1U
kJWdnY1N3F2S205c2xnV21BWGpBTkJna3Foa21HOXcwQkFRc0ZBQVNDQWdBMdg3N3djZjZ6TXJLZ0RuaTdFL1F
nTXVxVFRGWhmCamE0Tk5Va1Ix b3J1TFdSbys3dWtaK1dIMD1CMjJBBEZCOWtPY3NTQy8wdkxYdk9rc1dLaXVUN
0w1eW9NWDN2R01sQWxZVjFCN0Q5Y1B6R1FnR011WVJZamY1enZ0TWtyUVpGv1lnSGdPTzB0dDJUUU9zNk120GJ
1bENVQTFGc0FNeWxzzTZK2J oVVJ3Z1VrQjZYS1dPOTVY aTR2Y1hMTWzC0d dR YXVwb1V6SW5JVTY1UW93SDJzT
EswZ1RES3prU1VhaV k5YjZ5V1J0REZju21XV2R3dDkzV11kcEtMZEoyNWRrUFg0YVhCa2dQT01LOU9qbWJsYVE
4RGp1R3pXaFhGZGpTcDB5MmFCYW5DTUJub0100UF1b05jVklWMnpQe1NaVEhrd1NBRDYzaWRhN0V3TwkQnI3Q
W5UT0RnanpiRFd0dDhOR2J1ktPTk5PT1I2V1pDbT1XZxp0VjVYdHdVYkR1WvhET1J3SXpQYzdRR1JHSXBveHN
aSU1qTXA3T1R3TEZLbjV kSEU0eWtHbUVFVFc zQjNYN0ZNVzYyeGRFM1BjckFNVjhVeXNjamsaUVwVk1oK3JjT
2pXchJ0R1F1QU9SMGxxT0tuN0p00URzN2grZzYwaTQ3Vz1XVTNHUU9vWdk2eHRGcDRybRRbTRXS1Z0a1M4aVg
rZ2R1eFVHZy8rKzFNeDdScXV0MzFFazdYWXcvWC96aVFQW1U2UwPYXNmMzFQaGdKRVFYVR RrVGdGbWdpR1VQ
2NuVzV1SVNrQkdIS1YvODViRHRMRkRhTV1YQnVUaCtzQ0F6bXQ4Nk0zamJMMGJXMTRIL2VaQy9QY3JjaDZaUXp
vcHJzakpYOTFYcF1ETU96Swc9PTwveGFkZXM6RW5jYXBzdWxhdGVkVG1tZVN0Yw1wPjwveGFkZXM6U2lnbmF0d
XJ1VG1tZVN0Yw1wPjwveGFkZXM6VW5zaWduZWR TaWduYXR1cmVQcm9wZJ0aWVzPjwveGFkZXM6VW5zaWduZWR
Qcm9wZJ0aWVzPjwveGFkZXM6UXvhG1meWluZ1Byb3B1cnRpZXM+PC9kczpPYmp1Y3Q+PGRz0k9iamVjdcBJZ
D0iby1pZC1jZjhMMWIxNGU0YjFjODdmN2Y4YmJ1MTF1ZmJjYjVmNC0xIj5TR1ZzYkc4PTwvZHM6T2JqZWN0Pjw
vZHM6U2lnbmF0dXJ1Pg==",
    "digestAlgorithm" : null,
    "name" : "RemoteDocument-signed-xades-baseline-b-extended-xades-baseline-t.xml"
}

```

Timestamp document

The method allows timestamping of a provided document. Available for PDF, ASiC-E and ASiC-S container formats.

Request

```

POST /services/rest/signature/one-document/timestampDocument HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 182846

```

```
{
  "timestampParameters" : {
    "digestAlgorithm" : "SHA256",
    "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
    "timestampContainerForm" : "PDF"
  },
  "toTimestampDocument" : {
    "bytes" :
"JVBERi0xLjQKJc0kw7zDts0fCjIgMCBvYmoKPDwvTGVuZ3RoIDMgMCBSL0ZpbHR1ci9GbGF0ZUR1Y29kZT4+c
n0cmVhbQp4nGwTPWvEMAY69/wKz4Wkk1i3ZDhhD7y43dCscdCjd+rEV2qV/v5LsHlmUgHAkRXr0WoEJ3e/w7cC
BnHjmybtMOGX38z4837kviaHEvGT8fA6Hy+DzxC6DF9f1zd2f0aGcP14K5colch2xxGMdvVpxnNXhoY5UOFxFH
cnXvKZRzwgloB6Dtwwwm1Ai5aalf0UW9w6pWBxDtE+jcvGScf6enkclsVwpBNOJDPgdoqU0xTbFH6dArBiAes
AQXcbEUQuwBAr76FAkmNBXMdkwZn9Tyo0WngNGS6r+6SeVnDRdJ3f4SaYkEERq/ZZxUIA9INs1BHU39Lnbcj
1GHTVJE+EZDXk0ZmzI1M+pB3ILzDUqrnJ1XnOfJM/LswgSKqZg4a2QvokpaFssnteVoA0Wdyz+p2RK69uN1PN
eyrFpGZqWej1XHYPyRQMUB+s9ijpeU7Qmas2dogYfd3tAvKLzB121srVZ7IE0QeG0NeseSBe944au2PAABzgaq
F+vu9WwTKHFDmsxE1rbHSEdsXdLfttj0/+fMIWwrm/cra8gUWmGuqRtD5QDeyTb8Sq27YHKbH+lzUDKGfSzU/s
6tDU7yyBXuif3B+aa6fYKZW5kc3RyZWftCmVuZG9iagoKMyAwIG9iago0NTkZW5kb2JqCgo0IDA9b2JqCjw8L
0x1bmd0aCA1IDA9Uj4+Cn0cmVhbQpQSwMEFAAACAAJme5S17GmgnAAAAJwAAAAGAAABtaW1ldHlwZWFWcGx
pY2F0aW9uL3ZuZC5vYXNpcy5vcGVuZG9jdW1lbnQudGV4dFBLawQUAAAIAAmZ71KScopFokGAACJBgAAGAAA
FRodW1ibmFpbHMvdGh1bWJuYWlsLnBuZ41QTkcNChoKAAAADU1IRFIAAAC1AAABA9gDAAAawv3H6QAAAb1QTFR
FeEsef1MoWrx6ZcGDbsWLdMeQd8iSesmUgFyshV00imM9jm1E1HF013RSmHZUm3pYn4BgoYJjpolsqIxvqo5xr
JB1rpR5sJZ9sph//5F1/5N5s5qBuKGJvKaQv6mVs4H/u4//vJH/pLrRqL7Thc2dis+hjdCjkt0n1NSqmdWtnde
wo9m1ptu4qty7msz/sMPXtMbZtsjau8zcsd/Aoc//o9D/rtb/sNf/s9j/vN3/t0HDueLGveTKw7CdxrShyLakz
Lur/5uD/5+I/6GK/6qV/62Z/7Ge/7Si/7yrw5z/xqD/yab/za3/z7D/0bT/1rv/2L7/1ci6/8Ky/8e5/8q9wM/
fwNdf2s7C3dLH3tPIxNLhydbjzNj1wN7/2sP/3sn/0Nvn093ow0XMxujSy0jS00vZ20/fx0H/yuT/1uDq2+Ts0
0b+10n+2uz/3fHk/83B/9XL/9j0/93V4c7/5NP/6dv/6uPc/+Pc40fv7+r150vx40//7eH/603z6/bv4/D+7PH
06vT+8e3o/+fh/+nk/+3p80b/8+z/9PHu9PTz8/X69Pr29fn9+/b0+PT//j3/v7+AAAA//LxzPJgAABiTJR
EFUeNrt2I1T2mYcB3D30rPacU5bWZ3arU5Bypg10z2x1L4ggpuXZJReXWBxTDdCTBQ1VJcLBVo1F5Lq70/uA1y
3219A3H3z5HnJ7w13nzs5wpPLAF3HbQBqqKGGGuprpj6uEFVYpmzuKEt0dkYV0nqf0e/0nWSPz1hV6Z2ZybDi6
t/jPqrfJ5PHEx0HL9KUaeePyvJDM01zyuZVPrZ+xEqk0snN55lcr1U1jLZykQ61U1Rrv/qyuHJxvphLk3H68k
X6eNcNp0lw6NnuVQm+e7k3deHbGNXsp5LUXoj+zzDuohdgU/n9dm5r+e1zRLZdrfV71VsNy4to9dvVD/Gup2yL
9SXoi3yF6IgGiTLoshXt3ib52nHqskCb9cMEm1Br1YtWaZNqtFWzRIMoSbu+qzeF0/zhsjL1lk/b4nVTUOW2zz
VLFvg2/LmDokk3BdE8dva1ualyMb6J/4bEnijzzPkqmqyZVkyG72qcSkbtUp1lhTFrdE26qx02AbVcHgSb4QL
doRa4JMLmz5dpVp/906tH32NHqJvTfubsylg71wouls4s0fdBBLRdyXdYZ2ySWPVmN/uuSQ6x+10zgbGlydC0f
c0GwkEg5F5mZdirqRUCQco8Hd0KfRz79/FH1wIzQfmXf9o15NRGM/hsNRGjwIReajj0JMHXsQm4vMx4i1SGI+s
Rr7LDS7t3rD8f3b01/+fnvKs0T1YrdZdn5nVZ1IpBjTzPdOKBRMoiaL1YusNAtqJ9gsqn1VLxQkiSSu61KR8py
0KEkPOVOSSDXb5TxXeC1RgaQCJ6mLXkHK56mc3P5hWJ+xuurrWiJpgVnLp8xXmJFMTpWKK10fqpxaZNdDU1PiH
krcjCMVpEU24tJ3HPvNy/70EJNmp1k3PbPzN+jRad0sU3evM3S7zrpVaktc22y3qWmSp1K9rXaCfVUzaTezRaV
Teh9jvWXmP/OgF/b88DSurLT0Ukn3WooS11paSY9TvBVv6A2dVkhbbumNh1ZSPF1r0UpDa2j6D/5Q7y8rbKNSv
BRXtE5LIVZprPYUpbS0oi3FWa100kvLLOQH9b7e0jRWDXqf0WG0udSg31N03TidHZ60sNRVdaee4oWkNxF0
DeuSxQ60v3zrB7bEr9+3a07VutM5ykJwrc7T9pXvVpnXAzenp28FpmhoeHqoMH5n6Ndxt16NTAbp17Gbd4bvj
g7fDgw9CTh+U9+dDgaDX2zT9NTU15NPJ8fo70++GhfpnsjwbHx8cDoZPD2k8C9a/AVZ+1afntryqUa3/mghhp
qqKGGGmqooYYaaqihhhpqqKGGGmqooYYaaqihhhpqqKGGGmqooYYaaqihhhpqqKGGGmqooYYaaqihhhpqqKGGGmqooYY
aaqihhhpqqKGGGmqooYYaaqihhhpqqKGGGmqooYYaaqihhhpqqKGGGmqooYYaaqihhhpqqKGGGmqooYb6f6T+A
Di3IN/f+8NVAAAAAE1FTkSuQmCCUEsDBBQACAgIACZnuUoAAAAAAAAAAAAAAALAAAAY29udGVudC54bWzNWVt
zmzgUft9fwdCZfSPYJnEDG6cPm+10Z9LOTp09vCogshaFxErCdv59j8BqQYGocezZFYfof0eqc80++bDLqbPBQ
hLOVu78YUy6mMU8ISxbuX88fvSu3Q+3P93wNCUxjhIelz1myos5U/DXAW4mo5q6ckvBIo4kkRFD0ZaRi1NeYNZ
wRSY6qnTVJ1I9U2v2CmxyK7xTtswa2+FFT/aaK7DjNQi0tWXWWAiqyZ5yW+adpF7KIep5gRTpWbGjhP27ctdKF
ZHvb7fb121wwUXmz8Mw9Ctqa3Dc4opS0AqVxD6mWCuT/vxi7jfyHCTka5/GmjaxMn/Cwjo0SKHvb1VuMuuM2GQ
joYnXSFjnRgXuXm+Q2F9vkJi80VLrkTu59j8Dsfr4fH/IBZHb6tLYTqhiQQprN2u0yc85b03VDHWBVuYuZrNLv
"

```

3420NtJ+FYQhYUBjyfhMaJxG3GeDwUNcHMfEB7e6Drte18HQo4wLPya3IJ1Mir678/3D/Ea5+gAJi+DpcKkQuwQGaEvYdTtk1/gggvVBia1b5hwW4vWtrXK6Xi5a2oDzUSSDELBnMCH0ofC8zYEb9+5nU4+nRBhLyGqtvgSSwUy+ +Ykw3zma0xbxpAihYvsnY0pbxk4ATMn0A8a7AgmgSohVb1JFgZj3lrxC5n12GhE5Dj5g2HaN1aVAM514uIYG gUHgRGdzd+SDynZ04XRw8SfsSe40i1jJQQ/nw+NXXNE+PRxgAe03GWrBwb5sdo04f0m8PUtgFvBTf2EtTwTOxtT d3L220nftZ2r9x7vibKucMbxFCGBJm7DrTvBpwT+rxyf0Yf17/0kfWp60/Lv8P/oD9L5wEx0SrZw0yF0obIgqg YWvcGVY5a6kQb iN9UaMGHa/y08B4xEGD9GfxX0j+VMoqPM79qEsd1PE+xWsecwprxIBCk/ZqBfcENpKqr1+4q z5wwLeawcswAwBMN8FzxN7KuKnM7eEsTJNbIo/Khn4Z/mi9Thr3LBX0j7HugZcCjj6h0iD0r1+crzgrKRKjdo6 zvJnF/lhn3J+jukHGKBJ7lZy2ZVafHf9+n7e69k4USKBMoGLdE0BAv5FVD94+KrCWJegkb iNY93WvgMGAhSJY0 u2Yj6DDw8SCvQZdPs1S16C0irweRjto2Dth/0Ikxhux7/fDM/kVnMevXiM+k30XJ3Eu5RH0cA7vh0/SNAhmM/f /4u/Vqf1dLmezNB3wt70rnMnb5am9hWU/DIIBb89epe9P7WoQLJdh0HGx9Z5xJnevT+9uGKbn8yc8fV209Cfj3 3xbX/3R+b8nPPhkuX3QphsvWSLX1sML3YE7hXdWuXJTRCW8W9fEooB9JcGSZMzLeYIbMhhTffkp8X81ZoeV5Pt DpzpKiCwoevZ4qShh2KN4gyk459bk01K/UVpKVW+kwxp+QNhjsz4dJwX+PVrIXf11bXWD41ErahYz/2B63n7C1 HLnLy5o4swXxc7pzLW9wGJcxWADGNWWEhYDkho+q8F+/sJA3RPsxBxPSCi/2Z1ISYcENOWpwV/YK7jdUH5nVp rnvq/bNx+A1BLBwi3Yk211AQABoZAABQSwMEFAAICAgAJme5SgAAAAAAAAAAAAAAoAAABzdHlsZXMu6G1s7 Vpfj9s2En+/T2Go6L3Rsry7zdoXpw8pihTY5IA2vdeC1iLDSUKJGwv8+k7JEWJkiVbWW+B05zzsIE4fzjz48x wSPrtj885m+2JkJQXmyCaL4IZKWKe0GK3CX7//DN6DH5894+3PE1pTNYJj6ucFAPJdWREzkc4kGtL3ASVKNyCs yrXBc6JXKt4zUtsOKG1z702U9kRo2yqUH2pRV5V10FNW9HFm+nz2yYfe1e4MNUYc0LmPrkZ8q/CwZSjmKeV5 iRXtWPDNafNkEmVL10gwPh8P8cDfnYhdGq9UqNNTG4LjhKyvBDFcSh4QRPZkMo3kU0t6cKDzVPs3rm1RU+ZaIy dBghU9WVe53kyNivxuBjs6wmBwbhmr7vHfJ90W9S3zZHktsZE0ew49ANH8+PrWxIPKpc2neD1Sxo0VkNy23L88 5b0zVAjZBjbnLxeI+tN8e9+Es+0FQRYTHHpr91jzGL8R5PqQa8EUhcCcY12HquIV2e1TzQyhIyYVqDEmnFyhAZ 9mkV6ZyNp5emupYdyJJBlnBnLsQUg0CHe0p0XwXdcrn+QVY9RbA1KFLiobJr1NnBaJFqHmatIE1aYuq2DV1P+V VAU7AV1EDSJ5LIqgmYWbE1h0NfpQx/gKV9V7haegUUeqYy9DGpUE1nKncI1pAYPJy7U13qoWUd2po+T7/Gmoa0 rsH1Md6Hm/TXAbv3A6ZctgdUxwT1JCYyXdbvWVrhmf2Wxu5CZ54RtXsJ7LHBd5hQaNgBsXMMeeUHTfBP3HJ5b/ 6nHY0CM/r/4n8if9TzX7DhRzV7PHUSmeeypKqGArZhQY0ETVxQoA+vTijZrp+yp8FISM0eqQXq/9E8201DTyzp 1GX01zX+xRnPOYMNtWBCX3aiy4orA/m6y7sFZ9xgHfrADakQIYoNYLnuPitYw7F7k9vgmmyQOVV0VDPw2/NV/ PGneUiuTXWPcbrwQMfcAwkNn7T7NfyA5iWIza0S7yahayH5WxHrdHcedZQ1jcsfqA4TTXJu8ELjMaB463/kY1F HUiFIUDiXZTKsG/EGgJGYce9Lu7+x8e8H0w0zscFH3GGsqb5SqnIfZtvj6AKsRLZQp6wZH+rkvkhN+QGctJAo 9b4LF/C70B4nHH1FBR4mgASdI1j1g9h91XNCvXG9KmnX5eJZ5r52KT1mhY5mq9YR1QGcNMgM/D1R1yB6oUsykF 3E1Ftjg3UHbkDQ/wpXieg4IQ5oQb1kxK70mThkztoJgOKzAEtFY0YruGLVtOU9AnAmktp2gokVcdJumD56+M85 IZyM0PRA3vJQ66sbNbt13SfeVJIApVeVTN5HSxKVkrj1FghhWAdEm/Aj1alsqMMVzsKryDoawyAzE0PkpAx Hz4vUGIKGhQ0BciCuPd4JwI+ldcXEj2Vksb4UsixfyhbFB3Fjnj18xRatMc4f2nUyP0eYuR58GK2Ju8YbVY9CZ vqBntT9+QfvkUtCvaKRBTqkYTAshZWIY14QLCtZcbYg1rAoUgaL6yY5mRwiw1YjhJYKGmoaZcMJrTxreJ+VBWR awqq1CXG4AAQIGFv5wwLbRQqFsFH0SaEIfojabuy1VAtJtKt/i/n8n7v3II67e9cNRkBzTAukLDBeTpzFbVjL rsVyRVPZ86RveRvyQsld1Wy50mugYhP0GAorhUuoAv3ZiJPihNzmM9LL5CyElUnxHVkbvOnRCXprYn7BuqSC9E iySYLSouMVjWEKt0LnVZtqpgv8EJ16Kj6qDgeaaFQ2bUug09hk+w8Afy8UfW54ch8y6VP5yLKD8AGS1bg7u16Y 5aMe3XC19PwN9Q7TUJHsejyHg4f8Ks/623CyC6SoK01VgdsBHeakWjRQacxhwdebvPwv0S9p9m8YvrUq1EbXQ1 S24UzTWiDt669JYhRz0bVqxq01qmF94AGoVjHrTcIz70xZd7Vgb7hfSfCh1Yg8tGT56yTXzydek7ouzcjQj799 MzUjTVWSE7jK1t6bF99NReoJu4yXunyLMTKucUAw6wbr8pmCNxh6tk5F5j83R8RVjA6AQ+BtLd12Xrynz5oHMP t+Yo5bsUkzwuNedXxgZH2kG7zIwGK4YTqtwlp+8yBazmjog/LcFwbTGzr0xoddWTo6iX+CI+/yKMUSNvrMx9Er r/d+XqcZ8X1l7X3AC9b8tJegxMrInrGa3A0gBcKRpuqoc6fc3DMWzQVYXsVq0D61P4pKaThzWyuQdcRceeAtkC JVdod8rhtT2WGrdzJCFtpofSIK2R1t0oR/24Bi0Dahjbk8EZhkDpAvI3Q2QLiD3N0C6gDzcA0kC8sMNkC4gb26 AdAF5vAHSBWR1A6TxmC3+jxDpknyYCq6IhFa5S0muqu/BGgKqjwMp50p/DyEW1Q2/fYXcY1YR3fDbQSc0G6TqJ wVfxp4S9JuD1ud+3qP9nW4hKZIxAm+mwgU69RqS1Ygia0T00fb51V56rVfuCMoR0raRFgZFU1TRaxML8jFlv9N6 zt9HWvnbra2jQSWpkC050toPVxkdY3c5B5W0Z8EAU++0bSgHmuhfHS6j+WIR1TcBhuCue5ar+ZtRD+spAEcFu KD611P1Un0hBKYq6N84RM03Db1hjdHj0Kjvn8ZerHqPwDb+UI6fG1/0XVj7c4iaQZLSqbNYL0aL6LGdxL2XoS0 Bzw2/5okW0QAPTvWr1BBLm3+bQHJGm4M3Tv6spLJRYGPDjgtIYrc6D9+3Dz72dwgL8y/wX62H1t15mxGsn1XMR +hD4A2eKmoj8jQeA0K0Za0jma0e1Jr0voz4Nnuh62VCT304/Jvyd38BUEsHCLzDyaM6CAAky4AAFBLAwQUAAg ICAAmZ71KAAAAAAAAAAAAAAACAAAG1ldGEueG1sjZPNjpswEMfvfQpE9wrmM4BFWKmHPW3VSk21qpf1sWeJW 2Ij2yzpI/U5+mI1XynZ5NCj//0bmf8MQ/14PjX0KyjNpd16oR+4DggqGRf11v26e/Jy97F6V8qXF04BM0m7Ewj jncAQx6YKjafQ1u2UwJJorrEgJ9DYUCxbEesKxtN4bDQp54aLn1v3aEyLeer73u9jX6oahUVRoDG6oIxeuLZTz UgxiqCBoYNGoR+ihR0c/q+pgV1bk1JeGg34ZHpsFwVBqg3QteKsebeAJaNkXVIDPFoFtVXWcef7XwyK2W7Q4

2qnI0wwU3nDQeVUCMVNUTMX9+K+e7lo0p0V1kShxfrRnu0IVBWHmBakXpbswwUmBg9zfRHEQREKSz3WuM0pG8
Z3UNMNh7GdFkWabIi9KtGADf9/kKjBZA2YNi9pjnRobVp934cc4/TL7uA1fZ9FftAFdRW/oWZ7Yy31qY0tow6k
z6oYcGvCo7ITZuvYrT/s7kfpGLIcfQM1btV2R4UVTpFakPS6BzRzopWKLfhWzSI8WpbwUpUy2FBJ29f2RG9Ato
bbNDZnkLpoHrEGAG1f6zA8KPo13g1I/8jM/enjmojvvv+Wb/SZxVsC+VXIYyx7kw4eON8xbtvivXomujhDd++G
rv1BLBwh4q1Px5wEAAC4EAABQSwMEFAACAgAJme5SgAAAAAAAAAAAAAAwAAABzXZR0aW5ncy54bW9W1t32
jgQft9fkCn7SkJC2nASegwpbRoSOECbs30T9gBaZI2PJF49zuSTZoSvE1ttE8hvsxIM983N/nq4zoWR0+gNEd
5XTt9d1T7AhliX0X8uvZt0jv+UPvY/usKZzMeQivCMI1BmmMNxtAj+ohe17qV3b6upUq2kGmuW5LFoFsmgGEc
vta6+XTLacsu7IWXC6vawtjk1a9vlqt3q303qGa108vLy/r7u720RDLjM/fqip7+qUqRHxWZF/IFuOUNU50zuv
Z/7WjfJEvTN0otbd22G6/fZUryP4ccw0xtc1Rftku7bpGK1tPHFbPVqvte+/Xd77T84ECNsGktr1jNgndESjnt
fbJvF21iLeL7cPM+JD7yC0z2Cu40Tg5b1YT/gX4fLF/2afn56en5aSPF7gaQUQYg+6CytnoHQ1TRAfM1tpGpVB
0x63sKFxpuMcIiqtTPmNBvFn8cs+SYwjWEL021n6AuXeIGmrzNpFrjtL1UZxa2yL5UZ5VxZi7/1Z87K82CKmn
DUbJZHxnWs+FXB4rjixB2e2kzoqpIgl4Fk12R00BuMi/p2U9N0PxHhConbBtkBlyHsVcMY2mJouijSWu5z0pZc
Gx1Z6B3F5MFk/tkuPhQZVwdqb70sGIz0GAaGBqKfoQom177n4MrgU3c7j1f4HKDe+PZtmF1LFDOXmP0mrQ4pgZ
qhQJ2QAD05z8m3AGDIviSRbP8nusHA5V5jK3SB9KCvDjGei8LWHQEo0zn176JMjvGQkdPI/K5Yse0hr+S7M+vR
xj60fXpMzz8D1NwM1AdVTGI/BpLvZ5mBwn8DaDAULYYEigt1odQg1QZK1zZApNrdevWdq2UMVmzPBhzSegr0Jp
82NwHYAUmiCvInbDog4bSd0Zce5H8i00Q9onEhRisIH6fKRR/r/EfyjhmQnjQ8zPOWF9be90otnLR/vDKuI
nCURDHppUFV07goaeQGYMo2LqAWPnE9avskptUtLp2ZLEB96HMLGG003f0EsJ81M4MpTpJ1YM/kK420yvICv0
PXCQcu5Lkv0EFFRItRelo4sLH7RAIuGfIoTs+mCEJaKhzFyHDCbGyw2j5sWxIF25Y796WF98ws0kyD7f4Dwef
Szo40v7UgymrtwWxGxaoPLNzqPpuCuMnHX77SIahKVckgNdbSfUqq4m80ItI+E/jtXKKCH1fa9EnrLbUf0txKn
yq/JREzYEuuCcSJ0n+EakuHCq61x6ArYISr0wAfVaNLGxFEN4wyIzGvEqQyJ90KdSAjsopcaio0rb+7TISpcD2
GD09rGIgoJ8U90HqLm40qGJau/UwHFVXW44SFfmA7Zk/wWeCUPQcQMuDSE7T0CKxXnmCCGSUPT5GR5rsVO/mp1
j790GxeND5U6b3z2cFo0vcVY12a/c/esmorRdhVlneyeYtMXTuyWtjPwsYHIdQIuROWECMzYMOUBRj2booLUYBb
WD68gCyI+E9Iecx1+G90FqQgpDnYxTqhxsidmxSPdkmMdG60+Z6dx9kVqH0MXe10jE+P5wnDBj57WmkjrRskF
xy1LPh8cUwWpXrY5rGyteEdKBlozuQwldTbesyJdjADajD9h2LkEDW3mrwhm+BgeMiE59p6nE6NxXdPbKjICAn
FEAXuP3+Ezbk6Qcdc23R5aFFJ16c1oVEy0QcW+YkNqVKE8m2tZ/+0MVvh+YovEAJXjkfUV3eZDKFw8FSFp2gIw
hbOPoli61R/ZZ0z0u/mZZWyV7aL7RAw8dRuyTxpCeEHKCS8/gaq1YrTwJ3eZHUpKX0e0utb2efaR0Gtf3uQA47
C5R1sdtUQcS7001wytXnb8B+TF8gaL5St5p8bYQ8kR0kV908gDCR1AAr/PsxFikaIu57Iyvv3jbPTZtnvQmGg
H7AG5ixVPjwt02psrqi+Myr50Ld7IrgOzaYbNPu4d18gw9ovqba8NnGakk/cr04ZzJlwteAzqkcAWV2S8XAGGX
be6qY7ezDRwH4a7qigjZmMtpTzDrA1Sx1+8iiEUuXl0IV1Q9lsy5L7CGGjf7ERT2QtqP83wxW7dwS1N4Dyz+Mg
vY0J9DPvcm2cntU9Ki6s1VjWVCVOHnC/VXnwfwIz6cbP8LUEsHCCh81F80BgAAeikAAFBLaWQUAAIAAmZ71
KAAAAAAAIAAAHAAAENvbmZpZ3VYXRpB25zMi9hY2N1bGVyYXRvc19QSwMEFAACAAJme5SgAAAAAA
AAAAAAABoAAABDb25maWd1cmF0aW9uczIvdG9vbHBhbmVs1BLAwQUAAIAAmZ71KAAAAAAAAGAA
AAEnvbmZpZ3VYXRpB25zMi9mbG9hdGVyL1BLAwQUAAIAAmZ71KAAAAAAAAGAAAENvbmZpZ3VY
XRpb25zMi9tZW51YmFyL1BLAwQUAAIAAmZ71KAAAAAAAAGAAAENvbmZpZ3VYXRpB25zMi9pbWF
nZXMvQm10bWFwcy9QSwMEFAACAAJme5SgAAAAAAAABoAAABDb25maWd1cmF0aW9uczIvcG9wdXBtZ
W51L1BLAwQUAAIAAmZ71KAAAAAAAAGAAAENvbmZpZ3VYXRpB25zMi9wcm9ncmVz2Jhc19QSwM
EFAACAAJme5SgAAAAAAAABgAAABDb25maWd1cmF0aW9uczIvdG9vbGJhc19QSwMEFAACAAJme5S
gAAAAAAAABoAAABDb25maWd1cmF0aW9uczIvc3RhdHVzYmFyL1BLAwQUAAgICAAmZ71KAAAAAAA
AAAAADAAAAG1hbmlmZXN0LnJkZs2TzW6DMBCE7zyFZc7YQC8FBXIoyrlqn8A1hlgFL/KaEt6+jpNWUaS6p/U4
65GM9+0tJvtYRzIi7KowVQ0Yyklykhotekr0rsuuaXb0trYtisfmh3xao0lnyq6d24q0V+WhS03DGzPs6IoeJr
zPE+8IsHVOHFIIDMa0jggJHo1CafXkfBo5zuIJZldRd0ugkHn3ID2L3TqpoLIKYbZSVYe2IJGBQI0JTMqEdIMcu
k5LxT0W81E5waHt4sdgvd0Dojxg8Cu0z9jeiAym5V7gvbDuXIPffJvoeu5jenxtxfHfI5RgnDLuT+q703n/5/4
uz/8Z4q+0dkRsQM6jZ/qQ57TyH1Vhr1BLBwi092jSBQEAAIMDAABQSwMEFAACAgAJme5SgAAAAAAA
BUAAABNRVRBLU10Ri9tYW5pZmVzdC54bWytk0tqwzAQhvc5hdG2WGqzKsJ0FoGeID2AKo8dgTQy1ijEt69skti
1GGLwbp7/fBpGxfHmbHaFLhiPJfvgywD1L4y2JTs+/yVf7LjYVc4haaGQPjhzKkPw9MtWexQehVMkGcBEla+
haw8jo6QJJ/6+U46enNAPbssMumebWxkKf+rp+q62ht3iq61EwsuUxhB5VR0fUtlEY1rTVaUSoTV6z4CMznnJz
gRkysYThfovtBZwQ9DB5i80Cg3GqATHkV03RHmngS3tcEB7IxZBepRuotxA213VAantWIEqHuT3tyWntmtiN1
xH24sXzCREHFB4N130F1au6x3hX1S8MT1Vv9wcW4t+/PPwCUEsHCJu20BwMAQAA0gMAAFBLAQIUABQAAgAACZ
nuUpexjIMJwAACcAAAAIAAAAAAAAAAAABtaW1ldHlwZVBLAQIUABQAAgAACZnuUpJyikWiQYAA
IkGAAAYAAAAAAAAAAAAA0AAABuAHvtYm5haWxL3RodW1ibmFpbC5wbmdQSwECFAAUAAgICAAmZ71Kt2J
NpZQEAAAaGQAACwAAAAAAAAAAAAAMBwAAy29udGVudC54bWxQSwECFAAUAAgICAAmZ71KvMPJozoIAACTL

gAACgAAAAAAAAAAADZCwAAC3R5bGVzLnhbtFBLaQIUABQACAgIACZnuUp4q1Px5wEAAC4EAAAIAAAAAA
AAAAAAAAAAEsUAAbtZXRhLnhbtFBLaQIUABQACAgIACZnuUoofJRFNAYAHopAAAMAAAAAAGgWA
ABzZXRo0W5ncy54bWxQSwECFAUAAAIAAmZ71KAaaaaaaaaaaaaAAHaaaaaaaaaaaaADWHAQQ29uZml
ndXJhdGlvbMyL2FjY2VsZXJhdG9yL1BLAQIUABQAAgAACZnuUoAAAAAAAAAAaAAAAAAAAAAAAAA
BAAdAABDb25maWd1cmF0aW9uczIvdG9vbHBbmVsL1BLAQIUABQAAgAACZnuUoAAAAAAAAAAAYAAAAAA
AAAAAAAAAAEgdAABDb25maWd1cmF0aW9uczIvZmxvYXR1ci9QSwECFAUAAAIAAmZ71KAaaaaaaaaaaaaAA
AAAAAAAAAAAB+HQAAQ29uZmlndXJhdGlvbMyL211bnViYXIVUEsBAhQAFAAACAAJme5SgAAAAAAA
AAAAAAAB8AAAAAAAAAAAtB0AAENvbmZpZ3VYXRpb25zMi9pbWFnZXMvQm10bWFwcy9QSwECFAUAAAIA
AAmZ71KAaaaaaaaaaaaaAGgAAAAAAAAAAADxHQAAQ29uZmlndXJhdGlvbMyL3BvcHVwbWudS9QSwE
CFAUAAAIAAmZ71KAaaaaaaaaaaaaAAHaaaaaaaaaaaaApHgAAQ29uZmlndXJhdGlvbMyL3Byb2dyZ
XNzYmFyL1BLAQIUABQAAgAACZnuUoAAAAAAAAAAAYAAAAAAAAAGMeAABDb25maWd1cmF0aW9
uczIvdG9vbGJhcI9QSwECFAUAAAIAAmZ71KAaaaaaaaaaaaaAGgAAAAAAAAAAACZHgAAQ29uZmlnd
XJhdGlvbMyL3N0YXR1c2JhcI9QSwECFAUAAAICAAmZ71KtPdo0gUBAACDAwAADAAAAAAADRHgA
AbWFuaWZlc3QucmRmUEsBAhQAFAAICAgAJme5SpU20BwMAQAA0gMAABUAAAAAAAAAAECAAAE1FVEtS
U5GL21hbmlmZxN0LnhbtFBLBQYAAAAEQARAGUEABfIQAAAAKZ5kc3RyZWFtCmVuZG9iagoKNSAwIG9iago
5NjkwCmVuZG9iagoKNyAwIG9iago8PC9MZW5ndGgg0CAwIFIvRmlsdGVyL0ZsYXR1RGVjb2R1L0x1bmd0aDEgM
TE3NzY+PgpzdHJ1YW0KeJzlem1UXNeR4K37Xn833a+/Hh9N06/1aL66oRENEkh8PBA0LSFLLQQKICMaQSPQByB
AKLbjCNv6sJC1ZmLf1iNNonidbGJr143kZOU4G5NznMzJcZw4ux5PMrEdkvFMJicmVhwnm7UttHVfN1jy500cn
T1nf2yr37t161bVvbeqb1XdR1MTRxLETGYIR5SBQ/3jS111BSHk+4SAfWB6SrrZpEgILxJCxaHxfYeKwj/9DSH
cHwnRafYdvGuoZM/9DYSYkCUzfjzRP9j44b1qTbK12LdMCJ2L9+lw/6d2M8fPjT1ya9pn41ifwb7HQfHBvq/X
rutHPuvYj94qP+T41/RzvGEyNg10mj/ocS2xBPnsI9r0G8dH5ucGiT5NwkJHGTj4x0J8d/3PKbh/k04vinEaf5
jHzOCWtanHK/R6vQGo8mcYbEKNrvDSf5/+mj0EheJauqI1Yyr79s+3GWSzdqbb9/+xt568/3/m6vQp5rz5MvkW
XKw/IT0pgc1JEZGyBHE3Pr5NvkRytKnRnrIU2T2z4i9TK7heIouTh4mj/8Zuhh5jFwlf3fbLDFyiNyDa/ka+Qm
sJd9DVxkj74Ke3Ee+g1LfRdwdf0oUteBrSAWHbsH+1FygZ8gW+hZ2HmcjNEQF8iK5CHtQ8hTu8+zqjmv/jdBt5
F587yTDZBph9a0p+/AfieHm73BX95It5H7SSA7ewvFN+AJnRpt1kC+gTr+t4kIrg7oot59+ndIbj2Dnb8g+fPo
B907Pco2kWWODZwLRWrq70jt2tu+Ibd92x9a2LZujrZGW5k1NjUpDfV3txg011evXVa0tD5WVBosKC/z58hqn
8tpE6yWDJPRoNdpNTxHgQRb5EhcShbEk3yBHI2Wsr7cj4j+WxDxpISoy000SSmukkm3UypIOfQxSiVFqaxSgiD
VkrSoNQis8mXm2XpGvTs6EL4bLPcLSWXVPg0FeYL1E4Gdnw+5JBasoabpSTEpZzKZhP4tixejPLmTcZN8qaEs
TRI5o0mBE0IJYvk8XkoqgcVoEUtG+Yp0WewaZ0cv6V/MBnb0dXS7Pb5ukuDm5MWuVkdIptUkUntpqROFSmNsKW
TM9J8cGH2oWsC2RsPmAf1wf47u5JcP/L0ci2zs6eStkCyWG50Ft/9VhbuPJEMs0tyQCT2ta+0k/bR1NCUuMXZ
Gn29wS3Iy+9fTumP43R+oXfEwZGUL2zsxFziszGZ/uv3ZzK0uCPDtvs+0t6CGSawLua7d/MYzdzLyUHdSiA/
DhvRmI+1tSce03V1J6o9Iw/2IwW+D7Kt2+2zdKzSxPzdMUBGoDtSpz8c2fuaaQvZiJzmzoyvV18he9xWihALdS
RpnIwsrI650NjKzMrLKhPfRmm07u2aTvH/z0NyC0j7Tn5zZi/60n51CFpKWP7h98qzdjtWEulVaCve1eXBESmo
KUC3IdSsDegpjmRXUjuUPqWbJjRMU20xSjYximJwWuSWe/k4PZ6EAqTSYjAZSpU/oSirNCCj9aRu1zJeHkKM/j
iYaaVbNlwzJ40mn3LRqT7as1pGdXSpLmi3p3JQk8YE0VzLU0sxmllpm482pJTBZ8o6u50j45uJ8peS+GiaVpLu
ZEYub0K8KWma7BoeS3rh7EE/akNT19iWVbjRwt9yV66a0hhoqXsTpf0qMSbqpo6tt9y2o6er0r2Q1AATx/tbP
iZG7nKnxKDLJfV+vdRF3Vw3EgqIkCIIyE21+E7q/Hp8BFS4imWu21QrdYGrFDjMpLFUkuioU3H+rcJ1TB32hR
dkaZ1XZsKer2dftSn9IgxWEpPTFy6J1SoytDnB8jAeIoiLFTJdZz0e11jkhd8vDU1KJdbG9MfWoWk4rQ9V52
1Ydt/VuURaqifhweKXD1JmMBNy3KjfZqvZXu9GPDW9eGZZm9XLbz1kmXE4LJLjyzUnCXFiptrnV08/0sxzpx00
MJ1o9z7PzisL08jA7trPy5sFZeWdXrUqNEeRe991sLjtpg7a0ptIgBr0meRke3DGvWIM7e7qeE7CkerCj6woFu
ine1D2fj2Ndz0mYK1QsZViGZB2JdZikduzoVx3cwohM+ooryLU/sA1ICp0v4IDMnCNpnDCCo4ijk/hFBXHPmi
1rGHUMcbvFmmQ2edT3c0z8W7m40REjeAXkiDXo3bk+nmqWnPSKCeakia5ieEbGL4hhdcvyA49A0QoDd49K7T1v
88qVVM3acbXoKYTK2AdKZsHEqq9ouP1SxXzW3s3tVc4iCZ5xhaw9BXdFrDh7Vxg0HDNp/N77P5mqm0nA/n14c
1ne8/3cy/TFgluhGz+BmsudxkTinWu0DvBH0G6M2gN4o9u21QaWu2UZun2TPoedHD5XsqEeQ8124uXN1QF2tU
1ZYE130AI15xj2XPK94eEN2T4wuoimy4gYDFUx6ri9Dw2WLNtrnJhbSULFUUVHREGZve01oCUK9SxW9fb2BwOE
J4fVe7Kwt390Ln3DVelthlc+lw/Wn2kyXD3E+bDdevkyzL90j12nW5cs3/vXyjd0Xq9z8530qqnJu/JZaWfvBX
ndV1ZtW3XjJXZXSyezm21yE+w7xkhIypZSedsJ5B5gcZxxUdBe4qSEr06s46/Esx18Q9ZpM3iaJQv1M8FLwepA
Lsh1v2hJ1rZJZUhb1Q/RBEUQS8/u1Uixb006wiaRhqWFpdUu4nZ29wg8qQsJSek094HLm0XBFPV3vsnDymjJaV
bkuXJFHPQBrLNTlqyyjwLsaJnrzmpqrczIbt3WVHvn1YPAHL7Q9sLdm+bHqHVXZ8B1bIAo/sW8+ua90ozdqq61
uMUP59Dfu+s07RXs+P900F0077tm69Z5daqEFpB93LmB9bSCn1CJNa4CAicCG3eQA1pUXC08mu8nz5CXCs97T6
F2mF01gYpttiERZq+RWb4z0mYASK2CKmS6ZkqYFk3Y0gesmTiUsLU8Rmj0EqDZmQH+KcS5URUXDEgQCvYf7JgI

BZtjDE3t6D6Mq/CkrokVh7IvL2ZcuQSSSXVqaraErvr4N7XQF7WRCSz2k1B43njNSjRH06C/oqVEPZ/gLPDXwc
Jyeo1RL0VmjjxEd8ko8KvnJfzLfo41LP8XEbfWxhYu0W6Bd8M04DxRf3zfgu+fi4D9Qhi78sKka1tphBcMe41AE
htISrnpgI4HInhDdXjcf56TymgLvaKIOjSuzq6VNy1157Rev/vjHr7/2j8/m1A1u3hKvFsXq+JbNg3U58NN3b
pL13/7mw//5u/7HR9avH3m8f+/nDtTUHPHcykb05s91p/EMOukP1Ss6K2Ra4J4M6MqAQAYMayCLggZAz2mg3WQ
29Wg1Tq1WkwHtPMffac5wms0ZFNNotVksPR50c1rvQL1iF0wngHRJ0mkFnLjSvNx8184ZPWPdZ6TpxEp11kxro
ZUzmjmrlTNzWiJ+T4TnRfiqCBdE0C1CgbhLnBa518R/Eemz4osi/bKKvkeEIRE6RIiIYBCzxWKR+4XIFHnwJ29
Gz41PivSkC00i5IuVYrPIiSJQEd4V4S0RXhPhSZGJ4u4ST4sUBRTj+RELcJqT4tdFjUGEp341/i+RisqXvhJ9S
fwpgz73t1GccoTNt0ukBeI6p0TYhFfnHomqE/tPnYk6RdCK8AcRXhWBSfuuyHwl0MawIrJwEyrHgbEUR2hoJPq
cCPeLAOMiDDKut0R6RrwgPi1y42x3gyJVRHCKbBfqbEpwUyR6RgRFjImURyyteY/p6yXWc4rnRM4pTqmb1XUq
AxVzsyowwyIbsKn5RS5KBriGrjth0fmIIMDTjFnRYuxzcwWAQT0fbZ9zzVbGacdiSwz0Vmxcw5zUDZV/VUS1a
00txspmYe3cFBGsL46Q30HT580GALp6JpRSgcQhxDYQTcCDoxxVYXIfAzMdGHTh1I0zPSqAQB/LeHddM07BNyD
fg0XviuLb0mLhQKsQF8TmUJgVOBF/9ks7Yc+1b5DSDjF9TobYCwIQU517dULT+5fKFqufkIJd+BBhgphtuh/Ef
wTf437/+0u/jhoCaDhfAp3NPfLiHm0d4JZadx7iQRQ5c5Y1Ar938sRIyWKMOL4zBMeAADK3E1lgky4L1FcuiR
au3eHP6cqSA7scQw7q4L1o06VgMEcxOwnWmN1qsMTM6WjVEA4Fh0+Fe+HwBG50qSLUi5kogEFbLsRzvpJ9cAe
Z9VYng9syFWUjeIX15u0HgW7ITPW25vPfWd5VJ9hN950YhGNk7JLjzjWBvMwpm3BmPZLjMM5uItPKVVDRdNF9
LweDPrTenqRh7M8mHnQ24ncmhkgAWjFRwnMBBYCnBSIqwAfUMNx0Czqjm7XgCYzluNyxERSGDMKM1ZF09T4FRa
+x4JYg5qDAmrQFV5H51iNYn4LyJKtsh6DlphZxtUxk7CtgRrH1q23LYG8Rkv/R/CB7uVj4f1PjIUnqygGlc9D8
9TyH5e9/ub4xtr9/pLR8I1jEXk9/PzI8w+0mE2mwNpy63tZpe8/l10KL4/MdRdmCvSxesNraDe8yXMnNFFiJPV
KiWC0mWnMPG50mq+beTKqCBogGkUT01zSJDuavUY3Y9QSzSSXpVol52VAYxz0Weo9jK1Dgybw2zRV/jCdBPUne
DiWfwunKmMs0/0R/qo31Xjag/M9rN1K1pBjSqc/DzTzN82merM9115r32rnT1uhhAenkxrGctasyckn+Uo+VfL
j+ZfyF/P5S1+zz/rKscSj1JDpe9g0xC7Yqd1unjyVCzla6puG1NJ618Kh316bvQZVHcajksr1tprQ2nKCuYIdl
GKoqtdUVRawRGEF2VYPqHWX08LrfNzDH76U+0qpxCbf9Mm8jetCdr1p6y07Xn8jEjs6Nz9Irzyy57P3Tc+c673
/QYPVYXwSqd37v/6n9ofuu/fkY7vRn3zLW7kkngUfKcfKLTESuitEtR44bjtno1obHDedM1HOBHotHoo1UUuFU
gGkYqaC1iAQQxivmKt4peJ6hSYFcNsrlCDyua3EJ2DmfMXH632+3FieuyzmEEsKd/AGgcQ4K0uPYfWkLFWAekQ
qKnPVF1vJk2kfc6QPCZY2uFmWMwvz0A+oBwjUvKnV2fJQF+v0HwA9ddU0bSnoemhvuHL4b0fCh8PsJD+5rBylg
2sa92wsP1RQmHq+/kluKLt0vd3Jmtff87XpyeceiJhMzq8v17CcFQplcdU65naX2IqbNr3hp8wH70gDv+QuEzd
8S1kz7XzUSTXuE266P+funM/lcJocyJT0GVGrVY44wH7t5uJVBG3YKhkIGDNmM6gBq2BDeKTPRsoR4BkLx15uB
9FaNLocndNBmiwas8vsRsihRdjS4oATDnBcu/mvykMFxdEtGpjVAKfJQtgF7cjTznh5nZhTkeWVovGabFoXft
yICfHiXIzULCGJ05CjcTjUfoHo7zH6aGp8nvQc9zzJFbjr3re8hgYPh+RDPUsIt/yv0cx1jBspWfKc07F6irx9
So08Grh7vG1CvdKa2Y0hoW74FE81LHXAhlaJpgFs2Tm9GaHK4ezxLQzb1dvzBSsRKPjzDHOiKEyvJSJ1bs9swZ
b5gXhc0jwSv30SqaA8P09vRV4LGrYg4PhU7fmBwj09R7GwCScWlhIPfoFFFqNuSMQUFMUf0747e01valy0ZRyn
nQCcaca404aXd5170+W71v+L4egavn6GDx179deuQ/aDy7/sclVWpoJdyzPYyvAefgbjMju5XdbwNa1/FT67sM
PY901kEXFqNG5dNTA6e3oCzNKeXwwqrWL9gJ7xL7LPmTXvmeHf7HDq3YYsk/bv2z/uv27dk2HHdg4fc00K1NNY
0uUDdNme4edVtgBJDs4WQCBGqe90j5oP2d/1f6W/T27D1F03A6KPWYft3N2ZhZ/UVRtczxqq+Q6Mq00J2ZexJw
PGX1Yp/UZY3rQ67m40c4ifahXSJ1EDPSpQ1tN930s5acTNxrjB2vLMR8DXqIwuDvCXDqH0aM/BP29n8+rb7R/8
M94marV6HM+GLzoIXP5W0omlarFA/iyYyx1EA+5q0wkW0zGc8anjdvyjR8Y6XEjGLNbTc6Ak7Y5dzsv0D9w8qy
30fm083nnr51awanU1EWdXt7r9GLN5Iu5L14UvZe8Se+C159DgHrTdw1zXKrrSLgXUKz08rnxDxWZ3YscyVds
xvFyk6F19UK/carveq1AtRAw65ULLXlUe6ja8Y9trwiUSzMs9nyCkWxKM9mZPe0ExDgf3YrFqk+2LGSt9neCbu
HYM4W8L74sCKYZFS+NctKLVyeI4/msfJjg94UJVGd1C1RQqoqQSDbmScEtYJzQaoE49iZCyaDC8HFoE5SuwtBP
sfU+rMSKGH7d0DdpCQS84s5RqNmh0ewxbBoV+MrS+MQCrCrcTrAYrzFxIL1yJ5eggZNx/ScVTN4qgB3UqGUXV
RSJf9Lf0bs9avq7CXHAzPfurGmdMQAoyvpfdtW3i58uCXDpcPxHsK4PrQmV1+3mDW38ju63/M12WVLicda6uqs
uTAr94++sLJqmMebVX9IY060cx9GzWyyjym+A4UgDszkEktYr1I7LJGvXYS+3UbIcMGwAPHAubHoMtInPqc43
rWrXVM9XQVw1KNSCwttVZyPTgNVqihYXbneAsKFgTi0XmkNxHUUariPdJ15oYgkCtaTD3YH21N2yWcbFYW2L3t
NXyhsUL9QUs77CapnD1113PN0BVKvFgGq6qB4f0wrmcrN6BHymsdIjy8s0azjat6G5tzorb93mzr7ysxZfdUn
5Xv+a6sYzrz2wcVd17sPNAXct7M2DLTd0JFdusdaJGeVt02rrd9dXyjgqX+kpKUiN8d15GWLazmPp46yWH3Sm
4VxxnnzbVrK30eyySeUGlqtt0V5LTzjhgU3NLi3u6nR0srFnHEndTp1hBM4ieP0HG+0GRSDJWfQmawu2w6iFnc
N4R8E11aiLh5/TL8Ta8t7Axq81drkqgYIu8Iu2eYU2c5cFg62xfvuuTfR8A//sLHcv91rXbuxyTmxjz5Swvj3f
99x41hjk1HbaHRaja17aAhz5s+xppDJ158jejRciUGI8hhyrKxitgqtJt0vTVQ0+Z/0A+93+1/0v+Xna570v+e
nfmbFrEB5NOAHpx8W/HjM/TDun/HP+Tk2eNw7JqoS1TrEKM1pncEAQ1LX8EoIv0Jd9Gr1XikmW3MEkzdm8biyC
XG2a5jhwvXG3Yg2E8IFWoIYDd01m+E11khElAdANI1R7pUT9VdtxTu0Jx7R0dH/rqeRv/E8oF7d3Tm1tdW2Y8
t7z98GMxcXMjHc920v+3GoywK0K6dfVq9ib/h+CgmADmN/v8exs0N5J+VJ5zeFG+11zM5IAAbgeYAGCtzKunxE

JwsA1dZVR1tLAE+y51F9cUOMNrBZLKAWWsUjdSa582jQ16euWdtHamD6sW663WU1L1SR8vrFGy4oh4XGsVV71J
cMRevdw0VQuCRPBCE6SB0BAeD1B8EMQinBdg13C1QsxAs4nN0fesA1jnkvhziBa83h8d0raaJMFammlJX73WqC
4XUA4SR9E1MJnt6WdmqlnDpKg7UU16cmce3zfw+Gir0tcGm1zIbgnqjx/a1WBzplF2N0+8M7h1fHN+3cCnT3x
6oK526qsHB660NcrBmdiWA5E1dQPHThwbqKuZ/M9H6j+5v8cHI1/LCvgcxdG9G6J7GkvLqncd6912rLs8x7b8q
ye1Em19W6BxV10wtHH3A/HeRw/UmJ05GS1fbcB9J7mCeKDZ5QmgzBw6z19Bihb8jMyfb88YPoGRkq5WZ5U0a
0y6/Kb8nvyfy4DE5EdSCS68p+V11QGuS3TL9/nUZX1RJ0ZWXjXNPrvCm6BmoUacwJq9GVbaLatd8/kL0ggxT8
nGZqoi1p89Gn5aBsR2X0bcMvAzvyfC8DEy0igrIFJEHGME5mV055hLD0bYV2qf152V6ToaAvJtR0mXKMC/JHIP
ZNqZkzYYPZHgW10gvyZAws1Pqk0gggyUyCDJ5XJMnpHn5KS8KF+X9YIsYXdb5rMyMnJbuVTP8Mq/lxfz0si0
TEu22qPGfpYPWiAVQdi+VeNPBV4M2ZJqu/9rWDlt4PUaVz5ySCg/oig/oy2QqJi2C11yFXrVxzJteJIHmBxSz2
oP3/iicCOI5tLI71rS4WCXDmYY3z//ZeW+TNC19rCpv1fPFRT0r98j9HkbRyMX0z48A++01Jfyifwsf16yy/Wn
++z1v6eeFN/z//vP/6P+z/6k/TyVt1p9Br2x36aRiGfzrfcQj6xSgQf+8u3i75NmjV/RzZyZ0mM1pBinpBtjF1
311D61Nrgfs+z+Ptg48PHzv8T2Yi0AYQJ8kURdiIc4ifJaeRB/yVB8gTkwjT8Bn7D3c8t8nv4R/H+e1771K5W9
w19s/5NQ4XhRe0Dps+oK3KR2vSaKsBEElv18iZtDeFUbC7sWl13fHUPgJTxEyJjoynYY64ydE0zCPN9Kwhlj
IE21YS6wkmYZ15G7yQhrWEyfUpGEDscAdadiEa9i9+j9bymBFfgYZgy+1YQupp06cHXgD9hZoexoGInH2NEyJh
atIwxxZxylpmEea6TSsIbnco21YS/K4K21YR97jXknDellev5iGDSSXfzsNm0i1Rp+GzeR0zYr8DPKm5mIatpB
Pae/eNDZ+18T1vuEpqWigWKooL18vtScGpWj/VFDaPDpQJjUePCipBJPSRGiyMTGdGCyTtm5uam1v7Ni8fZs0M
in1S1MT/Y0JQ/0TB6Sxodv5t47sTUz0T42MjUo7ExMjQ+2JfUc09k80Tg4kRgcTE1Kp9Hgkj/d3JSYmWWdtWfn
6ssqPRj90/FcWgqvfnZ15LzhA5Mio1Fm2s0yK9U81Rqek/tFBqW0VcfvQ0MhAQkUOJCam+pF4bGoY17r/yMTI5
ODIAJttsmx1B5vGJsbH0kuaSkwnpDv6p6YSk20jw1NT4xtCoaNHj5b1p4kHkLzsY0xQ6C+NTd01nhhMTI7sG8W
d1w1PHTq4FRc00okLP6L0iKu5VWuRsVE0zsEUTVCaTCQkJn4S5Q81BnFp4xNj+xMDU2VjE/tCR0c0jIRS8kZG9
4U+Es0kp0f593GTTWQMz+BdZIKmkH1kmEwRiRSRAVKBQUpx3/rEWonCTKIBZT0I0UQoc1kFKnKEGL/w+Ygth9
JmFR7CWwT2E6rvIxyK3I1kRaU1kg6EN50t iF2RKXvx2cKqfuRNkE0YTTBDiBujAz9xfm3Iv9edR42MoL0oz16U
8WMIC/j3Ee04AqZxEacawAxo+osE0hZqq7rL8v4a+07VGhydWQtr0vprYxU/knevyb536eR1073qVKmVNkpyhF
VdidS7FSpYion08WU0tuStXxJ2bcjjMOIT/T3EeUA6rsKeynJ18hPJzW6n7U+IS6gkGVb2Vvkjzv7UB88EJ9
MKxj2mJrW5anfMOFT+1+hQbG1Z742QDZp0Q5g32rwxpbpc8kJZbpkKHkPL/lG8KT8i4qseEaud9SJuyeZkq8xD
619a0hkZVv2ca0nLLH106+X0+F1Hb1Mk5eJscZ1nWMt6V1U+m1z+kzpPS2ji+x1DvCVxbZSp2n7rHEbThCEK3r
o9ZbF8a9/HVrKz19v38v5ybSxURN30445/4zBvi3wIdq8DV9wvAK92weAN+eA0kG3DsA4h9ADPvzr1Lf3u92Pv
M9Reu0+3v9L3zzDtc+TtgfQf0ZE1Yi3f18aXLi1pja3wUx+DbZ/Wqz2/iz8Rueb4dc7yRtQG3tj5o3kGxwri
Xve0JsibwDX+ToneoUFaaF8YXxhZuGVhcWF6wv6mW/NfYv+t2+GvNZver9JvVe3Xz121Yt/Baxf8X6Fxi7EL9C
5i2C96L0Yush97vEy7+0ted7HHi30Lj56/VH1j0FVj2bYIn2fhW0fefgqdPzkzMm5k9zMibkT9JnpF6bpZKzY0
zYa8I62lnizw1mdujDXqeVuqj9SNe/1F0XifYq3D41295R7e1qlvY6wvVODi+WR0Mp51b/6jXEpcy9w0n17LM+
7A5/F2PUYtW73bg9tV38f6W/zoaAt41tmtncbI8XeaGu119rqbQ21/rD1Z63vtGr7WuEL+I08E3khwimR41BEi
eT5Ir1Rd6cYdnUKYwsnBdIYdIZst60Uqu1z3rMyn6wJ3RGBA1cg7n5jp2BQNs13c32tq0+tjsJDyb9091b2dG
T1D6YJJ09u7vmAf5D94mzZ0mTpy1ZsbMrGfd0tyUHEVAYMIOA4JkXSVP350QU+6k2AIEAgkfwTQJHELVnMoUkg
ZVhEpiEyUky0QkBNqaC1CGTAYZmGMYDyLlnkrAXGw2oVayanMza878BOuzSmgp1bmRzdHJ1Yw0KZW5kb2JqCgo
4IDA6b2JqCjc3NzEKZW5kb2JqCgo5IDA6b2JqCjw8L1R5cGUvRm9udER1c2NyaXB0b3IvRm9udE5hbWUvRkFBQ
UFBK0xpYmVYXRpb25TZJpZgovRmxhZ3MgNAovRm9udEJCb3hbLTE3NiAtMzAzIDEwMDUg0TgxXS9JdGFsaWN
BbmdsZSAwCi9Bc2N1bnQg0DkxCi9EZXNjZW50IC0yMTYKL0NhcEhlaWdodCA50DEKL1N0ZW1WIDgwCi9Gb250R
m1sZTIgNyAwIFIKPj4KZW5kb2JqCgoxMCAwIG9iago8PC9MZW5ndGggMzE1L0ZpbHR1ci9Gbf0ZUR1Y29kZT4
+Cn0cmVhbQp4nF2STW6DMBCF95zCyz3QRYZMQEgkhpSRILPqj0h4A7CG1Vix1yILb1z0mrqdQF6BvNm/HjmbisL
7XRc/zqRtnAzHpt1INpvDsJrI0bNpFImNjyxit6y6G1Uexnm2WaYahNP+z5FL/53jS7hW30auzgIYpfnaKnzY1
tPsrG183d2i8YwMyMR0XFpR+z1Nrn9sBYpra1s39bxs/cif4H2xwBKqRbAiRwWTbSW41twgyjkvWF5VRQRG/
es1PIx0vfxsnZcKL+V8fyw8J8SHFHkXuETeB66QU+KEIx+I0ww5I84S5GPQ75FPxDuBfA5Mmsegpz11YPJwCX7
o3GvYc0KuAuOs4IFxpwj+MzxLrP6vyMF/ukNe/R8okPXLMRq8u5/Imbw75+0mC6acMWFtgP3+BHa00EbpN3Qjm
gkKZW5kc3RyZWftCmVuZG9iagoKMTegMCBvYmoKPDwvVH1wZS9Gb250L1N1YnR5cGUvVHJ1ZVR5cGUvQmFzZUZ
vbnQvRkFBQUFBK0xpYmVYXRpb25TZJpZgovRm1yc3RDaGFyIDAKL0xhc3RDaGFyIDIwCi9XaWR0aHnbMzY1I
DcyMia0NDMgMjc3IDUwMCAYNTAg0TQzIDMzMyA1MDAgNTAwIDUwMCA1MDAgNTAwIDYxMCAyNzcgNTAwCjQ0MyA
yNzcgNTAwIDU1NiAzMzMgXQovRm9udER1c2NyaXB0b3IgOSAwIFIKL1RvVW5pY29kZSAxMCawIFIKPj4KZW5kb
2JqCgoxMiaWIG9iago8PC9MZW5ndGggMTMgMCBSL0ZpbHR1ci9Gbf0ZUR1Y29kZS9MZW5ndGgxIDEz0TkyPj4
Kc3RyZWftCnic5Xt51GRXed/d71uqXq3vveq1qqum603P01t1Vc2Mrt3SjGa0gxCSLCAgekYtzcAsYhaEIITNY
kfIZjGWg0jYsTFgGwIBTCdygBEkQMD2AXwi4pjNHLE6BiR15/fdV9IBzDn2Sc7JP+nuqXrLfffd5ft+3+/33Tv

nzpzfYD57Pp0sfPTk+m0Lc7Mtxti/ZYxXbz1xx0/ufxFn8bx+xibv//YxvrNb/j3r1libPFulBkcw4XG1vcFz
j+D823HTp571o/1dBXn38P5JSd0H10/cIo9ibG1Rbp/cv1Zt+0TV+Jw6Tp8tE6tn9w496wv0/1zj0nKbafPnlt
m9hpC1S+l+/jH8Us/BRwa0hdSaWM9PwgLxahUzv/f/RQZ29oSr5M+22BPy6fYwfZM9mzM30vYK9nd7A3sXvZmD
FPELf6MnefZPF/lQzoX09xYMy+yeTEY4hrutfN0uLIq6GxV0hn9Sfq0q7yHq4N5MexNSxu5i3XLB2bYsYN6RjX
HwytdbM4TdJ53k+7y1nSTYedum1yKoHH03gvyqeDYQ1XxKsqs4VCUgxUOp+0V3hR737i7sAkwaWpT0+MRd3Zg
meFEqZorA2ltr6SVtpGVRYL2qtuqxbjehxpXy/vuG7RFoyUWkq5dHguDIU1Qizjodi83Qh1PCHeL7goezIU6kK
j0W8o7uHvISn0E4SIjJae0GLMk3j0oAkDabzDz71EcK9oC/s0XTqG13hZb99gviFF+rCvsc8cVd9e92reBrly
/Wo6I8tNiZ2715u6aIxodFRKVJ+1b9fm9nLZmvddqvoTyxcsafQmmkV5VjRePxyvP1mJdQNwsjwRult3i2M91a
ptdWJl6Xn+fCaCXVKSXQcTmBDgqYvFAsxrw/xH/Kv8BugEcdYcfZbew09jz2YvYK9tvsd9mb2FvY77M/Yh9gH
2IfZZ91X2BfYWy2HgnMXUZm4D6TFBNNc91PM02DZLhMV/gwn/pi2kWL7yQ3idgkzorqmGy9yjMzHKzQ9A5XM0m
DZbIXVGnN0qL9qftzNAdm5uVpaJZn0vqTU5vdE061p1J1R1z5mmK7qynNk0kzAe0kX7cD1yDZrnw6xt0vxzp
WQiaKxdce2uo0pXC0r61khP+SYs1isqtFyXa6ViGcbheFaWXHbdMKau4sU9B+cExnH52sNrrWiqpmFsRQy3Vpv
/WmNUz3MuxKtQnifaN9rTAsMNKxHK8lhyDR0yVERJzvUMZkmFQpCBYga+xnGRozBXnGPxLzVZTyBsjxsTifSV
IvlnUfo/eY+qubd5cm0ZquzDVXxil5hfGqmJpQWwnR607LURFEw/KLkVTSRAGHeepQ18aqc4f3LycKXSrVdvR
rfqp1WAw9rWGKuk0KoU0bZaX5NIylwIc6ULDGQ9IYsQbT4eIo+oe0XQUrLxpYoBCzaCKs7APoiuDzirqExVJDc
Ce4Q8zFPmE2PyGcPXIbj01Gf1k0Lvwunpb6Fk6YvBf2uMmb4n5mgc5/zf6G/S37DvsB+wn70dvEuPi8wqd5h2c
ck8jX+CF+BX8sv54/hR/ht/AT/Bn83/AX8ZfzV/PX8zfyt/P38D/mH+Yf45/k9/Gv8K/xb/Bv8x/wTWFEKKoiE
ZoiLbaLedEXe4BvK4Nhr25iCStNarCrJM0Rj/4k2Z6tmymepINev7u0mCDL1/zm10YWcTj5YTlwBZdmxh7AWF
kD/cHKNwxFl4xydNON3NVE4r1BkM8nybLae4UhJz0DF2EubZtAk8zXbr1rJc+eyjU6zv8paL9HIf7+SnMvJcmd
Wo3NUQmabc+7K4AQocdONcaBi9d6fYtYRruaMlgja8MM/g0ILWXLQ96QFyTrfl1ZnjuZg5rcaV0TwD6exiFrI5
qUSJzz6fdFTggudZ+XJHJ0mKT2tajaQRTaoVNuk0U7Nb0Miqqp9koUpg1NMJI8uQUV9CygXQ7xAiwEHTbpvwH
mOT0fA20Xg0dDkbjkMzdK+0PQSL3rDfsTE93+52suGKG9562p02s7KGwe2ihTQ8ie1Q9EhM3M2/LXVimNTtIj0
SpzVc6tAcx01Mp124GpWrubLDQb8HJMK/Tt+2484AVZuMj8Hi+evIz0+CB3A1CnA8EVUCLbUXcG44XB9eYs101
aeUJu83cGYJ9wEoFxBF1EIxeIjwDD3gWQ1n4kYqbgSeAlgAXJw34bEQHwI1aWEKCEkKJ9IvKFuiwIPoAt/3atJ
XJSsCpUJdb9ccbkn1160Uyh8bTxEBne8SMH0ffupzAjeLs20E0wlhEzm0kJK7MMYjRf3CDwq0SxAlraucLq3ik
zoe0Big9wodk9RnuDSXysNXSaM0vBBDFaMUHYm6Ky5odLgk+kUhyXIrNL1SKKw0B0+ASzUVNLS41wg1qIo6pS
G02u0UUbHVJLAfna1ErTMX4sITAacSNGU9JoU4+55e7VU1nqXAFwjza1TdSwQ+401TLX4UPKgjRykQbCLtN8G
WA7P0d134WCH8Tgo+zDn1ZW++KwDrQf21BH1FyYaM4NdkgoE511bRRKKnRZ8jk1NbWMuF73uex6XVpoR21CJ
btP4EYgVmnsYSRmE9rmAKg2ECAxERVgvREMVjEYExApgMd0tSYXKEQ3QeAQf2wz92Kja0kovvuat18DFE2GUq
s3VPFgE7Ad1MGNAcC0ffhW6Cx12Z/HtNxJwyj4+zEnZzG1Wj4GQ9jKh6RqDsISZc+4GUTxv40hmEaF5zEfXF1
qIFFgDv+Kghum8Vp6CV80FDsOS/Fuah1/CoU5zMYt0PMx0JvfgtmpEG8x07gcTQ0qq9EcP9VNH5Bd8AM05u8mb
+Krzna02ayLnw3f45e50Cwkv03cXNI3QG/ULmBbX6U7LZGvskjFPTNMzEvEgNQhq3pFLhbgeJqntM8BUe5yY
kNt9GVv0jWoCwuvnv4Fqy+HfS8+RQemSUUuxzIYvNbW1t/SP/mhiwC9jF7FL2GHCpJ7J1doydYM9gt7Pn0hb92
+x3wKbeCjb1fvaf2WfYf2VfQnz7GruffYt9j/0Y0c1DABnnLd710/ki2rGfH2Bs0E/EathdFYTXFxB+E+pP8zz
smCS1EEGY5eKuoz8cgD8Nb0w7HjVIQYocBAmZk0Qou4CRRpHo9Ke7aRDh6RxTqSS5V7xsWyAP06ArfUOBZgL+
LCbZ1m4h4gesVSEKSokaS2nVHcQ+SQCAJ4YpDYYT8bcTabhyI57FBQGgwtETjEQ7o+HCwjkrKDwguD9iHH1g
VeG04K8jZqxxHExzVGtRuwsZw0wEjTZLKV7xCBvs0omA19WTOHVBNNtbaFQ0jMCv1sXE305KFAZyFt43d8mUi
oMgKehKtSxNmfcwFBUbjpUyWanfWn53WRPnRpmvE530AexgamJaYCFr2HSyvPosjC+VbBDThamVKDIxj7ijNE
ZnxC30hveTx+JIFsXE1VqUxbGoZRIJpk7fMEI66MAqigk0CrRuqgs8LuvREk4wkU04gHZJ6nJF5XHKj6f7JSEt
y5cY/AoYIBiC9m3yK2Z0BJER6gazkLqTBX3I3gLYERbYOQwShrNrN15ftZWhabUyMosv0xdutfr9X88Kg6Jd
roY3jqgIzFGtPj1s3IIoV/jqCUDNPL5rRAsj19xgWIFgHrX2sgZ4W+s/RKCnv0/x9aM97iBeK/fg44zqvIPvEf
RiuLrzJC00Q9w9x+zPUAfUPIqBCpmBFpIoUHszzFQH0c50+4xv6adoqkFZ7qxZlwh79JMF9sReYJnZBZ01p8G3
91fH3Y3oVYUWRlgCn+YWs6fiBeqb+CWuBo3mjSKh67aKDxd40mcAn/niYnFw9AGxsWsNe01PA72B+w90IV/
Sf2Z+y/sb9kX2X/g32bPcD+F3uYmx7wGL68Df7Ygx9fxC+DaoJddub5bL+ekheDa+TMMldIzldyghdDYOeyCp7
dNtY9J0dSi1R2Mi5baYJEiLQInDAR2S2092QzQP4+aDvRNmw16R0i2n47zCBg2Zw0vgeKBrIT2rJrd1b8Gc
6c0Eb052gRB93rDSCjYhLdI4G/ln081JR7zmWJKo20Gzz6kWdtjSRe3TrK08CFbFW6MiQ0M1ekNqB6iQ12qBE
4Bgs10mrEcYezfPMawX9SnrG6JAt+VNwGKAICq2wAJR0oMI4ai2NczeyYqHralsx4twvti7ipZG1mMkWgxUwX0
jLRK10fj72kMVbQ3PjCVY9g2qiHcvnQtG750oiK0PBA+AnhBvbLsJFWFfQTImzaffjl0okbEtouognobT01sBu
pKpgg1DWEJ1RZVcbYMNQac0tBD/gxkLBis1yABtcyuAOPe+Azwh0HA6Jbnj4qKQcgykYf4i6ag1hdEFTBfTSam
9ZAWiD3Dfih8uMqBSQ0ivMmUZiecjSMgEfqAF0Hwfk4op3yvuoBbA5WZsfrE4XsYKf12Dbe5o1PTRxwhnC4d0F
5ZU97YTS9vT/b2B2NHbr8EtBauq19cqW0c7mXXP/B68uD4Tz4io5q5VJQ3NbtRs3V/Xsa0sRrhw40xHjz808/Q

Sbt5gyQC+I1BOPUK5120L3yisu3cfipFs1vobt/hC6aTyiubxDGe6rhwqm4+CDAz4NULvcDxGQQJ1Us7qF4j76
0Nyld13HxhGE6H8mUgxxgDfZcpmMYrJDfmneT4NCVgWxiNa415AZ4HcBgVbn5WhSHmAmc6VAX1XaKISP1CNP4CI
1V2xIffpIz3KQwlJWzOC/M4kt+CsoJVaNEH+Zf459l17Cp2LbuRPZkdZbc1qt/OnsNeyF7M7mKvARq8kb2FvdN
F6owczhk+PpeHiJ7k3E7mUdZrWsAN4MoUuB9xebL77g7EfMQ/eqq9wEnruD1L4E1+ZEkrkSwiH0UcxBMd0zCZJ
F1Xp5yaC6YQmUPrtJBBmM0ooNIDiLjkX+OUkPIxxkZ7kUc9B8n0qvVK2C5QpLI1mnh2fu3KpGhtVL6zUkvL4fDY
dQmcUECY9EidK+/V6jt/DmD7lpcx2RYqEBvgEUxhvxRAHOY8JPB3XJtJFyIu5wMMVipeiAPN1IN6DdvFJz/DJU
qjjCeuDEffmdi025q7ZEcZhrjaiWhHTbayXGD9ZvKiFqqam6yrcfefCidrC/HaoJ4EoCY+DILhMqBLFwbokz0S
8Vc8jeSG+pcQ3UNcppzW0IFLLt00cYAQuwOcamY85r8Vr0YLLzzjVhlarl/PAZ9Wtza2fwyzuZxeyQ+wKdg27n
j2dnWbn2bPZb7KX0gu4B/GAsmTvY4wD14Haj4C7dIwMHyXuMqQc9oES1CnLEVuPaJnskLw1LDVpRgC46vCeKFI
WQR4PB/UFThre1Aith4Nehu/Y3chq0dGjRJftgTI10Cb4pmknQAXXImHb5z+Mu60JX9swICeoL7e1MYFFGkuSY
nGseN3E3v60o0FXs97eiaTXW6hpgCymqQwZ5Fmz+cmcZyCuCz1F4k7cS6Tj1ZV9Ky1fjk1PGRCEgpje0eYVZj7
kWDpc8wISeRc69YqBBxW5k55XbvRzhh5wC8SAQE1IW4AR0M99sJlaUphpty0v4j/2nkNhc3rMBqT5gnqwzSt0b
itve9L6TY/bF4a1qBpZf7I5FdRazU6sIvlTTbp0b740tlaJBniY0TV/176LJk1jvC4o66XkzgsP7d/+EejNP6A
im88h4jZGsoXk5gtwna8Tl6jQ9XXgK5fPKZqgh8ByF2kioBSMhbGU2a1NRPwvsgXWZ/vY1Q4p1tnNQIozQIrn
ZeD+9/L3ua4/4fYR9gnwP+/CsSgFETE2x1xfe6kN3LZS/zlgLhiPpPUsegMs79s046xvRRokBE2pHUTgxHHeFk
HonAKSu1SKKDEFgHe1F2yKHZIU0PQX+n20yU+6K6RVrAgzMmyyzm1u/Q4TmoDSrxn40YEZGmHEmKDOuh2d5CZG
mrLEsropF3K/GTUmjz49/qOUeCk/ZDy66Zcq9oqHJmTpSgRlumYgPgSbHSqWgvqHqtPXXEAC2/4BKFVcV1C
aBPwMAwugqhFAUCqBcxx2gCt1/CoH25iUlwEBjHcdntYOSmBNZYhksFvSw4iv2iCIGHlfcPwiqXHS15CcXASUq
SDCTkKSy4p2BgtT90LJfmZuWKUrYXdxMFGYAVRBaIaBH7WqyqYLDa+kUTkvhDu0LTavej2eLgvedZkKYuBkRZQ
UEa/L45PmJ9HFCbxz03qVeifU9rxnCxeEe9LuP+yPCuCXj0j4E51c77541wT08zmR6jsb5Cw1kBfs/kpz3uj1
qkiQvzHm1TtPkJ2dCsogtmEGnFsG7Nsij3Av8B2sxVo1EOIZb/B/hU7AoV6ip1jd7Dn15bdDW7j4t172F/wj4
IXvsp9jkw26+zv2XfZT9kP0WDA14FrwUfHuZKU1LE6jrgmS8IAd+WKcpJ64yW0G1pkYwU0EYSDiZu2lbmhkbpY
H7m0quou2gdjI3SknmaftDGm2DydtVsDsi+02wncU9U5TKTvsXzGTFs0pX5QZrn0LsIpRQJKfx10KdwnzqFTxT
pu4IZ2oVnY5dGRcuku9rk1IVMhuG5I79yuhtFVH1+g67bm0jyHuM/fvS7uXFksKRoVwcBGEggFW+FZ5RtST1Z
/ozYWDqcV3vnutloSOQZLVFaWSuF19KJuh+fRFQtCIroFOXW5TcJeUoe+hyIQgrhKviJRKUsb0WXJ15VKGUoR
aFp2oDCQpHh1oSvi7AwSS6CunNQmX/aM1L+05+cNdVu6AVk6Gc1/JCGYEcyyRF8FNVdGK0CNWJTNBA1i7zBOIA
AcjdT1KdXESmXpB0dIV3hJNT02GIICFhd586IV4ptk/0K530u2arJUm3eJH0XVbF13qjr/MEVv4HJcbZPd7YeA
mdYr5RtewZbc10qTwfiOunoRba/VCXeC7XVwfx/Wh83011zTht0/10dcKaKDv5fJJ5HhdGv8TkvjLZA40XUoXo
tFbDIQSQHI1dcjomESn119041MNkyQ0gyRpFPZpEr1fRkltxZvkEk2o5Gxia2vzo/w7iAnPA094BXu14wlvBk9
8L/sA+4/sk+w+9gX2ZfbX7L+z/wn/eoBtYkaLgLkx3uYZgoojD5awtZ0twPZi53Lwgm6+jIYT0nqJcaa5SrqKG
GYK9HYW3c/X39q0QqKCHm1GuFTiyImrnLzXcb+uW2XgdQvvcCn9nluMTdphY8QBV31PVq2oKWF7vKQvHUFYZ
cIU1W8ny/k7L5atwwk04doro4Ba/FUzkZGfZ1ngTqrWQu24M2d/FKuQPeuJInotA5uUwEBj3I2hbRbJhTaAprb
4vqpaJVkF9E+2idrBiFBZpSbQp1Q7rHVyD0kPseEBcYQsqJmbqv07FyMewc7J1cAJkBiYg8MQWbZ/2kpqrKIai
DLtanaqV0grLhBQgZiM0x1Pen+ATCxCR1YjSgnqphZ6T4HMhLYL1YJdCFxhC8GPoQcyFaRL1WDUuWJzSJQyN3k
9iEtRtdIS818BvbFJQ80SHE9dFssy4KWbd1bAEC9sW0mBph1Cf1Myh1KSU82kQ5uxb4jtK9q2uT6IJfbpU1qoU
vSQ/9s/W9q/uTykwFw2SbM+MABBMT6G1VrZ7t67NsYL8XihIiFGSyD24NhTIXQpuG8F6rVaK/FQ6V3d+kJta
q45XiqMucyScMGThLPw+DJI2wxASZtX4caT3SIn/YyrPEMLFaVL1BHS3m20SiGwceUkrTwX1hZJ3KIY0HBBf13
oq7mmHL9S19PaCD+iUAfoli166fRs3fcL1Uo0CaTafBB4pvXLTWSVF2wewiP2MloxeLw0jYVpXIp2MDa59eDWN
6HdHoJi044o9yz2XPYC8LB8Lw0x9fey/8A+zD7GPs/+kn2NfYP9HbzWb+zH7GfsIfJExnR02dtkn4B+0WL0iEU
d17b1Hg18FAjJ9WDzF0t6cLR41NuBHKbTewy05R0TV30krwDXkG231/M3LqYJWbVzYxb18vg1m51f0jckdjWc
tqhZbTcvyztirBmCX6S1XhviJopIZPly4aJW3Uj5yYpCQKZ1YcdtGONUw2IyG4ZsAsaN8wQuJLUzjEhSAexOU7
z1sT83TM7m42SgvjSNK6qGNEikFCTM90YKQsTDQLPm15YKYWNkBakypWS6Ax2duud3XEB7jLwNDJhozC1NF6q0
soyuF+1HFkkoSXwgHA8Ihtockl+1qGAVDd8yrEjWu12q+CbHxypyRlaEgM/w1zP0wKQFYsUFE03dEScsKo8E/L
QbcHwSN9J06dxBjtV8FjRCyWtq1B4k2qXA/JVyrfSYhvCB4c0hreoxZJWxFLZXBzUSQt4Vd8Lg8DCTs10qwFWJ
nE09+8eXtvTKE1TXDStbTNWx+3txY1rdpDxF9I4CYxf9Sf3N3m1MYb26PLsJ10Ngi0irzcAoTTR3sd7tDUDTk6
MVixZsdvTysX1vZYw36bz5Sx1FbyHVnz47QrcVtsewoC8d+itU/WRhS9MRsFFEzpx8o0QWIW5cAN1NhG/reKv
pyVfmxyEukfbDkNfL2IHeEWmze4xbFwfjrqe2HuIPgy9eCab4hBFPPAs183x2J2Lzb0HNEE98F/tDp2c+xv6
csh88T35QACKrz+yHrFKH5FI0jwb+UjGMxc1tFAbqe4sBZLBmnBr0Fpx7dYhLWI428D1EhFEEB/c0i6dkQN1V
E+fUv6Dvst/dCWq6HdscgE1SFKQvAV6NYJZYnJDX3asqz0890PBkLj0YK6p24vw292xA5AiLaZ55QLkZkMWfOG
bdNovxY2Z8bQ4tu+iAy3iAs5fg1GD0o4BV978gWtuuJefr/TB0/cBkTDCoSUMpCUqQcpD6E1H3mhPDjEKn1KRX
PRcImYWLiEmg0KwLFoYKAJrjd2uXQAAQ1oacSUBwyFjRYgWfGLv82uuu3FcqwM41oqFXjCu1XTs6PohmpcSrSV0

W9x++JOZRrREmBw70ffSN6EshKuHDejWPcvwHL1hM7VdhJT6/Rwe0asGvEf1Sh7zDmQnJKtEHB5U1B+9cHCCWd
if15Plu0SFwpGrzfqV0V0A7kbW+mvjrnSh3E0zrFlzqws18se62tDDWZGzry/y74v7R7rLXw5be7rLpf8I+zj7
L/gv7iuNF1E//EftHMCMJ1lgnC5rg2/gc7/E9/GJ+mLItox1pGYRo23Ty9DmtSJnuSEcDo7NI1rjboQHoBeo54
dB5dTf37qNA15CdVYGGQERwXp0Q4XL6Siyu0jGTpA7/LaQHtkQatqC7KRZbt fzDsk7J16jbI6FfqG04HZ8NH1
3kLq8Nm02IMGfkQJ2+y6cxbezGuhSXg10Ssf1BcCdKxxKpbvcT3d1ASK1Qytv9Mywu0QSHceojNyJpIxZI+gd0
jHdpcUF2tniGjGkpTL+ru6ErPs2KEtnfqJAbIaAopIVRZ1W4KUNrJzqTibQJ05bS7J2CASAD+VzdS20oUeP2wC
GYdQ4UNYzpaqg0KjOTNGep3KceIXpydRA81cqYFYQ32HoVWeTdHCOjGcmX0hT39PGgqYo2uWgp5QoaWF7tPqPQ
mW3cg71IsDYKrn2F1P4LLp1u5CYvZVubcy5CjyWaB1ZVHxpdM76LyTQvw56QjXQBcrGwx5Re0nmq8igcm4hC7e
4nIBnotMEvqGTBC+KVq9+3MzESs11CDzQSgJvxCbfKyXFYhJ6szvmivHicG12u7AeGirdQjdiR9mzRUgD1IQ8o
LiH+oLYhzIUXffG4peny1fedVF1t1RdWt0Z6bBcL/jZ8nA58yjMVktV368a7b8FbVtDr66miPHwzyhPAW9Flco
uyMzkronAM5nvuaLBMqT1RL4Jqwe5VqAMm1XdAv9TF1E2v2Xt1W5p//v5UuEefD41X/vavIEk010Uduw3SWiV0
Fc39NaC09PrHkfPPJ6KE1ogGCUerYwa4zVIyNGSWeL2cn1f3Adm9RLEjdey33Mr409030tP2afcfsK/Yn/Dvun
WzX7GtrgPr85zXm7Nt81HuXK3q8o5e76XFJdWnEHTTkK30ux2pXadViDHJ/I2SsULVzBPVriE+CAZPqLvnZpKE
GNcoKBsalrvEkeb5/GAGFrPpV7JS2sgSq0EPX1XVjdu8Qtf06NdrzVoHdvJo5KTTV19aTHtjj1JA9fCJF0e0F1
KREiQQCJiy1m+T6zEM0C/1eNTRRHphSt3cF5vTo6XyuNJDdbje8rCUI0xxJ+YHDPClxSAzFqrSiEMOyFQULsTk
pa1TYikT+r1sts2U9CU41Je0a92psdcYIIv2mi8TB4CgtYqa10ZK0GcfXMytgLhDjkIg0PwYqIM5AktgXpF6R
jK7QPBRJL0y6fGVhiSHm7Y9bqijZejawikzB1KBaKJwHttcED0y641CQPXW7hHnguLb+hsZfSY1eh1D3YIqmng
pNTgR8W61UNz1Y9Ndhb0bS07GZkFIxgvyHlkeK6Bxt5oC1G/AoUvxw3KhWU7boeVOene/toN0/cbEw0ZtvLgk
1ZFKEUCdpPxa4ww4fMuzrFLJCNPHVhrsYxg+Hvme9q3D9x5TG3nyK1Q3u1oZ00JdQvycJDLuk4zyFkH9e2D0k0
1u5070dcqxN1ZSyCQdpKTkQe6j76uHqvHcQps171SCoeoqcx/6+dbD4jj/IptnffCux41Wm57GbmMMgFk3iGG
inS+YunTxKDUcW0rtmm1FcUYjNAH1h3mBTr690t/50s3jfMcc7DCDHE8yBCSnD1La15HE4smeqhnzV94qFmZi
KauuvrS5XqxeuRQpV1pHyiVwoKHTvvlRmXu4Pjm7bSdmAPtKntFu2LnWhUpC43iRBy0HexQrsiIMoxoD+Ws1Ju
NjdJeo/2vXzdiop3Mpp16a0pea1q09t8wfMyzKAPDj2qGKLqu7V1bW5kR6XL3Mw52t/xareJBnLZ39tey11Rod
/b3PJFEJm9i2mFK7yRbeIyktGkzT+beQJyZKK3b749/909c/2c31fb/A/Plt+nyF3/3pu89sgV+64HNj7pVfsw
GE60LnFb+t8wv7Jtnv7JzvsLvRhn6nw2vxhntkv+drS137Uc4fge+JVAwYhba0vIf4N79Ww/yF209wL+J+3+Bf
w9u/dzV9JC4RN9r77J3+S8PXhneW3ggemf5WNXUBvG050Ppbe7NFbGX9sj+mnBQzxQ/8uj13qN10PrTGx0LtGP
/6FiyBjs401Yo86TRscHxzAnjy7axk6Njd2e/NTr2cfyG0XGA2t800g7B7r706P+uaLIfjo6LaFt5dByxJt890
q4yn1+FlsCycHYL350zFmVf2R0LFjEPz861myRf210rFhVhKNjg+Op0bF114mF0bHHInH76NjH8Z2j44DdI14
z0g7ZqvjZ6LjA4C6j4yLbJ8+PjimM13sPnr7tjjPHbz12rrX96FxreXGp3zpyR+vomfPPPn5qY7518YkTLXf7b
0vMxtmNM8/cuHn+6LHTR0+f2Di3fu3GredPrJ8ZFd7XevTGP1fRo6f/9KB1/GxrvXXuzPrNGyfXzzy9dfqWR18
5+h59/R8079i5c7ed3bewcPvtt88f2Ti2furoxvypjXMo4f+mdvXbjw5e/rU+onW9Wc3Wt3WwdMnT26c0Xo8v
/DPPPto92CCp91/0jjDjrNbITXPsRbbDuibw/cyW2RLAMQWR0gd+DyKUrT0SimdDUBLi13MTuC39QtPn3VnG/j
ewPcz8XkzSh7FvdP4PI3SGy1Dmm7gSf043wd5X655n3uXb/6xL5faNE/vfsvudJC/dQ+0jqHN9Ii4QYcjFrwd
Fw7zW75Nb385fNfPvt/M3p05xzecxZjsIDf293vP0rdwL111Hfu1XjK9XPhV970f/f0Na41Z9HPUyhLrb3eta7
FuvhH/T+J3w03h8d/qYTDuBzbP4DB+DU//xvo2dLfCmVuZHN0cmVhbQplbmRvYmoKCjEzIDAgb2JqCjg5MjQKZ
W5kb2JqCgoxNCawIG9iago8PC9UeXB1L0ZvbnREZXNjcm1wdG9yL0ZvbnROYW11L0dbQUFBQStjaG9jb2x1dGE
KL0ZsYWdzIDQKL0ZvbnRCQm94Wy0xNDkgLTQ5MiAxMTMzIDc40V0vSXRhbG1jQW5nbGUgMAovQXNjZW50IDc1M
AovRGVzY2VudCATMjUwC19DYXBIZWlnaHqgNzg5Ci9TdGvTviA4MAovRm9udEZpbGUYIDEyIDAgUgo+Pgp1bmR
vYmoKCjE1IDAgb2JqCjw8L0x1bmd0aCAyOTYvRmlsdGVyL0ZsYXR1RGVjb2R1Pj4Kc3RyZWFTCnicXZHLbsQgD
EX3fAXL6WIUyDwRZGmmUbKog817Qck4EyRGoIIIs8jff5tpK3UB0sb3Wtj0qubcWB0yVz+pFgIfjNUe5unqFfA
eLsYymXntVlhFdKuxcyyL3naZA4yNHaiiYN1bzM3BL3x101MPdyx78Rq8sRe++qjaGLdX575gBBu4YGXJNQyxz
1PnnrsRMnKtGx3TJizraPkTvC80eE6xTF9Rk4bZdQp8Zy/ACiFKxtR1ycDqfzm5T5Z+UJ+dj1IZpUjsj2XknHi
/Q94krpC3iWvkHXEukPfEuwPygfiQIx+Tfot8T7yRyKfEpH1IeqpTJaY/nJN3g/yMn7rpMGaUqR3SQ3eOsFWc
Rc/I+Tq6n0cHy2M5oYTMxb471Ld5NBG5xugzJDCCmVuZHN0cmVhbQplbmRvYmoKCjE2IDAgb2JqCjw8L1R5cGU
vRm9udC9TdWJ0eXB1L1RydWVUeXB1L0Jhc2VGb250L0dbQUFBQStjaG9jb2x1dGEKL0ZpcnN0Q2hhciAwC19MY
XN0Q2hhciAxNgovV21kdGhzWzQwMCA2NjggMzk5IDE2Mia0MDcgNDAwIDc1NCA0MjMgNTE1IDI3MCAYNDMgNDk
2IDQ4MCAz0DcgNDg2IDQ1Nwo1MDcgXQovRm9udER1c2NyaXB0b3IgMTQgMCBSCi9ub1VuaWNvZGUgMTUgMCBSC
j4+CmVuZG9iagoKMTcgMCBvYmoKPDwvTGVuZ3RoIDE4IDAgUi9GaWx0ZXIVRmxhdGVEZWNvZGUvTGVuZ3RoMSA
xMDAzaNj4+CnN0cmVhbQp4n0VaeXAb131/b3dxEgBxkARJkOKDwEMUQIAExJsySVGiRPMQw0MCFMkySCwI2LiKg
zQzie3USetwmsRJ0ybJHH8knaaZjJtdiG5zdDJKH6eaivtPnQwbe5pJ3ZmmVdLU7SSZ1G1F9XtvFyApKUebyV8
FjN3vffu9933f7zveW8rFfE1EJvQk4tHEejqam+juIAihlxDC9vXNInLB/xgH9N8D72w8t5H+9onyPyHetcNP3

Ehtxz+Rbv5LhITfQ8g01xCjMffeA4MIWb8D8gMJYMzeucUjZLPCuD2RLj52TLfTBuNTML6Qyq5Hv96YcML47TD
uTEcfy/2tZgDD+I9gTDLRtPjXn32Jzv8aQnVv5rKFYhAt3wHRm/R5L i/mEukPR2BM7VuCH4Yv/ZiA1NIxxwsar
U5vMNaYzJZaqw39v/twJu4vuFcBnWvwfRCdQm9FK2gRiagNuVATmkQL6DKaQ0E0hiJoCE2haXQBjaAGNID8qBc
50MMS8knIMSudXAxD25GJ0QZb5S03vDpC009I0K+LWGHv7FHwj7ymmtY9kicb3YpfM4TcfdIvC/ZSKSJxbBbm
oj0SIKPTnV73G8Lf9f1VxExYIVvu34UcXncksYb1qY3I+xBJALraXzmq1d6JK1PPo6fBu3k6atXXRKCZXQ+uZ2
xJqosvc9uI80BHsngI++gSr4GyxCJ75jxEEnoffBCi+EdcSdKKDHkcrsjrh02W1JGVKFRsc7qsrfphxRofeYW5Y
/KRgKTzXg0Tct4zHX2EhElTVmCypmpZ1BNdsj5nemoZ4fseJg6D11cmgBJ8I8ypAmRDmC0hWk6vdodrvI3g7
AAJNmwJpV1TY3E6v1ecieqtxDwrPLLreEI+EdcGjGs+Mh0zM7nidoEyhtx7JSSNgB7tt1AFK209yYIfePNFHH
j7sCZ3q8IET079LYXsw5tnRSWQxPOa6CU/qfDfQBj44cwbPft6K1hG7UuHVML0uhT1rYL3njAtu2HMGkJ9YCpc
RQVPrZ8qYYLhJZf1qElsquup9EnABF7j00FQV0d0ibvNE+RBXciLAuj9qBzgkXe3W0CDglc6Edj1KVRnYLdLQ
K1AOQMyEW5hqTcgndjb9WnQGDB9VrkFe2WX5pbkssqd2CsZg3IHjPRBqcMq0+AZghGyyj3Yu+t1k+Q+4HZ02uw
SGZYcNlmoHR5GsrPLZr8hGFwt7c7h3j5cp9PSbyv29J8aHKDf/lNdNz7j0m29Bxih0LBr/f0a2xuW4fb5uYeP
VtTP2CpuTJhcIxYzaePdTxqF1Z3HXWPD7XduLL/Ivfa7W3u102X0P3S1NHR1+psDAX0jBvs/W5Xy/Qxi9h07Fv
13HJgX5NI7PGf/e/VaR66J/S4Rajrd0JV16BxVIYGCf6FygIgJelDu1rMC2avrKWomAISvydz4KsmKHFwQag0
AVlo+aWbMZeBB65bR5bCMwN2Txc+/57rl3DW/Cb29/1Xt1/EY/e7gJV6NidW/hnnAs5I5zqAzt1StrhFtjcn
ovT Gh0Rm85ToMvFbh1m4jNtWZvbuNAvIKXiy5A5J1T8ZgQW0wjK1UHi0DFyqaqu9wNzgbQg0DIQDVc7yrs18Fs
c0twop/tv84N+v2xT0rDQnfwNPjXQ+dXv1WHbf/J04a6wysdtf9o0RzcBvbXnvm3Pon5vv+q/nEJX/LbAfDaQy
uzwBOPHKjMmxk3jIPIGFJCEgcM4r9NMwUBwDxZAr36u0umKD6/B/gcy10yjFU1l0fG8FnRB21UnChpkx7ch24V
qfkwmCLuWEEx2doIaWSiCQ0+1Ncd+OakmVIHLgeVzMHGjwy3xocXnh1zJwfnr3U6FsiJK131S22c651nrnz6fNf
W1PU/ne6Yv9o78Z5eR7h38gN+KxhQixD/LJhpQjVUFqhlmA+Fyhpwrgw31IRCISyZA7Ifoj3Q/MXTX3kj0q9R
knvt0i6mzInvKmR+Jtf/Gr0Xz/P+Ngv85xe4m5aZKMRnhlu8gCXzuj3+/GfwR5KN1C/8pEmmgErt6cGe/iQC9P
/uNfjf/Kp+Jcs7/lk7I9/J8Fdvv0ZAFHiFulPiQFyjj2o7KwXkAHJCjBnFEZEYbQGJ02erNHc2jVoUccUoo1ci
LQAoYFCCk1RT500foD1Bh7aaG9dmR8ZMVtMdfgFr/71v1/40zXr1kgtmrcfgzY2NFxdF2Nuo5C5ILg2S1CtVS
1g6r2BKSApdkJEXRa5TZAyxbcxUoLwdZdToNcQHEB2Gu8cpsTDNjBS3C4bHbZxg9T22ye/s0Z2+C08XdH+McPj
1yqMf32WGdk+0KjA7+/2t083eFf7mq84Ka5dm5zj3PnN56dDeBFXLv/k6XQ9NsGXEuhs0+FniJ1cBwub+VehFq
fQuUa6o2eg5bI4Ro91BknoDZaZibmCdR1ucZia6wGG6ApWHf1GmQDH/SBSr17+lnbGoR6x/zH48e01XPcSBi/0
W52NpzXam+PUp3H4XIhdLahLCpbKXhmAI/jVBrdHFxhgjJmRYD25DaAsM0q0wEnC5DaYn1poTJ0K5S6m6KhpPY
yNrc0Dw9LT1tZb6yj1MUu1QKjLjMgqq+jid0GBmnJaGtp2HXuejfY0o4HBpV2UFcf1P/D9sRJ/Aexa416Hpf2T
+iC3vWr1/GYp+cpE375p3hxc0ZdSwPBIXvD5SenhoZ0Da1c6Gr4FE0NyI2x052cDXLRCn0sjMpm61kT0GRuosa
awaEypsmoRlioxniZu5Zwaf64C7P81OyWuWvCmkA1U67mNwE50Mfks0m8eCCk6VqpfA7u2y0u1JibH7s0SdoP
iysjX8QMuHcce/Kyebznte471+8hrMXWDbM77/Bfx01e04dp5rfEjr7RH8zjYsRLq9wjsIFT1TW0L6PQ6ycop/
THGBB1nUagJM3sAR1eLp0Hgcfcg6Gc0vz0afeff0Ln7v++OMPQa79+ze+gWsw/vKXUXVPGYa6MaCJQ72Sbi070
sTxSKPoKCDGSuuQcYsEbDGQK8GCHVntZVC60D73DV8a7+IryobytKtJUVX751bXj7Rzh1PofK3UjpBGXUTd
DPRCFWl7tto211NfooKkWYjneAqp7yJ5NDgDRYpUFum3THkx5DojJKaXr/csnX0C0u1kk91XS3pSJ4U3JfRMGN
/QGrYP2SS09uHi0Fpt9G01q9Qbzi00H6K3WEkN0LySkaAs00NYAB3dqk71RMirBwAIMjtu0Gg33eFKAtf+kS
jkDGZnn8fTkxekeCzd0IzT7q5Et32gqfvPH2vRzcv9/dyFbi68HByNWa1zHoHv7Xrq7UPHs31nsq36B1zHh1rMN
X3uaLn7YHxx0krmp3Se1zXA3kcgA9A09vZQvNhrrQrkE5IPUGyxxNYi9g6GVR8rJyHQ9IjXvyKO1apUCFDJ
2EAowNEetFDzZ3hkM7uqUQ9EEjEep+3hYPhWAGj7W0zRMD0ZeAxyMtPaWfnYwUsvW1uAMDtYr3mt1alccHMWDo
xzU8Wmsbuo6R5C02FGKYtSv+6bk4DivUe9p0G/G+PJbM17Hgu9dmH1mavi32BwbNg4vh/305msPtTimrPi3Po0
LZsv+SycwN24Y1+nm8KynZ629zoYNrXpTw4SjzUY9r8y3Nrf6kM1s21ZPjmuFypkAf5edZWZQ2U1RqwmxTb2sh
cuuTUBTtKE2B1TnntwAuxKnVH2DVTbBYbFGwaWJFpqNbqkcdB+y0TXQnapy1rR5xuYH5uqdyZ005dE610pH/Qp
nCq9ahGnSDUXx6HSjYd47vP89pSa0w771TbCrD30J1XtoTQgQ06GHxk7og5qop/FsAV5LPeW1NNN4B90SbU8mE
LxqMI1V1kPUuoHstkpN1N0MdHNAb0LqCCnV8caHv9quVIffKpluyu3Cm1LnTRjcgG3V4ZXarTc62juhT0B6UCZ
1eEir5YbJ3N7BaqVKsUr7Rts00hztEqqJ7p6qJP7VUn/2wy9rSdXWowDZsvCVpt1JTj1AY8xrIkt0bVnaq2Lm
07dlf6Zr1rwx87VmyAB/Q5BcLSGwk9e09kkutzLuK+2tqcBeC3TDC0P+5o/adAreCL8sZHab95++cb12rGfcm3
8P1P2K9+JfP/g3fh0J/8svG0gpK+yYB7/rjtDKHaYc+Rjp+8n3N+gRa4eHUN7aIxrgPt/wrzPoFo8B+MA0sb5U
Bx+x4Eew3+HjNx1kP8o6uXa4FkT8H5E91j0IfQSPoGn8Yfg+wPu0LfdFy3v5Z/hv8z/RPhDTVDzuLZTtcA0J31
BeaGHMQfv53DTPAOwU26Qe6hq53zVZgxP51WaQzq0qtI8akZXXV0AmU2V1oCwX1VaC/xnVf0PMz6i0jXwXvZr1
TYL30BfV2kLate8H+SwwIEuq+YGo3mgTzoXGK1h/JcZrWX87zFax+gfM1rP6J8x2gCrPqv1VRojq/bjKs0hi/Y
51ebRKe2XVFoAmTdUWoPatXdUWous0o9K65FWF1LpGnQJznMKbTz8UPd+1bagCzY3o43UHluS0TXUf1uJ0SbGf
zejLYz+AK0tQBttip00o022Tz06js18gdH1QLfYXmR0A+0/zuhGRv+I0U3UR9ubjHZRvXYdo1upjN3J6DZGK36

5mUyA0ccZf5zR7XQd+yyj0xk/wuguo0vsIqNPAq21bz06h8k/wehhRr93Kpvbzic3EkUS700N+ug1xK797DrEr
sP02tfLrkFyfmGVT0dFMZ7NFMk6XPLJtVIxmy/4KXc5miks iRulVDQPgi0kw iML2Xw6mhpha1SY18R8IZnNkN5
TfUF/MDgwVBWHuY1iMVCYCQQK0c1oJhNN+DcyJX82vxHI5b0Pi0vFQiCuWhFYSSQLYE06VYqKecIMA0Yumi+Sb
Lxq8DTw/YTMsId0L1lk48WtaF4cIdvZE1mPZkhejCULikc iSRZJNBMLZPMknY0149uUUcrEQEuxIRJQ1S7Q9em
A6jgvZsR8NEUWS2up5DqZS66LmYJ1omAJ5RQSYoysbTNxag5ZVrWT6SysG10CEj4iJuF5nmyqyIQqCtTVfASs0
REtUoPzJJujk7rBym2Si hYP5vmNRoZJBYoDp2IkMwELJrI58CEBS4FXW81UiqyJpFQQ46WUj4AkuTyzcuHi6gq
ZXiQy5NLS5MLK5FRkCwmsvBU3BSVdZLpXCoJy4In+WimuE0Nnj+3NHUB5CfPzMzNrESozdMzKwvn1pfJ9MULM
kkWJ5dWZqZW5yaXyOLq0uLF5XMQmGVR/GVYx1ksALKYWIwUmWxMwKRK4BjqrhJRDrDf1OC6mNwEg6KQEbtXyV
AqWxmgzkGkipoNE/iJJMt+kgBzDIS+NCEhHzc2tqqpmJKWaIQMBonC6CxkBPXk6BBfGxdzCkBTcaV5MqLECAQ
WXXS2kRwrKVS4nKOKFA7U+mnBETK+B/VUmha+UiaYgvsD0ZfN0YTXzqjH0FLPMz8ryvkMPY1nQAb7Q7EsWC2I
qDrk0itmEvFgopYpJQKBqGSwFubCe3WQK1ZT9+QD6Ccu1qs8H+hLZLc iTPF13G01YwwASNUsS3EApED92EpsH
7GcpFk/+t9ZMK0grNbpITwZN7oN1hXFjArqgaXgKSukSrmpyaJMpUVAp9MSpd5kQeVG1JYQTItloXtU9cKISmx
BkVcf+iBJU2JRVFtnoQhrMEfj+Wz6iH7/fVKMJnFKjBcDG7kUPE+nhNPCA8KEMCb0CqNCv3AaLcBrfB61URS1g
FpD29iMRPQIyqBb8Dt4No+SqAhfEejn+ef5F3iZ/xwvoymQyaFtkEuiDZQACQKvbr3wDSJf1Q4dovsP0UOH60E
q3Qe/AzoI1HmwcxXu06BHH68ctGaYrnWVovrXUAkoanMB+auyy2B/BjhLQG+ARArGeXXFkUNrVuTIEUxGdt1xt
+QloKmuJL0BgNQpJueHXxANgHf3rq7oTTAsczAegZe0ANyjcNSiEhm4JmCFDaBKcKeWbIBEDu5ZiIsIHhdBPgA
YHMUigFZgZpJ5QFFJw5wSi1ge0AeIKR15hgIdZ+H7vQhPq/J+4BF4HTqYWFdFLYETnFtEWW0tkWG0Dr8Qs iDJMK
D/Gzh60kchWo2tSqRjYnmWVpuFOpeOwTkWixCQUL4rgocjuIotQoWp/5Unfj/MwyjApGkWCfGdNaCSYBn9QzW
9iywiIrNCwaQiU2Cr xYC3xiyprD5d9X35L t8JQ0yxNQp8JSdoFovMj4TqweZdOR06x40jtvnYczrzBFu3gnCe8
XNVTd0qltTaFJ08nz4/MsL3IE/uzor7RSrGnmYOWZhgmkWVE1Xn0+sW3FMM7zX2vMR8iL0q8zFuicldhoxaQRf
QRyjVCopzPzVGGH8S6nSjJVeAM6quWWQ61bk18+mwPUMw6zRuirVKTPIMjyJDREF4H2DtdA7L+JDoDdsxwW
yJVnKfZeAFk1+FLxxdhFrWRZhG1bwUkpsDy0Ta13FW4L4LcMsxSKmZzZNFly/jh+pcyTJaCUWYS3EuqPGMqDV
XUFFKMRQSRKeIag2uszzcVBGKqj0idwidX7eCu iyXNg5FTFnzaKZV+gnVSPeXIsuLgoqWkT01zHqlQyr9cYt97
+2KqSNW0K5I8ZhkGqNs3RzzPan6IKLHmHzuSIUmmTUHnYviFVW7VJRhngVuCaIgqtWyxfaZz6WVN3389Zx7XB
Ud5pVRey+kpXsK7EdIKXWryKdY88qFh/teffWcYbtgQfxvNt638+ZSeUUP5S4VHpfku03dN+Pq329pMa8o1HP5
pWY1Uk1B+7FTLFK6Qs09zYPeXi0y/5fMpDm1UFFuzf09/OPVs uW2k/yKnabTFdS7eLioa6aPdTF1QwpVONB82H
7F2BOK/jgfPSbxGDmSC4f3U/vn58Hs1EmRbErsjWPZur9MFVierAj3b27He0sh7VWdoKK9souWo1NVvVyQ41Dr
Kotxq4U+3v9VZ5V1thSd/J7Z/rUTppi3VT8pX4WWM8tshquRDTOTmTpX+C//1fsYpV0nGJ7JT3JbbD9TJmfBur
wifSAVjL1fs9eZyd25eSVZTmUZjmt+Bw7MucijN0sQ2ZYt7sCNm6jR/EZGN1fb0o4JvQJs8J5eJuYFYaPaKW5+
nNmqdijrEPAKVCYhzeRi/AucomtoZyAYx6m9Ra7IDzbuHIW0iJYVtAm8d+CONJtsuXWDyihihDAwY//E4Y5g0
B5W+Zy19yF+Ewfp/P5/Gdd0v4vWhW0i+GZYzff5Gn6f/hI1kjs1LdEhBPRl0lrfdqGEkCOSfx3rNY0iiEpCXnv
og0T9xBnFFmpze9ZyNSnReh/wGKk3dsCmVuZHN0cmVhbQp1bmRvYmoKcjE4IDA6b2JqCjU0OTEKZW5kb2JqCgo
xOSAwIG9iago8PC9UeXB1L0ZvbnREZXNjcm1wdG9yL0ZvbnROYW11L0RBQUBQStGcmV1U2FucwovRmxhZ3MgN
AovRm9udEJCb3hbLTExNjYgLTyZ0CAyMjU5IDEwNTBdL010YwpxY0FuZ2x1IDAKL0FzY2VudCA5MDAKL0R1c2N
1bnQgLTi wMAovQ2FwSGVpZ2h0IDEwNTAKL1N0ZW1WIDgwCi9Gb250RmlsZTigMTcgMCBSCj4+CmVuZ9iagoKM
jAgMCBvYmoKPDwvTGVuZ3RoIDMwMi9GaWx0ZXi vRmxhdGVEZWNvZGU+PgpzdHJ1YW0KeJxdktqwzAQRff6C13
TRbBsJ3EKxpA6MXjRB3X6AY40TgW1LGR14b+vZpS20IXEGeb0ZR5J3R5bo33y5ibZgeeDnsrBPN2cBH6BqzYsz
bjS0t8j+uXYW5aE2m6ZPYtGaayZM17yM3eLxx1UNMFH1jy6hQ4ba589VF3Ie5u1n7BCMZZwaqKKxiCz3NvX/o
REqpatyqktV/WoeRPcF4s8IziNLYiJwWz7SW43lyB1UJUvGyaioFR/3JpEUsug/zsXZCmQs rEZ18Fzoh3W+Q8c
o28idwgb4kzbw3hbIBX6RIe+jfoP8SJnyIfIpHmKevKpI1MPx9jPDvkU/XPkJnqiTyoin5Bj/0V0w96nwrH
xLj/r5PLmXFg1HY92iNvTBvjvge1ksYzeN+4VkxgKZw5kc3RyZWFtCmVuZG9iagoKMjEgMCBvYmoKPDwvVHlwZ
S9Gb250L1N1YnR5cGUvVHJ1ZVR5cGUvQmFzZUZvbnQvREFBQUBK0ZyZWTYw5zC19GaXJzdENoYX1gMAovTGF
zdENoYX1gMTcKL1dpZHRoc1s4MDAgNzIxIDUzMyAyMTQgNTM0IDI1MCA5MzcgMzMyIDU10SA1NTYgNTU2IDU10
SA0NzYgNTk5IDY2NyA1NDMKNTM3IDQ5MyBdCi9Gb250RGVzY3JpcHRvc iAxOSAwIFIKL1RvVW5pY29kZSAyMCA
wIFIKPj4KZw5kb2JqCgoyMiAwIG9iago8PC9MZW5ndGggMjMgMCBSL0ZpbHR1ci9GbGF0ZUR1Y29kZS9MZW5nd
GgxIDIyMjY4Pj4Kc3RyZWFtCnic7xsLfbTVufiZ+Wb2cTZZdkMWCOQxCQY1LkkM81bZPDYQCE1M1kC0tN1kN2Q
hya67mwREDOoFxNKLVQ1qrVKKqGgtWqoBqdwahW47b9i/7W9Xmulf+u/kXJ7aW+LMNzvnJnNbKgCPi4v3uzZ
0fMme9879d8QDTc6SdJZB0B4mpu94Zc184sIoQcIKrIae6KKrPhtP4I17/FX7U1tLy9u7vIRwjqPX1qedvqlsd
mvtnF1LSZkBGmVr/X17t78wxCRt6Hz6e34sZUtc0I94iPXNHaH101x/DzL+H9cbz3tAWbveqEX2cQk1qG98527
6rQ0+IiCe9DeK90eNv9nT8MV+L93YRc814oGIneQ647Q8j1UfY8FPaHw05Lb+M90pNbC/AD/tJwqWB3YsgyQa
jyUwtScnWETZ7yshU8j/pR5hG+shB/LxCdpGHhJ1414LbN+H0dvE5sp504s6PhYPCJnEK7u0kx8mbCLmRH1RdE

hEWkKm4S8jbskh0CB6yB3HME1KFUWaDRKQqaY9UK/VJH0iHyQwpIh2WGqWIMBUelevlnfg7C34ippDXSRbpE94
1EbIPPoSpsF8qk6zkXTgMu8jvkQraG2lsITvIGuQ1VQiSHnGNWIIs7r8mHyYP4CeLzw8LDwpvI3T7hDvIWuR8kc
T55WHgL5TpI/kruAI/Yg345VWxB/19DXIfx/IMkIhH5LYESVbwS95B7pNXEvzNgivwW/xwnPUjZQ3YY+gypxvF
IhWlsp/Bjod9wL9103oQvwU3wG2G9NF56QppPtmgagEayBXE/yM4YWoTVKDv7rGHYxW6pUdhFPpQajU2I+ydMI
qS5R6xFiVrIfvztNthQpjnCetiEnLKnGeSwcYFUG0cRg3EtSk1IEKaRFbhaQ54hz5Ep0Eu2ICYur2GG/Fc8+ZD
0Hsq8Rfia+FdyGMpIHmmRjqGuCfp5LyEvGA2yBKJAnIptt5hb4dvtun6p8t0G7Cn0IbeKzajsJjW7k1crfWf01
CyVxskNu+X03ZBr2i3ljn/vXA/fm+JcWLNU2X3aXaZjdTeW4V7dUlyy09zGfxCzf8aI7pZz8U9F426luVW5y3b
X+N132fyzp6DaSIvaK7XIOzAbGc1YV5L0MTF8LJjkH1EiBQe09F9FbEf6j/QXjrRn230z7dktEjkVgXGnfq/2G
q1/+3PYkIfx/zrmgjnyW8RCnK5U01bx0xK5jRqMkC7PNAvpREqyneovwj9XkYJrjuKi8LmaZEYFNnIqIp1qH4/
f419/Q/y3N944nf0G/Nbph0TfySvF107PYVzdd0Z9qQdtneRgkZDrCuIqzBtMd8q0JwV5b5Lw4pi9KX1Jm9PH0
USTw0QWiikj301I7UT/AXvKLGT+6In+o7Zj+D1xDDcKXX1zM0Iz2zN+nnE8Q55L5gpzxbm0ueNkp7HAVGB20iA
JCKEx6Ai0My+7SVgmOLIzhalF02c4rMj4hdhtZGoRMeYL43MMRqnn1HNJh19Y8VpT889XqifU14S8U78Tjh3iY
3c+uNcqfvmG1167+upnJjuFmQIVRgql6jsHtu155mGWMV/Br9WGVNR5zgtkmyiYSlmETKN6+gtdyTbZJdfIjXJ
IPi4buJbGv9KHgfL3/pg+01EfFjKaLHCNNOxNIXuT+1I2jzGnjLgeUhzuMVx8XTTbsULX+L1pa8gaQ4+xx9Rj7
qE91jvJPck91p4RPbYe+5qU7WnH0+woaY7BkToKRZ129YSJRUzS8TnsW4zc9/RTW+99+u17jwsp6rHj/67+SbD
Dux+8/voHf/jpax8+pP5U7Vc/QuFnoYypwkyWAAAnWFuN09IkY2rQh5rkM3ULI21ZhhLLVQcK4gWCzXYjSajb
Jdkk8ko2kFMQmg7lpESKotgAPJdiyNQs0mLZgsRpJs03J09CzmQv1FRaNnFaK7G23yR0abSf+V41dt2ZDzHLG
in7n+bBANskgJf1loJPkKqtDrx0vkq2khrRQXySXURRvEFeJKet1tpGvEHvEWuUdeR3vFrXKGkZhFExDJIB0ZG
ASjZCImo5mYJUqTiHUs0CSHKS3JZ1WkbFkxKEbFNN58Bc21KFbFeo04G6ZJU+VC03TzLMvcpEJr0SkXFoguyY0
2LjGUGEtMLpPLXEYXJbmsLutSsd7UkFRjbRGXg1dqkhsNjcZGk8/soz5LN+kS1oiroFuKyqsNq43dppBpVVJPU
o91g7gR7pQ2yevNd1m2WldJj1i/a71xGUHHMQvsjzDeL1wv0yTMFma9z740q5tU9Sfqj1T5rY9TpGP9+SVs3
kcW67fZjTF6B/2UmhK9WAerdBVK01z7zfSA0oenmK7dQBHsnoXge0HgJBtadm5CMjRfQhHhg0D0ipjuswbEaNh
gVZFc6Hht+7d/a+9SPz02FPiv3gS6efkxqfaWmWZRLz5xswFmwkncx0pY3dS6ype2XTZmufsA1GS8Qkzr0nWn
Z3KuLG0H+oyc09Ns0Hct8vjFzXeb2TEDKL1MwxjNEXmMjhJivowMwaN9fb0fveXgGXlm4C3Pnn7t8XvuuEKe
+55HJ4Xv/z3/1d8XqFMM0GnzKs6Dn7wwUH8RT28hz3WDejD1LhchLYLomb5duMZLwZ1GVI5I1WGxHFu4e4Vm
6F4FdMxv6D2gBXHSCpUt0zoYcYc8IGCGjy2Zk2+VpuZj5hNmqsEB9QPC/ISw4tWOXFJnfN//kW7u43rFySAuM
qaTh10T08a0gzHpd1k1d1mWSmzfst+xvD316xJ2EMRGRYGmj7aBAXWycLfDs3D3KM+NC3enem5EXuDMyzMbMG+
//DLPf5yfE/0DwSJ/J0x0t91Hz0LuXEWLpXq53nizdLpcNW5jmhH7izRpLBaa9CjpMnS0jYyLpt90NqTdPvb2c
benP0GeGGdHt8pFMaZNjz0uEzT1GqcxW0u0VlPRQLCpeeVUZUT421Tvos3f0XNVTcfWfoHIdV9Y5p6YteuXd3
C12e3b6vo7i0pPXRV0R9+9KXHQhnqH7kf7FPrpR3od8wPrn0Ni/1Bn3Wz8APYn4E+MI97QznhKIiLcMdjbmC
y6z5wm8zJWFZ7oAHYgYTx+dMEDijWn4TIuiNz645RM6c0bTmWXEmesPj7PeJ088Y6C6fV92v/g0/+73CH2P0w0
zB+2DHQexCKWuctm0wkp2kIzsIkuCKIBdxDR1R0hqN10BXSwU5zJgmuhBo1ATPHDzELaSvRZEhiDkQ9zInQf
Db2Zec2NDB/YgZ1a82eB+yx1HeuzDf4nnp3fipJdKzkoBPotdJvdlG0xLiUttAu4Wapyx1lX5Nupw9Ij0jbjP
Qu+104Unpu9Jjxm/T7TSdYguP0duC2U12mMda8mCCnGuebFGSMX3ADP1qI8tnhckVUC67zQssruQGUi80iA2wR
K43NBjrTfXmBktNcjB51dCT/A3hPuNTwg7j7uSfJb+bfCa5AJ3ILLKkxPOT5FNXCrvveVep+94WvqeG3xbyhD
p8fS7p18R+tT54gJx1HqTsIX5C0szdmNMmjA75bscZKv5NmGrzStaKJHTkotIullK4XnCPgt/mPej6z/X0FLgG
YLX7+zcbH7NE4R7TwjThCz1PfWgWiI8Ijwn9Kqtao3q1Qs+7hbGYDjzCqN3qtvUdeqtai/z0T0/Uet53baQEaT
M1W4Rjct6UpJxo/wDsj/pWZvJJhuqkwVTEim32U6hrY704j1IP6/EGI/Yf1hsdpe9xt5oD911XttTDVcK01iuQ
pff0v/t75dffFajs65MaP978y5cf8j5gmpQhj481mCenYHxQkkv2Y4bIsow2W8mTow17rXZ1Q9a+9L3j++ybRye
R0TAm2WyyZIEp1T0BtXHoC1aLnavjwNETp5CVV7Efss+yz2L9UEdhrMfmYVahUphdmDN3oivDlenKcimubFd0T
UZNZk1WjVKTZNTMzE0cX3GxsyNWruVjdnc+6euH3i8YmZsa0xQ7EDjZmNWY1KY3YoM5QVUkLZ6zLXZa1T1mW
PWZbQaVwrzNCT9oRpV0+fmpp2YTkaJL7379G3BB/b29c3df+fTB09/LIiPb2t83uN/6Yb/OC50bVnTFH17T1716
dt2tXhfefQHL6f0fDU/f9fEiadYrNrYuyu6iYg84JqI8WmQMTrBwC4S4HPrjvFZApLwXWLA7oLIsmC0HTkwWku
YwYY9hI1JtZLNBGYDPOkefINsBbuAMx5RtEkmQ00IVucK42VJ5MjwgQxT8qTcw2KaSaZKkwVr5GukWcY5h034
BYrpAp5ngHjxdA1BqSAfDPW9S5xtbRa7jSsM91PthnyMD6ytcotLjj96pvC28Kvf3H6NazWo6UPT16JvvAM+sJ
S9IWR2BXPcaWDmYBVMGy02jdTYX9qXxLaP9VSJWJb6WDhMEur10eLMFnaUzCB7Ak6XnZgwR6JZVmrlAP1c4LwT
N8tt2z9T19f6b0dr/xU+JmwT9x52vvIIy/tEND8vP3plubj8ATT8Ubk4VGsVxPJra5rkpNEq0XMzMo0mUJjFb0
yMkuoJTNLcgjE8a3U+8ZstUtbYX25X7dvnpRJLvnjjCrnXJp1jEtNWeS7V8P9J/qP8rCY5bWsR/FaPnrmdur9
lius36ES/2ChhiBwbvs+ay8grzqPMAeXws7Ukd1CZmCI1XKzpkwcRpv29GjCjC0p119BXqUND9y6CuPfa97582
/7/q0+oHK/60bk1/+Dv7Nz645ndvCKP/Evi1v0Mn6av62r2Z6Vd+fbzb/+2s0Dn7vI7b+24JWvM1JefevXoB
L0/knxc99NdaXHN9yXtpwJqvIrFXnkqq1GsROH7zoDS6x0/Y0oXsDpp9SjeKEk+pvSn9+4t+V7nK6+K005/SXz

4kYdf2nF6I7ZKft+fkG7wzPvwGtKdSD7Q9V2nq7surm6yQdgkpW5wbBqz1y7tze2L67t2nM1qNKXmuCex/u3II
H3329S/HLP95VjKYH3H1E0mMr9vT6fp1vSkfHxJclqcSXPMc+gcy5wk10IU4QpxEp1kmTyyILXAMXnUpMxJWX1
KXvYVEzfQDZYNSRuSU1gFFUUDNVggCZLBCiPABmkwFsZBupRhn1iQNzfvK3k9eevy7s7bnnc8bwgE1DDWsYf
7Zhp6Py4KtVT9ywaVPTfXMPPPafv7rhx20tr3pv3+x/yvXU/b/915Y90txnJk3yeFwV2dbJD2x66Pnx41+aNq3
h+oU1uSOu2Hr7w09navUdc2zDA3ev/+j0V0Zc8xeSZeKznF/8Kn1rbK7Ds9xp4Ens0zxYQ++6bSrGYNjnyHjI
Id0mLQYJPK61E9uEmeRV6Tf4/Wre05DUgYLyD451dwkjybv4W8vrvdJ+8kcfKt+Xz15jfSB2SNvJrY5BTyDD7
biNd90pV8TpFDatAXfyVMFiaLJtE1bhYPQBncJ9VLvdIReYX8a0064X7DEeM/cY4cZBLmDcIr1Y08wCSQHOIov
LJZ0FjhugG+7x+QQcA6d7++FhHu2/oacP9xfS3h+j19LZMk8gN9bcA69bq+ZtX6TX1tIRnkqL50TvmmeJseWsn
VIx/W1zZiGfkLfW0n0sh3kKIgmZGhwpHv6WuBjHLY9bVITI6J+hpwv0BfS7h262uZjHHcqK8NjNUR0dcmkupYq
K8tZLbjSX2dnDvb8b6+tpLW0Rn62kZGzVmvr+3EN0cbpcHQ6nBgeWtUmdScpxQVFk5Vm1YrJYFoJBr2e9udSkV
Hc75S3Nam1DKoiFLrj/jDXX5fPj3r6HR210Pta18R7FiulHhbz3GwzL/CW9+pNLd605b7I4o37FcCHUqos6kt0
Kz4gu3eQEeMps7bESkJB1cm3CYs6/3hSCDYoRT1T52mbScAtAQ7kGoUhWiNRk0zCwp8uN/VmR8Jdoab/S3B8HJ
/foc/Ws7BGA9MigHBlukRv19p8rcFu/PylQvgOF+Z17Y61BpRAu2hYDjq9ykt4WC7Uhz2d+msxGhwDXVqGkokQ
2mc0krmVTTWBtRMp5z3h55tkAu2pTKEciBCvUo07PX5273h1UqwZSgWSmv84fZAhKs/EFFa/WE/01oe9nag6E6
UHcXY6gx1LNt1QYVb8dqJYQGwwPBpihqLIAq8CrNyDRFyGirP6an5uZgewjBGUC0FbGjlv0dEdReD1dJTh4i8
yneSTYHPAiPeoLNne2+zui3ijjpyXQhkaaxDDyA0pdsCXajerPyeOchP2hcNDX2ezaHwBFCzQ1Bn1Mx7ooAN
ONHNzW6ePcdIdiLYG06PITHtAJ8QohDVVItr0CM1zcZxKu59JTbmDRFqdCTScjGZBMKxE/GgHhA4gq7r4Q0g5
hBtiCk6SjXVcULdrehYZx1gZmjPdhcgt8/6AsqkaBTiXQ2rfA3R9k0k6812Ib0xgRqDnb4Aky0yGxKPYj02xT
s8nMJNC/iDAw4QUcwimaIaLvMKqG4B2jP1Eirt62NNv11rSEbGCxeQXIG09Avwkp7M0wfVmwlujrkb/EioXyNq
cFP272rMVrwu/CQEoC05m2LouvhApF6fT4uuuaY6FqDeMPLV2eYNU0b1548E1ndwNpZrsYqHmId6mxFJhJ2I8RM
ZSomhpEiAK8zbNjwC/UyMjzg2ZK+jbbUSSHbzysQJ+9nfWHFytogwRTK7xMLDjz7nD/ND3cGwL6LkDMRHdQmde
0BzWNjmcJWhZSr1eGnyYyQxrJ1oA6aTrmBggDH/qihGj0INhTC8vE1tfvZAkx0xswWNG6XVG1VavRHE608YpBP
mdXHv9imdhT6d4Tir1DOnSXg+q0aCbsyqudmYkbxKG8seGCsxnJC3eaV30QqGcdgRpMxVP51TDSKFCQtZ9Le1M
Kbmu5Xy6iqPUlhd711SX0tWKuqUmtrq+ooyd5mSU1yH9z10ZUmFZ371Yo+CELXFVZ4GpbpcKa5qUBZWVJU5Ff
Sm1p3XR2tr1UqFtVUVrhxr6KqtHJxWUXVPKUEz1VVe5TKikUVHkTqqeZHdVQV7jqGbJG7tnQ+3haXVFRWeBqct
LzCU4U4kblapVipKa71VJQuriyuVWoW19ZU17kRRxmiraqoKq9FKu5FbhQCEZVW1zTUVsyb73HiIQ9u0qmnrj
Mvai4dqFTQWTVKHKtwkHykUvEobj2eG6+cWV1UpJhafOU+suXsRgmXbmVVUvctPy6sVVZcWeiuoqpcSNohSXV
Lo13LCU0sriikVOpax4UfE8Jk6MCAPTxImrg7ID89xV7tr1SqdSV+MuR WAL1GNFrBvUwyFR96iJSs5uaXVvnf
6xbiBcDESTrpkvpvputQAGK8U8p54yLX4XiMjye6l1rPACtLKurcTqW4tqKOWaS8thrZZfasLucesBj1yYxXpfPLb
MT2zvY0hGKndQHL3MWViLC0sYEbdBAsepd7VbM/FGW+rQe31hp5GtVyp5N7rZYE0IXndWDgant8iWUJ14tXHS2
7xQs2K8d0LfxY9IHejZVIS72+Lj9mwAhLJcEwDbJk0h2I8EjHETge1GqeEvG2ITE8xaKIQ2Gu9LbhscgAm4MCi
saKYSGcwCPd4UAUk4ni7cTdc0BmvQyH9TLFJVDiEjAq8eSg8R/2R0JYpQJd/rbV+QgbZrWMcxLowF6tXRedq68
50jvWkkSV5Ry5Lxi12NH1K5TyjuuSW6cL7WUvTx9EtT5IuZg+iMbj7IOUi+yB6dh+kJ/lmjikSqnDNKjxh0VeS
q+kxH0l+sXolahmh0+tV6JawF5Sr0QvY69E472ScpG9Eh3UF1xEr0TP1SspF94r0YReKTF8B7VLWM8xSVyudon
q7ZJySe0SHcQuf2+83C0T7Qgql9wy0cvaM1G9ZVIuvnWiQ1sm5WJaJjpsy6R8kpaJeorrFy2oZmwXz7+o7ojGJ
b+U7ojGujPlUrojmtgdKRFvHdFhuyPlUroj5qyDAmWg8aHnbHyUT9D40PM3PsoFND6UNz6De4d/3NBEY/Au3jT
QfLzkn3dyVdAdWBkoCGAGWZUfag0V6G1sy0SM1J1gCZHVJEWcZD1pJVGikEmkmeThtYgU4mcqrpoQQiE1CBM1E
fwNE/zxknbixN0K0oHw+bgqJm34UUjtAK4Iv/Pj1Y9nuvDbh5D0AqhOH6DqQUpdSIv9s7U0hGZ8ePHMJ6NYhqs
VeK6edCJEM8J60TY/P+H1EimIpQ0/QwjThHgDCKfg+SBS9/JnQ/HUcSwR5CiIn5XneDr8bj3nMIJ4g5xqEfI51
UwbBD08hhZ+QpM1qluCyR5FzmeTAzv4dPgUhM9HuCBewyinN58Nc7nzEYcfz5QnY1vpIWaLsy30njHd+r19/Ki
1I01GWGaNy6NjhmkPlmNMK38ZACfhTjfUW5PpoEwP8E8gGhtGqKVoxLEfahzkA+dSxr2j6mGk12zmRdXiV072
5spmX1JH3pBEXL543J4e8d1DuAtYldRvs08rJ3reixuBdEC/4gXJ1kNx9f0scW9P8B5auXP/LpcyzmVDt3qTt3
umrU0apqPaf7s5HwFufU7+PmQHmEahSBijeo+FtC9wMtxaJqmOs4o52KoPzVz00aHGvYYBgt8a75sp/Hq+Z70
Q1eksMtx876+DXC+WrGM15dPsqj0Bk9tJ1jifInMf204KpNj6RJAzzGKbC8wviPov9q3s8oxnXcdkI8anxIoZm
fjnHj4xJEua814dMof6rRo0eh4NSjuRk56+RYNJ10cx9o5Vknqumne81ShSTITzIKzVu07k0nQnWYet2bk/N1
jQhg0TwtPMccjgH5CzgGUTHmLV40HAHdK00tv75pY5pTuM2NODRUc5X30viEnVzfbRfEIVYNLTwrN2hS+hPo0j
j34yGk1+ZJ1YgRDPHp8HE7Mf8uE3PbDELNXPaPs5xQ0d0No90j86dFzEGeWaI2yAxF8U1cHYm6ED4qB4NkUGws
ViJaywxBySeU7jMXs455b15sK9p2tBqifc89gzyKqfotm/n13j+uBBBhRhk1YpXTq0uUP0hT5zvLdLJary0adab
zFs6jT/ekNu6n4YEdjV0mU1+CzR09L1ZBvbwiBnj0aON3dEAiH+eU2asjQRvLB9VVjVIsh3q592i+G6MxVD+Rf
yhTjEuqSxD3MC+30YVzMJj0UH0Mx5tTt3cbPxc4RzanA9YJ8zzr5Xk1jje2ExnwyFi8Dk0efj3P+bkUMUrdXCo
fP58zTD3MGZB76AmKz2LVNifBy7SYqRxSX5p4vAcTe03U4yDmJ134NDCMxvxkFddzhx7JIfxo1cvLM6p/4ESi3

TWeYzt02Ehp5R1e4deIzq0fe9K5/CSW64bL3T5eCTq43RP1NZxWaYLmEm14sbEa4VkjVqvj0RaLJNY5tA30HmH9xGCMIE7RK/F7uW4xrR4yf6IDWfXTzFTnlpqJj5GoXg9bBjQ1n7g5nWpShXeMTjXeeegS7CNr+bMK3F0wj6vFJ/V4V4a7ZdwuxfwJe57Do3EJrhGarkY49Jw10I3w92A0wy3wu/Z3UKER0Jc7KybL0U03IitDjmrxjXDvQh3K/Hq1uHYiVLcWYZ3bD2PsC5Uo1eFpz8dtg5xovGqQf341QHc1XBKcY4W4R3tYh/vv60GHFXcHyMfyfvj9i6SudT01wtx850xDAnXK1USW/Y7uL8VqDcHVcn8VcZo3bKi5D0T7XZHFzDjRLaByV4rUGaTOIeciXh2uBUfLokE5uRyZPGT/PqC7kUBpn1bqV2Tq0JV/XpcYH03/9AOU6Ln81fhQuvwd3PNw2xYg/hjfm0/M4BsY35dpYz0Ur5nqo5hRKO BzT1tNn5YDH1S2YpZTr19mNcV7GKRVzjdQNK0kMW6J1hvM00kBhHpfPzTVVyaHrUI9uhK8Y2NH8sYLLWqrrWs0p+b3mE5UJ2i31MjLLXo9U3bpPFXPdDzA2WkJ5z8uhWaBYv27NEFncetX6daN8ePh1D3DaGUJj0U3hyrmq4biJFyHr+LdM4XD3hYPAcs1v2zeoCzwfqNxVEM7kJyh4YrRnuwBcu4P1XqHNYNaE0DoOfBq+UuN9a1Zv6eEx3I24Mrd2LXG09GE/t0Z0KuTewEtCw8j802D4GL72pvS1rNir/rJPZuw71hx960tV4+1vXGuw8td2vvRIldr4/351oPGBnoSoK8Dww0dCbd/Gm8pof02Ulw0Hseo+z1td85QCtWi+K4tL7Sy7sFRi0yjDbPXaHoWW+GIV7vNSrdfB3V0xMmX6c0y/ZvHvI2HJv/nG0DZVgbxGQZrnNI1H+Y2zukv0sFuIZZP5mv4w2T2HtZXcdMA9pcrX2I1ePex7DNJk0nCkwHyxM493FdU6LN6BhNyvNVbMb1+U+dLvd9os0D6KD5kFD069Pbx5Eh50HKZ/xPIhe0DxocCffnMBTfNYRg7ywCepwExb6uc2V1LPmSvR/50oJc6X4h0G/51yJDqqwn99ciQ7ztvZFMcvRYedKcYk+m7kSPc+84LOZK1HySedK8b91upxzpXi8DZ4rnav6nnu6pL2fa53EF226RMng6dLw043PZrpEz6NdJUGDX+wpE+U+dnY3891PmegXeMpEh0yZ4u+6n+WUif7DKZPymU2Z6CeYMif2pSJch3UI9YFnFtN28X4/L0bHdFhb55zY7oWbMj5X0bHdFzz07iM6BPf3ZEP8Hs6Hx4P93ZUSyznruinD3xoRcx8Umc0lz0iQ+9pInP2e9sFzfxoQkTn/PNHS7HhCZ6Fn4XiU8aKKfD7vIv4d9cFXC9rMTfAs6bj3dN+bx/DeHe4G7s/P/mjP+fZe3/cN9Kbidd/PSJ61xnP1bhZCr8PRf+VgT/2Qt/tcJfVDihwn/kwp+t80+9cDwX/nRXsfwnFY71wke90H8S/ngS/r8KH86GP5TAByr8vyL4/dE6+fe9cBQbj9bB+78rkN8/Cb8rgPdU+K0K7xbBv6XC073wryr8JgV+vRbefhF+pcIvEfyXa+GtI/Pkt9bCkXnw5i/GyW+q8Itx8H9U+LkKP1PhX1Q43AuHDmbKh1Q4mAlvFMHrKry63i6/mg4/GQUHVPixCj9s4RUVX1bhhyq8pMIPVNivwosq7LPD3g25814V+154Ue5T4YXn18kvvAgvrJ0e/36u/Pwy1x143iV9Pxf2qPC9Xnh0hWdV2K3Cd1V4xgffscLTT+XKT/vgqV0p8105sCsFnkSmnzwJt6jwuAo7VXgsBXao801HrFk3i+BRK3zLB9sRZhsvPKLCw99Mkh9W4ZtJ8NA30uSHfPCNB23yN9LgQRs8Q0F+Fbb1JsvbV0hNhq14aGsv3HevVb5vEttxrhXt0wtfvf1H+ugp3b1km3/0i3L102vLPufKWzbDFJf1zLnxNhc1fzCz3q/DVfLgLxbyrGDbdazE3pcKdFt1Gxt9sAE1tSEX1tvhn1S443a7fIcKt9vhNhXWqdCjguvMrWvXyreqsHYt30KDNR6HvCYXb1ZhtQqrrNCdBF0U01WIInoTISQifhJt0QkiFoAodKrr1w0oVvthL5BV1EFchdS0sx5sWFFwq+FRoVqFJBe9saDwJX06CZSrCqMINKjQspXLDSVhKYcmoNH1JEdSrsBgpLy4BjwPqBjtcNwZqU+H6BSP161WosUC1C1WLbHKVCotsUknCQnnyUIUFFT5wUioyEiWK2wwPxnmqVDeC+5eKFOhVJwi156EkeheCG4VJirwnXXpsjXpcK114yQr02Ba+Yky9e4zoyA0ckwW4VZKsyckSrPPAkzptvkGakwfZpFnm6DaRa40h0mJkPRVRa5SIWrLFBYYJELk6HAAv1TzHK+DaaYwVkeV0701a/0weS8FhlyLuSlwKSJufKkYpiYCxNyLfKEEZBrgStUGK9CzgjIRjmzU0DxQdZJyEQRMn2QkQzpMF0FcadhLE1kIY3aSqM8cFo1NRoFUbhoVFp4FAhVYWRKqQgQIoKdpTVxgK2tTDCB1YVkpNGyckqJCF00i1wqEBtYFbBhGAmyFyypYPCBhA819AAH4C6oI0K90AUEGxAvhD7Bt/5rwpX/HX7i583AeX8y/gvbpQKHCmVuZHN0cmVhbQplbmRvYmoKcJzIDAgb2JqCjkMjMKZW5kb2JqCgoyNCAwIG9iago8PC9UeXB1L0ZvbnREZxNjcm1wdG9yL0ZvbnROYW1LL0NBQUFBQStEZWphVnVTYW5zCi9GbGFncyA0Ci9Gb250QkJveFstMTAyMCAtNDYyIDE30TigMTizM10vSRhbG1jQW5nbGUgMAovQXnjZw50IDkyOAvRGVzY2VudCAtMj1C19DYXBIZWlnaHQgMTizMgovU3R1bVYg0DAKL0ZvbnR6aWx1MiAyMiAwIFIKPj4KZW5kb2JqCgoyNSAwIG9iago8PC9C9MZw5ndGggMzE2L0ZpbHR1ci9GbGF0ZUR1Y29kZT4+CnN0cmVhbQp4nF2STW6DMBCF95zCy3QRYSCEVkJIKQkSi/6otAcg9pBaKsYyZMht651JW6kL0Dea98aPMXhdH1tr1vjVT6qDRQzGag/zdPUKxBkuxkZJKrRRy62itxp7F8XB263zAmNrh6kso/gt90bFr2Jz0NMZ7qL4xWvw17E5qPuQt1dnfuCEewiZFRVQsMQ5jz17rkfISbXttWhbZZ1Gyx/gvfVgUipTjikMjTMr1fge3uBqjSyEmXTVBFY/a+XSracB/XZ+yBNg1TK3X0VOCXe58gZc428Y26Qc+JUIu+J8wK5IC55HwW75Af1LME+cBmkkfW05yamTic0Q95TzzngNww45xE8r17ZM5fY0aE8+cZ8i3/CZnzFxkt5Pb1uBq8u5+VC3X1PqybLpj2jBs2FsTvT+AmhzZ6vgFvv5oEcMvUZHN0cmVhbQplbmRvYmoKcJzIDAgb2JqCjw8L1R5cGUvRm9udC9TdWJ0eXB1L1RydWVUeXB1L0Jhc2VGb250L0NBQUFBQStEZWphVnVTYW5zCi9GaXJzdEnoYXIgMAovTGFzdEnoYX1gMjAKL1dpZHRoc1s2MDAgNzUxIDYxNSAyNzcgNjExIDMxNyA50DggNDExIDYzNCA2MzYgNjM2IDYzNCA10TeGnzcwIDI3NyA2MTIKNjg0IDYzMyA2MzQgNjMzIDUyMCBdci9Gb250RGVzY3JpcHRvcIAyNCAwIFIKL1RvW5pY29kZSAyNSAwIFIKPj4KZW5kb2JqCgoyNyAwIG9iago8PC9MZW5ndGggMjggMCBSL0ZpbHR1ci9GbGF0ZUR1Y29kZS9MZw5ndGgxIDE1NTYwPj4Kc3RyZWFtCnic5XsLeFNVvu/ae+2d97NN2qaldPdFKYS201KggJi0TUuktJ02PCoykDZjk/SRkKSUtiKoRxEBGQTxMYwyDmdmDuPR4jgeFGTQqXp1ZLx3x0P16HzHGXFm9FS+++D4zUXY3P9aeydNK6Lf0fd+537f7TbZa6/1X//H7/9Y/x01ERvyIx3aiTBy9Ax4o46b1lQjhN5CiMno2ZoQLv2sJgTjj+BzKBDtHTj3r3d9iBBOAM2e3v6RgJlyrbcR4uDRXBL0e30jz4wsQchaAh0LgjBRIw4q4fk2eC4Jd1S2/UXNrIfnnfDs6Y/0eJfP+20pPJ+HZ/uAd1v0NFMPzLLgEQmD3gF/7Wfvt8KzgFBhWTQST9yK7ryG00KVZD0a80e

bFI1yeA4ixH8P5hi4yJ80hgryzGKOVyhVao1WpzcYTeaMTAv6/+mP0aA4YtFJ9Bu43oFRCwqjLWgX0grjT9AYnX+ZU5ELz7nCe076CRTDnQsuZhKYM0iM8CnBtY+AfoAPB9F3XT9mJ5Hr8fwOXYYsbgdRu10x1F0Ei/j0Hx0+tBdv0E3oRfRcTLmz6HDQNeB3o0rHrjfkgk6j95m70THmXbQd7UNxElUon9Hw74IuYdTNv0uv/4Zuo5LJXJh/V2EBSWGw8zRwPybNM+VMB96MAXBrY2DBcbwSZu9BYW4zXGX0aqT2STaw7BjI1+1F59jb2HKujDkOcoiMc8D/OLoJ9A2ApivhwxL98afoK1Y4Qzb+ZXSL8haFj1EotyMPRI7hGuYxRT7ajLzJ3DBohb19qJY5D6QgC3pewXOYZZBdMI2zpW7fu00764X/1FW4wD7jUTAphXHUPq4fEU5eu9a+nsvju8b5We04VDX01Rb/4esW/7DAvqp9vTD+hqtR5ura3AhznethSJ5gGuZdjQuInwLiYS7AH4MKoES5Dh3301K8zqj4jSyHKifOT34Hmc5Pnp+syjQXmksLzYUBD12J47wrn4iH1Ya//o+YohxybgtCit3gpzz0PUdF9k61UavEOKLnu21WM2Yi6jsykUVXos1V1KgyDCVZGr0JQyU5KjNb0Mt0abL6QvV58mWi/3wHVS6/eDG7+ov18GW6W0VQC/nR/J35v8rnmi01tTXm4tpCazFoI91rrIuWVwj3LWf0sM1nsIJ8n7n6wpkr18+EedTx/UKy/X1fYnCYfaq3eEeq3XPubWQgxYQONWx2xWZR1Fqkf5/bqc/egHGxt1m2ep1NzclMfkGudRDQGCtwCJC1cmJ03iBdPFjLoqR4aQ78jfkY/fRm8zb7NvW9/04zcyG51CqyWrpnrR4hoDU1yEaheimqkrICxQsmtvflnJx8f+UHBIfvnL38qfsJk/fG/M+x5xYH7djxoYpHqvQ92bH9mnKkRrzALxfd0vPjSmRdoEqN1oHQHRLoaORwZi1HVFg6Nsowd1Smxna/SmC5dqL5yAYCbrL4yWfXsf0Deqwz3rWv4Q0184CAw1cjiXmxUu6SiXAADzm2NNXPz1xggmFw2HeKc11DLhoUTaa78hE+3WK/QSOHJxrbVQDGdkAxqVJCQnwzXOCLWJjmY2l1YtqF84BE8H27KTRVgtig4cefVjQoYcf0fgX8U+M7dNPgZv456bPLz49NMLFz7rAutFplr8rSiK56idGoRUNRBHFuawYwP2aXVa1sew0riZMKs36BmTwaA3m1ijz4IsBp/eAPmkNem0HM+p1KzGxPPcao0WxSjTqMZ8xKxTdmNkMRkNep1Wo5LSzpihsJr0v5VdRwLt0vL1k9nV5uy6Ksg0pYn/PLtaaVJ9Ln2kJ/7z9JmuomcLsiR4Mwm8VgqvFS4JXsc/6VmN1jfksgVsgTbXUMqWa0sMN7PLtbWGFm2TYT17m9zJGND6DHd0E4YsFaNgMVar1GqlVqlXGVRGG0RdFphr5Wwqm9qizdT16G2GOWg0M4edg8u5Ev0cQy27EC/SLtQt1ruRm3Gx9biJc2mbdY361YYNaA0zn12HN3DrtV26DYZNxk2mTeZNGZsyN1k2ZQWQn/GzfhzkevmAKqDu1fRqAzqf3m8IGs0msDloGWLjeCs/pBpSb9UMaYd1Cf2I8XbT7eYRy9qNaGNhsZopZmjmqZkatTRS2I6Jx8Xjx8SRQ2zh1xixvMkcytT9xTz9EHuzJf1fBi7wT34pd13G19fdTPt0AP8Q7EWQ56yFGkUmfy4JZRM+Kz9urVo+Yj4Ct2ATLZM1R10ruhygbxDfXh/KRpAnx28WK1+Nprpoum18x1ZkjDJSZkYkyAmIkz8SarKcuUbcqpzHUgB+NgHdjb0fh6qyPLke3IsYJolhALjMAqNk41iY160QaX5EXGXewz1mp0ZYtBVNTnYXfcf0tq+47807Vcz8Z4KLjo8u60HKxS/aWk+Pi7nhMPaExor0GZK59DHkrAUvojbHnCzOZMXIdkhfcEhpHs3cgkaVm/V72Qy71YTLIJVrZ+Vq5hWR9LpwzflymswXJkmSmS5CeIKdalRsKt5RPF4MZZAmMakvb03CDFArm2i1Uyki5YerGRcfWfTTsYn/9ddXx3666GkmcPLg3gc0HHhg78Fw+Ce+Tcziy5eZRzt8bHn4y9+EfvXmR7//8I9vUN9cBt9cg1NWg7Y5ZnE+Xsfjn4JfjRUcwytkCkia01Eozh1it6fzExJUJWnUuTVBfkBz1IE04SBOAvk0g1SIHMMU17bWzUoKoBSQwlepKDUfg11IaCr+W0FKawtpCM19bCmXdCtHGi8w/fczUj+6dd3T14eDUs1cQ/0na2g5yy021GosyCjHiVYX66fD+mHzRa9LhtpmGyFyqAxMRnafN01CXKMAZh1dSTzL35xUbxoArUBWW2TpknbpGuyNFl5i1iM3E1Z7XAgc7Qw6WWcFI4DKKaN/7wU8Zg/jbHJ24i40MXbmJefXwzh/ev/8It4XR1C3/4ysfiG9nw9pJ8dgAM3H4y/sfPv4AiYd141ruJMS8DhWjNQ47g8z5WM8dytWPGotmjwqaQ5a9uZuFu4xHSgrsSJmbXVtos0ebUJm+qoScQHA+Xpo0QVYJidp8a10hcZz00rHS6EKM5LrJU3LkjGbYbWwxUVzyqRQZo4d3LvvwIF9ew+ehMpPAuSvf50gAQJHAbsydPaNP374+4/e/GUfhIj45uXL4hubfGIeZC3JdgQuKIPyN07oJTX4CYvVYnhCb7DCbYXRofFoptTpWv0K1UmKWZ5QKBXoiTykhNsKqJBZVlsum7ciK8u62mpQIk6Vm8VjZjQHH8k06rBq1KzJ11JpdYqoUjrMjh1RvYsCLEPj8xSqqYY6feVbFOuZczQsD6S6Bsg1GbRUJsF11yvP+kQxmbFeYZSNM/qjNyGdWidssuw3hpAAWXQELOGEas96A9BouGUbNqrNYrjLyNz1Xk5TK5bC62aW26XL3VaLPZcq15c1E5Y2ftuFRbqpurLzPMMc4xzTGX28pzy/Lq2JvZ0nwzNKNuptnQZGz0u5XpYrtlwIE6tcZ1hjXZq21rc1dmxdmwmwY9yqD2qAuq081+I2BvG3oduZ29nZ8u2JEP2xIGB0mhDlh2WbdmjVkg8odyrsX7Vbu1e7V7dbvNuwy7rLca73Xdm/uvXkboGozmcmKTUJAK93k4s16/nk/8+g+8W0mZY9o3fXnv9wv9u9jSsWX7mY+2MHuZmr6+sRhcXE4zKwQX4HvQ8wZUsRZ0vPx1+CsVsHd5PDp1NGDN0aFFErGasRcYYMvboEmbWFWRc8pL8jeVcnFYvJaqhmVdlt2Tuzn8im1UxAzhMqLGakUis9Mzdfh0RfLL4pfvi0Ay9XQdzF/OM2C7uCDMtZCK4V14+TBo8cav4I/GomEDstWViHfMs/zEUktko5CiabcrWYXV2Vt7+TLv1HAK84cFOQuyEFZkLTkW6e26qgI4Wa6Qg+Uiyasrkx0mi//4mum35gzIqucrhc1CVMBTp0QBjZ0CuKTYeS6eHc9h0cbMhYugBlstCkgzgrSzjspqa2YzJB3PNVRWNDirKhsWQBX+9c9X3r1ik3JNx4Z1/JaKxsaKyvp6cbTvyy3HThUWfrD51rE7k2fHHqgVGjQX3eGozcvU2pBJx4zaDqr2ZxY+ULxv9v45ezM3g02sjUeFC4r09lwjKtPyuUVZ88qh1kGpg4oB9SFQdnYXxItf/MkERQ/cYK4j3WuJaV5k3ra5Z+dxo3PPMmfZs3iCm+AnFBPWiayJ7LM5Z21nc/XkiKxNnSmZ5uSxM6cWTC6UCo2yuFY6Gis/333vfYn4vpPHxGM3He078Z8vvs+wB+8ZvMu0afMv1/7+z0zrH7a0Ru56gD199Xfh+c1Np4/+7fPukXt83e+V139Aa/q1SYipTyCmD0hBRx10a6wPqgrjQwwUFTi2GbVGrTewyKTRqFczKg30Tii47pVeo2aVB0GqY281gjdNvQzLtF68bX1Ano5Jc0gjdCCMCqWww0kT1MDpujnjWfZzjLztPPM+iVTBzTxmBAQs3ShJH7HybAvMCc/Bnz/IGr//LU1YsPsQIf/vJebrSv7/Jh0vHQm2ox+PEDqJk2NOQoyjYrdHoLe486B0o+0qLezCF7N1l+G7cxSg91UpqvMJa1Cjn9y+sNtEkp8Nc2bquRjmUtP01yUK5+mz+3IG89jUxGaSyM015BJfUxhmsvMNDTpCWEuZt911j8tP1bzZN+jn/6E2bRpd0e9R7myKw3h0+HIG++xdX1XH3rl+6d/wXxM7XgK7PgEztppyp

sxRD11HSaEJevPZBWY+YILufLVWJxRyWcw050MHyJ32WH25gxmhss3zBK2uMF+VbbSyRqQqtypL8suRkZ1HA/S
C1NtkSHXBd0ES7XbEi6n6bvgchvINXGWaD223QyvMr5rfPj86f+d8npid4SGN+IZV4xZa3udRf0ahecluIzFXU
64t15XrywvKhfLC8qI16jruNUu1S3VJ9naFu9tKCpcLSwrqijCvujVt7i86tX1ngFtyF7qIuTRf0z136LqGrsKs
or07X9Gv7dX36voJ+ob+wvhUPKYZ047pxvQjhrGCMWGscKxopHjx0fnj81kouYvN2VL2L5xTyVRA91AitWLSW
2nJKmBmw3HNFbNORPnT/3q2VNbnn/1H59hDJeCf927xXzil4//7e1vvPCbvcx/Ez8fu7aS8TDHf7xvV39i8TL
nK3935ncL5ouvPL7vzrG+0eWLb3rpifc/JD/aQd2AvprdTOuffbkd2VidMw05YoIyATUP611ZptFursqSXhfJq
z19YTz/FrwyZqe9MmbRKMqCS4qiafUt1RC/TspafWVlw788ffWzcWiGU/Xsy7nQCHf0yX3+FuIBZ6EtjjnKLJS
hGVWhWdx+2xZQyqI6Y1DrmQUIjvwytAh0s5qroFFbfqEagn+CtGqkdFVDv0+q8sVqUpWf3T87TdN8qmk+XJKm/
7AZbWbYm2cz5NCTsM6HF/LCVBtfU0ttycZbvExnW6+HKRffm7z61Pj4+Au/aLi3nretat1yYHfflaepBfc8+3y
0DdHffNmuRxcE9m7bZfz+r6hARX9b+91/1T+U/J2NnDyqGgX5cVE19eMbQsoBMT/957gZP89ZuXMo0L iMtnCnU
YDdj5Zx/wXu0Nmqq6pAGv4Nu4QtQgM9G1+GzB8bLuLuRTRFGW9ixa8u4V2Ht9WuTfBlazI+hp2D/LewXBHNUhEb
R7xgF08Q0sUvZp7AK34af4yz8LD7Av6loUjytzFD+SVWhepBqZIWEYZEUQSb0KFJcp6dl/ idYQn6Cy2VWpPTel
LIBWjR4YuRdH0qTxxjmI/KYg/GYPOZhvEseKyA+D8pjFTKjx+WxFvz4jDzWqw+it+SxAS3UJGWZkE6T3GtGnOY
JkMhwang6o/mxPGZQjtYrj1mk0o7IYwzzd8pjDsaPymMexifksQJlad+QxypUpP1neaxFS3WMPNZnztEtk8cGF
CxIyjKhnILkXjNSFFyhIRIdiYV6gw1hbk+5UF1VVSN0jwj1oUQ8EfN7B+yCe7CnQnD29wsdhCoudPjj/thWv69
C0+gPe9c0CT1B72CvPy54Y34hNChEh7r7Qz2CLzLgDQ0maTr9sVCgPhLpS390H6/1x+KhyKBQXVFTK82nkWQig
yA5AfoEE4no0spKH8xvHaqIR4ZiPf5AJNbrrxj0J5ooGdGD2JKyQZgb9/uFbn9/ZL18QvgWWlndoNF0bQT0vIHF
0YavZcMM/jebfjqqwQ3IIVBQSMa/PP+CN9QmRwEwuGk27PzYQilP4gDroj/lBVm/M05jw++xCIAbGwzYwGGCyc
4mI4B0cEaIA0GyIdCfA4NBgL0jpAaUJZSL0lxH39vREBqJATggSqeAOIPkH4wBwEYWkqByY+QRvPB7pCXlBHID
YMzTgH0x4E0SfQKgfMJ5LONINQmcckBggzIvKqSYxfzQW8Q31+CkbXwgMC3UPJfxUh2kb70C1nv4h9Fk0JQIR
oYSoMxASBZE6GMS1MB2KA70xBY7MOCnVLP/xoP2NB12IrMyEhPifvADUiAVdn8GaKJcsA2SoB0yNBRQcPByMB
XNxA3B1ZigyDQTzf6IKI18YhfiQ91hf0+CzEgY90NIEoN6Io0+ELEjv1sJ8cCStzuy1U8tKkKIkpAKgsFIAtwQ1
2aJV6JTESCtCfGgF4zq9suogRoQ5N5pdkYGIS5iwkAk5r+u2Uj1j0oPeEFQhaTU9NUB7wjhPxDxhQIhEmje/gS
EHgyAqdfno5ZL0JH88sZAr6F+b4wK8vnjod5BqkZv/0g0GcebSIR6e4Bjhn0xI6h0fKUmKOJ8EmLc/jcEMjvK+p
C5THEHwf4RITQt1MGkmJ/8+zxKSwzxAibxTTJF/BB3fsmA4UjMFxeKUr1YRGQnF4QikrpFFDbwToucM91+yCb
CdQj8QIzYggm1FPNvS0DWCN5oFFLM293vJwuS/cB5hm0C3oQ9MaBo39w0i4gbirCfcLQoE9WuGh6XSmSLLyRZ
+0RfpLZ1HXEU6hn1QQyJckYdTb0+ftBcMgFwCjqr7QNrmigoWqCivz9A1FrpEpraWj1CZ1uTZ52zwyW404X
2jra17kZXo1Dk7ITnIruwzu1Z2bbGfBh7PV0yW0NQn01i5h1bu10S641r3d3uDo7hbY0wb26vcXtgj13a0PLm
kZ3a7NQD/ta2zxCi3u12wNMPW10q8zK7eokzFa70hpWwq0z3t3i9nTzhSa3p5XwbAKmTqHd2eFxN6xpcXYI7Ws
62ts6XcCjEdi2ulub0kCKa7ULjABGDW3tXR3u5pUe02zywKRd8HQ4G12rnR2r7ETDNjC5Q6AkFaA18BBca8nmz
pX01hah3u3p9HS4nKsJLUGnubVtNcFoTwuj0+NuaxXqXWCKs77FJekGpjS00N2r7UKjc7Wz2dU5JYSQyeZMwUE
2NLtaXR30FrVQ2e5qcJMB40jucDV4KCVgD0i0UHUb21o7Xd9dAxNALxQBDlnpoilAACf800A1o+a3grmEj6etw
5NSZZ2702UXnB3uTqJCU0cbqEv8CTuIjWsAT+K8V11f4iMy99XoACqyWzaw0eVsAYadRI2v0EJ0ubb1+KMJEtt
yckv1kZZSqX7aadRKRQBCuHkQE1eo00Iz8gsevJIFW4quciRbJfLLykfEN1wGkn117fVD1UwTkoJ5EeEFJPhU
Jxm0hyDAxH53It7+0EY7EpRQb309s02eErN6QmVPBcjRsBsGY6FE1BMB08QzMZCo/JRHJ0PqpkWECKz9Y/541E
4qUJb/f0jFUAbI+cZ1SQ0C03WgGw6ha8nsTRZQxNCL2XuA80hKasQNDfs1yqHQ32hyhDUqG0V0WC0U16UM3tG1
ADtehSNoBgKoV4URAkkoLmoB5XDVrPvWvVUDo26gEFA90CRQHD4x5EdenIDsM0tGg0BfASMn6odLQB0pXnH65Ie
7H/ZshW8fUGpQI4zCwGEtGgKKHqD1ApdeSinAmPAXgMsfEeBphv4hoB0gP0Rk0ulazP5dFIPIRQATSNw9X3t+
tfNr6Wj0IwjVHY1aFuDaqfRfx2XAN0j2ZyQ8SEYJMCpagSLp9MvxXoK4AuAvcYWOWne2PU/grg4Yc9TWnckng
k/fJVP5A1grGf+soPaEXQMNAsr/yfbwqC/jvk60mWMPCKF3nr8aVbI34d1xE+n9ErF4f7SmbQzKKA133Uh8PU
FT7YC4Cnv0mXYh17ZTfAOU2FX0S7yBd88t29VIpqzTCfJRPgK76U9IKD0vRZKd6Ra1Gg3R/VI5wSUIEuCZkD4d
oVEi29MhIJ3kmqBbTY9wLVD00QqIy9yQHQi3pLkWSH+bjcgQXpUVJfUc2eu9jzVqwf2eGX7pBjsgagcoFwSd
CWJTwBG/XIcz0p0CWBZDbRPwG5IMU5kTfCZmJwncEpAxRPae08VELEjTwumE1QVeTMr5eg130pR7QbIhykTA
ZpjEQpDmfkJEZoHPpFiX5x6ZFpaTtEMXQnuYdMh6g/kz6eip/47Db/jV22FN2VtK6I1DOUj5IvEMyqt09f20rk
8hJ2kZTE2YEXVTFg1TPAa+1YRkNgRozRyULfSnSfTRbyLDTu8EiTBQ9FB+Ek16HPfLVTLpoR4q20c1DsmaLqX
Z6ZF3eYfjhFaGKR+k16IpBL5aCQaBPiFnQ3wabTJXphBLrwHp+wRqs1f2VHeqbidjTUJDquTeG/gzQs8YQfb9A
L1P1Y9v44sEWB6155ZTqhiG1I32kswGUnpP0CzL0Rz0VnRi04JuepJM5KmBFNfms/Toy55fhEpE15DwMVL9yU
t81FNib8G09DoBTpiTCe16XVUC+NHi12kzJm4hP/RpvSa5xvWoR5qY+up8GNNZkubyYu19PRLvu9n+4L3aCqx
+QK5Kf6DUzjm5yJpyIzmTczTxG/X0/80zwwTK3y0f1F1zkXi1J2z9xB6J0nb1FatEm50zLjn0mmeR9J03VIzoe
kJ7bCaug6iPnRnorzoJzRUbikU8xLk6s/tSPd/5L0N86YIK30Ar3HZR39NKK+P14k665Xw8nqEKWajvD1UBXSk
Ev34b81Z+00eibP7KmsS2YU6SD6Uz1ITN4xnWOURnQffPfKHpP0xUGK7cz+4/96xfp6q7r1HENI52IghdRK5KJ

y21ArPBE5bfDkQeugn+yga26YE6Cf64CVtfdUCLON1C90ukLWi2g2roMx4di61lBeEo80+Ca8u2C68BboM3laB
fStwIvsdaH1VIYLuHVSy7KezXMtsDdJdORHQ0wsaeybgZkW5UktcKuzw0d8g+ouukqQfmp6R018pNJSY1Ww1
PHcB/pbzqBN5uyo/ob6dIKXFrSs8mWVMnxYhwJjwbQKMW+kRm18C9Heg6KZ50ar0kbSu1oQnWJVtcVAPJE5JGD
XBvB9mEohn081AtiCSPTGmnFhJ7Gul+InUVnZU0a509TMZTXCpkLCU9CP5rU5I7qf0tcAnUfg/MeKhvnMA/yTc
Z082Uw+pUHK2h9jkpDm1UQj1dIygSPFtS1B1pXmmgeBG/Ec0bqSQuRaTzupYkuU33zvWiIymhmdrnok1iU0p0w
NEF907UjBSPbmpg4ytxF0KeykmWtLQbaA2Es9+F6S65JhyUuymWf1CNF/ygrJA075uyENsynvt8rebuj5uo1
G2VdRWUdz0UWpnNTXnSkUmmj+rpY1X5MWYUk/rpHjsy212XR8k3mUpPs2tUPi1ZQ93YONN5aZA07U2h8M1+pd
rngX0uh7zuJVN2efnKnd49TXWl6/21Pq7XpnYBuH7zSp7cAMuqlZqT5LZ9bU0096D3e9kyv51iz19FPdb7L7kGq
39G6U3v36aJ8u9YLxVFc inR+RVGcyTFenznTpbXCAUqS/78WpXMmyIXnHTF5Sf+m13QKRFr80mjcb6oWa+IUbpe
S9JGabhNyZEPuGZFoyPzrjrTg2463qm3yQt0Wb8I9Rf0f1d6oQRZj0kxUy3xhKvp9NYUIQkH7dGpjh9anoI9y
Wop19KMGgN01zn+xx6ZcyI1Pz7/h9rZL13QefSqqjj3Z+FbQLj8Lc9I7ym35nRFj+70DuQLf0/H+wyN9Jdqfjo
9dF/FomfnWihH/Vh191cBM1+Nd6/MrLzfwrPvxyGT67Ef9q0z6txae0+MUXLPyL1fgFCz5jf9BxM+L+Jci/oW
InxXxifFm/sRLPN6MnxHx09vx34v4KQP++XEd/3MLPq7Df1eNf+bDP52Nj1XjHz/p438s4id9+EeHDfyPSvHRb
Rr+aC1+YhV+3IR/WIGP3DebPyLiHzxm4n+Qjx8z4UcfMfCPluJHg04RA37EwT0MGx+24Id3cocN+LCDe6gUH/q
bKv6QiA8+mMkfLMUPHtDzD2b1B08yDoea0/B9DX9Aju+cZJDDzX1fg79/1tsf2c7vP4UfuFPLP2DGDzi4fTDat
xTv3X0K3yviPfdv5Pecwnt2cvfvLuXv34jvd3C7Qa/dpfi+XWb+vtn4vpPXzjqucbvM+B4QfY8P/00VvjsL33U
Y36nF030+fqeId/Sb+B02fMd2A39HNd5uwLePGfnbM/GYEY8exiNmve2Dh7cK/PBlvHVoFr9VwE0zcaI2JWbju
IhjIt4S1fNbRBzV46iDi2zHgwMr+ME+PLAC9/fp+H4T7t/J9elwn4MLg8jwZRWKnudJ72buSDp3BwJ9cbK0V
7N+JeBxcoxX4g81/GPh/useJuEXtFvH1TBb9ZxJsq8PdEvFHEt63CG7bjW0Xc1YjXi3idiNeewmtE30nDHRb83
Wrc3mbk27fjNiNudTrqcYsW3+LD7iIV7z6MV1bjZmzimzNxUwZ2sRreZcONDz18Yx9uqDfxDZm43qn1603Y6VD
zT112qLGD4NjJ3XwYr+AW8CtW45uWW/ibVuHlyzT8cgte7uCWafDSugx+6UZct8TM12XgJWa8WI8Xibh2oYwvF
fHCmkx+oQXXVGv4mkxc/R01X63B1ZJ/vqPGVZU5ffUjrqyw8pU5uPIsVzFbw1dYccV0boHaxy84j03zLbx9FZ4
PRsy34Pk0bh6oPs+Hy+dW8eVOPBcUm1uFy+BWJuI5S3GpPocv3YhLijP4kk5cDNuKM3CxgytS4UIhhy/ciIUCM
y/KY0EsVwDCCsy4YCc3W4Nn07j8YjzLiPNKcK6tis/txDbgaqvCOSLOBqHZIs4yYavFwlv7sCUzk7dYsMXBZwB
iDKDLOIXNAK9ZxCa4meqxEfQ3HsYGWDOIWA8M9D1Y7+B0ItbC9gaxpA9rgEazHat9Wku08yoLvpqxygq/mFdsxD
/v4aswBM24BBqasBj0dGImY0cn47tnHzP9/9g/9Rytww798hP43K53LZwp1bmRzdHJ1Yw0KZw5kb2JqCgoyOCA
wIG9iago4NTIwCmVuZG9iagoKMjkgMCBvYmoKPDwvVHlwZS9Gb250RGVzY3JpcHRvc9Gb250TmFtZS9FQUFBQ
UErRGVqYZ1U2VyaWYKL0ZsYwdzIDQKL0ZvbnRCQm94WY03NjkgLTM0NiAyMTA0IDEExMDldL010YWxpY0FuZ2x
1IDAKL0FzY2VudCA5MjgKL0R1c2N1bnQgLTizNQovQ2FwSGVpZ2h0IDEExMDkKL1N0Zw1WIDgwCi9Gb250RmlsZ
TiGmjcgMCBScj4+CmVuZG9iagoKMzAgMCBvYmoKPDwvTGVuZ3RoIDMxNS96aWx0ZXiVrmxhdGVEZWNvZGU+Pgp
zdHJ1Yw0KeJxdkk1ugzAQhfecwst0EWF1gERCSCKpEov+qKQH1PaQWirGMs6C29czpq3UBegbzXvjx5i4bs+tV
i5+s5PowLFBaWlnu5WALvCTekoSzlUwq0VvcXYmyj23m6ZHYtHqayj0J335udXdjmJkcrPETxq5Vglb6xzUf
d+bq7G/MFI2jHeFRVTMLg5zz35qUfISbXtpW+rddy9ZY/wWUxwFKqkxBFTBj0wuwvb5BVHJesbJpqgi0/NdLe
bBcB/HZWy9NvJTz/aHynBLnGfIucI28D9wgZ8QpR86JswK5IC5S5EPQ75GPxLsE+RSYNI9BT3PqwJThHPKQ9yn
MOSE3gXF0ws050XLIX2DmJ0TPdshri/iPymj+nhaxfjqvBu/tZORN3a/266Yjpz7hhpYH9/gRmMmij5xtsNpn6C
mVuZHN0cmVhbQplbmRvYmoKCjMxDAGb2JqCjw8L1R5cGUvRm9udC9TdWJ0eXB1L1RydWVUeXB1L0Jhc2VGb25
0L0VBQUBQStEZWphVnVTZxJpZgovRmlyc3RDaGFyIDAKL0xhc3RDaGFyIDIwCi9XaWR0aHNbNjAwIDg3MiA10
TEgMzE5IDYwMiAzMTcgMTAyNyA0NzggNjQwIDYzNiA2MzYgNjQwIDU2MyA4MDegMzEwIDU5Ngo3MjIgNjQ0IDY
4NSAzMTkgMzcwIF0KL0ZvbnREZXNjcmldwG9yIDI5IDAgUgovVG9Vbmljb2R1IDMwIDAgUgo+Pgp1bmRvYmoKC
jMyIDAgb2JqCjw8L0x1bmd0aCAzMyAwIFIvRmlsdGvYl0ZsYXR1RGVjb2R1L0x1bmd0aDEgMTI4MiAvTGVuZ3R
oMiA3NzMwOCAvTGVuZ3RoMyA1NDQ+PgpzdHJ1Yw0KeJycvW0wL2XNXhs+x7btm3btm3btm3rHts+/bxV3
1vV1dX9p2NH7MwcY66Rc441MzNWRuzyJATySjQCxnaGjQj2ts40DLT0nPiYfjaGLk5KBrbSNIomZi74/4CsUCQ
kyhb01ib/i4UiUTVxdLkws+X8d5yQo4mB8z+AsIHxP+HK5i74Ai5m+PiM+AzMnEwsnCzs+Iz09Gz/DrRz5MQ3d
bQzhCIRsrP3cLQwM3fmxP+vXXxyFUU1Cioq6v9GGDg40PANPf7N4AubOFmY2eKT/g+Bf1BXE2s7exsTW2cufKF
/YGrtrCyN8M2sPe3MnfANjYxPjf2moGlibW0GLW1hb2NvbutKTC1H8T5V/EmWg+eeLBYoEX8zE1sTxn5L+Y+C/z
BK1czQz+Y9a6NkYmfDJzZ2d7Tnp6Ez/oUz/RdE6mdLamjjT/aNJImJrLGRn869snKCg/jVY2MLRx0if4j3o/p9
uW9naudl6/S/Y1MLW+F/S+MYu9nQqthYOLiYSwv8n+B8I6r8xMxNnfBZ6Rno0ekZ8Ewd8E3cjc7p/nVLZw97kP
0iGf8EGtsY+XvZ29vimbZ0Jj4Wpib/bKC8nAxcTfCdHV1MfLz+78T/PIJiYMA3tjByxjc0Mb0whfpv9X9gE9P
/PJYxcHa0cMfxoqf9x0N8+n99/mtP55+JN7aztbf473BZAxsT/P9V9H/RgoJ27vheNAxszPg0j0ws+Az0jIz4H
CxM+D7/U+q/TPi3Af+ByhtY/J8E6f9bUsLW1A6f4z/r+Mfa/1MLPp3rf7YzPvm/2pkC/3/o/50hnb0Fkck/nfJ
f/ahNz0L/Tyv+s2H4f+3T/8H/V7f+/2vQf2kJ/Uvpvzvf2Uo6mJt/R9ukv+nm/j/20mEL43/L00tDRz/d7yBj

YW1x//LiP8VqWbyn5fk/5eShLPBPzUI2JpZ/9tofDoLJ1ELdxNjeQtnI/P/bLX/Q6jYGps4W1vYmsjb0Vn863a
BT8PAwvD/JJXNLYysbE2cnP6Z0P/gTGyN/+dp6URsjeYMLWzN8JWc/+lpA0fj/wL+RRu50Dr+Y/F/zPI/Y/99b
GrxT44mJu4mR1D0fMHIj0bW5Zp5P5/yzV1k9JQaKtE+48cTgDjXyV//1rHUze1oj+UKUXZDJqUzeGv8DbZsjH
xg8guDRFQBRswHqRCBiwCWM+chX+n0sysXSMrhpKGFWh+jQi0hVmHISYjh/078xHqbV9XfIQt9tKxTHz4qD75
on2PHFl0BuoaFLNQ/igY8bIgo0spdn2+wHiPwNIqoTg8vhhIWDPbPPFAYEDobv76vA4RFCsnnTc+9AX+LZWH7x
pHWAVhYTl5ABgPf5mz+YW3NVX0cuskMrnzBzwV4QNsBH0nDck48SeBd4MP04nWJgNG/lpnf10p1wNnnwpF9rZ
TBZY0g8hKYnViWS6WmuhRv7h7zrB77anmrscxB+fM3LPgE2QfNDvJ/hW6CDyvqBT0oAeDs39/kxbI29kCjWw/J
Qctsyes05RA1JH2CkBkdceh6juCkm42AUGhZN1RPUN0XPjyDPwEVUjAjW06mD/Bp5Bak3itygvnmVhmgYfoXVi
x1XiGBXT6B718nfrkqrHB8ri0MCjr2qndzgxyJKtD0AojoWwmZ7mKEX0bU/nK6IhuJqG72kxgQn18Vfu0Nx60gF
/BfmXD9aWY+hz/c5azHidHQH1V5kwxufwRAHPp8bWfeo6GfeQFMaXqk6HAPcr6av2txuZrpA2dM6ZHXHRPCTWS
B+BamdgjcBREwDTY51nxC7n7G77aSb0GAxVWrA7eqdwzNtpjp2H2yujoHPcgouRS2+o0MPoF1usDUSUNh207i4
f2nU4BYAbtSJuzLkUWaqxLcDNinCCdBNdudcFP+p/sTX2gWnLRRKA8FvN6RAnsFHAtsSCFjxJmg/4xQHYbJhW
nZMu3PM0X/TyJMu3688CUVcrk8FvUgC4o1kYjA8pQ01pVKh9ad3hxcM76QS6UL70YwxpJyVxNchiJgtuu1yu5T
0CKC74jlxIonpDLdNsGL2k1Q61k6iSt0V4Vdb30s7WhGdxK8uajqRp/9LQY26Qhy000EEmYViQWm/83CEVoo0
7qIHAQmbEPLCeFNurhrbV6mEc6tKrbgaeMS0PTMygLaF9dmYkMKSUTrV1IGmSb77eukEPPKJ0ytRHHfWBI5EP
4EY1FhgG+0ItYQt19d11cIhaNbMmKx6Ei7gVhhZiCu/Y8XrDau110ked2nI871+SvDwUzP6mJuNWgfLZeNh04e
i9xBn4w9hj4R2XP5B1NjG09aXIs4Q8ztMT0StTezbm8KifGOE3I3Cx67AnXVcSzoCilt/cGcPwJRDyYVpBEf1T
ispP/mVyKGFrp0l+dis0A4gYkkAVNL42yLVUEop6zFjxv/iLTzgohkQLswjMrvJ1U9KtHHU/RNABgUQ8iRcNTb
B8+ea63o0+ABdD3WSZXH4QjIo/dK2UWEaRa9fSPkmqNESyanPnq38U3YsvVYzkkktDce0XG8bd1/Fxdt2r66Th
0VnkMwPnNweZbXh3HH073mD5rIs0Mf8sa1rytgiMjUyDj65U7gIx143EuGn04J7H0uYs3ERPDy3MzW2DGJ5CR1
9vUb0c2fWtwIgudIucrNQW3tFNHHsLk2fI3jr+UIlqXPX1kG/1LKkt5R+3X6ymNwfhtwLa1i/K08m95NFy9o+
Ao7r2CdbSBlesXFz3C0E5XWuR/AbwWnaAKAbIeAu2YszIsNy5F2LdA41ZQXNg8LREETsQDng1PerjyYCBpt/Dz
yQJzq2pb0rPn/j54PHzP40IYlck6V5eqhrGbxRXIeuBQkm9DvQM7/V56CSPsXU1Y87HfijTeBU2C9IIkKq571ID
i2IdAWOITDYPQQL1E4H10t8FwXtXzt20r424DG4IOA3VIzxDUwbV9Kbw7zpdZb+hvU2T325sPf8ynH0yD5n2B7
Zakjc72fzf5oVlabd1qk0UfzRymLhLV8UgPQWLGVZHeDE+1vMmfV4eytYwYizFMKZNyBdiWbGbQKPxJJfpbDR9
sqMx/XJpz11ThQzrjW6UuApq9bHAhyaqGQk2oViftB1009A7Qe6AW6uzbm04dD8AjAmPymJ4cyjVL9uZddP+W1
fp7S2Ss8KHDrm3jtf7B1QXG/1VqkGp7JeXSMk8hI2070PujxZ1/IUJsvZ1qBohbuR/qoPd0mdlzvBGTGsMdy
CZ3MtSipoKgGGkrssV2JyzpITcmXvu7uJGRDTqWOIPVbivYMsThuX+N1813U5r6cYP78vXa16qJa5vjbNI6cuZ
AxPpn5x21CP51haBK2ojamc9r2Q0pqzV6e7TLuoT09FGtocxf0Whj1NdT2dBULPdx9b+dxeEEkNrYyyCzH97Ed
CN6WJjljDrfbp79JKEVF8N9idcD1Qj+XX6z0U3y3BIPD0vML1Z13bT0Nkis2rHSicWjW4LIWQo/7jSwk40w9wy
+Hy0bKS2Aa7tA6+h/6oYPo3pUtXvVHYB9opaP371XHRvZXwpoKLql5nQ1vPb6MzcnY/fg2i0To/FXyz8FpfwoA
k6r703fd1Ug+WIVllyUceuChRyMN75mauo6EqB+xldTjTbUPMQHTXCbpjbYwAGkiSSAQt2XunQ8c/uYh0f1h93
eMByoZAsEq8d4Hw8y6isI2gd9t9YFqeMHys4Ni57h00p0jU9xtosoxIjDqK3HSEYV3v7PPDQYJx8mShgeHshi
d5sTIqTiZ1taMgjuLcgE2Dna93FFiYUg3AhUxAMsnUuXPE7dy701CHunWk/tWELSpmcxYoewA4lgYfBdu0P9p
F43JWZpfIfSvk+VZMr2U0RVfAdxVbB09TqenL91ydmLdYZa3LR5qVgh6S1kaqhI7v4o/0f04gDP2+fhKVRB1Vj
6GaMdZq7RxCz1oD1b1S6nZfw7hDWYHVoEotF/b2KWyuuEAkjIYX54r83UzvnK80E2T/2MnncIgrPTX3qg8nWu2
ibK9uiBtGVVGT6UDIM5eD4CsEKJ1yJ9kod0pza1w36Cock53vD0rVaA9/hK7w4kYhTrvj0La/XVNXPIm7Qbx6
iynn94nt2eGAtE1aSq0D8BdmfrU3HfGcMFTiJu0ERqvMkXCYLLXrkPCF7pYDY7178T4LzfS2N/mWwoh76SPyWg
7xAA06cJH6qAHkvixuqK2HkLy0Btt0+mGYNKnXcX38WpEhMX3KmW1EZdA1Ygutq373nJICq2qfVmA6JmNXWP+1
dJgK0BLvZ8JUFUYiMEebadzc7N94mNmjcjaMqyE6YI1VrlmKX9RbUUeUs2yisMNJ1kFW6n2Go4E8eatcyeKzIw
4505eY/w3yUhpKq4KqhiACNjjb0mqjUG+YBoPuAXxPh5qq9HL/pT35WyDC5+wtEacvB0m2lZMSaePYmSezbJ3y
islfXZeGsweQIZwZqCFRSGq2UXNxs+9ypmJvH92br518Fd07FY0phc3SVUdy30TMDrLyqGqDp10JdozgGGHhm
qBal/3D4iUyjfvwo+PKg+hnv7mm2i94EW/N07FyEQA0cFwTijcZVShavXmgzIM5aQrqpNSaPr+oI41GYLjvh
6QR0hIgn4S49EdmjDMHtyNff1Eo4oPFz+Sxg15cH7XY2RiZwOsjnq1Z2XYchA1Z16Kn160tqG0gAPe5mxRPFbI
PzxxXg0ZHIKftz9KY1xynICf5DSMs+JwXWM4HD2ekKsBTFXWsNFXglmP2ukzizH0BSUpC6HVUDLVOfdiy9LMZ0
/HvEHc6t2QCHQmHNjA8AKVeeskaSW1kfoF1v21BRVvg11FcWZTGAintthk1K/+pe5kpq4uRibNDEGAbXN2800
K+tzVLUHUWipRmNvkRVL/fZV/WBtTXcauwhaM2iG0cdqz8Pehb33i2QEV0p2dq1itvZrjDDW4T2s+0DrVtoac
0CUIsY31uuEJZfkPNvMvbvYJXXmTkLJa5tYuiCzLnGqFjId2V+Wa8ReuZXkjRPvY3jZU7i418/DpHb3oFfB1Lf
uk1nIb702aAtv7Dv7gLubtRQSSwXL45Dkm5hpQ4+sF0q7jqmNBryXGQ+8Sf50YIXWicg3u6WUimGD4ZI8c7MV
3Vhv4tzLifSrRFoQDCS7iBQG89RY6J1zqj6wDV2L8AiTMAudKkLuVX1MH9f53Su4mxC0olfBcwttnSheZw69a1
+HMsHnQ1r2HVqI8iqGkBd6sd5+CYMpR6xYDNYWGab+EJ/LNSRgt5vGEos/v6djJ3IoVWcpcjNob0o2Xw0Bwb5m

MgX810n+B+8ywTUQG/h0s9MGwbeA5Rn5L3HNeG12Jm9YQWFE99AF6T+/3Vhx04EjcIV0ZvqUALPKu8qCuWlV+iJ/xj5LYmyLY50j6DV33kTTq/R9cwfE5hzlyfBgyCvrm8BDCwKrPOPiNKTDiDakPdmhhGjIiPC8n/j6IhUuEreu16Xv5+DTqvxE1HkizXwhay0fmc8E4mGCEnSN1G6021gMIEfuAvLeb9LSfse/uk2xijFljdADciOFmFRGALGFWh0n3EnaLwlpFE+jjtN2FuDcw6bMXjBRxB0c1L79Kv1evs+8IKEEz35N0Qs9NONe1sMMPEf1cVN2ye1THppJ6ley0mCpjTd6lsV2KEGNY4d8HCWrD0gE1fGHdAcKofEmwSW31G5VDzYmEVH0JiPdp1QuxoJtNrgXzI7sQoEZYZepvkqMKwA/ZRx/qutr+To2zpPRLZgYzdGj6g5zYwDnx5LSTZgzvaQxBCSZKtUZcyfg2tB0v3Mk+8IFnZdxSbpwyBsU0kC1qQ7b3+iu70vmNzy4e4zVevq37aqIfsDdNIMYZ+xawv2znrBYYbgQWB5h40ncsSzLQYIdxJHYcULmpT4g2Xh4fxsEGJ12ZV8P+qCZjsq0PBeY2WDIpcnkdlH115Ydi39Y+JQzMPF4XHTAdhpj3poRLwCL54/YnZeKvpMH6T+XDyISBy0fbFLLWwRoYpDGKKtsItxWe01L3Ybsvdd0wfUgT0jHx/PICiE10+7Gy1IM4J+nEupktMID+nQTeHZXoFFC/6Uw9L03sUwJ4cwP+RaaXrq+5u1TR7NhTrXEOD1IgTyinVxPiib5WIoD1P7Scd778AZQeULR1nw4yP2uTDyBAgP/IeIxt4k0EpsRkP10jpirWEEQMquWF1Z1N4YSD1KaF20nVZD7EZr8SGIjNzIa1kLNVM4LRavTJkhg3sKFZwgBWoCncUZByswt/07+BA5+xz1/W3qat1BayR4s02yLqMr/QSooqN2FGqkThaAczStdNN6/1SO+YwhbZu6U1TuNRD0U1Tzgs0rHP37nqc5hUT/MUtbZ1fz1aSd4418MOfi10PFFBmWVgdLbQwxr0Pqi7YijGb+/GmL/Lh00BjRvF5hpaVEDejQKpYbjMLFIux10d7ndhwmqgse5sr0swm0mSSsy3M+a2xWTOLwcblxdFyaZMHfjuqZjEQ1Wkx+0MwF+RqQEGPdJYjeWXTRQZKJ1ezjXwBwQaWQcBYdZTKwyeEug51MhZRXPO1UEvZkbYppMPXX+1pXSpC32m5NPT7LP/7Y39AP25ySwAuR79IwFMqBjtahLdFZizS4Vz7tFD+4PEqd7Nvo24hmEInuMo9YCTUScKFY32sEkREPVDMjIqLOmkMWSdWCtmtcFh2ZTLcJoE9HBpm/yaIN2WZ5YBE6HRcw4vtMFr3hRM5gNArNmTbbV0apz/1pe/b0gE32Hhza dd3hVUe2xEVfyswAy0Mj7KxsXCV4W5SasnV3iMudr5y2ID4hx0v2Y0gcNNoaGEjC1tg+Eg02ncgWfkhnJNg7oIjL4/0Bt1X8Z2+mEIZf5xpK0j4RbdnTy9Vph61SCdzUkVvLFMCp38CnfKy6YEATwqegQa/GkHw7kWdyn1dCIEmNGU1hmTy01oQ5H5g0fgeY8C7Q80MsDsruRv70+NoTjk39hLJzvBWiszElq8N8orwAUdzAw42qDEGPq+H5mThuhFdMnoehpZJmfGWbghj/2xYwJBrvdB5v60MKdP5BiXyA14r2XLRsafY132u81TjoZ+wSeANMwtwVjd0v/ecGj2+dUeP1dH9baVpVsTBCQ8P3QctwlFDecoXiqjCkjbyS6maLk5MXeRaNfhsDARb+ccv1QumegJ18GaCEdI+71uSPDx9zNt+4NFr1o5dj8T1HzvqbD878q2X0LZ77yzog6i5j2k6Te9Ww8F5uxVGfz0V13++qYj0GEDJs4Ije5ziuvFFaeZsqgC/MXWcXh+OBmqHAcqH4B08wpRlwCBFpgAsbZt1dFEloa50TtRevgxvSk5Fca9toRErKLKb0f1KYQMFTMSQJ3ZqosnAgUBaSeUabAH9v98p7cSXzahoa/MDqCg9L4LxCKt1NHa1sS2kbp8cd51d1PXF04InRey/e5hs2L7D2fZ6yQqhtZ+2y/SgNTAwct4oNrL+UyIX51DQC5fomqoyxSpa6ZR3IboG1wuuDf9GpzJKDUE86e4NPWpxv9N+UQb12vs/5/KLWCN0osUq44aqSJH61xG1wnR2E8+15JkavDhE1ujbDzEBGpzmrFc2nWfEoF06Kxm4MxLgJeS4YIk4mXDrjtVdKF3aeVdyyzpmh6CYGoFPhv3HrrBUCt7mt2cp1uR1GAkok+NSXWeX4Fj2ITPv+uQL2JXHzL4tCPwmf1uUjXZeNqdyo7WjJGNUS/UV8T+Hxzsa/+UqsGvxSvoN00Co+W04+BvXuHa61rjRTfKPMm8G9knYIsHdUF1TquAd6TE5Mb2wFyzeDIbHfijJXTioN9aj5SXa9DYKqNQWQ+Ncd/Ic2hN7j+Gwt1/thb6X7ezmUUFhDxi2rzo1CZ6PxRb6XUcgmtESefRAMGz5Li+H+r0wdwB0in8FqmsbUZjEYgKiGTeQovNbX/q9a9yY4RPGDB++o8g87jbzVQuw6RW5b5wuJv4oEy30neplrEIhNEIjxIdq4XxC0g6n8j1/hD0oQEGv5p9Q5K/HYmk3gXz7zeUahR8HeREA+j+1Xast8gKjTnzeTpksYeqJbBeIb2KDKWS+RC+pozW+Q00Ny6o1cPNmNdi4ksxq/KxVXoFJB3Dn/8EQBL10emMBQxMt70o4xFD7luesYzJw9QJZ+iiAf3KWe8AX6NI++bzXH+MezG30p4ycMhCwH0qWKPL1rfCcelP9/MSs1loPfHooDw7mqStIyjtcnNPAYVRx5qrkNmfrqimWZTBrq72uNyFXBc9E1hH0mu1zpoXVkf0gD33QbrQ0w2RSoDw/KV1MSEqX6ARak1keIHpuXGy6KAawg05oFm1d1kNRyatDHMBbjedFg6ay7j0xVn1cucrN2ACQjeMnQe3ZeL21bRxMj5PSVTN39b0u6RDBLz8DHqzQ1A5A5iWFTCBnSUvVsP/xgCN2hZip99xqoQ+0QULCY2U1v8MntC2t1PAB2RGHlRkAh6JArchA7b31AttLtnGZeecf1pjDdyj3ssxjs1PTunJGkJYb716Chipa+Z5AQfWHIPQxOronQhrYpSG6C3p9t3/kDCUgml9/Pjn8GLEm4DE3GXINipTgE60q0yvs4zF9oL87TRG3T0m8loE9/nF2dRmRlXk08gdEmGX/mwj1J3TaDIDdFw9RLA20o5EwR0Zjm95xw4dWte6JMVxepJatc161eTLRS4RkHf13ECUZ0BLXuuFu3aBdQr3sV+ctVvP5tvH03Kn+7EzKoeKKnZQAOMW5mVIVF7788DL5aHvGzqEc1cfwx3dTeItVTwboHkA53sjXGQd72g8KpZSt11Nw3ghVT0a+zGk4hzH4FpRVgE2r8c36AA8DeqG3FhWXrjZIAOIiC5kxRcsHn0miMLquS0yJ+asxi0362pIZmWAD8LdMKVZSVpATRASy/kLxRusVDbpxwco095L1vN8i10iR0uZb+nW5nVtHP+wC8K37oMqwEbp0AeQKkWUZ/kaqw3xyYujfcSbnB8rsQAQdChUnrBKDA3GjWkw0suTp4wi9YR1YUydDjZqZHnB8YGcSQsuRsV4scXf8I0GWcd0wVMMhHMCKbcI81tGcu1C07ZY3vw7WmuWIFCstYaWQZhf2p5e/RzF/4qTrhHRS1TemX+vgpc0gs8ZwvcTA8Prw/nzFgmQtvsYqjZBS107YPRQG16ZdI+zxCOqsDjCg5iZ8QR6t31kcM3SS0jFEj7Tjm1K1ysItRHg0oUdQIz2I5vpsuo7MaEfaKcBVwpyd3uKUNuXjTgXE+kJwcwFd1SYn6HD3r0rGTND30Yo2A33m0qC6IuUYT5gAvjZhKmbs0/ArtUQCriFyic47abuHdPeVVG7gai5/32qMQAsf/rx5rmkhUNmeNrmScMjD0IajvB5SsGryNr7YFH2stHBVE/1T+nN8SntpTdluCbf+1RbRNyhbzqu54IEZRD0gPeUbPCb74oDGEIApN0df8QkYcmbu7n8ZHuvhDe03BtkgXab6n+FuJtVY8ndGip+6uSrs8b1psfPJb2Gem8Ircds1V/NuBhUhD3wLZruSwvQxpifrVrWaUmPHkr0UYTdEmBbfSpisJmMtRz5cJh3R4iVcoDcIsevxnxcxIS40J1h0gGhdjNTF4bTem8QAQrgtkpSbevyTTs0e9Bq576y21Xi9EMPgLueIm2RraFKgKu

2Fa4iQ1G20yVQUJ17hNexJ0MrD5X7tupgd+xHBqFvjY3sH9XjM4u/KWT5ug+8QzwnJ1hhCeoQFgCvTim8t1n5g
4DvhnbafCLnE3y0S01JSJf4y+GafDuDt3oJ2H9i0va58eNoPV0e04eV+1eXdJgwH1rxZvdDThkhCTKSdbES33
TpQScMyRbve094S9QK3U1N5m6UMsYPrlwfyim/FxIDkbsktgAwvE5qrk+2EAtJyGJI2H68IciAJXk2xaSm61j
nU/xqlbpLfGvyg1Kre6R8mGF9rX1r+qpTF/jSB/FFC42Zi/Rzd9VUfqhL8iyqh0XFteoFrRZ1/0Gd07kijhMB+
qrmvLco//y2kqnMbWI+cXYXNuf6ojfcYoXXEP+4qPLEz83ZFcYdCmcvNb24k607ymVgC/9M2LIB1q+oXKPWb
QTz0KIdcIYaqbUC44jmAbd8z7g24U0wTv1sfn2c4X7QL1iaBL/cmaZeC7rmzEmGGBeIzjmV4Bbr314ME4uAYuS
HRmz1s+jFDF8En0fCR8cF8eqr56XBTxHipMhQx0N20DuD/h7T2kL1zg2UDXvmKTiAsWKTFlttuMqPa+eHijvfr
FJuhtTf1XpBx+A2W5fbKhQfgMbTQG+w4+UWHM2KcSLnvIPigiBNTH1wn6Mt5Q0J1eYCvv74PrB9Qn8yK2aFc55S
JUmkqdEGSVsnt0pYpNn0AtwySS3m57L08eGs3KavQg+rjGyle0SHmuKaIFqA7wS/jJLUSyi1/wnHKJEX/sf15K
N8Z4WBQF54Fc94nHH1WyKq1dr61kWRpFg91EfjnEEwm3dEW7SiPV9N/JadjWQ+FW3bs1S6EInZAejoetqaA8ZM
eaarc18Yy26hokz2gIhfVLi19eylvyInfCoA7tZgagAFY0/fYf0gXq3t3RVB+Ew8haL1zRfGF1Rj0XrWj+lamF
+3BsM97TkwY6I1FxzfkNQlyi0Jdh1jAo+0o4SdHE5VaXf3kd+e3P0F+CvqHrDVEFts4YwlPbZwneIyzvmPsqMy
obPQub8rwSrvNB6J5s4SN7suXajjKabd+FxtTtQij0f9GccnVXBssRjS1cpZKPqicitbXAHDxp9y4Srhc2uX062
dbgCfs4RVx+YXXn99HFoCLYCpc+NeOWgQ8Sot82ICn0BzXmYM1KmfR03D8SYw5V7uq107c/+QYE10aUIDWcdn
BhBt6im8VItUXIVf1IwHu0H94kSK9Ih2kZI5gGaSMs6JIqGIJGSLR6RGxr5BSW6JRHs+7w4y1UouV0179kYIOG
DkoiY/VL0/8ojDrfm3Hjq3UXvLHMu4gKDekP0zZe06+CtQ/hcP++GcngFa5FQNXAEPu0FjwqrAf2pPZKP8/ksa
HZUQ+psm1ltUuiGWQF98noyCwEC0+LemFddCz84hdUVZNu2NmTTA0nQ4vN4A2fp0kgSjVeMCLv03H50E8bMV1T
GOM2E1WbYeS+VUiVjLdHcS0MBF60K1j6rvhaB7C++86BrS8gJ31LGUQx12QKMne61crWv12IKni69fgdbP9mN7
X0Z9BwXYf7DkyPtRQLstL6MqhSutxjnXVpo7+jiyGKHWZbAe4RwDW2L1v7WIEA22BNDz6hhVt3LEtUR1dtYZ1z
w5SSWXIPRw4ghzJ83oeTeAm2UUZTtjMGx0U0yPTXT0BQGw0tLp/pmssPuF60XfHTqjqU7ykv5SHwz0Hfb10TrF
Kso5fxCvKhNAuIN+PHCntrv6Yt+pt/U50pNW+k7wnMNB6FOIo/NwBuEBCIeszGUxDlamL6LYG9i6vWg4PM1Tt0r
5ehtBcJwUB+Lg4U12xE74eSOXBBMndZzV+j0etyfDoo5T5uNmgzHRgVRF0z2PukoD483Y9LNzdh66rR9BZwVKX
x6sybs0XvvkLre73HT9foQOHxkMIF0+mQdKV7Nps0ddz0ww8mRGaEGVXAEjh0Mf03We4Yr+1DMFUWcEZnjtzM
Bi6p5C2AvtMb5+0oTk7BcbGS+JjM83mRWmDNRDFWpo6HNRwvVrEof42Cnw6JPLMhiM3puRwbobrXbXWg3pGiY3
CF15z8vJ3rhBMGpHWtAy8Rv0Ksneke0pam3WBKbzbeDTTSQnak7VeJKQM9aA9QF/i341D94kz5Magq1Fts1JaU
zmQJWNE+GtEtCG8Pk6bYKxTVmbpsjYTrxPsc6Zna5BEPMxZ5YJ06UXXuBKw871sQVtdLjTknoECN77sztXXw1
gKyileuFzG6UGdQDrcessiELglc/f9puZ18tGTsDjeDditM0g/UcDe3TQ7JB7KSr9N5Y+TH8I9F2LFaD4ii1JuA
BZq+0PJN30y64w5/jU/hZAn0pi5kgj8b18Ct3AiS1tqmFmp8efBjS3ckm5Z0iQt/o4G+KqSFw/a0afvG8gDYrr
5DFkHE3wGJ+B1w9abkZI2kJfFv3vaQp2n/y3a+WuVGHuWOHCCYAvnfqmSOfqVKoaTSonryAoa0D5xKYX3HPM18
2Y217nri93fSsP8KBY+eQbUm245TILExmBGqRReA8JN51wUvYykwLNFNSTAf+k8A6iLPugFJRH7xcAIZuydV7U
Cfp9DSJbWh3FiaU8/Ca7XgLg3X0ZVgtYxy5dLxeF1beTwzxCZjeIxtQVtPjZt+8RftEWJihY/cQbA8Vnk3FQA
cxEsQE5iKQ1D010Fg0vzds9GBAUykn+oWkexvWVn55km78W61Llv/dfUd6olGtiH4iuNAD8dcqyI0jTcwB2Am
JWTwhyR0uoMBV2Uc9XfJrpKfxr19dtCSPN9YnMcRJ+JLvgH4d58hwdpUoo7paXURs6MXIxc8Xsash0mTJx4mpg
wPqxrWFG6TGjwZpbQLFSTjruU3fH75BfW4RkVPin5kqe/K8D7U2rR11Td9m4cj9EdI0lmlVzGuts70yqvXsAX
t6d8/GXpUU/w5A0zZzeU86U5YxZI0WbiRbxohRXW02Lc5L0Au9ihDz1twJLB/KhWq6jE21GDEC9/d2PMTz1Pg
g107b0M4bN8HBCyfZMyXK2ciS9QjfBKfCNIdNpu7Ycov6uInGpe4YI0UQBnTak9Rqlv2Gc6/78o4GFeSp87RbQ
QMwSj0lu9S06d0KJehYbtLbgiu4Zy3Hdu0KIdoUeZx60bj99/14LWSgyxFb01x/B8NtpxfdicajL0sdKsdGvji
B4Kr863egIyS+aEHTA/I0z0psedvTHY6xzbpk7FirOpI04Rczvek9pXbnNeONQTL0FGa9M1FP7KPIsABH8VnTD
vgLMMvMmmpo1dGtDla2mkjYNs0a6AdRT0yS+/Phaa7sNu1vC/GZb0JTxp+1ddU2SPbFkJnS57p35Lo9V40eX/o
n7HNV+0v7oebHSGGYUibF76nEeqoQ5jLCK/phEDAtz3sEfeBVBHhAnQoQVjPZhJkxxjSjkgf1M1kaxb/eF4a4i
xwqb+8bTP6k/F8er8IIIt7HD6j1+puYnGoMRo7m/gx6CXyUwsS2SbJa3asZ+yo9vs155w1d1Y/3zAHya0Z67D2
H8U8IR4VKRSnAC3Q5c14qEmSBmDbKPHnSATQnxJF/vpr6I8yaQmD3XY6tCgIumyE+0z8Rzzod9AIn2PXW0i+9U
rSrwwUF6M9rfXBPHgfk97z4it4H1+TLa+XAf7aIAeffTT9NspaH1kjtstD1KWC23J0sm/IsxY30E/k1KfNi/V
SqVT07UNGK6rHz0dg1nmtegGagtU32xvToic48XYH1Cnkp+upv3tSXgSJd211NgjqSE21U691Jmj0MxkLxiFD
MYPbsKtPXep+BKotfZ4uBDoSUBuZg2mA+Q3ZkdGmRVPwadvD0YFz+R3sGwXL/Q65emGcEvb6rUkBHdJeF1N0o1
VNcLYHwpEnWUUtNXjrer7PaG4AzneGvDTXZHSDD2936IiDf10mq4VD2mkA5w0q0Sj/TbT0w1QYmjJi5IBZXvc
ZXWVvyRSmRCf7o1jnq0beQguKAbC3c8IrQod0oM9i0/yoE1EymtSN6GnjJLsJKbWukIZBkdUGQD1DlytiVoHJ9Y
FG2sJTz5sE+qgRf3SJJYQLR/zm7BnDYN0AjvN0bzxS2/We/PgMtMkOUYXC49m4cikU6axnK5vQx7b3SuLoioD
HDL/SAxyv6uqWo0TdstZTJR2cRstN6mpxGnt6KZQJqtliRbo+M70kMIN0ZsT44fGS0k7Nn60zRXoBXnYgIHH+D
8BgC1mUFt/HpEzbHsL/jQ6HTt7+TphQKxvImtNwTFSf1r9de3sNHAWbp1gJbVTKtRDMQbH0r+ADy9TjZY3ouCw
2U20PYTt8mBF13j6b6k1B2mj+ajH+XacsGeCTX41gAoCRo28GEoL2tomHZWMFsJUDSPw3a0vLt7CN/t2nnnkoE

gzIw7gWHGcz4U7AztzbGbyvLkGioeh4D1L86gaS2p8qFHQkCaQtADTkcalcX8bY0A4/iVh1or2NuohgFN5C6IO
eBEUL0eHnwDItKbBFoBEHxj15V4DKyxcUz/5sxN1p/crC93BcszIyPP1618EeQghqG2RnsJqDGfQA1RRgi20rc
H41+Jya2yGVfvSFJrIC3BbwuqMLJ02Pt6HL954WHJMI1rDhuXFdsPrj0DydzNiQfcBz9sf6W3ra6nPwgHa5hhm
eKLUvjuNU/dd4MYqVUN0oyia2Y/a5Qah4q54v0tQAAfYHt6xAtiE5IsUthAODN7NX7dQmLuS7SP/46b3v6tk6
yZRStwmokQc4JtizRIKUMfvoEQxzmCmujWIrCCdzhr4gPkzbzu6R35s3/mHkIlxShV3wFs/ZHrKLJP1fWa3s
jW9UUy2EWAHBUQ4uPFb7HWx4GmGAbqXfu6tgz+2t37jMMx3IW9Rz58CFqqi7q0G5TvRhAGrGU4pyUAKzUXnt
FdnizL9gBZE1raTUV3FbR6I90Bwpxe1B/T83A6g9CRIoiJJ/oJ5yqAR6d1gEhVb8jemIpTv4c7y4ES4fnhX3L
M1FqKwbPMtWhpFMokC1+lzcXuSELw6pm/BTIC/wYy5mCBjohNGD8xu1EYxGUTx0cDOLi03BsU5Pi4x8mkow6+u
nSj2oILuCqRT0Xv69vLiICg5Yg/MHAGp+7DB/RbD8TjAjeFC0Ds/aSpOKjzqF4DSnZaYtLDbU1YAJSKXClnI20
eKmNy+vVd1I6tpA51W7jev3kzjyRi7NqvPiw77N8LXZ26cG1C/CJELdFwFSjizKW4kYtSnTrZK570ntpJf09J2
3rz5RgaSbV4CLfv770WzBdqjAMpo8i0kNPCVFmhdqgvBrBw/05Nj0q0fpDm++9a5X7tI9vRYII/W7z7mEZS0
r6mCmKiJqkxcQSp4vxLE4UguKvDXZT1ZZWDKiouhH9xsQx/x4LP21XHKL+vbP0hRjwLNcvURqSVk0daAo2PGBt
wrVPtNqGf16VmHyP8/nmv0FebkMd6aAFeFJryWe7V7taNteES+MJD1fdwpFMDzTwG+W2o/udyzwEir1HPY6CNI
EFpXzzXq+S0DfogU9NFM/bd3xuSqAa1WX3u6Mt0nf72bJ87T0RdbQjJnnhCfb1GYGsabmAvhZwoVL1RXhvZBVP
fYc1UHYH0L1uGrw6tCg+EAJ02VZ11Pkzz49/dgp0mp0dZKmU8M3p20q8e+6J6y17NICpwaQ20HBq7doD62hDp2
7vH181s1/qYSYFBPa05GMP7/0x46wSAgJ9afkPgYuqEX71AjUdBRu0ue72Jvmuh4wbKy+owjHs64rAnNzMqjsX
5JLTQFJU+qjGglszQxwAMKHGnIcgU24Wv4PYKjXweqOMCCSwI0IBLF45sno+bQI7jP7Vv7E3i7MjI3XWn4038a
a1je9nzYoYiCL5o0QK3gmb0dQWncYoXIR1scE+EJUbFEDSH5UsnMmZtHXXYE5xC9LFzU/PFis21PKnRNX0Ex4x
zKVS0rn+Bc4HTopUoNgB6LkaN05qTEYA1tpVrg0t+aSZeBmAL0G8pJ9r3Wsk9fkknIXlk4CySfX1mPet5Qt81
rDcHXcC+wc8vftgQ+V0K6npXL2tV3BeCVw6TwAmIQ9D8szQhDGuN7yP7xQZ1bDw7PRw5yVbP/SpACCz+hMYvbr
A4VV20bTqQoS0iS1DUXez0JBGk4I6FPz37wEdbxK4Y7NaRJAiHpjPz2NgV06B5B8k3IL3hUGXLb07xQq/GQViQ
Ui1N7KeKL6qu9JWnv16KetoDpz1n2G8ZSPdbXTkfCAR/34LF+xX6RxVKhj3wxccgS4j4gZoCuBS77YM+HMSH8e
UHOYGTBPedkrFmgPv03P0zFz0mW+YA+NA590MFyPQgrhSYyGLfSLBH/m97a9ZaTrZ7pJS+3P2EC67FAfDi3Uwo
FFeg3+1206rP83ydm/uv/y6es08YQrnx49b17EcrasgJBu8iFJU0cgz1z0i1zwGhjMcps81F6rRU8z0rwyu2
Adyrg3WDrLZXPmcegYk5G78PHaCrgxHV4wqw5wE4nZ505NC2NqS9ubXxXDyEa8xoXg8foBbFWG281zSn1PeJ9wf
Q2GQKugjopDeh/o8XNsfdg1FEs92q7i30qNXTlp9qMSSwdRNAEagzRqnUa0Ys6u0021wJpuuuR6mLSpe2vQSD4A
8vwMrpQwK4KhWJD0i4P3kjSzRrRfBG9H8uYhEqqfKwt0iK/3+IQYybV7PK5jUGMtI9IUow19d5FFv23vzMzUbx
JrGaG4o36ZdyfiDGfJHsg9HQaLCJwxFVtzBQ8ESi1DDKV9yF8QkS7RR9iTRcKdgcS5F5RfFdApvM8tISwWfd
zWw1DFweaC157VWP+nhfErn6M2qn8CBZottDDgz5xQ0+ALECw6jp17Td5C7F3q/0B/DFVswADzje29QapKdsIO
uNpnTTsEEB83jedLPVvW62bELz/Y1PapSkNYFdJjmp/XprBoGoULq79sdBcwI5msBderQThDmoyXV5+KsGDY3
0GeAknaXjJ871GRSQ1qjT/wMNwSX9BJrQ56gHIoP1I1NQ9znGY6oHDyov4EE7Z8E5007kZ+YU7X20addCIT23b
VZ3JzW15K1hHCZ7NqQm1y2pGgN8vYJ8JAuW3ywCSeSS3n1/ef66sHFKY3+vX9Fb03TrZXBPsqaA1yn7GnZHiUK
U8qtC7Nn0ieG4BWrYEIk/BssnEpnc3Rf3g0SIA66CIvqL/b3N1jjo0ux8EWiAHdG7GTm1NeihIqcfm8p1cmF
cHQzD2kRkPqIEEjNeW816v1Ir512z2vkCcvM1h/g6ZN10Ksf eoPtpsmzW79wTCCB3upTWGEJ5Z/u5RhwNPeptm
hsA82xtcNQfcx0NEB+mzbwUdX/Gkqg0RevVaTU0XnFqyri2i70CL11U1zY1bwzZ33t3xRpK+uXnIg0yao/nz+L4
wJPVHjh8/I7ckLf7g2Bt5uQRsTbNUrJqf0oC+RfQKDqilwbf5At/kRxLWznbawNuZ1L9PtcrFZBr+pWmVIvyx
KqNuUir0LBsa0XVNYxGi865GM2YK9/IkISk/IoL5BKxN82ZKM80YmyhPvKctqzIqwb56BdQC3ZkBCu/ftQLKW
/brsJpFd6TUh//j3WmB4pneR0hc/yZuHydGYN1JNjdJhxBxNinJk5vvZ0dLct0JBsQx+/0wF/5wDcAV4YEDpJaI
Q885oa9szdu7GkZIpT7aAc8DsmYtrFN1YjEd0kcm5Tf1BOZIu0YUZKACTfS13g7Wd1R5+W0km6wthmMQiRMGy+
RSvvsv20arC5rjQADHFeCEm3Lmk3sYN3RQW0119HNewz4P8tHeR52RFjKu2CFb0bMIR26swRZGcm1Ee2WP0Kw
2rY1DjHR3EQcZn5mY1fDv4M1u4UeNHzhjL14rV0ySN/ZQW8EACHD6WQ4cyLSTG8iH0sysNMejR0jd0Qfak8FmL
11YB9Tm2vXm0tevJ6aGQxFkPLYrvFBs+rYpshuIlyYqBU6a05mn3Xbwg1czBrmwognUxL20FSeqiJxTdrSN9TY
hzuAI8vebRtPinpTaYJp072S0ocSdUwbJTyvBcm5XfR1F2UAGOEdZHUEVdx+hFOZ1FeeQ/p8rALXnkM3UORwW+
GaIJf1XNYd1Sz/53IDFCTkc+eQZycyUdE/yEPLLAE+athQWNVuFVZQ50PBA1+Nx7fbpH0J89WEotm5qSkbt+8F
XohONT+J8zsPGvncnkXM62ejBhVVfJu0gio/wNyuuwh16TbJ9Dgw+vJcCgbK/Lr2cPyES2TteSMvRuzvIW/hwm
kkUJgy9IBL4W41gndDX840rxm1sxu61vSA0g+u+1HDzBTf9dT34aIL4MGKuuSjhYCh8b6f7jwP4pQhwUFyFQVM
ajMNKwpzDWU0apYhOS/sgX8w214MoS7jo5ulck+sbMQMCr2RdxV/kwZdsPhKv1M2pMcJJxJtLQF6FxzNcNNU4
9wjxGjGor6ux4aAIrbDLuXN6+g4jC0qTw4UZZd2qgTNlm3A4U4dcceYku/wgYdcJIjtGNLmiU0PkX0kF5AfL1j
7w01g2d2Njef/XGi30DYiGZ14A5Mfj235e/WBbkMWJvdNa4U7CkLK4jx5A5xPoRIR2f0kbMyyvJdX0hGzB+cRQ
kHaZLFTTduNUGLNUy6x81ir/mmX03qbTEypisefjKgE9g5JhXmJAtWX/YZtNw0ZeUwZxhCM1bfkkeqXQkIi20E
8165sMQX1HTjATI2LIvpweyJHCSyMseHqba2Qd8r+mS0vhznc9q63ZJAFj11kmOnA9oYCBhPC/1ow12euoa94

gNJq83bHFx5w+x2Cd7nGyL9zhThtFEKghuvRtsr3SvfQeq41XqIYWcnc11fI1cf4SM5KGQ9epeuXYOKwnm4vMG
FxD4PtyjZbJXLtNlH0g8MtHJB9fZVV6ZcG7kv+1sBaxDf2gHt6JDFIu3/Vdk3WqA3NCWpimWlHQpjWA6xWL6Df
I571ZM8YV5TB9BFndIJMguCwn8agCuRDT2wsRLCjD8LH4fYANBsEmM09U2idy6tlwIqWz9Z36UQKKLC0NL510P
zQ11RzpLNLHqJBQAoiuGgv8payNsRsWKks7jHMomrt/ruu+uz7rp+vXcf7UKmUeDmzDwK4dRLSuIVy0siGwJbI
ygr/ZxPK3vi4J6PKaYnq9/miotTW3Kqny9FpFT+txMp0/2mfXF1KxGpKa6/dXac/qkeHy62eNuKYzh1rDboYe
ofNbnYTudFHTGMU6ZR0c7Rz/JgUL8Igaij4zHQIPF0oxCREwRmeTs0zPkRC4RVo9XmbohX2Ssw4VrGEtAIfAJX
gEY0iYpphGS4/bjvp6yhvuhzXV1o1PUu3w/keU+yBomHTIdcj/3d02Ac/u5oFfLLF8BYLst6w24oqa/p6zT4Ap
Muq07y+Bh99tgW8mf1B0vL+BanBWVq8B1kF1kckbIJ2UVkDy2D0bLWcxs9AYs/uuoqQ0i1qgCGDgJSEUbVbMG
q13nVjwSbCDzsH4ndETSLdNVPy915Jm2AwNQ456NH88BzELqDYeITEkcGhqnwfhXMUIh1tVCz2GU4uR2U22N2H
y94iYCDoQ11seGtKhapz7Uy1gJshZeDN+2RaA5TD6D91PmU1oL+dYwE/9Uy0euvVFHcWukfszuP4abBT//Tf
a4ALsZSVj0KTqhhNmmRrxD6DRF2154qJHr7pAn1c/2x3vxuGN9Sg5mUU3YUROvOouyQYD1/AtxAXLUWR5wu9KM
qqc5w+8YX0edogBxHEBnjY+Ez5EQJ0u7+1dkf9TaSj0aUoTALdUAWK8yqz2AXe1m/g/qJstJLu+mdGWbc8zvY
nqCuk0BmpYgtTeL+A0h1D6gZ1XP1HLwWIdGY14y21G7cz94NYxHj70AdyorL8QoQLSuQSgtPcn+FkuLM+H9S+
xQ071UoNCysksvJUihIH1XbAgBb4L9r/N90sW1+h80bLXn9yWz000SmVG36fBTMqgvDWTN0+WFRg6A1xz4YCM
hA0hj30ymvDBfHQ8h0D6zJS8yPHEjxWm3PxGx/Xy9ME/HoawNe4sv0LNuhf93aCZNCuHeVzFJtZUA3knyp0gaV
zQsXDF20RpxDGqRgTp11o3HcJW2HsbF5mMk6GE01SYXMxPh6tzEP1mzRd7gXQ0W3917eX7YLu/8EW3HN6jCYE
6xZ5B30mg2ad1YhCQC313T1rKJZnaGjihzEvigb9TwKS4jt5RTE5Gzx+9jdDujsgyoDyrLxykI0H3TV1NNRrI
CrZ6p10akw3IYpNDR/TpUzAQ83d1zL6vHdAldIEN57iC3kkpC2vIuizjD9BT2E23eH/XCKNsrbX1aNa+T57Bm
IhdkgGnYq4okG2SBm5sfGtYLeAQvG+TLXjEnZSwRIBzx00Mhe/oLjxDp+ognY3MqaXF31VEZkzCG+C3JB/aQe
4XaPjwEgSIKj8qjG1as7CudpeE0RjCWHBH0xGpr1knFfwzB24/eFXU+3MxTDBUdd/6rhK2SCWYWhNY/HyXnPD
Y7XbyNkucp7PA3dbuB7n1h145w93c2KsQavR/6Y0/8MvpakAIh0DJDp2w6nTIdfctwQPLG5PW0a7FvEK0etx1j
guJzPn7BLwU7PbkhrTMMoReWJiZ08NKMndexgTvY8YCkkPupYWrZGAuDvfUyid3iR/uGBOTRF/p2usdg08csaH
q3wXiBjorPPqP3VazxWpcYhhu3dXNX9zImS2eXB0zIQYLLaJ4d1y5KYEbElfXbx3QcI3/gdmR4golfJ90NZ+
NXuV7JXbSoTC0cCb8PKEQSuy690rXPJD3TM9Nh24hMmYnKDon7AAZ6ZVti3pgNwtzh4JExloFjyVl3Smjnd3zJ
Hk1qdQ3Ht/JT5Nx9yPCoj2z2WL0io4WDQpF0hp+1qF1bUfKKvxz6FjvD6vP1fQ+3CG6em9W4ov6hd7FPr/HwxB
PkGu5s/T5LTZYRCvp9HOXx4QobUsoQX8jupyZq11B1GUYf1DcC60kkPnR1uF8ge+ysT7o/93RzPDY/dpRURCKR
H3NLjbvsC0mS1t9Sjkdm8fuXehJ709CAxS+27op+EcNjclCtI/4tdWdVE++T1znbPts9aluH4IqZ7Lv6WomSwk
joJvoJ12swssWzOPVaCIhpdtYCpn1xWQ9VGLV6a/7L0c5yBtv2UTgCQYW3eox91jFuCAVMLGbpX2wIx84a0en+
40xodXluT0yhEslRcIiuMp0+k3p4A+Fa5kDw1i+FABwrEQpbpoT1wdceUDFDwv4B3CoFJ+VksuVEx2/Z9kGewF
1L9FF1RPrngLAyT7PxNHudnvp7ZmP9rxr0R5x6geVvzG8JcfQimSf1R4Yi1CsJgHl+g0dTwvAtrkXZLHnIumgW
PyYYMxn7G7z0U4t030zpgtadTfqKGyHpuWq655vXYSDafSOZAW4qJisPZgnhVaRbptogVHUqRJmk6wuw9GdJt
1qYZkrEOUTBm4VXPP+MjqqpvJkxILjuS885NmL01U/17rDhYJrjFoorjmU8/a9TEaQoHqQTIZuRjbPN32hKEk
ONF4vuEwE5PPmaHs6QnizF7HXJIVoe9fts4JrtwzenII/n+7Md0ZiCWFcDyhYteJCBfHg420G5Z/TKp29jxkZ
Zh5jCQfUusDSwbU1loU3qT7qCrZ1eXB05vopvTjNQ/b3S99enDgQmDwB2JqKfindHooiPrm62nRS23oWCX0iq
fRv/TtXX7rRtQJguh8013Nu6UcFdU2ZJFqhc/ERNVSRakJWpAlk1gT4kQ5X/+sCniPyah9m93iQMLzK/qSbh
Uu4YTRL9GbDqBKqw6Lhs6Uty7g5l1LwMrzgdZdcgUfGaDcjuLhwGIaNPKeLaIiHJuRsk7w0zS9E5MCT1SbKMN1
K/LxKGnzgqvYwfmn8s890IlnukvYqrEanT9/WMLV+zTqUb+9FkhUWX6wkq8ffRKp1FkBU3FCVLL4ZRgf1uW0eA
Nf1GnRL1BG0Qg4Vs+YuRq3eR4UqRtI8J5G5BbjcZMgr3knU4ht+7WzSR0kZgKvQExnBPNxWBvynG/ujkhSe
dF7PqMgHe8hAuYN+Lj21BVUriUN3fI110RmR2jK0zxNvexTlw/9KKOCom04Z+d8vzF9kn7r5Wx7VBPUBwJ1DCm
1RodV9X1g8sY9YzjxhymBgTUCXx/ErXVKemffZ8Co/A4UHiRnJvQAZFj009w6Rjmg/LJCH+UX/RWibih5iAMSG
ewsNJUz7hq2qpHA6CSx29DIocnU9B9daWi0Gk27y25TGYXuJAZ6GZP4yUGq9ehimZFDjZp/jXSmQMytDyVPI5
YHDWnTr6DWDT7bWtA7iMk1Pzd6q7UMOY94bYPU70GxW7GEou/uqshZp27nhhPKkHktNf2QivLrltErhgGtmPex
EEQuV1hD/UY1E7uZi2BqjPFSWcY68WGL9X0c1j8pdYwr4UcNqZ5RY28LCa++ct+8mpchQkISKwPnIrmEw1iSn
VNcB0IN8bBzSC0yEKia/24rW0HoxMsfdfm39pWAnPH1+VyyLiyutP0zsCG7m0xRG3eLueMt+p1NLwG/68ryymX
V7vxRtKnQIZ3/u/85M/cyq87PsxnWZZ+m9RcR0FsEnokWz1bj9VL8tEVfYhGK10UyUoo3dHEwOABWfnReSeWx0
ZvuqXzyr9HPo+wtQXJ2694EW0GW7UUf0mtkhd7CTkM6iVXW5NwnXFe7o8iujX45pIr3CkwhkTAG/k5ThmHxsqy
VbTNy0kEGjiS6DyNHY0RWktKtacucuci05Z1Zciy817g9tMMfhD1b6BaT1CmGg5H4yJ0VrCXfMnbGQa7csyf
VNauAvGMP59uoTcv85Evb/7dCEuxCa/ItwrI8UkZf3ia5p7zwDY1fgqXjwYfZPGJQJ3B7ACYsj2/7qcosz7
LDMxZQEnBn5kgxtDIf3gF4I4HZzzj8SDqP/Sq7T4JYBNUu5Ir88aNZEwN8XfySWm5isk06VTDZreuz8s5fKMgM
3KL9Zr/1VHMQvPijxPk6mU0kfgguUzeGu0xqf6uC+jbWjZbxnaRcNIsLC2FmWFPEMz1Ifv4TP/9SmcahCoEcG
MLH1FK4EVbFtomAs9Rod7bd09eM3b1DCCxIF32ce4RILnx+R/0NZ3u14Xtq0JI1v5kdbwxeRZ4JvZ915cdLrXN

WKLsxJ18PHc6m6k7uog8EdJMjyfc81ttb2luUe6YOMdU1W99Yrr6YquUJ14ICJfQWzfbtNDzfb7D4XJYxA4j9e zwHL0SbjFLMB/WXn2aEUx9XdURubbffhwsrr7tweNxLoSfqFwcaiwik1g+FaUucKv1IJZZWYHvmP1H4MrmgEO LP1hky60fRNv/4UJGfGoM3caoieILjTSwYMHSwswihvobeN4N8sP1mkjpmksq/J2USgUK/Vrq1pi8cS6S6rdW 0HYcMhmch7wotyVfIc8NTer79r327zIJjn4EH320V89B8RSEGyyJZHn5JGc4uPMZ+AdyOnr3JPzuZcSwog/f0Z QvEEzrUnqzsT9vMFtv8Jq2K3DT3huyJdD1LeZIZA60LQdd8seLUHulvdkzyKuWv4J2UXP6vXJ80UoypLi7TyTf VGcWnRWApHsGxwKn4ZEr947QRmxrhT4l1n5S67kMqip/i4DNBnluXIie6MZQxkp9eVnIFGdRU0fVUdu5IK/t1L9 SfyH7asWXFRWQSLACcvgtOMIN96ZBfqp/LNC5QSz0XRUQWJ/Z9NdGqavT4a0DqyeL7ybqg68rhyGX+A2aj0wR QnK7cJ60cgLfRXfMdYEhSq0AIoiXq2PdhHxn+Ita0PxR5uppvSixnk3ds1wKu3G3I1QEFZQ9GcRhsaIc6Rt97GA 8FpvHs8oPcv8wPutunLL0D7f7zGjzLJKhPpW81FY8gs2mVZtr/T8ZSsC4mjIzyn31hx2SwwQ7iN2AaVm9ytvTE Mh/UuYUL256pmbQiB+7Vx2c1nz40xxZyGG1K6hedxMc0LP4LqhiDjT8+5Y1Iu0bMsmSND2irrbo7VL5Dtof3cT i6esykpjxvZ9sylBUcwyadgk6kqr+JiGHcTfUyMwLwQfHF10ZwCfMbYowuZ90d4EUopWUDOTQ1zqMb471kgML4 0i30k0ngISnDBU61IytZ0EH2zop0fvwjCo0L9aC6UAt/oDPLFf86pQQFdj6q6j2EJKbLoD1ZHm00yhg0dqKvGh /0h4gUnaUuLjWIvNDnHnei7T0LQ0N0r0HUsJ0bEUwe80mwtabjDgFuQWpm0E3wN0RYHAyFXhRKzGNVJJvju0Sji SSCufJkYnu6xDNR0lxMy+1U06JNcU2hU2sN7HzfPH3p+gMW1Czw6I/HVCcaYC9XSKBRqxRz88V41uTNsmADfMN KMRT82id+BgGyRmBATLo91h25tk5sXx/qRFwxz2X/vLH6HQjwSn7vJBj3nXPfasTMu7NVcdI7u5jtFqYTh4Rp 1m6T3h1mJdfRt7n6Yd0WY15F7ikRSokmB6q0UC/19XS+CvOKW0z7laooIbS0guozGFTtVB/1mIRr7d4IZrChtl V9U+YZ3bHcgeqUZnjTif56yBK9mhI1DqxiwEBXrnTopF1N4pLLC05awYrk7XtCoMz02YUkRN0bXrusLwUpX/in fz4l+9NUqPNGZf4+ZxBFZXp+XSrh1EcPoJ4s0DJ2zoXxbFizDCxZ9Xv7lwE5sh21u58ZobUvR1MUUnwo3B1oSIT KwcC6MyGps8fZuuhbTqPbW2PDn0RHhzkZewg77sD9/cvL0600kqoWdQn19RvsgIA+p+01ZMnxB9l0w7wIMJI2z rLcq015xx9CdTau0aKB3+2zyCECPdVykW1ZYcjxe7C6LUsi6Sba3Bpo7HLW8Q42khez7dbMLrui8sMBzTJACL jAe63nKj8hgzbCQk4Vm8vG4jYe4Hpo2BxRQdkLV+RMj/ZWTpQ1gIqWUucNzxU3ts1VUqbUI0g6apetmVWprhDd jXx3+8pFB12d1kZ6FD0tVJVsY+kCc50CF0Fg8p0vE6Nu9IUiLg59sGMBcQmjF0/EqBjI+iVDpw4Lq8qot/mr0a PjEYxiRI4FFG0x410Kb3pcBIDF3fh/SPyqtYrU0csgEPFgK0U8jHrGqZsd3kIqm0Ruyz1obnRs9KaGyuEka0Vk eJ6ZTnyHjM/kpxEd9bBwWRR61lvqMYvTE+3fTj0fR6soqZMVlzFoY0xLYi/M+SRFYrZewm6noGzaNPgwc1zX Sfg2qI+PsCsdYQef8bUeXY/cxAQXQ8JSI0SXZP1yxvorEXG1CFU5A3GNeI7jVwfrWTpdTyLrain76ffHMOmkb8 cdFedbXIjeBV0k1kGtT835F66PCXe7v91F/qWFpEu19yA13naVrjkgU2q1QTa9Hadrk8PX1IF0oXeYhzc2nI2E oyVEhy1x1Z2R/geqSM5kXaUKJeNh7XRmxRhyfwbaUZ1jekGfJJS EatEYVLaBusXzdoakZPJ8dpZn5Q58HSQFb /VQ1fLzeZWzujRFB816xm28H3Nh9X08Lt4uMbxP2zCgDokF1EJazrSR3eCgHsmkI++2K9JJZK/8/hh8TgrbcZ 5pW6SQ1P9IYEmhKpcOyQzMgTAXh5WsdCwv1jNEDyQQtrPpEb5RJmDrK8QEhH+m19ke19fFAmJ+T150//Jx+U4P FEF8LLvjjfI1CpsDCPwMkoL12UQsLSXQ8mvZ1j0R6BDWah/uSefdRSaCDkF/1A47tA5oYtE3iMsk4pNjGZPIx1 HdEu9094xpISOGIwjBQqt8Zh8zwQUV2N7QR3LhIuh9544W8uQvsw6cfZis/Co5TzacJSbQG9pLYZjFD33SYA5N c/00xEXHZVGEqFB41NQkSB6Tj+z1QE+pXmxJuekjNwYRbhviEh3UGw7Pchct4RB8Njviri6d+Fa94vpx99o1 Pi/t4H4v7z/KEA490d2v9INiU27mH0AcSic/qqCgBSZAL3cYVN/BC6041EBNvSr/PXdQTe6VXe95zVrcDIzuD 7NWNP97Hwn5IqJH8BVudjqQrY/03eZERgZWYUJP8xIs9R00Z/8dUYBwh1ZiwpCgDf23Mh3S5r57VaWR6SBtCQ 1Z+w/yU3h8zugH+5f6Nv+Krdt4ZJw87sJ6ZpEdnI11l1fuST2dTNPWDNbvh0LSOfE6E2Td4d2t4jHtE0RZqEIN B2Bn+52Z/bJGUTfpTZyooAoQ1fyPOuC78G1VEV/w130NTMU01PkKGq9I/g2CJbYR2ePGE+y21X0AZud9wKPgkU 2P90fo/zpeL/sKNEVSX8D0VSGagdFzdRK8WSLQj0CqP78YdXyEgSNsBRZS57NqkpKJT9gC/XS6Nyg0Dr6SyLH7 1oYsVtQv5W0xbEBnmSBVgMN48tirkN5LDVnKKdTndJalnZrMeBRp3cCPCCq0k/dv5S1zEM1Zok89GLi65jGTu4 IZXk6417JAwaV2JASDB6azFw5q3Rq3e/NxVvzFum21Qn7h7BkTX1X0QBLUSvJuzcjqhMcbJ54uD5ApNx+n/OPM 78FPjNnYHxDAYNugnnng0MNI34aI1GjB1SrmhphsYHaqmrPxQ/dAVJxnywMPsGKQxiPqoPnuV5I7vkY70kT LEfgBDM6U0g/D5IrdG8HvBCY+OnnJld7vcRFCuj04dM4CalHEfdG1DrQSgFz7+I3WNydtFz19k6cZaeza12w0 o5/wZkKdzT+qGeCqAi460gX0n7zN4rVJKzbMAQV9DQCDaUtz1o1d6k1YdRHWS/bWQriNE2X0yWyokvobZnLLf1 CXAzeRErXTfQfapPeYdh75160IoD/4hKBk5QcQwrbW7XVceQBoPilRXnshU16y5d1Jc77VCuC+QB+R94S1wZa woAKtFprRK6B28zTgt190qsDg8q8u2s0wPqbnHaa0r7hJk4pGBFXqhrpagR7z15Db0TxpjnfY0rKVS5a17lUMI iD1oRhZ03AoqZtv7U9C6uc6amN3PZKn40txNPgbjagU39Wo+B5Texp5FEmsiugin0X40/cZ2BbjtYQhWo0qce wj3R8q3UcWItcKUj/UAjtKzqaFoNm0yUczxcJe/OM01YqesN7KLJTp0PgeQ+7SMkD5cL2YD34S0ciYiBD6qagn 5x51gi7pv+wR3W1Qzw+zfvFM5nVKQnW9mhWi7r/oIPj4y7uY+zG0Mt/fXx7V3/FL3dQ19cR+fbRjR4KZP9xMSp r4MqspA2hwWy/7dt+dRRut5LnpjcpOPI6F1Xr6GQS4xBoTmbTKh3ANTun+nzsT51N6whkTUqHXWSGXQr3vV7E8 GS+XV8mQZRb+A2FZ50w7hE9K2KmM4S1FMr0fGzuPTsbn2bjGxZoW2gx4DDV84Qy1ezonX/twtk3RLwUhGt3NhC o9aj6R9W85ygnxAQ+/w8U5wPypTssRCxwWuxT0sQMLAtUEFPkozJinnoi1yMRakyhHLL6SuAbpCift0A2PwWJX U4xDiT2AR3wgXL2S1ITZ5ktDf6rmTVEtld103mMAM3dLnqda237IgZ4+GaA01yB4MqQHsa7LbIj3VbIwST2Fr

ATGBOV3YcgpZFZJGte4RIIzi9f5J2guTp4e/3ZXhKM+95uHIp7dcnbDJUnThs3BvNQbuIfpBcXMpWRnSxOPzmZ
Zbju2tRc/mqvcj6YJ62jIp+NAbrRAEP4txx4Nhn1pb9yAIbeYlkSbwPZ3e/p7os578qB6ByWpqZuwlx03EY4
fcynKeB0szrAehhfH0qRY5pv0j9gc7sHb0/fg/NPpPsTfRebyW+oQ/av34MKznlSJQexthBMnpQbLBhgjkiLrz
o621bGaylGGv01WwHvTk6A0AEyRkPH1DEKnFjLqdTEhg0dWWmxvZDaHSnTs0bVauoG1M9F7JNtDjS37Zp2r5Dm
lnP1m5NlwLssD+E9Ax+VZfh0g2vWbheh9ERWg3pSc17S5wJW0kQBAZZrC1Z8qzJvHm56iEZ3YfVWSivZqusMGe
sL8UvtejdBws51zhZ9HBIBn+pCUOEKz8nZD0Asq8Rcd5q0j6UjPFZKJeOznTgc6fJHTn6xifsfT1zRRg58E18
t5T13h53gd6dgWJTDRIExvbxN/NfDudv rkRp2IxRNpmT6RT5FWREbIQeVXMii3RBI4Mu4+CXRW6/sMgN1RNQom
xW37C7LmmAND5uDY9Bxr0rZUDh0VWP3ME1E9xecBfkkUhp5Ja0pfZ3Ao1r0IokNj5cL9He0i+ZVwm4pdNz
mGiuvF2Cm0G/iX1vYqHeKXVWN4FkRCVDFUpVWRhAVqYyZSXQbQCkt8SvKaxAdMPAg1vAb4gjd6lHseUp3Jgje6
FFJ40v3cBh0sRQB2dbRjUmuWILhPf0G8Eg9vbPZD9C4N3wCom3xFM ZyfeiV33CBm5Kcu+fhtVtvgCWdfbxGwv
D0yIZI5UNHv2BQFvXNyo200M4+MLXogMwqqm7GMTukq14yzh9jGvWhY7mMMUF1F0AXUPBNCjif3Wh/+F4W/us5
2DWknxVtZbeeyuxJ3CWi0u8awokpwKmBpbs0KZzVmeZn6rDTHIXgiQqqqnXpIXaNCxDa0AnxonOsbahMhZWGz
yxS8GX7DLt6F3IEyG5pY1XifhmlwtawVFyqUyxu06po23c77I8RFsSCIfe70iccEK4wk6Wtd6GWijJGGnuoP/T
rJzheazeEaSwH5q51Hudzj1rW3v4xgE7dHYbQ+r6td60D30I07dKZ2JWc3/IZeWwlJoiyJun5SONp7q3m/Moel
TWuxw6d9JGs8Am0FjT8aK4K13oS1ckrrbSD50+c0CTJbi4Tt61LJR0+ikq9GpwMkd1fM/bhyavWPLTD6bLP0B
QydvMnrYJcF58o9+6HunNywBU7dUA7Zem8dmIuye1rT80H35a0vMBbDTvdK28GPK9vAW1uwqpDygL1TnSH0KFK
yq8FNT+a1Qq7SF7R201MzpQFs5oeBo3RbvbPXgff1+CH/UnMvfzWXp1wFstyp9G/6V633kfg6a7Vr1KMpzdvb5
4xArnmdExrqKmickFAMDz7BZbE7JN+G6VHgdzwFRwNVU3sCmQjdpB/5GRUD5154nS9eglurn8WTUuGrENvsF
BbvWBIR2tLWCQT0DX1mbjuVvRKndjHbVTFF5bPRj1YsHInxWo2B8p0eswhtXwhAAy+OA/RDaCi1V/EL1M5DqhV
sMN9P+NT9ZfbIcZnXt821XG1VpR+oXazzq/vM3IVORDdYhGwSpVPIbwkjKyc+17kc1SJv3Wo/rdLPj4BZ7xbt
krYYzPH1dBxAVHXX3IOHHmoCKjKFhgJdeUBmBNIsy7dBpt50rs8d2BZA0wpjdXUiwf7FvthI+DACxkPF3NX7K5
9g0oCg01m1+nELLyT9m5FCvxd3x8y0MNT1lV5qm0826L4vgVRZJLWkZgvt9tbHcT3s4wp7Riz/kgqxs7ENXgx
F5VjHiS8RQVe1B/HzP0aDX+9rV1RUggPGgomYk8g5a6w1CSw1f9X0pkUNrUzW0k5V05kWYwSJ0aGSC4026YtQ0
+TSU4wGFp430CIC+/9AsenHN2zsbbh0Ma1JKXMtw0qfoG216AiAWlyNILZv4SJ1zFXV8Hi0Wbf2HVpKK/Sd20ki
7ftP6xcI6dua8AL9Db49drC3xWFhtXshhNd7xuDADIhbvPSJ0x6LJBCc1DTxu5+jFByDaP08G5114LZzx1kLD1
FQ8HtmRrzCiszj5eSIECC3fr+BpGpK418n53xdg7Jho9CNbqzUj/4EtCTtx/xyeRtJXGHRpsyETI065B5VtWH8
8Vm5zeSe+Jxi6fEgdfwKwmuMUGdBIWaQ5Gh02Ha6SX2+C+BQWuhZsdyFa/HcsuLpDhUVg6BxEuxTjahMDODS
4PrFZtSXmJ6qYjheIaPxhYYZczRLGqfia9a9ap1R+mB8aSbzNFK4RCMbWJ6hKFxuf1JgAgJMzgK91p2s0aV1Jb
iPdh39v5ZoaWH9sMBAqdh09rEZmPQhB6r+9H6y4pQbmvS01zzJdfNk9XzHq8k1/AC1KDyupUIPiYa8Jt0+OYnS
rJnxGXmeBHLke0IkpEjhq2rc3Wmecujjkcths4PdHuoIuc0tjAua3iN+DgwtPfxnj04p49Npuyidpbka0zyUr
fXX4BWpOIL9PpWSOVTBmU1+Eh60ntrYjvz09mBEP20vwqZeT3oZDRfq0gM5tPfeIuivSsoLuH5es1pk3bHSr49
V1015ahmrFQx0J129mk903b1XoQXW+/6iIiUV917zrpuCE/vkl7ox4KnnTPvXUOVqQdXHs8orbw19tpGL8+w6
B2gAh0912dEtBsBhvURONpNwcAd3/6KzV1vu8t01UdhPQE8tRSdkZb30S0jgatedUmIeEveq5ozSvvqy+d4/B
fQy4x+sg1aJDB8g+Ka/z185P+fDupb39YEC8bQ1A6TKDBLAPYhvznMpJPujA1/ja9/TkQELoJSq7eSx8wwAZfw
/aHRaS2S/Lf/mn/5uu17F/wWEeCz00ak1cFL+Yt8FcE018HtU9ThZ4UqHqtSPi9SanjVdS2gFeAtIwNPMpGs0s
1T+J42Fzqny119zGffwaGHyuCeZTBsk/GHAMT60MvC9h0ho2kryQZ7XANAA1y1a4pZS1lkmp+yi50HscqSdq0
7fgA0t+NRqCi049i1S90650fPiHQGfsJAEoy5oZWAF2wokhTdrzSTJ1Dn5KiE70zUzfiP0vpwi8WCgr057e+
30Xx0Ef/kJJ8StgR5clxXAU+z69EDH0kBImr2sUm6T2I9PzYtKhwZRF5WFx+svlQ44qcUwOSEasvKBmce/oUaG
SrQpXMSznW2cwJywBrevuAQ1U6wwLutg0JZQHicYpeQhUycI0KIS2Yw1GMdnQReXyAWW9JgnRX6wV7t3rnD/Ga
kn3KzpuYo6ZdIrByFLEDgPpj5MLHIC+MbWUp505crB7eSg3Kr7IUFuCTvyKKVb2wLgzq1i1hPYDgLwpauJBGs
+2kCyTX587ADPIWK6Y6nue0G4L/wHWW6qVvh15MqNNAunmY8qHWRDyBiAqr15GtqK92J11JprzYgd0g90rR4
5Yny/hbTKvSw5kHckAXQ0u1mtYPQJZ9R1uW8aXf1fGMk3twKZubmKqUNCIYsfVexTz0BvaIF1NwuIKhQiwVx
5t1GvP6Jt0zdLoeS9juJzqYb1HQOnRR7CRrciUDKvq9nWCeCxqpwuBo0JDLG7sC62HBDXMZ66J0uSmUsvtx/
wZozrayCSmEYBDGhr8ixjRPKCHAaela9TnEB2F+0HbKx21yKk1TnobR1Cq1VTuopherz6qUsabFLdyZ9w4iEF1
2qRH8uGjr4DI5Mu5Zz3fy0mgz1Y9xvds06fcpaVbTKwsHWKrq7aThWuKpAuL10w/1sBFmcJrUr0rdROIaL+
YA1A9YVCK3oR/yPfxfpzGgKpBFH4GEWVYvvZdA/TrcxcgKMTe0uX6W4Bpj8ygzaFQNB60JpNf7PKsu+AAibL1C
Iaw1ZgL3ZNot0NEEkzRq/giGkci0vkdBGqrZSiyr/PZ/1A52c1CaqqEKbL+U94ds567t4pA9CZRCbmJ+44TdAm
xzZTfV+rzDtyaAygJDRoDaiCAPWxeAE6zvE9BTDjgZ+1yQXg4UMHzRzPfXMZ+RKx009iVM7RRivfmpT0Fr09fbG
Jvn2+IMc7zzES3iF/8BLeT50agkqU69ww97CKSYnb69egbAg+/Q/ntXEHUiFeF6J1a03riq3+0c6Yi31Kp/CQ
gR1Byp0KU13LFFtJ10/CJ6MCubKM2G9vAGe5zervF1f9oCvxDGudGx6Wk/H4szgPDFq3N2otAYxpeAX4m2q6vy
dAS0sbI1TIL5Uhf/AlbiQ+AIn+mRwTgIwg2e21uybbZ+7jg3L1gtVaffR1zinTt/h1419BoyjDfCGEj5kvD3g1

zEn4o8S+MvpzXTdxhDmRuw41Ty2RAEyNbAoHp5tGLupoZJw/JyJ0DxnQ7mMIow+pLdqEa4fIxFzEy6+WkJ+j06
1H9mt1Dwe1Mi20jmrnZJiw7vdpxrxki6Q94nGLNQKD29S2rHR0b1vIaXevIFB6yj1SR3dvh5c7VcFiNHnQhDNM
vJIWnCk9HzXZn5JDOA1RYCF8zcG8YhKMACem55UomEPqhfEySdXuUGydep8xCh/qvIPuU3BmxJB79RrA3q98Ks
Anu3qsoyV9RjgnqzNkskre/7djaR72hG309VREBEVM3rnJ7gsmH4r3vLwL+hvuhAdL504TNgv0vjpruDPxIPKR
zXD1M5g6UZeeWCam3upgf3LVGMAh2tynWFnQGKqD5nR6AW8TulwxmK85VA5YmgzN36j+GsjNwN0tmzgtsFXDn
02wJc0uv/kUu7YH3jF20H6V+7w13C1LEm7WzjRL1PBAsBkpfUGqT/tQUIh74VNWj+ZKDU0Ub/LpoBpSXgEh9F1
pp7fFXdU/8XJ+r8gmVAucsSU10UTWdJ4J+anJBjCHYv7ZJBINZGryaWtYaj4Iyd74yp62jiiHe8UWAh2458WP1
0kewJTUhX+/yiQBPomMJdAXn3brQEvv3ikYx1SZUWIMdCofzGDavDOEhCL029EWTHefrVviNe0q1R17mpfb3o
CRBPLHPebfof9SjmW/6jFBGByyQUsTbkGSXoQutBV1/KrSyo1P8cI1Jt986uZajVrUcyeif9TT+8rhFTDBD1xwi
/RPO+4xGQZNCb1I4Q/C45j0JpvpVYGVntykDPaLsb7vRbiWii5SNUAD74LYxFEj6o+aIMKwbZPU5UsuUzTNWgI
80Ds+eCV+BEgxcpXYcxyYb5wdAZ7Lrxj11IERkBtDcBxOsuiFnR52uh+1TByGqzfmMnFuoqR2Sm9dBnq9c684
PMhdCFZA+1StgKmIf1z5/KJNFF4rYcuTarKcF5y4Z1JFb10GSM+nIi67mxehdxqfORM3f2hRx0iMgXFdt66Nda
bQAG3ZuWwkXYBG/0kit1glCvGcoFDr0v3IePzhsx1pTDLVupeCcJbwCZrcfxKh41P8eCiKQmqUVvamcfLH/xgy
Fju7B8hJS1xRsOpRupANDeD9UcESAQgr8bmNic9VYssu0Yo9igVery85ux2/o3NwBF5NRL4iojbagpDvf0sX0y
RBcYfTD0wIcNsZip0Ge4vuNaY0iKZz+1f/kPLDgrC80UzQkfguMnvPkWqjqwnnkUmeNoSvxk1/hVSst1p01Hom
GQZDIQpnMV8b3MNJGMHD3AxNzZaKe/OqfAHH5krBJmLucNPeqHOIH7gPKBHEYeGOHnBZba1/3r7rGS8h7+nHBj
kpyfbCcB35E1FLMzJqB0G7mfERevDnGRErgUj6qH1WUAIF44+BJEshR4iC1vhCK+SazG8WKpcTemmxv2AweLj5
8v6sUe+C6aZ4takhWdjAkB2qBqWjo/0Ha087cz82mRn/8vA4CqpxuLqCgPvRotnb96sNS+IR2z8iz7bZkzjYpn
E5YXP0QsoBsk/J9DIO22YNh9iAqpeieKjsyr9bzZdhDUNFrQgo9JnYhTAqxFK+SG9Yxot6LeMK6ySaGXabeKA3
9SE50E9scgxZzirMjwk0Nqauf0n90EdC/LpMEw5zbDpmU0+MtjGjFk5BTh2WNia0wjqyVUcbsyMXWdQY2u4fC
1hh+g61xTrMA8qN7dWGSZmb8D1Z0GRc02fqaHsc7Qu9mUy67NPsjXwmVRxPKoTFh5aFtDX6V7k/efZxKx0q1
x/y7/fz4aEniuy/l1HSoVT+GX2HA4rrkcZ5jk310m2iuros49qLd3xx28314AiaqZYx5DFaw6sfKVjE1j3hmo
pwNnzBcEYM+k6vnc+vNunezsvNxzuQhWgKxF7irapQL9nvzltvk5Cf5uTt0/jamkzCvhM7gxFo0bnozJmgPje
zQpwrT1Up1yr74pW/tAXouNQhjrH+ONVBj+akaYjeeuL5A7Xc0/M9VuY88bmipASz1DgvaDH6goa2WHX2i6rMA
mZGkDnxjTvx69dEJEd7m7BEZgPWXM6CvyyWbcSiRsQbxINGA37C+ecfwu3Zn0K5sCBgktx8eumtmk5hV844LpJ
RpDmQmX83zUrmEaSpAPwTZk3DfsZV048Xdh8xWikyLbqCRD/TupfpaCBiVq2EtGc6h9WUMZvA07z/QNksxTbs
/JUJrV8j1mAsyggUMThP2kzsInRPqg9Fulq+5o628k8hUFR+UVmBZA7zg9WVytikmG1BJQlxehVLeT8HjmhxL
7z0rruW8kQXXd9+2FFFa4uo7hUWJR/m+Cgezik1XkStJoxiSQ1ppv4udCba3phFH76XabyF3Yf2CbgkbCAZpZk
SG7knIIxTkNXKv7fQUSScTEzVJ+IhnGXT2jAreX8ma6FpZop2uxM4/KHx0/dU0nWD1283Bsacb60D5QTX29200
PXShKa44Wg1ZJYkaQeeR8YcHxTsXQNV AjLMZ0aZVzbY93F0Z8f0WUTQPCq5nf5U6V/p0tdFplJ7RMJw8nB/kE1
GBaj7I+cwy+DLoGmZKABUL3RKPDP2/zxIGxtE2fgK2GwUGkaCq9VMXctWnzEnELmjPEOpEkkC3F1po6Hzq65L7
L0sM/Cij+/16XP9M+Wv8juAjPIkFec0M0xGpwQbFKere3A4kYjDn1lwhszuRZMrfjMeXWTkKTnb/0Ge6giRC1Y
7p3H+V1v04ZtbL0+5m7G71xUgETXzCXI6rMVMWD5hgPaZhUHlwjrmB3VAIK/ut0061+HK1yMX5I01MhbwBumJ
A1kZcqMLdAM/ZYPeq/f5infbcrg8WfxAwqvgZ2ZW6ZRV72kQLCp+AGrywN7egQTpBM1Qk0q1Am5D2XipWcdZBM
26S3becz2gf6nQ0isSeHEZZFYvJQqP027CP07AKzE/ir10uzACePxpjirRAXAFKuHnvSf9mayfuKcmRMewR7J5
S7VvSII1DX+0Z/9axuBSRCTbT/4xtI8U62NOQFWIP7R1oYyyzeNVPkL0dwDvEZ3THTn8QrCcGBHn+MvcpRtEU
xGmN3mUCBG5t6rJImMLG+c9au4jef0rB1XIhopV/mHwVLF1Kw0gzJh9n5otvdNKh88kAftfbF93Z09nrk9Nj4p
URCtUyIMHzsCh4ozcmSKi8EtMyVZPAvMzGh6uEGJ1bdr8fxxesITrTT1mNpD2LvdrrV5GRx9A3d1CIXKiR92Ki
CnDqcobc4IwBUApthrVeAMNDcikvbu5aDg5goZReEnfAjvoMKXk71GVoSaQezFjIunzUPMPd7uU60mNAUI59Lh
X6WehMBDZS3uztgvb0F1J5JHo7RhW0tuzgHryfLWBOYbSrI106kKgFdmZ2Fw7CzhjkLEzYEqp7dmjViYRpmI
MAy4v3fn7t1VdNdrVylaxXYPHDfVPU/jAKgt/5RKe4po17Umy0jpGKu460CJarhAQxIiFgj1CF0ygKjFnu1JXq
LoacahUOpkYACZvfe3901Xp4pZ+f4ySzHkzLpNjfkTa1McW0ulfGx0rvIuqTuDaIpYgMSrQCHCzGRnFboXvGqm
WMhag1Dd3x2KbIoR7b1z7Wkij00uSwPi4AHC6Caev5y1Io9ddUda14LppaaEiXLM29LC1ZqyJMTge+5NaCMNU
VwQdXUIXMBdDnU4IdwEIr+Rq6xMU6Xhh+5MB0vYi4r3n54eKu4vTFsMdcm01CUG4Feg6HqkMeG1atn6xMgf0ur
4yW5th7+5CNssesvpzULv2VdPrSoWyx9YiSYERPnjYu8wtIIamPK9j1N0EvA2M1I1hZVQVNWFnMLMN18Qr0/K
9nULvZMM1AZm2LPJjNzG1CGwVm5GT2ms1vGgop5G6e51snV7/4bNb0A3Z/Pf0RLNdCGRpKwmRrJsQ2xYZS/YS1
tYuxBcS9/2Ym60QIVn30GAX1pMNgdlepDWkbbAka0Lz/D6qdRRQ4PfdtdnKNqSQ6S7c075dvoE2YtdZ2ASy+T
SbSM9TvYBSjsQzng06td+cS/Pv+PniUVGZEBxCd0ls3WRBiCfnXoK/xS82MJCte68aG0d7UvSBv3fViFQP56m3
5yjj7KArYdJMHemMapgVDG4fAbzRx0Bjatz06x++/Wc5pLj4r8rujbUKeTg7R1CFd89YNzZT2XD/0Uu3mBk5jv9
/1m1FaxrHT7PM8AJ4IXy80RZJjHfUzw6Z0ZXuS+u0X+8r8m4JNdyZSD5P/8zL05wmIRzPSKBAUSPKJEwGYXZw9
sB/vIUsxxHJMvfBndOBxNPSF3uu5dg6xYzdd8VoL0Qmv2HeNkACBXj1HzbBrwKdPeqpmZPM6Jap0xpeHZwPrxQ

fsSAK2bFQ161JEf0BzW1Jq/BdFLa/QxsLxzCE2Wur01fBtilAuV5A0WxDMokVRpJTo4LZ4Z7q76VCi60fbEe+g31rhUvTJ+Crx56q01TaAQvRQ7dxtV3QQ/h2wGwKtJsTIMk/Xox0G3PWEsNy2YeZ7auffFMA8IV3pE2JiPeez88c96kqS9VSSW6nyg0PsIYh9pV8vAZ1KY0THORwb0FNLl+mrhM03mxjMVAW00aMuT3Z/qcsFMJ04xNwztnrZTLUv/w1KqKYT7zMccXRGLEL2dyTygogeXUGstwsM4IJUBmW0iu2i8/LfgKRK0B1v3JoW/QUiMQX+Ci05yevTvc1A6mnPPNvhPH7oC+K95h0LHzmS8YSs56R0bmX10R2K2IdYmWzWc/0B8tDFf5iVAzbY0tptrRtFJ2/rLMVgsfaDT3C9WifMuGVGQ4A6HPqhBB1bo2tKDOX19s6Yz92fVGFeaN6+7Fqep/CxZY25H50I1+bGP/w4qaH/tGv6Ip/+rr0BpyFCHL30ZpENzZnbrrWtTppU9hEid6S+T091hpHX10oNX4ykSE07zRD+4o4DHHaSj10ZVOWPwEBKpCdEb/ar0sjRfe21/9ca0Q7ktCu6gfsjGrgKhhgyjVARDhffVa8IeQTN7yfcmCBcKH+90ZNx/tfbjC0oUhpBIzTLjvEmpq5F0pwpEUJ1+uDYqKF/+MP9N0Me8zfGvxXYCdIYX1C16Q/N47IeKSQm7VnZLjnS1c6rpQRGm/aiBTC1MU0gQ0iTZ2RDFTFQFuVCPSHtTQC293a0vJ23t4/cMp39/VYq1w29tnjIEBogZKris6a1Es1FHPMcc90AzPghrSEmprbbSF8a61c5e4xms0iVRuu0w3P3oSEWsVRkL82JD8mjbjaaHOMw47Duv17PXEsr2YC6wkPnMLDZXzTAtg/QaAA1b1cB2DLon+7Y0fmlf0YjfBcVJqsRx1xNLfUL5UUTnLsShTU+myxBcVixgQIYZzzcvN0uL+8XMqE0398hAeci84xp051cZJvqtFgI4V5x+CM2cPrcP49bu3495e1uxuWYd2atPV0n9NUNV5Lr9ErsX66oteXSF0voccgEjgZQDtnLZ3fyq/EYtYsg0VfD03tJSKaR97vwe/f2+bQWidQS1s23h8F+/XopWZHIYGJ/3aXD9Nss8ss1WZRbmFevU6mpuE+hgX34N0HtF1+Ch690ST8/1KFzueF1vqU7jilk0cMTi+eRCAoE2GB0s4jMvj0he+Taa6No0dVLWsdthZcKojoPMuNvrR09kcadBHRPFZagP9nBy6tmBdRosP391N8/wB4irFnv8SIeg0bqGhPCuLNVf6Pyrsch1lqrg0PYpi6kyXf9gsTpfpjpI+LBgV3YX10C5QpAVXWEMhS1rtLbW5igo3VqvTxnvvM1twkoTwn/7K020JMgNmZe8x/VXnRZF83tzm5K5mX+mfBFLxTuMCKI/cj+Vr8Z1fZY5xXuvM5Nv+X/uBzY4VR1D6AspVo+5mU5Hg/gNtloh+Mr6UmuFzeVn7095cfXZGFnDSn4ahN2dxUA7pD5AX7M7YLMMRoMACsNQP5GSG13gd3bJyYw3inm2IbU6y0WRKbgDtnloCQ/vqLgwxGLKL4SzSKiFDDoJiemYup9r6LOJF15fp2QGw2DrPSqo/M52s2Ri0VFwIKk10G7hEwHdHgpWbo3njka97o8Tzx5DGwcmECRx+s3DDFgbSn+P1VqJesfD18t5dcxsxRWc6Y+4m88wZ8Rx4oN9eeSVgHgzk46e89VVFIPFusTkrE+vV43UNgtIt/mQ9+6apnb4+Ea0fnnxAuTgncTVI/uF3fZq3wha9mR66mKL1xTxuy2NdqFN60/5H11WwSkD5+xF/PtHhgAxtiWr0zw15hzcV6m8MSQmIBVyU9f7EAzGgOsNATe0UNz5U2n1yaRum0RoKg100J25KpH3V6brkg5+85DReNmTb8b0TsNg9YuQrEo2EilkArugSV6w0AaEVd9BAuGtTMj0yTdgZTVq1V+Bx8DCuYTt4K6vX84NkDOC+/rWXYHFWrPOIBYecLgsPhBkg8ijs48ycHoXeVV/T907nYJXxfHU/km7Sg+t95JUPkF5kzGA+DrJhoQbuQ8sTakS0jZd+s3P/0wvG9K8Joa2s01z4Qj3mrVfIpU0m58fZ+jyeFmcvkG/+zYwaJ0rzxGoljxbP10GE5oaXLdBpeyvo1dk9U4IcPQ9Wp6XsLDQ0EstdZeaotFc8STcgLW0nn0UmsvV5aekd1uYVs7XPm1cqq0V/uONK1QsmwzKvHj5ZQ5FptRUHoWRn2TgyI6UhJNCuDyUDZALFcy1QbYyzK+kyeGwj0wsfC+cnPS64M14TWHIVEx8b44s2LXJK/KhRxEH+0Db/tHE91Tm2sXMp11gQh9juSUKLqWDHcjdKQ/Mi7GFJatrCrw19k7QtUbv77FtAtI6ozBm38sEhDeLVz6pIbp90qG1+GAHYQeWzRQFYhWG1ZLCdgSCP4wbFZ7gdUUiEm7Dzppule0M1qZABQ6De4yvx56ab5fHCEvVmnuaW1HqJDPDvyA9yqoq2UFbvE29P0FlguXAR21JJFd4n2mM7k0X6QYTWZtaeV/mMo4DcdRpjFNxyIXHL0ECUueY36pI+i9n1lVMygU2BjX5hnqr+K9V0U0r0rpmb9Kp6kA31cC2ze236nAS9I4Tu0rakI3CEae7iTeVX/Us37lax0poMsvLYCu5QmHncNe4CfvAC4mYy8IT/nN109ByN9aNrWf16jya3iyaa+saGgDXcFILTR20+Nfss7q9/44inmp5xvEgf04DeN/J7zHr06jkBpz38WHRkYzvqSTNOB2P3Vkdafacjw78jP5fQ6MLYCvK4KesgKqI8R2nEntGX1mrXuv+yHfdyaAtAaL1+GZGh3msp5Et21MLcPmGvmtXQ4KxRcQ0Ch0XhsC8KUqsus4PBSC4ijWiiNiVLubd65MDZvhn92pGszecojhdCwrvZYCaHwZIacHbv0fyEBNS58qSvQHFQ/UAPXJiybqmI1Q0+0i1N4cfTMb1jWrkxqfNqcDeQ0MaraB+07Qmfh+Lc8amcMWlwh+U6BpAFhfxdpzuoAXYJIAFaz61UpP5AeYFgjQmjZpfjN+3901qqb/dLaUePYraxf0Bk+6BqTULwBK3rkT5QCA0o39rTTAC7tvBqTc6Hm/EmYI+Dz1ne8vJUc4N4SmtAAJd7oThc1JPK840e7chiQ2UUj0ykLFYQ+eqkFTRpCGZSW0mc6CersZhuGYRmlaTzb14oTSuVQQjjq0m6ecto8FOTIhA89p5Hib4S7Eh6T3y/5C1vmh1h4Xq10kU2DbPtHpxKgbb7MkAWBxhymdSa0j3Lg4A6gdy9kS9I7r/0bUpQ/ET/VpSdopvUBE+RIyzk/M9A3YuEa07vyjw9Shs20g56QnMWsTEEIP7grDFry81EsNvMe50wTusIk28j9cBKmHHUHRPCLA8v50KgMITdgml8MFNEoxtXnAkAV15khGtNhrjG1rgIJdY91m0DAP51kSXacue0qzGTJDYSpYkxTHReFjUYcXThy0izJ0/XTTLRodV9+4U6/MmdWgt1BUcsY90gtYIT0oxLKhJCbWkzfoIigDS0ZKQjV3pYo90zsLEL8vTwHmcldZqjn+zeF+TSYIAduY0oH8fj4ByJcdwkDjsb2LkqkJn0VnRj8Ry+0jsTUyUA0SPWDZahDQcHSiHUYdpBsLry0rYyzGU6y15moQ0srPHcWZK6LfdJ45gvAtqvwnd33yKrWnH54xZfVin9+1DC3K7Ujh1xtb14u85bRHZBA1mH6cVRrmsj6N0px3FzJNz5ZQi b0a9GztW0j5XboD6u5nBY9rwANSzUeUYALBsj9Zo4IHBid81nTmo/ZLz9SoX5j2zTpLfkGMXanGwMrE1lVxSa++n0V/k0rBmJWEAsPSLeSxrz2jBERYiocwqFGi3kUsFeBFQEWzY1HTAv4kSBXih+m3oKClw8sof+LNgKvFpj uGUvRPUqEpAS+Zm0RcWCd+0xFMraIc1GXXm2YwWQDmW9tRt1WNx6JDSoRHLjYc1V3HINVB1qjthRfoD516k+s4XZ9bt+CeR9E7HU9fmVT1QttPSLsfqUcLqLCMfi4YqIv2W2TzPHhgLjwnS07nm7VnFhg8ulxmpAKsI0kG0Xn+R38pphjsinASsH06Uf00w41Dj2HqQPP/+PoF4y5AjWB0b4psqFjaWqK73Zszhws0t4KywcqB5HbmP6nqBZFN5+z+ft8MPM3gg91orUGMFLSIufrLb9tKzLzQhrGJnuzzXsBpnUeBJTyTiNxEzHadjjX68EkMkEgAfVxqLKks+aKhI

pFp9yEmw9DdMZVQnWAf13+R/YxFpEW0bjdeQIHc5ci2Ci5lcFo0RhVULJG6xc8zsSXPPab+W6U4fhtkrIjt1em
MyQAu0rivW31Qo0RAzMfrsmq1TxqdZAJX/tfv016Wp71DdJrNEXZdV4NPQJkmKwzhNpteuu3XUUQhWAmDaJ9d
t1q5fxAVhCdkKtMpNc7w7jATbwXog1Q4dPuCy41I2MLpsA3DmUIhCvt9aIixGiW025w0CDFShKwf0i/YoWpa+P
/fu6CH1kNggRMqSsfPGwDbiI4jIEBIhWj1oU5XM+5AIbwmtudNi4vKc/2GZ5Ep0X1EmAJGfggpQk5VbyYoP4V
aMZRq9KnhWVj9y+YmzGgv7Kf5PDv9/Zjo08mCrjaMu730TcX9AaEesTYYqvTWkdPTThxhDdSFVniKKGQgYohwv
/A3ni5SAZm+pXGcLXb1jtPjG4v9C23UoD1Y3CM4K/1oop6WT8XxJcNWhsEvSm9d5/P0Wn1h0vZhIyYLtgfv1aQ
efWZ1HvBsfv3E1h7VJmUlzYHx0/Fb404ktRgTtz1R1CAXLuD7M0/sgb3s09kFk8CYVzv5x7cw8GjavnqN1fJ2i
fyk4I0aFmRCPerdSgsip4IdmjZSQYcZB8d8C6g1c76qyUm5UQj1NcZhmz1LujQ0qPTuJBDpAfPyzscEjYYsHk
VuMlv6lsBMJRWZI0fGaaqS+ASfsSRC7gRejICvMht6yyBcMkI2R321yGX1I6QztX5ELvy/cDj0PxqAugodocLN
yod9mSW3rpun0dp01+Q2mePFMiU/pCcHzwXJHJugvXG08dqkWR93DU3YQm+3Ygy4S6tZX+9CUMRkQh+THiA/5C
33E5xVE4IcqHIBadALYgeZz4lx+kNq578Dc16EG22Y62XdKw1sVXGn2znEQHSZMXQ1BkTUxxGcbWIFRjUs27EM
9JcjdSBzIvulz0UI9p9ycEipDDNqPPd2a60QMdVpcqUvTKAeNxVD72Y+qUpqT1E4Qxq0wx/JAIik0L5iZdEvcQ
AAFb8/LvnD0ZI3fuI+gix32m8TzRejUMFV/zMTu0y+qzsnSxZSkubpSHpqrQZHNN7x7Vh8QZmshqU+rK9tHzx
+Swyc6PFs7qbzooZgbHD0oEbIu1p1WCJIKWsy076E1Ud+rsZ7qaSsONIhFw+3LoIcX+StilvtccUFynKVNHME
hGweRQZ3dRjFHFBNzjmK8M/KCw1ecQwdUvapHX0U5Q1JR10pZOKIttDQy91Uorpa1BIs8auY26Xx+GsIbcyhzB
6ZFcMzrd1lgUurGkn3ebFDHdW/w1+qdx6a6YBgc1omsBZS2s2DiKOX0bp6nyrV6Xh0pQore+gteSxbZGbFiWbS
n180It8hpg4dR7jZfZelwKhs33/4TzRINv305wIFKMMcG8fG1Z8zU7y0KHe4Czpsg0A9HoWzJHOTJ0p7BQnBm
S/dMV87eDMK+viPYUJJPe4p/LMug00BEmxax7jdro8fIDA7YWQsyRLcDf1PwMzVbXAX5mPwN2Zf3fzqNrs1arL
jkEa0ynyrHomQ501A7U/2pFYna1DcEAGKmUFYfrjTUKJ80GNcmxbVVAojI7cMPWWK+hkLkguc0omXQECqeRwdz
KxNpuwKPF0kvIxWoJ1f1bk/oFeU+mmlnSUxyZ2dsWA0q8FtYSJIiDjdqnh/1yQS787fvfNETU4tQyME5ihYq+e
VWzfeNAL1UlzHi2JS1g2UbTvVddjVREn8wRSkRvNRW2Jjf44Ke61UzAecwwbXhI43dKcM31TporCeky/l+XTBh
b88gVaB/NVPM0ysEdh8rNbgXcXxV1q8NWUy0a5oEYb0wR0fYgsBLf1Twdn8rc0qirkvm8TjqhvYY1P0mnnyvGLP
upMCwuoUxYGV3udB89126kgkDcC/PG3dWwRZJ1r4E87SGBX0BkuPrer7XLmLX4WldQ+gqEVpvq3/EKoHRVrh10
JNvRU6TJtKzQc7+3svDu05WRXhCtyaa0n68b5T3ek5t5AAUrueo5jqCxqbKMaS7tBC12QBDhB8WjtqoB6pwH5M
M/Ab0LOuD88W/W1Q6ejCLV9NTqLEogMnFQcet4G4n14aso4hK5Mu/coquv8br1MCzFjCFRBfpckRpCcKAru1E
FZcwyuAqLhTqyX/+08gDoSiWxtY971EC05Ae8YgV7YVRmVT2CU5fbbvHCRWAj0MHymNPnvjXijkw5+/j6X61L
Kb8mSIT01cCSXyc0c3vPvPOTnaHRUsjj/VbNUyCypnBsBUJZ9EaAfSfXQq0Z5f4Na/r8NMUBpZfUtY6u+6ZmE
d7/Qu2E8f1ZJxL2BPw4e34Nfia+3UIQVyALKfwBIa+G/25A9EtTQLYxyWxy2DNve5kY58Z4RQvDu/t5PPFDd6B
Itk2qHYakvoQJhM2Jso04Jhv+4zdMHz7RhTDU1/EYw39kSIdPKSWkFACjkNDatTUo1EYoZpTSaCMczKuhn0cDa
Ylom0UN92Aq940w5oSNyqDuVxran3uSVoYFeiDqDxam9jsF+0hhvLE+BmZBePXyWM3bY8WzfgXtGtZ1T1J97L0
3q0JM1zg9WGbvDCeMgLWZOEJcaNMN4rP0TQ7cJXon9+TBJNzWi3p9kjXzcuCaWJN2J7HTL0D6G7+HFjELI7m+1
X1cCahFB+1WB2v7aE3wFc24VXv1527RX3CIk2zA6XrRVSmWzh9qkkZUqukxUV5V7Ce898CKmTE49exTGA5/Uw8
G8fojv0t2dZinLYuLv0vieu2x1cCi93FKQzik14904MXmoT9Sb53uC1Aja2CsXJVC36rQ9ymIPy35NQP8phgSr
rs8gvcK9WMTut0/VEkjicgK6VWTUgrrZ/It8AZ/PEyB/w1Vqt2gHA3yU17M7VIZ0cWCib2JMnJwra1jAA10xGa
ecYAJwAb9GYab+0YbknLc3E00rDqieU+ZL1fgCithioKy+G9XJ0EtUCA6/Tt6BLPZmvYBI0rJsT0E41Xe11g7
TE56Xz/Bt2pSTFD0dvK6pT5x+HBAP1GSArpwQEWbjMYSAzWJspJpe91NDjpzJj078w7sUqPjwTum6eFQayJNVM
IiPpCvoD6o2p/gP41KB2saXy1k1fSKWg13eezYhju80erX8SGqR2ycEf520x75jsW9qzxpJoIDeCg0DdX/TzU
U5MMngtTcMRUgFVpydwBsXRxICIFawxaI0cMxRm7WFoXqg+oR1/UWQg5nE1hrhBhTQf6rT+gZ1KEtCZmWmSGsx
8UsF2p0iF1Fzgrrwlbp/skjmByy63TqCraeHkHqInXmJUs4m7wJFjvZPfZhxBwzGUKM2FVCx2vVwNrvzIT4nQP
kYUCn/iXQ8N0dP6QGCZVmCVb4ay7xKoTnkY8rc+KSRZgE8kbyKRk+DTQAggeeW83Yd8ijJDYE2HEUAWZkgqA1z
2RnGZS1JmWBCXaLh9yu9moh/CjiWwAfnyQbnQ1LK7UFu1q7voiPnuS2hMTMBLs4QjvZYeS/6ktD9y0C+NzTRVE
/ZUPHD10fCJS2jnJ7gHkk7BnhA0KrrB3LJ8pZJ0GJVgdNQkLLaZP5fR87a0yX63vTjX7mCK5nkSJ1n0yMV4W
pTKwv6DnXD43fqMkWnkoK5m0xy54maTRP1NHJdhndwMlm54uD/xcAhoDhf8hG0tnd8h6QgmB6fd5yVEyv6vBR
DsCtH9B/xK/FhvR/dTWi1LzbU4DEN91i8eW2xVcCWs4W+dcp0oWPF5bE0p3c3fPzDCAFJ0J7yZr9dEWCPnwmcu
/Ssf+iY0s9Xba9XisBJL4TImzS9VPMmgV8V6TgoabRyKJ+S1EgNbHmvJa1fEq0VvBn1pK8edE8kdd5vTBsftG
WURDIEyWe1rYtJVRm2bAU1vaikwHTU5Yv0DVIEdGdb8errwE1JpnJgNKURUMy1a2GRR947nQmj13qH5bbN/48F
t/CSR1ViFu1wkATmLB1YJcpoZTYID0MeaDq1iJ136L/dBck7y5n86J8guXpBkocm7Z81hSkHuLQsSfSsd5Erx
BWWyt6PDLVM4LPd11xx7D005UTqXGcWnhwxNkdZY5JyPhNc8RFBYNofHg1PS/Mr0Evvlzpqa9h/pKwf/2LFuy/
r1hF9iJRKnYL2cXxxKZ07UDPvbrb0u2tym8Rf/Y3scC8xkXpKxMueaanXjF7hF6habzitk8SyAccEXa4QRkI+S
h5NVxh9YobYjLn7joyTaj1DfkFqodiE1rwIfy1TLw3s1M0L5keTUX9jDvdh0L5NP6YeXkCDUibWy3ZRIvmb
px1mtfJZeax8RWvs1wpDwbfJ/wG+g0e0hMVGF19aBvDMfZ0rijdy0e8taPsig7kmvsuv1XjMpra4N94jXmPkq

1dFIOJwpF2VA4XhP0xw2SRytWH0vuXhSdxuhmtL5TnLAQDWv/VpFTqj2BwAU83ZIq60Wvt66d0dfeG3K6NWk7a
Rkk5aL1Ge62fan4KLiy0BBigYErUXGp0/1nrX5k2JP+NV04XxRmeNxcBph9VmVNS87wBRTsAij+etKLRs7+sj7
/6w3+2j9aWFDYN444cBERroUBB0tE6HoHMaS0bajKbwm0buebuCmppv+8EhS6uCr0BrLGqEM5WBzR1Gezkn7A
BqT908poGvme9m0B7VV5n7mEgVigjrg0fngMEpc5EP7Lw51XPyzdASY40gt5jX3MIJiyosECgHncUVgmnS6hcD
tLQMzEIq0+i/thh/Ye+xfcMhmY0e2k4xZE10T5rBLn56oQWCqQFtB RJkZ7bPC6kNtaG1YprFFMNTVENx+maxpk
1vFKPND6Y8/7KoIkvcK6V0ecVcDYPHwu1Q9HMfaRZx9pBoPfQJGB41pVvArVW3ddGFvK9uhLPvcDZ/Iq8WNVBY
1meepnWg1W1HPBs5d8zg/Fbli+1kTEZR0kP3JY6GFjMKkbW71CWAXJVRZ1w3XwvCM+DNCWYMeUV5sEsKDYy+c/
x82h2P1JkzQXX8xKQV5aif0+sc+fUEz7MwCZzzs0DWc29yQBKcxQKDzd0cZ03Qg/2gqzeGAI4Ps+in+hMkRPim
KM6TzWNC1QU9bvHBcT80rnwYhpHX65Q3cFQPR5JgS8aQyvZzzjY2XuPxBkGHBaEgvJDH9n1A6K1RzkzqijBj
Qhw8i+/XGRKL/STfs0qAr9SRW81ctn1KRYirLbk+8cYkoSxjLh00Ik4MgvJKiLDEuEuArFFLUDnFqjwA3lwujQ
Www7uSmkPCiKz27wi1H0614XdsW1SGGaFfMm7bj1zypjwKrCNDgNPaNwomTWewkYHy6aboNq5z4lsijgtoAxKC
sAqfONi91xc3Bh514ds1qh3ICaBMyZMIB19tm6Xg84pNw/ogN7oYLD9A6371LqGSKzrZk3m3pTnu2AvzvZdYJF
EPRIEZOGXT8xPXLLibd865ctnDtYEVAHJ3ldiX9wZ7H30cWczchN9j0DAXtj/m4h0AwjBDjb/BWe2eqN7tc6nN
r7+OR1ooRB4DrdiQ4mwy3DAH4h6FbgeI18eNNxj0elrzjs15eJ5E808Y5IiKjBhs47+XcnIjvbjPwf12i7k7b
PE/N11I4EAG6Gb0+x8B8UmK03Xohk8Rjv1ZK9EhtjMaZ9dvh3/aUEWTs1k7LZZZ29V61W+206Z7MioCRsPqr0s
GH1tzXG5Ih/n3oj7hk0oyMTuSsZ909/ywWSLFG5HNFPW9BWQB3LTtDV9eJttg2u/3bk81xPB8wpr+t/vKBumA9
9ZDBr+PZ5CI8UwczK/Rzf5IdFmpKhrJfKazpp1X/rjcBimsHiev1N2eSzqeL7gaXR31DVR/DgAXcrVy03n0coZ
qEn+TrMMKcuyCq/hY7SsWorC9ExMzn6aBw3oSubHYb8cK+m701zPnntBtI9KSD1EaLeEcA5v9amzf1Smz22FN
XzMKTfxdatxynHiGXhWcvXNmSpr094iPsbIviGTswUAJUPWaA32ABo9DtahWWZ+8LerNIAdWN2ppE49BwTcj1y
F9FQXysSN5d6B1q3nGedHR4KDRSRg6psghh5M7h1LJF3z4AHQJA0j1Keyne4FJ+Hx0WmkM6pbelrqv0n1oocEy
xCnh6XcLy5+0weVpl1T9osaI3palUCHdo2nR6dVc+Yws3oKdw5w2oUsJCVQxS+ZkpNIhgcVot1IV4EbPacegyF
38iSrvyxLh1cr0ieIHaYre7EmieaKsHJoaAkqCsvdDS/gABapI9TXym7vNR11YEB14QKq9ACup03mR018Gp5of
r3TuhAr5ev1kN0axdCrDy1T8jF8vV51EPq8cvb3xCpmVHeRDwcqsNPUUqSJSdWfChpx1Jar6jApz5rn1vrPrUP
y96PNRSUeAr2U59Hhr94CMFq+wQp7Fi1HuPg4ZQ1dcVXUrUMG0WBD1CbKsCL/oaWeEIm+/DAefepMk+0sy04vg
C81IKY+LR96uCOHrSSvpwdBPcx1RV82Tqoaqk7DVyqjxZ1vWBmSa4mYR1hrkW3PhzT4jU0gNBe+1cnZK3MjkeU
Wfd3spCdYDW2wdSB17IZswn9cvhqapzJGQwuUo3nSRteVKEXYyo5+NT77Id6tKpFkOKjK/ETwzvFI0jxGgvG04
2VcM7i/XBycnZkWBx4Ewfbxj9mMIAofG4h/kgn21AfEsmi02NvdMnd4n9phLIQ131Eix2ML+hfYqz+TBK6GX9y
Q9TUEB2u0TsjsE//mczfGVL10xdXosv++LjiyTSnMstKs6g4Sj2VNPBZUmBw7vIPZsPaArTUPw8TQuYFqXF2XB
eVhptFF+E8D04SWrlTL+cabmuHFuPwkWK/hLRBdzsf0Bd0a2ARK/DE6vf9jIqx83qsyavQ9zIcqMt0wlqtIO
N/mZCgpm40PtCp9q01SRo/ZAa2KYeZd145bvDXojd4zshVWGd6d8AI80rvhmHuGzn7T0xSxkHIm1Z7T9+iCqX
HJ1zow4G7b1AdwTkpLxpTgHgn15sfK0+NPXWG1Lia229J9kEtk9cEVV2ysiCqFnDHyFT9i0Gq78pDxFsxJ9dNo
vU0m1+7AYaTx5fSSDw40J1Qta1/sg3YEF28W38jAfeRgiwwURd6/SjW1HzaLuFJV0c6B7wK1smNOzbRbcYLxF
Is3yN78r2kwxMRdLSfdsFg2VPiU7LbU70pbjLaJoSUx3iet30r0z7abh/T2sE0p9hvYC1vKrThT1zsd34g1F31
RqDN1oZiijsLWHRhp9Ua2jbInupE9Nu5wqSud0Ebm1ISw1ZROUGuelPozW8+YPLLQ+PjxuUaM0z1dkELN0cUC
u0UX+MmRWhvqE/UP1wM9068soqWpgmjxUdFTbzupGmz+ApIsYw+2A2DI2zsMeGkrb+q2zDu6305zpIEvjp85ca
hmLQZWGYdjILbZNp/CwDNW9djmMs0a+25PPhozEkUb1K9/b37Tin1TLB2EZUwBm0mR61/gIVYQs4WH439ecPG
Y3UKaQozIacmq04MKdVAfvGzNtu+2r294/50AnFts1mV8KyOeivugLWfi25s9B8m5AbsZyeKG52C70HFuLhh6R
rmVIOXVYR7cvQnm9pH8hz1PUSU4dFnH+qc8nRbx0hpafWnwnbVbJ9Mn1nG0uoNYIttK4hM0413SAG4wyL3nF6
ViIhg2XE44Wv62UEUoUOLsBULZKec0uBzqTc3ZB25pDcJI60ps537adg8N8FzQ3y80pAs3CvL06FoVYe2hKQrr
j1UdFXjc/H0vJek6Kj1Ls7PAa9B/MdaQ0RVZK0t1ZedTaJpgBQk+mt9AzMuM5nJ8xExTEW06gaLtv5SZe/3v
Vp+1/u+aJIUtsyvzXPNos5QnQMosan1fMcLvIErfAUMkvba5y9gh/H3BhYty3e4NMAj1Hfg1tSkt4BBHX1I/Bm
/Cya12DjPA4R2o01xGHIHEw7hEIGM/eVzpqD+gbZC2VdgZiGj84Lr8EX3gcLuJ/oKmqmlhg+G5WxbzrzuuUK
D1fUeVWVJUEsv6adwLRbn3cfocsi49Wu1jqsXbpvfm0g8VCNJhHp/Agl1KI1AUkyLjWsL/9aEUWtGmPkzUx3Wk
afcigiJ0zzxUOC/uX1BEbG8mjuem6r2vZ3KY7Bco+iwhYzEXZ1r1TI52XbhAcq7Kv+W/whAXm0qwTU1ss040p3
c7FEjdhpP+njCSjaykegWMHq5RjjTt0TVnt6quJtwRqrCwdmxgHhM45oUaXr3f8D10iYalPAZln9UDSH3XGq6
90ItSTYtiuwbtmWh607r8hvz6WSGZfv4HUHD9Tz0qvvw5T0Ev8P7ME2yuxtjtpC2P18RrPG98ZY0RGK/15gy1
0pTa8eanSWEenj6A7NpfATZcEJKYUpqppN9oQWa1rSb0meosHnGrWK8s82g2ZI6wk9TLcGjYijQo1zfmNkxi31
stRASTgjTMfzDPWdIH+Mq6hP9b/Hik/7dUcDZYGXFxNs8aRqa80vH9ZP0mYCaykEo13RSSipFMH7H0n8AnmLt8
n7eb2MYzp4e9nZLB9wcbJ3Haf7pQzTXgaXShQ3hm+PuJpA0Se3pvBTyI8fMzzxQuApyiNaBRX1eExnq3ziNjb
LyLjdvEPbdbpzy92vjoWcavD1ITHHo4Y76u8CA13t8w/gP08AexNQ9ePkTcewRCJ1cqs90MA59XayiJffgU0f1Z
sLwdrYytLQWXpPfKB7rFdxnZZfJv7Scc4VA1EVyKy0/EjD66NEo0PWXTDzU56q8Qn68ydtu125na1DeC+oQEMC

+u0zVgYwiZquoq0tL0vFYva5YoDmbjmip7jlAmetnV26KdYtuuYWXn6Rw++vLQJpL7inzD7P1EHijVYg+WWXbcKXwDBAdPLT5jc30VRZUnS/Phka6839bn3A4ckdWsHnURI8vF1FIbw32AyNRYcCwtwoRFcUXHbo6vyVIXnJNEnWLXab9/jBmP/+BuBYVZHbpZk1cNRSFiyKSTEaeqo+kxGQriaCz1Mbx8Yoiuc4dkeUN20dVj1pWCc/c1KxGUeMS1KTelpIrdcKy61r2pvQcUXpMc2Bwzp3vY02tD1I9WiZGF1D53736NNTQPH0/d9Qe6rwnkU/H6fTgAVD1UkvNJDKuf7w7mmdj2qu2y13qbmqBp6E7n4Tzi0T10chD0K4630+AyWhkGND2U0ZieTbPYHdwJBW14/z/HSyYmk2E10g0EG7RIdbSNxHohXAozp07FGR/aKxqh6H8PeVz+ueTcdVVzc3XBeCKPpX+k1ehMtga7+S6wBZ2H0W92p7eNisBtMCrrvqC1gypkcr5Nb66fmL7hJskPe/9iSe3cGXhFXHwDjC1RiIVA+1X1VGpfoFC4tvv4WnwbFPQ1vDF1zcULjK/Z5i3VGI//SMqVCfjMGG/qtzlpMhDE0XiMF33h0Ygf1zTMYHvtHQUEGashAIcKCFbLs2J0G9K9EudNj4ISYuVPoTY8MGh+mZSfkVcVeh586bHih970fvtmGMVq35v5Rh9sknRgo9a100a3f1I60XSVwKy+DSirz1v1yCor8fuSQt8AkPExmAJGJpPDkAJEt4/Ds1EZ10xTs+bTJCrz7I0ry7n14NgzDi7wAhW5dDF9T7F1SRzj3jUIrg/SofGqCC/sWhxv/+u005zhGxmEokS6T3E71gVXodAkJWcV7y5+n0g+mquG8UHeW07fndVWGoSdqwdgbzBEEJ40kcFT16XFkcVqBh/1kUqtvR4wttxPYHHrJJB4BWdfqbTKCF8/Rx/K/hY+1kw0ucbk4r0d0brPAcESNmCQHyLsoHwfNSS7q1yZffknujolAxkwLLX2DDUBxEYPLZ0VAisRrz9iQiJFdIBYZAMOPBCLfo+HuMg70C4mX3haXr074bEb7Ecn4VRqPQrE0g5BbqfkK1MlmrfyYC8ULRT+9IYEX0KescrMEMmBr48iYTdEAOWWEIVgi0iMW45krWiUY+91jS8CskVaLAua+KkByIBjvIz0ieTkqMIkgTYBdo7Efx5K1xTWVTMzK5Y40QZ5nBubTAIabR2c3TV878tJPmkDba3SpTdpdIyNUEDXUoxVZfyIynr98IXqgD7UFiNagR4KHBUubgdc18NBox1mQnkdnyNtaxIyZASrJedwptpUGC7Alb/a1xtxLB0S6b2eCwIqNPhwfxP6VIy9EUuVERYLj1s9mpyTX9mQp64bMNkANKiZX1gHm7Vtu0uQlq15CFHRaCCJYT4xkYZzx+xAz5rFCrE4hk1bwFdSWpXcqsUQK8cankHw0umNecc0Go0wlnncB+CfTmm6Lp0tM0q9Gw1Yua196EK2nFZw2B6CiwHqRFjksydUJ5Dz4dn0cJPunFE5b1EEpp1wG+E7J0rc8BjfhV75BsMMdZchez+gvrQ440P08hSsXT/kSiX6XrfZHIGLRelfr1BFJn51WMkIgLYueoDPTSCQeJF8xam9E8K80bPI6yzSrnf41zrjU9U17a7Gn+04F+YniHyZMlgk1HBSzzToNhCLp+4N2R9MAUX5dSSmSWYDjIgM8ckGLgPdN14V1Db/lnfISPGEJU96WsjUVgoAP/DW6+8zKW/PenQmsQbTfckHWthP6Wgkx1LuM5K9kmd24cLha3dgnmUTWA7+bj6Hy4Qo+ki+L3isTDBxEM2sDIHw1G5f9P+ixCNvuVBCh+7YAi01DUJDS1Y8ts0yA/uHmqfjwQRG7T47oBSwWnH94E6GNsGM6gS4MmFaOBD7RguE7duUj2XnbB4i7pJ1wdMcZ1DSox21822rF5zr5cH6mRzaxB/Pi8jhaPsLT1Yx5SKKTxB3UPK04LorJch2PPnqkV1F7ff9oQ0z9IvgVNF/D14jHKnGvi1KHHFBuhI83cf11V0Z5VDn2HAbNHQCjv926+WvL056M4op0pJoa1WFU5IR9d6DcF0Mt0oC1v+vFALZ1d6GxQzEVT3LcBuWCntGvMOG0Ak9kes6WuqqR/aKMgv2zLBIinG4Iy+Kfw/6APiMzUQrooI616dH94pICGoNQ5JGbtmj26TDWJ3agwVmUqppSTtXG8omoDEMqfayz1Vhr8e1gEWd9um8oXJzcymmlqLLR6zaYctLWVEBz/+CK7tLxf+jw9ZmZaNS0tTWhg+mE1mvmpemBcaDjSvzcu+FPN1HQ1THpA6Ry1hpuK7M/DJBKh18T0sT3Kc9Sck15CZJ6pR+4o9AZDUjuAXDkOZ/mpw7R8C+z5pnp7qfVgN9PiPDChvwGuxrgueUUAQbY/hiEscWkE3ugIHZ8gCJeD3W0H3RPbAhyUAtHD0SbhgZ5jY+vejUgu0Jc4gZ/2DAAA9Nza0KVq56fzNE1wpC8Sg456+2KTgCwrx0s3vkRya/iz7Lct2YIKi66Z+l0ER2xqmFRiFFKevGE4uEMLSuNoifJSugGEhv2zKFuCnEoJ1wJXpVoXf5BKBwMmp7q6up2uhS1lylsMSkj6v4V6VgR5G/01++kNDwy4t6bTeNk96euWf8A5RQt09t31LwLwmFccqfw3kJf5xBVUXaby75DPqM9ZQJj/7J+/1ifU0VS4j31Z0G1Qmv+44hnV1AL0JJq/EocnHcWk7s/dp6HDp62KWH2XnZM1s56vzFEYVqb2U5p7tsyAzGPIGZfnU1EBpeTyMSyMc+sdnTcJPDA3ieHBqfUFeEaeqrrsPxx++sA3xxcd1CbMKRg4EhSDZFu9KBZYnek9BFiPjobkEaqR/3WfEJkv/8D9AxCM6TRshjN0tFFo4uYm/QGyL8088GPxjje7pRVYigBmxvzAXcNRjZPOQcZaBB1Roz8Y7uFyuAiTk4Z7qLjpktdD70BQ9afyHDnUG21QNAWraY6zZvD1+Rv6nkyZKzAIbKTcjp/+YJPh10Z0SPhxSD26+Cxstj7F/UoQ8+mvnVWFGoSp9F22GsC1RfGTMTwZCFC2b1eQKjtVzln+5xTjJ16V+DydMRaQ/q3tVW3jSNnZsFapqzNFJ6YshPjz1fPZRFRTuk8eY3X0AsqqeT1A+AR0q6K1PKdv9RdnS84PzSaTmdMVcAZplqlb2mk7A6+ZnpHmuCuWitILz6Kw0sHF5J5WJT6VP1B6Gk6Ef0xQGmG/ZkLMZLg5XKhx2S9bTo7ZRe7JZJcdhuHsfjVQflMvx0zb1j87BwHSqB7qE/x2nk1RUzyTshi3vPa921SEvFXnZ3E5Ib4tTddW8reYuqAZvCgfW2k51QmBHGGqbY51x/FV30L86MPssiJmIj68fJURvNs18nMYU73i4mF5GCUC32SzhjFv98bTbwvWkRggfMgJevdZDjsa0CxakEEb7AjJt4eIX2c6hKmetly0/l1sgkWxtq/JmSnjoK22Qm9j0Dpzfz/NwmYpA89a4msfIfm3qDBfsib4A8cRGHWSRDr3lnsidXcAnRJYwM9d6YJD+QX1dH3SD1IY3TMTWscDcedYR10IcnrL1rk+XQXe6LWGHW+q1xqfWwuQGI+ZSoRiZhpoUSiRbrcRFkN6XgNnjQXbYKZ2JUbp1yZ71rDdAZ96ZBXM7fSnT1cEAmA6+BDDVkfJy90mCT/ddc5SHfWzbUmXjXt59X5x7XUgnqGcrfrGJntcrgyyLptK26GYP1Fep/zXuG0D6T2zJZpemA6Sry/i1rjrxq59aTbU1L0WvH3s3d+GX5cPVdnRK/r6BR9WZV3Eb1zzndgprqRnB8qB23cGNyxHk3W7p8IFpP6TdtbtHxsuyptH0g6UTVa0bqZ5ZkRAEpjYTy7q2cs1vgvKsuFmXtyrQZU1ei7HG+4XmywrgV8eE+q+u3Cqza51GK8L1t21czLkmaZSeJ19KKielzYfsApqg13TCcFqKUITHg9196x/pMFnMP6qsE3FbTGSmlGtLpGB1q6U6kuhHr019VmWD25fGBoDpL/rUxjX+hjTfv6eymo2NytrFV5+eo2J711Lfn3TsWDnh5nHT6h/uDKkZ2ryF4r7x0Byx8VmQXt+v9ZY5+BTMNP3CDGEoyoK7sgoIHu9CkXeuRAQ035ub1F8TMQ/D8B1pW/6zNGQXnmbfaMPeWY6gd+anpr1I89usFBGB3ojPIuP+kQIdhH8f7gfdaktUYPwTjm0zqfxkyao0J1rb40meiuRDY4NVA0di1WxWhk6UCyr0k/BiyJ11JnUaXSwB

Bp1LmcvxHonY5+J0HyYTA2SKeAJoXQx0RGYmV04c66u7esKYjnBl+1nW/LMzxk1P9LR1EnY9Qx5mer2qyb2nUGGj810z0cLNm6D7RFsdM11r3sW05vxhaFPgpIL16EpE07vIk/Kv5I6ezPdeqsPuX6fa6nDM087ZqM/1HvCNW2X8d5Y1ZjIMjXJQBCt9bvAP4zBK1QQSdmB7ZJGJMeYjysEDaEQDHjQYjyVaCFNdFJbbLvr0TrmvFXWVUWbbZ6pvFAZI2BH/A8VMAV+305P34Bb0X4Zxq26qSDkzq/FYoKSIswxa9C68Z5SHCBUuWv81aiDFh9IHkRoYBFFhva8vle0NGd0u0rWHWN6g8Hd84zUnHOLzuN3hgTEgWWYRXNKA91jHf/Po2qRiQMQtD58zxE1cGSEp7RSRIqdcUujxHkfUsHqYNhduEETU111Iz4au0S0AJFmcehkIdD7x011WixJ4RK2s+N2GC/KXa5EKUDUadLg/bD7pWumIK5i6qW+z/waA6j0mdlWNBPIfCibiLxYj050Qha2uzi2+9TfGFckWrgssijWKjGFZh6ugYM1DhVycuf74/l33Fmi8v+IrRTqPueqj rB0dkXnwG6rk5d9gFLy5e5jY8rcGiZCrujX7ISVUEY3RGJ2ivR0bJEwjM5Wh6yjv3xrMxjyp4Eg+2DXzapIP1RkUApAfb3hk+5td1vCSZqxVUdasDisFocBjHvedpJxBJUsJFq5NwgQU1cFGD+oKkQVf91D1dLWHR39gJT1T+dLh/3sVpI9p5LIZ0tqgt3dUP/5MSgA4NU+5mEyPeMTBcf/0Rc6PW/rKKf6kVIyPuQ7yD0o/6D5m6Httzc i0IiZNLb6wQpyDP0Ry0ratr2WtxXlloJz6mN+HiiyVuw4TPUJid0R5v80RWIK4rRGRYSyW/P/6BH1XgqPsmp29qwsX4LIVGv0RzLpLpp+8Z07oEkiRYvPSXYTm4Tsw51Ibmy3BopVswVzj1KnAG80F3GQmp/0W4E00VtG3NVBbDEWm0dFjTB2wPJfpBW2AZzUg9E6SnUqoYFucZq/PIXYjHBSYz5xrW/eVw+DV9YJ8AN7bwBTZhRB33LaVWjSPUYZm0iYYcXrzC9w1HmVqbsqQfVYsWnpg7a0AqjBWeBdFCqAhGxpCBjEc07Giaz03+YUgMY41oVGJFysCKtL8ANVsRgNUqGy0ZKQULEuqXNHwRpaFwVQRedYPNCAScGjeIpNnk6P0HYTw5nCHCbkIYYYJAvlC1JeVHNR1rk8d7DN7sy4XZ4TXXq14rZhNY0/w9w8Mo7BvZgdsHpehquLQpdvEthbqnT91ds80rJweiwzQjDN6VY3czwrjY71C+4edvxs6j4jeyMpcAhdRLW3vzis2rZnLSLhVQZKr4A2RV28DbHkeVlaJyxCxP8T+V9yDF+v3aNcTWaD1Rtq19/ZsnlwmmruS2VjC0CHB3K82xxSRNCI/oZo2P5hXdbf76ZiruYe3EdXUg0kSdwUy4jLaL9/IuK+2kNvt3aYaJ0XzUCPSuXX7e57P0gxfktFDHakuGhD7b0t+35y0NxSC4qMMFBaEwcWbqH+rBS71Y2103M+BYGZ+24j3GIOIFr3Q9mBQFlrA3Cjbq2cGn52zNFtFEWJDzFoF9yH1K6qPxBa2vGyR0GHPx8hWeFmMWSXS0fJLdGr1dMV/XZ62XDWHi20QLnuyc5oHUBStcy3GNQ+/70RJtnveISB2tSEUBwh5ZBAUeh74KBx8phbfJ3v0M0zPun53yAoTkp04VUw+B/ft3hCNkVL080drmBUB0+9f6e9d0dLqjItrnPKoZWvflk3KNElHBtgt2LvvG+inko01fKyeL9Kvi1RhcNC+tzMkq9Vt0FzaAheInZQdEaJ6AYDY8bEwwzGSf2WAio0zPRGrG1Cj1mWhxFA6SHSVttKPhZPaiuxeSuXZR/sFE1CB/yxKo8M0FwjbJTn4qGdt2N1+lWtRsjiRyL9uE4aEh0i183yy2FQzVs1me0V374r2Pwygds95mn0syj1NAMa9BIHDFnuXemCGj/KBMJanjC2mQsXzS/E/yHdOUXrWC4DBGLC7onxt+yfNTd/zmCIhBv3UYjNZNc3JX/cex3Nk7CNOMMCgtpjGgwEhh+FXqWrhJU0brg5pWKZ5wVeaRVsRwS4v4q2ASCEkUm65gtxKoM8IctK/FS7ugDb24zxS9dWEGty96QVgo0ym/z1VMV1eV16qp0Y8Ldu18xf9QTIBoJeXgEB1dv8ne/MjNZU8WAQ0KRLjpDMobyMjb4MuvqzK801hZFrTc0bX0N9ED+TzvUMS6wX+HP2K9eJmf2kH0US7an98uWh6vozqKxdFpKj3NgxZxK2KiIT/KWJ2tvoe5ts3kUfUCx0C0UCLiW2Lu6B1P2+3MUN7//mDR6HjGT96qXjtHFWFQKA0nhfw/tbzDQnB+zAW6dE+kRjCKo6tnFgnVYNF1htuQxBZ3qh23hNiQB8pRww7oEK/R9aI0aDyj3Gn9iVa54Y3HgTowakNvasHks/wr41TRTy4bD951JsPikC0QrsrDd1LnzBPYVbLL8nsvRjFI8TE15JaZyBgIxXLLe1YdPMnSJe1oN97CwcsuQjR49nF9czch9Fh7eqyXcyA61dCJBor7RP7PryDYrzn+J1nF07j2/MsRpHGmL7QVE15NuEB8sTKAv52UC20yw0Di7bpkJcx8vyu1RJL2nkjKPhGaPx4ylna8L1cVe37Gf3ipEfka0XAr12YWi40GS0Ep75YHYTEs8t0UMISKX19pGfDzw68+nRmYq6TTASJ148i6atjNshKDv/1sTmDSpi9pvU0THgKnuELFotYClLdLSBBHTRjYa2hU4RC24k0p/+k10L+t08GE/IZCbMMnWH9hTqRxZteIRH5E/18Jp7vxGYwnE1Tgn00FaqTS3ICrLWK5yVra2ME4fxzD1cJrb40iTcD1GskcTS0cfJPKhFnx6ysCeDYqmF9v0ZwfrG4YtpksbVqBDkS0jMMyd7SCU9QtmXXd1NMfxNS0iX0zypKyq7VLXQ1RPnG74mn19DyItG+uxc5V688qqm0QeybZsCT1IchnAxQ15SMDvh+UuPj5c19t4r0IdyMCsPXjgbXYSGVC6VA53DaL+QsrXJ6fnmGH07fc5h1H830YNcAqo/peJaMgtmXIo0hygeQ6AGccZfRj46UJrqGYB0o077h9KjmGMYTq/S4XAT+g3fachXhQ2gjttGGHXv09p1jMPST1ZCtSkCGdX4iqXDe+L9GdD1Ys2pD11svAJ1E8wHDzgXbpHOIbbafBTzskTIQ7R9uLyCmEUU1kU0/D5731GvR9TaKfW31znDy7c1YzLOADNm/Ec3IHRbbkCyipD8D2t8N84ySiCmHxMcA96IDakMT4tyHswqUez0A3T3bRbtgdhwTev7fcZ/ZQLGmm5i+q0cmRjkceA3fCvaKfiW10+Wi8iKx1LyGh+y60d0/HJy7u49J836hYwS1sod/VQDgs+E5+mYtjNJSDBitp/RgKQmPNFaLTiCD1gU+XM1/0docrW7b01vPTDhY8AqgtR6np9kzIp7KgSJvJxhVNEaRy3YbewVfBGFXW8uajYhJWjucKHKgDBTkbTu7a8YFBxGRJttx0K9hbZ2Wa9dsPDFF2T1BrU2tSv+JWXcRi2K8/HGc8Bs6qBvoGQs+Eg16VjQFj8DSBLYrHRg7Bfn010eCSkJxTH3C+ideijq1rqx6ExQys8y18ZJvDytX8GfHKqU1Q3R541PNph/typhfwK0v/yGw2kYwsCHBPJPJRVFnS1YSBMhXMfVh+fWvZi5cNj4mlzaIT6MScNaj8CvrED0o606k0r081mrJkk5gQZkv7deXVJLLUGQU8u8mkkaYfp5a3kwYe9wZzLttkwquJZ2iNr11G0vnFWC3HdKfYzeZfbks2LBf/wcPognd01xEJRTu9VFxh7KdqysqGWnrd8X2FbUuzIwZRU6UyCkcamKtW2j1Jbr77zC1RdtVmJzdb38sJx+g26i0L0D0//XgbUqia01B5y5vXj/3zzmAyEppCI4ccbLNfSX70bGzDVuWixa1M9Vf1uI5NI0AH+uB1d0ukyaGxNWLcYff8rQQWw6zyZCI3Qj06XsJHHXj jcWqpiuks+sQvjMIBdNt7BQl4ESNKNWWZ/NxCmMTjHI7LITYprwk8rN1MzQ1mYop1+2FfwxtzoTMAvFeskIrmZTkp3njsMKAfz973m104H0gYERrb0sEujqbQYUpSg6SASC5++Z2KqBs1YzWalGsCveR+BDTTo2XML3093Yiv2m+ASFUMM/cxR8BzAJZNTiZAV1Vo3GP04WQ/wRQNz5DWG6xp/bLKR

FrJinBkoyyS4uHMI8Sk0YhEIOMNOSzkeiyu6G5ThxPXyJcpffB3JM10h7ELuMwxeBzc10JWpAP4p6N+KqDWSuW
EMTZJnL1gj9JXGXbIfy2IiU9mZE7MQGSJWdvZIt6ekpReu61z1dBeX0sgFiuEKhBBob6fKx1akcwI9y+iE070W
27ij8b3XbkwCP0oZbwFpswHr1XPR1oHIBjjo79HCR+EakgBm970e+N5tHzcBLuFhap4cyhWJEBgnNYzocz/v6d
X8YeD9TDN4b+vvQJEdSA6Ue4f7gLEihuHwZxn/t0f7SZ9emXPruzUzRVxh2G3KaMxPsTAEJYPwSd8L4c/2T/f0
djBvCG0btgeKX5+/nGZVm7tuUxJM6yb4patp2knNU7hruizozjuzMj+QPjRXLJnigPmjnmNaETh6nXXecX3v2Gv
m2Se9bYo9ivpDlzVpSjHwKvwEpBo7leoD79nrV6jUSU6kXpTYd60+SZfc1by8mdni0hANxBUHIHuUBmE0t3JRY
o/hxejR1zpJycnpaJ3CYhu/D2mBZrhk7FTMwkzoPUvLz6KBzvHyq497FCwte0Sc6jmNS1a5FvYE6+008HBwBuB
nAXTWDKDCRNPaL1NSxkTK1pgQbXux041sNpjhHHUcmhwh01AvBDMDMz4gPFL6YdtJ1gPmWg+HJyHHVeZQP6X4
IwdreiS3ns/jMaf1RM81kgrQPEtWLrsYHD+CC8j+mrAT42JhyaMwMxKhT7+BInRacDQF1sWXhwZvu75Bsg70
7bS7iXir5IAmqtUzfehtMKo/Kk7N9fiKLe98jM6y2s1RpKq/z1I2/kdmRfnZZSjDs50AAy6/pXTjM1sv8BISO
1I9BRaUONEMz1kCNbTm/HYyFKCHjd8wWqCQ+vg45m+DV18xc1b7XLuvQRUE2Ebqf1odPDAyphMhIwrmwnMGM95
tcgby7rpI9nLXCprZgZQ6qUxCVXUY9Q36wmmbiNr+0SzS3kYNWrAdVoWTVMAFGr04M82vwh552Mz5/aJpAR01t
4PZY09xsNNLHBi8Y7k4pkvQ95/ebVTuMkb0Nf+y5PtJ4SgIStmXpbkQAlgpPoGViJyWh51fkYRjBfyhyJs3B
SV/I8e81Ih1QNqV3uBiVqvHTjmSTiwFs74rX/41BPLsz85B+nI4QvyTS0KghZyKzWXJYdyE8TrMAvzcvjMZ0WP
AOBxIQTbhzbvN8nI+M22re9y9cktxRSrqucyySj6ndDQDqB8bphuphev0+1KZJct1A5Fswgp3e71EiQ+UbEOV
cJJ4XkPTbwYnvNADkaw4Tn7R7GeizweEr5EJ4D+LS01p29w0bWXZmiP0v+ABecQdCdtF/HkGRvCUzkLusdpp0
ISzK2Lig6jonlqMCMQyTHhJ3eHDLgWn63K1ZuzdYAc1qnn076KN6E0ZzpwRdgv+pAxuicdYEZF5XzrFS3HJ
3N1wqY6TaecL/H303D+rebk7JA/JdiHsdZ9xwupZtNLS+alkvSZy7g/YAMM3LOAYamTqy0Q4LXZntX7AQVHNY
VWiL14Qu3boQEpIEPX4QSpbSu9KDTDp0X0wN/MNNxPpXjQ0HjF/c1dUet2fWHvZXtnBGAVdn7tCgdQf4dKXfP/
MCL5A15rYbxw0qbsZPs15poP4gfmY/GKGSXIeLJ20vnRNnptLQd2ap4c1C+p8odTdmokfETnvqjUkpASqJHrh2
KPhzxWmZaDESNXLTAkPhWL+RA17NYH7k9ZQzxuFxXCB55rzR1/W91SFxTEctSy8W26D1WT67+tb1YBvhbctLT
BEf2PlnGx0zAAFwoRyxDe9ROAkMEF7sMhgWdeagTkGDdwohWOWwFtaFH0mhxTBp3zR1Davm/f5EqJRC0Ny7JF
ua1Uwv/UiSsAA4w80aSJcvRoDVJnzs9+1AJP0izby3EEZCjxCADMOVeV64sJ3SikMLgfE11HV0fRIuQ4U+rb8
0GNgVuy73++fZLq45n8vQ0yBBLGB80vt1Z6ymBivJ00aKcD4WcPWN4kjvsPQwPHGL1xTw6iJD3pAiCAR7VH37Y
+IZJ+kBGhdXs1E9Q06drx0vIgtuJbU/qXnTOTqURodICre+pkhRG58iYiktvHa0zWx3GiDv2zVNLNFCCo121z0
QCvWBtIm0Trj39F1GY9x7yE1/Vy6g1PNY5+847C8ajXjuZROSgMWmSGFXShJaDrTzUlqHaFU7m7JaDRor04Rs0
MK2AW6v+R/vnsF08w+HF50gnGaNE6NwMy6L27A+DPTJb4WsHju+egaMQjDPyJp2LsEHHTh7RaKXhhWpqjkYK9
4SwB7x36XaKj46KAc01HbP9F68RB33L5ErRrdJY50wL09JV2pKRwFnv+App+PJKWRNYhQ4gTXtAjReueAyURC
570YL2qtkGk/LOTbbjH+sRWry3IHTGamgEWDtVfb14leq9d0aBMuLo9zKbTdxGWao56h42XdV2jEaRSFucOPZ
YfFNLY013CcZGoSgcPwoM3Bu8iqScaXL0xj60dqBL6m4J0bBc3Q0sDDSnftw3VcrtReEwgWJY+RUKFRovd3ML
9vYtpft+sz9t8gXvI2Nb1Lz9KdVQBLJxvYwfb0DZT7qxUVpMzufB89//8XhWqTY6QNDm8qtVQg/OP5TCWp1WD
39CVA6AvMvQJQ/pvUMHrV23dCCd61PVEGQuek+UocSex3R/QFHh1a6d1J67zJpcal36w4ahi96mTPQvb/Idk1
5yyDYj4v85J6dZ71vUxhj/JYZGLkCn9UiX5feKEdRKQHHF7vsRRUEPEb8sIrpDh5NzNt0zEodsWVv509VMLpqD
Z1SNJ9Ma80qbb+zThfrewGVH4eeexXGSz101Xh+/PSsxTFNbPXurjt28gHdi7SIQsR+SSuIFIrdEBZqc2EONoY
EbjB5aun+ClyMcsGGH7Rum6jsrJqhaMz5Re/UwSzwdRwCKjUORhb1qzDX+JZ0nHigSNtr9Kq6z/92efaS2dhG
z3mL7Zkb0jgwjxKLxbqI1W7NX+Ia3L+gZ4y8KoB0uupL1vTxXeeA54V1IUC77z6sB2Q0im0zjd9/yfHDVzBK
/GwKpIXhZ9QMGh0qrmGcMP3rnrP1DnDEDI50ankSLQsd+/i4pIhd1Wgvh2m4GS10mLquHMxrwBy07kXST3ggq
WCMW0jX/WQJtv2xmc+uZgRgsib3bxzxHTI2D92U9QzLbVwhL18tHSHrsLGcUk8fv89U4L5yaE3XcSmMF1bp8U7
xmLRWv/9ixGMeNmkZ8a/9NSuYsVuzbVt1zm1RQ9hQmhA0drkdrJsgqeq3bDh1U4jT3gdHRYoQsg90gzUbC+Tz
oGKIY7wGxLsRo4oPzr/Vc6C1uBEiF9MuTI6z3zI5vxmG49tPSNaDj1xHqGciSrZqEV3010vK8t84N+aB11oXk+
3+Nptt0QWhM3xxqwTTWmbJ0+617nEqYdL01w51qf3aNL2nJ/cRe6BeLhQTE0AvnrbKMFf1ndZ/ZBRBhmSIQNjF
ChAOFQ/N+i2A9f9kN1jVxziA11NSDrwpHcZoNJKPptSfhD5pK7+EKDNEFAR3xKEYzbqgHUFxsYefoaidRs5St
9Wclek+0LSJAo0bwo9n4fubVwTJBGj75oFhC/zF2CJ8V1NHyErZg0u5IZXxGtu6cvwPCMA1pEJi0uQeD8/wqyQ
x1KZKzZDLpUT9nj3J0WzfVCbyhMiyZkvrc9NU+w74zK6R3KLJewx7dv+OCFKXhK7ZRHnxo/gfZfNb7u9fgkp/X
qmdAtVurJ5/JDCawwb7vR5qV7sHGF8vTDZV+/V/Eq1E/Ea0HxxNhyl3Mb6DqmYnFucU4S0sVjpfzYwnK8K7be
rQ3ujaunGWYsivqv0+UNMN5I0DVKHvuEQNSxZRcnw7fUMq8XLm3T8KUg6ESwF00tjZEUNFR86kb5qhv1H0kIVj
f7oYtG1sSZyFEXOTQ9JmmT0aVf5n+dpHQNK6spRNMoiiBN0CEg9kXmEkaQSy+4UaBmfpxuLNJLwSjJPIRB35U
6ff64uKA/Tofa430arHRufpFFbHziNid0aegrZJHgI16V9fRtSQU+ir0JbswrbtmT/cwX312j+tK//InF+zgya
E49L16JMDI0Q88vzR4yWIJPehG0+/oskMeqqMn19nM7cpVsYJzJyENMrIx0LmjuxhtY1gS1H0S6YAbB0WJi57L3
FLvfsfP/TMZkJ44U1yqp45t618qY54d48S9yyx9TC1Ia2ZdDYjR51Z60PbowE4tzUAmnzNH55HvQH3BbFU7q7Q
JAut5DcnY6eqS03WY/RSm6d+0Z5U+xtp71bgyj/aXe09U9HPYg4wRp/SScaR1CggdHiuz9u69NIN2AsTaFhj4a

XpawEkwI66Zet2KRI8s6qhmUigfDeiweW3eWrNPWWhJMP0UCfrM4VNN1XgdWKu/GUdhy8mG2db2LvHU0pp3p4+ Yah7TPR1FFkyuuZB5feMPYIEG/tc9jvoTi2ZhVVUELbyt/D93ISZw+dqt250cHVbCW0+4ajU8UJ1PVRKfKeaj Zj5LGEVN4ho70PEC9+peTPggwg+WvACGYmTW5MD2TXALzs7c3S9pCIxwFLSrQqOK3wW/PtRLRiJyRj7TETeqe0 cPQCGxw7XgGK CfddYA6wnX/TDZQxp0LUz0nDtvESj90BtBL+KAXIW3/KLH2u8cJKDkDg3TVvCNNGK0gXT+bhVU nrC0SW2/2/Jvgv7EJxL6pgd+0zJUDY0Cx3dbFbjQtHBIxnVUdK7MTKgWSv56Ttfru2UDaSabQ2+mkEqZKcHKSe SJVioJYVGJ2Q+Vx70PLUPE9/NOvTo9ResVqYz+HGxwf00utovGeQMW16IL/N1/Dhj4IT1N1F3I/8oAd8tX8IF 8Kvz7NNbi1n/i2IaE8cdmGT95MYHxyd725wzwg+4133V6LWx3nYU8N011PG3/mgljm3jzdbw87vp2XmPuEKzYR OARKJXnH9w0GJ/dA5PDGt2KBNUICsp6GsBZLc inIJEEMkLT7bid05o3KfmBpv1ECTiK7yXS61FMtwu4Fh/sR6o YJ71goMU nIIVDb+IrGf2Af4iqIZ3wjzQwcxEjBZy0c1PdnL1WUZb1vyrfvTfCeyMe5YjjHuXxxweGatush+iF pnUZezeWzD56i1qNjXixiAvpqNHG4DMRAMoZVBHWbZsEoz06GGH2zNK1fU1aRMzTdfYcKbq0SptTUPRqDwAodW xL7tw/wZFL6rqADVzzauEgE9Tv3BvDQCmqAL++2zSB51d+NahpJ78TuAuzqhW+egWJsryD0Q4MeB8dfgUPyctr kPt1ptIBQ1BHQC379KLVNBSyHI5vDZ6p4p8PkTu7LIJf6itUNzH/kh46p3ktPTqY8vUN26LJ4uJvU1iWSSpigb Y6vpb04i7jX33QKZxB7xiZrAcpd+1QbZ5jXw9DxkN12VbyjIB67uFutAaP8VvAvjuYadY8JBXiGdKdGbRqzobK 2p24e6s7Nhe0eFSi0gDaNVW8a69pjidje+1qzWBixA9KUDqm0/LG9JcyZucYZ7DMdeM2mecye1ehSASN5SbGrZ LGiEBpeDzcrY/LC5a0eqeLZ5IsYgb85zdnpSIXpiddF2LUTshrbC5dQY0ZmXA1Y767cNh5tIjpitVkfUDJB21J 0wsfxsYvWvHf00krbbBHyexQ1kXV1jLRF+o8L2fedIkv9L35HtQPdB7Fiv0OT+RewJQrsca5DY35AHXMNLP Rud 7wGxUD+/5BkJ+tr7x71BzjejMrymNvVRYej94ep6ENxec22f+PcS0+100h7ZxFrbIEBstLsXoVFqKhRe0hW23i RmB1cRP371AmPc8ZTsjgJZY037qQwdNZX8Zh5ouTW/1LW9oHufPpnLZLbphSWzBJ/FCONWUoC+T036+fkwYNDk 1p8rNx0TcCnbQm0N3tQkGfWF5bCOX57ccN+z132nGdxnQ4pJ1qtnA4GcTMfyap9+0XB0/P+LgP/WMy5ynD2W2i D0Me0ZIL0YVmbX7Uw6h1fLmqV01K10Kb9HtSm3JFp4m98DX0971YcVPER8qiao05h/KBK1UfcVBF9HB75dgU0 StnyThJug4Is69AN9X3KAoRjqlvotZGsTx3fAnBr0jjDiHfv0J6izX2YpNaTbNHmgRCpVZiYibUSVCGRpRN1 +DOE2CTFcWgCFCj1Awa6AnQ2r0H4XzrhB0o5zXf2s+h+xRBQMvtytwde0ZVRCcr6Yb9tHNwMv i6BMRGMUgMy0V9 bNkA15DoJbwGRUWpZV4fESWDuiosLnijpVV1hCx4nmQGVdQpuBi2FzU/eBa/cZCgpFk+pnPQo33DBzu3v0PEhq Dox00zInTf5IEm1BA1xVR54Yn78Qeqfvf1KRIZCeeTxEMnWdnYPQvKJLsTdcpoEgQFZGVKQ/EcVN+sRpqTnqZz bTLY3HDjvJsbWJFsZ3BgfXpZAz6Ws1aZg9rej2cHsfkYADHNCbIg51agHany0opQamRcxL8XcvFRfLkMzr/ DVj HDJ5NFdqwTPx1sfEH1IyJr9yM0RiX20epSW4z2W01PwmuV/1LpaSdHwnv4dd4oZCML7nX1mtrd0TjppKVAVZVI ObjdCsRX02rw+dxxy/4Xln63QBjV1GHKXfxVaZZ2MIS3nNfhRSPQnf56BBFpugwCNCg7nwhgNwDHE516x/BGx7 RQQCg2GzHkVWAZ7EHetuzXgDR6Ape6BnhZCDGC1QLdEV8sBU8S/9K9mbaZj0tH5FwqDhxd98d+tL108DYfCam2 IhRSUEiDR8VMU1q0rNUcBJWiY3x4LSDXMgS9yNLdE02Ja/1ByL+isqEjXJKYg4HPrgeioCa3I6dAIPY67/icIM w03CTS8z/qwgkd1cRj6PZ0eDnFD+kivvDbecJbHs2DCC9xUTFr5u0cPpj+1Wn0fWYvr1DxCbt1LCcVT6qy+jMu S03io430P+cc1XuyLFFU1zsM09Et+NemzZGfg5MK3QifqA9RcTo9+T4+oIi84TX+DgjpWeqrL1tJf7CIH23zQe 8VX1CZdgs4E9Xj+HkobDH144Vehf2yZtkS9zebc8H2EEhvfEbQz/+CarzsraJwIILfb32/2IPYULFkZBHjEMqF RbsIVk5k18cruW70WrXQ+zvKUiuyjvnxj+5ykYg71ZprwRZAGw1V0YYjQ108Pf+xrRsn+xtLkfBGEX997Pq/nyK 6PYXGk0jIqr76KLKHRcQrC0rX9fwSWqaQSTRQLJcZLs1rwWMXk0ht4H1zFvJ0AnmadYUA+YIWz3SuIN055HyX1 tLZZTVf2i8fIy5q3o0fqDAjgWugiZ01z0Ms0KPCb1bmmP+M/SojLshUq2yhs755AE+sSr1N01qTdREQed5cHA q0ox05Q5Qe1+UdQdBVmBbdgg6Ig/CdNNu5Jf6Yztor/4AjQAJK0rtzBhn8ni83/ds0F0hF+iRmYNWHeKmoU+S KDDRVkPqYEMSKDh26kKGyhvASbDtjvV3mUwirQq8sFmdKdK5bCwMpa/YewlCQ1xkFT6K6xQJ1Zkz1JU1QMDcP K ShCJarpkn3Htn3Q9La+iwWFngfiBVXXcopztTf3SRrsZuStHmtjZ8JuI5n2Xxbdvs6RDfodRhxTMWAXs0Fq/aB 28ZTfN5yaqJa2NdA6yk1V80KhfqRDWDvN0shqcNcVIaUPEgasRWSxv1VjkowehM9YjNhQ4TwluEhgSQw10Evlf /8e65aQuJA8jHGrAF2TL43W9N0YkqoepXs+8Ect6F7bbn16pwANhVUynq20ff1uyWLEHNDrnwoXugPYFBRxcsf kxxDeoCz7RoG8UxaATjxCj6wWkYINTIITeZCGX/k0Cx393BpoSTVWaFbQKCTY60GnEInvvsyq/f4mnzhIb4pkj juzWF6yyqz/5MFHqrthX+Zeh+rjouPr/japA7IZwcqz3b6q1fu7nYevCes2dGT8wVY4Mc91sLf6bh/b+nQ4aQl PIthR8mUGyBa8Zt1x5v8cM9KBih0QtnvE/tPVnbrIq1n3z xvNUT4WHQdcSJAhxusXQT+7iFiCaLi6lkrF6o1Hu SL5CM5iP/VOQ0hZN+5+m/SCo9c4vhHBw8pN1btguSCXEAsiRzb/fUZB2f/C6iVQ8w4pihSm/f9BfWSWM1FVW/7 URPdMkbbfmgvY2p+UmgDP+DaFxTGQXA2uaeNFT+c1aVkz003oo1NkkuxRIq7tc/AXqgMscNM0ZEtibzILWvZmR HZtcLf5/YG2mkntx3sooGuW57kAiUsRsv7XoJjGChLBvTLBnUcan7RSX4bvwVRK1t9cHOSFSU8BnHYMF38F/D PG8Rsujp914j8d3/SuEi1V+Yc1Zbifw3qq9M7sKfm2qLJmA1J0MTNPf+IHwaR5VNn3qRopQRXqErr1/ExuCp6g 2IXcDjTBFteGvigi7+f77U4GJPQvUrtxEZpQL+10DYDBvmbMRRto0Q9d2+1LRBzst7IThdjM8jHICEcAGNHT JS86ur5yAtVHNCvHZKVHfy1P198APc31oXMctxV2Sdc0Ugv7HRqtkjjfhbxM7205JL2bbkCMMYX7WEZ+3zKDLK 53KRQ1GEI8urwcrYzsav86pASE9TjpLWxjY2bQrnUVXJuMvVG5sUfDi1Tz1vdgU/CsnPdNY2E25AEx0ALia6F UrF11ST91BJsm1D6uYgnys2BoY08BRLg3/pjLyaID/KMxnYoR0UYLjufkptDnwe30S4chLHewtp3Mz/AZHMrQo

VpXb7RwMY68zJTnnusUonkBggCn816QGJc1N0+kNgiX4mkqv6fDLPS0cI9cibaeF7kjy0xK9GUPreb7aM/DMF
MWWKpe1NziyjYrdiWjvIq0LCy0z9SeND2gx+5CsNnDZrfDmaD3p6iDRvEt0ymWfNWKoxGnLQFv4W3N1/MNpKVm
PzpjvrgghtYTbgBZJK/OwyJpakcodRWt5AERC+Q2S6I7rfTYzW0tv/9h1lQosf0qxK0rBbH1Lwn89jEptSQmM
8QWpBvDOUAAgoCvF6SP9IFE9pXcA2yUs5rF1v98/Uw0Hmsi9/TCVHMC90kQbUQkfRmgDI7L113H+1aCiMIgEgk
y75m1IMitFQHibIH2A0IF3Sc5qc1z0k8Ym3j9QUYRgTDxi7nQSksJPv4N0J1YSSxQ7S0A57RIHg0q8zitam5h
zavH8UawU5PI1Lpm3H878LK11Scx0GU9IB3ApJS1ZxZ0j1fS0dnfZDJ+ssdZEtTTH8YyENDyRqfcldk32WAPb
W+1+86KduzdG7aJvL98/oqMiUDowKWgcbVDjv23M56ajfZgRPmRFpk3DfdvfHz41DcMmfeQ8kZNkCL3npsBxp0
DqafdB8+KpxerPbN+K0BZoi8j1RsJuU/G276FF6YMBrFCvsc02oqXerMcxXJrb3Uo1v0BA+IUakIDcEj0wmsY
xAZp3tnz9bnnJWlMAoikDS9HyEf210+nzQZfasLbnzQhf+Uck1KfoIEbm4EHKwG1rPOKKQ1Ic0dIYwF2XNeRN6
v72/b8W0z9a8dfmfN6fSWeYBMQyneUvXX5GJZ/+DpHA76z/+C37UqrExw1TyfJmkNi9lH4MQmHHilHU5UQPcWg
+RGDjxFmu6SOQUBQ4uJP+QdLkc/vDsVKHw2KksA0tDndZM2F1/0fUj+GCM4testCEU0BVf8J1P19z7Y9tdEGw5
RfCDSMhDC7IMjcSe37xHNS6hP4aJN5aBmPlqgHP1voYIyz+X00DixnNSxTWCqXmxLkWffMbDovzUztaJ201F1p
wgD4qY+ppEza8rBsi3LqgEA0A7rTB8d8XzzE22A6ohyZaqCgPEaxBpGwOsa/KDDJCBEGAyEIQe1oAQMeXMT1d
Dd3FAZjZ1Eaq8mWYIkZd4NeWauju5gAzVBM18djiJ1Fzpc2V13FyQcStN8I90Dp6GpspsAWLay37ZnikxjRwp
Co3cpkDpoynrv145b9I5U0zBj1Cw1v7dMpBt+eyhJnGyD0ROXs3UX9rVXRUEkeymUQZp1f1muHGf4jzK1ImN5S
eLNEwHEdmiQBGVtg4C0Qv5Rr6LsrzK0wXsPWg1DUAEKOuV+9nNRSqRVaH8rfZepkh4kkGfKMNEHFo92FrDaBU/
z9dPmC1gIeTaekLwdMjbHQqfo1AR6dwz/JM8h+Dhnca7k1BkXfe1s6r7JCvAok5bYlu2Nm0yfiktVe1GyIr+9q
oZ5MNknyocVj4gzhSY+XSK63vvehB7n4C1ROXtE9Z8dj1tnxWb/HECCOHC8BDkZ3kzf+ZNReW1e1mSnRefkFw
/1CTR2fhD00q6S99eA9e6jTLvEx1v8p1vsRnRhF68IzXre4W2/Bnb6ppTNAunCigdmUJrnwn1tLU6WdvFhMb
xoDhRiBqNrqafjEvhUH5zCnbQV0ntxp/A99B+2WpEHmBbg8oyVKodV2CQmjE4e2aii2oybU4wx2SsdSZY5zwly
LXsNrigrizYYlp0UUr68Sa1GRw5bKSwzzKESAbNQ01224eDThcwE3jicJcrJ7EYLtt0DphVQffZUvCR7drepG
daLnLxP3txMbGUZT15sU4+RX01nyI8jeQDt6j0e4Tsjjnh0mySp/0tUc0Fj2ME4UsVsrfqWPnCLucy6X0pyzW
XsEedsnz+kt3znYtzPTzTaHXqPaJFxF2ewt9L38sxn7mXkfWgWV9K9dUKCCkD6TfrJuhp0pjesYG+F1ZhrPEp
BD2tConchM9XCyhXWQJALOJP1ASLVSRRhYZnwjlhnxI4PnKZ/ZyVpyCsC19JL6P3ekZLOLva1PzaVSkBNGXT3M
jb0ia9F0D/cConJEsWX3WHscD4pRJSr1Ynppzm1WPdn9kzxD10Kz6XX6T80qg0UAQERqSAMbGKpBPGlxddggt1S
AGH+/6iQ69wkLzv55sRG3evvJNu0iKWoJNAKTVwT6g+JwA/N/swcQkRmRUTzjh3H2r1Ys17rhgSpRo0F6Zw0/9
JcrIJ+DGRBnpXb8+3L1kQpcM8PiwXfq2IpDoGRuw0XKwAqgyV479b3e/8TzU6xQCQ0YryzKT1Z5G6hpPHkei3
0Y1DQLp8uZce/3KUXUsaNSVwAk85y3Sje58G/QKfm1fIj0ryKpF9tNaIyWmASdM9SAkeADS4JXaHTAf1mT4j
Err31TXCIVIN/ucyQJVy02pUQQQh2SsGIYAYtG0wulx9qMHURd+nBWez+n1QnU4nxQ8MfqzjP/pEgmUo8IWY
KJk5CaZkYr/HЛИHe1WYLMKD5b6rV0bCauRz/XgubW9/qDL3bKsC960yPDqIw9CrGbxjeUGZck1zQqCMMz/YH
bHCKK9rTPSzmaBss1aR2d/MH7LVEuMySSEajr6r10KqsmkhZxSm90JBIsoawiNhJmqWgBpVYVwYX85g4IQU5X2
aBB7oijefEvJbuZIj67c6LBZZand2edfGjs2mhzvignDDjLgajrRJzigs1P9chqrz5xbnHbCRz+kn+M8eCfJ9m
SehnYmvQ0E3Xy69ifAAiVYNSU31RdWChitB3J5UkIPyGY9yNfwJuj3JmoR75n/+9T2NjyV+BVKSknndUNCiGc1
wL/rsDDBQrSpIzZfVUvet6xYAcTyEMQGewuDNZauqXv5igwo5NdGrTVns3RVG5VLiFCVgAKqETW80B6J1IMIn
x+FGRvdIwH7FktFRMQR7wzkQEmwXWM7Zk+ty8BsRbw/pNr9Eg43q9rcSy/qsbpReYEkvczgaExwWg70DK1Yw0x
DavfaXMwP0Cb0Z2qjk5zs46ijck5h+IAq+2nFj0qMqgWt5+0SpjVLZ7Ec0cUzL/Q0Bhkdc4fCmUa5ekc1I9
fkjH961j6AicNAK9K6JhfJlmVjC4Q/oFcV2xx3vafIT//1R14jqh/hMcCUke106C2D0m8ss0pRIS3o05i0eM0q
VZen+xamVX0jgC63EA75wn7UutB4bRRZ9ATZ8kP7KVMCCnxUdyPFrgs0MxEJchQ5oPjn1557DYSA+W+otMvMBy
t4BnoTEt4T09NLEBmnui97isofkz/1Qip7aJxzTdszSiVAMGcw5YoNqxRPf1m8Y3PUEvSPogysmMSIag//NK0o
LwivhmacKbfH6uhvL720F8rn08ed15pW1DRkfNj0cSwC//pYg/6C0bIznrsjs/ihtMzXN0ggbGP1HQwahzauuWs
LvhwmJ668H0yUnJalyaMfY/mInaoDGFzjYn3CF1C+qgE+YML/r1BOGxpWnrfQfJopW0uvK411t/fdrNu5wYcqQ
adSZNPLVmUDyuDcF+x25L9DouBBzB0kaGh0sY6A/6pRF+kuAbFIPmSV3ZcJjKZu+enFdGiZ/koLz+1xU84b
pyhupz/MiQiMeyHeVmjcVrS0VvBdRMv9gvEAUNc1xRQ1WM4kwus2U0KqX9L4yFvD0kpy0p28iWVxxip4offlqr
UQWgPE/B1gNX7pD0zwTh/upAo/jUbUA6fPQygeo908m+LeddoCfcEt04jCTIP+nq/ZvuVd3zf93BYVar2J5qR1
uNyTPfa6dX56YHRT5I7Ts03jJG8+r1LRxFvVoJ1fkI5LtgpzEy7e9Z6h6Gvizcb4ebqJppht+39fj6R13VwzC
3wFS/Hmha6x8J91onuWV9LIWXDu8YX+cY8s+ZwRDriYGMvW7D154AkeXCD2EgNyacZ2xyLaaumUbCOI2JBMaqb
Amcoj/n7U26A4azt6DMax2BLzv09MZCCRnoNtSEhevGXzo7P1CyrW+90+EN/B6tIYMcVxd3ATJqnYu0afqMYw
pTygjsEt7eGvutPmN1n5r09itEU2aUbLwvD6LhbmqJpsZDhoNV5nAcK7041j4VR3Ej06Z2hGpo+R8EKF0uXuyy
Zf9jWdItfthBdGFq1EXaWHsKWjEBhN8fm+Y16it+l/sVTS+YckdvZfyHNbPrpwY2Rt2bME+oQIzo9zuJ2LCw
BFyBxpxLLMjDGLf0yNbtT6LeuCuFmQfyC01+9HxdQ2TVKjr+2jAfLIEWHPef/6F+TwYDXHc5hSLQZJ4Zk7yZC
ARzCOnsvqVokU7zq/X3VagpUITSz/10AVpZP3BaRkjCj1weBjhuyTrYco0wLyJ3qFAitwfZDyRG2Ma0fCcZ+0

1SRv8KY78j1zdIgMGgKSUudh70S19ICuPeWe7cLTf30IHw391q4sjbrxNoZkCmqPee9be1D7LyT/ZPlQxeJJDP
Qxfqvcdmm+vduNkryjyk9Ngzt2Wv7GbVxQ7NbJKfbPB086JMcI64DMD1EUQr+CBKopZNVYTTcNQxfBqUIwdgoI
j6g5F9xV2JzVaWNRIsavxSnnMfz2C65Ln2W5HRppkh6sqq1xL4LguyQxqSTI1TRzI57FuZde43EvJ4Zrl1hwuq
XGwzZ1vee/+84QGINPU5LFWBDQyPyQkB+UdkBh+5kHRLUertHrNDYRV6zCmsU1M30379BcRPA11okLuzLMicVl
7EGkjDw6+aUczVkQASKOGWoQEM1u1i8NIi5Nak8z7FwRMQRsTQsopcDFARHwC3uwP70PvQZ803rP6JX6HVE2Nf
Cc7oge9P/G4d1G0EbGEctA+zcDf7T3Tjoxmbwuh+u25rGkCH+F2qZTXLKxYYbv9/atYv8NiGHw+JHs7pAG3wQZ
+pHbulkSGAiVcA7UgSvvi31ftnLQzvCZt45Y4SWHJH1EgSVVX1sijaHW8+05Zu0f19XTTRRH+7AJItwYgZ1i01
/2SaSRTqh8cdC6a77Wy+he7+7i3h5NkD3CXHrV+awPqRw7tkMba/p4YHl1Rsa09pUAhdRsVacvAleZSox9q06i
z09PH95XT1bQdRBw3RhrsP/sVWQA7s/iXG13WXHQPiEFk4+4As3YMyPnqM4fNWCHngPwKcfGg/w8Tx5KtXuv6s
j3RAeVbhzR657iK1JbG4qgAJETIgNGqXDQ7auyaBP4Q1XUWzyod9D5Mz+xMq9o5Tgp4zpSiYNG13G/mC/qwVLM
sp3v0ERh44tRhjkR9oJXa6NHn4G1TQ9P3MF6gpEFXomILXD1MNrb44qg5UGJM26tq44peBWELCxGAD1j5m1XdI
B4sb15x5RPiHHbrRkwwy6VfSybQR47CJgduZiMUqIfWUKOAiIT536FnjDwvgU2rzEoDiFnrluX88trV5D4Na90
S7cN75baGwSgmvIYd252q+R2BFFBewz74NnVmBPaadwB7zh6/qw7+qmN+Mzb/oayEsxuz/ZGExL0+whavX/v47
kj7qFypcCd09TGZBCwP7d0KqrtHwp601y6pY/XAo0y7xUrxcRbm3EuVmASu0/mEJadtK9W2g5McbA+f3mjsRHu
XbkgWKnApCcxXhvu/30ZoNneM3+e9aXqqzxV3AIzZJf8EHAKsepVHFc1gHLgvVTf00aTOXHik9S9FkyaBt6x0
yIsoHCHdyrh+KpvdT6/0pSpV8PGh7GUu/L503T8e//msyMVKFYvpz4phEmS8CRUdQCZrifyktNLhP1A/octiR1
6RoJH9UKMC71aCLu8BwR3uov6IaXKzJrlhd1K11WE1/FDZiSed0tTw8V6CLnFeKqiv7cGwd5ctkvkwV5K47K1w
3JtwGVYjRuplqpzWgtWH8oWHvnuj0LeVsblgh6vCJU39f9jld74K1LIBCaYI2SWYamKHmi1eAVLRJssTq2Yhd+
0yB321WJ1ZN52nKSn2HfZw0+HAgXsOfnGouo9FRvRqkrKcGreUrrKKBjDB1aER0bggXJE4EF3uL5QTN6fjF7yL
enadKq5Lew+z0ubVraWY0yhjkbHR6sUjusHcb1DoWIWR3jITwlumr/1AG4ltf7euPy1shbQhDdtHkdRmAsSsD6
o00RMMh4I9QhSd6PH/T0m6XqpRMzbw5UXm6ltEmbG8hrZPv6HP+/NKORv77tBAELGP33f4H3dvmLYM0mDMPqGS
rvYa5EuKV1Z/JGQ6InmPHo9D6p4xRX2xykxiy6PwriMsg0u1pm4iU1IC+Bt9HkvJE4o4nkAcnim1RpDBTGSGr
1m0VLtp+8Cz9PyAVK3A1E9t44AQneLFpzjBF0hzxkImIqYRgpU7aeB4Rwp/cxyybTwsc1tGE3G2HzPhCcDmEa
rrQmoyC3wElFyKd3KYy3bxYKEAehvJU2VQ/TnHvJG9pNh8uNm4Y68vcSbR2Ga+n0NjFGVRC44wMzPtzJ09+dR
9aDRm4H7XMoTSqBGfCm1pvZ1c+TuVYKNBkhHKnzFSsLzkoFzseJE/ajqiUoewp7Wo07vR1kLiN6YVvMwW5Ig
Ntukr1Itx33VIc+x8g8krQ1KXLfi8ng6WRp9w666Ad1o7W09h64pqIIcxzWQsjAxIyME8ECV3bmtdRPxjpdYkE
cvJpa03xW319sVM3M4zRafrI5u6oFnVw5kZ951SKy+ut1t42+fJ79LLjvCARwccUArcom04JcwTvG7d3R8zNr8
PJhYIH9Z1a1jFnGYhb4YSc9be371yfYGrYVfzJ5vd2QU1U152iD/deXwdCIIfuJX37KUS43WWpd4Q+IuYN1MjZ
u11S/iaQrv3KUTU0tbKYuDt6P1nK3u5DL+U3ACdQ/c+s+9WA5oKDfyK2vRVIVb60afrXCPWDV9MwCiTo80T2cT
DUS9R+mWnHxF016zC9rqYJ6FdEFh/FpH0gYARhDaVqEA0gDWDB0996JDN96Moh8ujNqkaiwmpQ5znrP1DnBID2
T4oglkigFJcRx2vDa9fmRwYyoPH1y9uqdpDzv/w9KoMb2QvezoRJbksBnWV+JknQVjEJxZ1JV6tDe8fUG+G9M
FnB0EIMy8VpgIJTpIzQpLOEJV5bV/OSR4wYLQ04ULh1zba8Q+oAlmCBs0r/HUiD8r0a0kGX40C+3Mmjbdle+T6
J1z2p5mAQdcykrMzsrd0zaeuX0hV9Z1tbdfEkVpLWYamtTPBt6RID/+Jtn1PRfwWR5ab8Qht2PqzNoHmykP
1X9Bb0x8LrqJ95j2wDXA+3j/IyeSrrGDi+dFGtpHhGKZXfcLvuuvApxI9tg0oRtByYNNNhJpoeXkzrxTcNNcrx
mcK+0sQfqMg35IUuWfoSkJyUf0AJI0qdK2mAC+iPCjVdoxKQRikXUpVGdGZzRo/qd/PMWrfKaQEdNA+h8000In
GqRo5889+87sWY0R3aRTxKR21rIpL79ILEqrTbMoP0vv/NzVk0rEVKqqETjVTwgXn1fqUb3spzye3YqLCBh2Vn
nQTiH7uKW3428A2DSp0pCYFYcp4/3fCYB4BMOoVNZDQYcJcygjRjzoo4hTDNidzaZrnHDLknPhnSK8vuLPOUh
wnEFhx3vwJQCIa2rzA880SIAZHWuBChicR7Y/tE4nq0za50LQqmji0p7EUE/+oAEIeGv2YIR2bf0t8scV15u2
Q6I/Me0xcbCIH0oxgEg86Hchohx88HiBTfsj5M2rK+AYSTVUQe5PYPZqBsQUKUQzx1i3/gjMGkjqv4k5JmHqQe
Z0Pp7iD7/KNQV1tP9tWs/mxXZefqqzKpbkfvQ+k172ygBX51eJnQH+yrWho4PNi1eTxJGqvoqzePZoj+vhad
t080WfGE65g4ai97pRyoqbujQNAAS/mkRWGrZ0Pf9Nsbs1/TG7DQfcfNytF6vxk/LLzaakYNYOs+fwJphYkhfg
xdq2eSWLIXPoa1VHAtwn7fg8EJ8y70vjKkGy3TixIJz77sQqARXK9BdrrhZfj7iSbm6yh5o+sLd+4FB1EqRhm4
x11T5e5qMbtU+0WRetpRTDIIhUoTP9I1sUSAmbHxfVuX7cnu2E2HHRetJJa4Jqz1A9Z5G88ZMpVy00F0B8I+QzBf
fGGKbYkYPV2xXcE0m3vDgjIaea7210bXxOKxRZea0GM50xJKuHGH4brWZ/KE1zTJL1t0BUTb8MIkpCHxNu3V4P
BkXZ+xyw8C0QWNn+npFX2t3VifEmxNWn5pS3j1SP1P0tba1NSsxVHyQqkAI2QxCwQcj0WzsIGjPy2dXtj6koU
hrMXBKfT3ffzVcfXW+INIQImevuVZoq55n2X6sPH/z8UIb6UiJEWnnwayaKFslrpsGdWyaG8lajyQ602L+E7c
oMIr4MyPfmAjHMPK13i3aw74Y1EV+m9cQINVM8dRSpKBNqsTn6VYH/HbHkG3i1BnihT320w1HrurWHfeLj1e6H
jR9MKvSd7QULstAhWTsekxc/4zjXgFeSm08UcfSDucwP3vNIeYjELkr0R17A7k0k5md+e5zPX1kZMuwrWx0jk
Rqfw6CqSi5i7ftaDHHd1aX1PdVtS4F5ntP4Pz8SqhFeczTIZXnBLsgyBP8MX0xnomb0B5HWFZeEAI6PiMq9uCd
giYXPx1HsUWgIFZYjZ2Ic8j5n3eJCT1gSDzu7VNs+9I4aTw3gBbgML+j6s30767JWowxPbfJMeV5YzzvAlrPo
hgu+Asyx1Atyn18tKuzeTU0/ACTXAT0us09fy6jEcf3r9m0L0SXvAiCxTT76CaUhk0nkGxx8Xw3IfwCmsXn9Fv

psS6S1BGrfXA550ory+x57+wNqYi/AStwswfYh21yVsVWUgP2ZyufzUtT1PCjG3oU70pKmoVP3c6MQgtEK3nJm9TSTP9pyjQ8IeJE0m95jt5yZf1LbzNhuJN221rC9k2pnI+MzMquRlwV6J0T+6gtX+wT/KWGM8Z2Wv+80Fqd0XKs6aGG7GnevGsau36CQTmo9rDqYDGbIaeOLcTol+V+mjxx1JEQBHDf9V9aIIoeJ74qobSVkbZGBvhFS0tnF0txGXckCcwKnxWkzddDbQUUR58095erHaQpeF1MbUmwxjg8t/7o8JVSG47+vXD8Cp7+0yG0uZ0300zd+A0euWeX+SF6k/2M8ru/yyaecAMj0xDUTEcvyim4RQerL9C30f1cZB/3Dpj0oVA1F9NG4LJR71K75s4fqmG8PwyTz7PZB5VQ5Exeka0vPNCsovjLvB6b0KeG5FURoE3ssI+1m+8RCxfG/1LZX53WSjjRYXLJkcrswbXNkbnBC0tW9IquqRh1I7BsVhrlsldwgBhXvAQ/o1gzhWSgku7BTfXk1G2gi8T1ru1iA5HyG1B8naBmkYkkBwkKTWx3G9AGDtZjK8jPi1Y+uEoIF9DP1gbF9yJ9W3TkdtptG/ghPoy/4DexlrkHfGuaYx/hyoJBOP2FTK1Zc4Daf1Tmd4B3MjkZJKIeXAVKAbfE3+Stf05ftKCZfnmItW1Z8gXJ4bP3B0H0icBolB2Cgj7+4J2yeIkTK4kHsj4jNCftKRmUDbfioHcfaEuXJGpRz0t0ya+YKXWq5W2jnpkyvS+7XpTxyRATbfD1b/0/y0gGRwBcYDQqnR3TAK4ycCLEJ9zijuxc0W+9itC9WaiJZoUt4WFcp0f/qkwyrMoAFRtmCx8haJiQHuxLQOZIP+t/oJpD1UTxxd3UfYxj1n3HWWXGKwEwqkeZEXNepMKL/y4/kLWYBWwHFFJ0wR6+4qgYpw+IsKNwpwC9wlrtfc6s1MVYpDssU0hw0hLQGSWXjwu2WwDdx6h0m/KvKRIPt318reEgZWzG850WYLTfD6RFCAec+UxI9078TzLxbEmm0Mx1KTOPHRXWsYfQ1N88PA2i4AJ3ECMbniQpN9fUFQ5On//ewj/Y/Bumnv02wpieFZHDP0Pd+w8T1TBjXvXYcXXooVlg+aimqxDdz/bJHdeDF9Y/SM88Lj0MC+eTLaHorMh3CsE8G61Fm55NANfifc6SXb1x1mTfp55dm74nSYAZDs/FZ13RQkiuWRHn20gjQJwDS1EMA1QGWidI20PkjbtEN2TmGX+4NNTJnxZ43pQshpxatHvRib1Mn/9ecso0D8YTgz16mVwu7WBx400RtK0XuIJIMLrGsm4YdkGQjtv4Sw5lw0hC1aSVr1vdT6915GMzb7dzSuhtyEU9Si1/W9Gw9Kf8g5NGzF1N8vidJXc8fwnw1oJrpMVvBx0pDbE3P aMo6VduHKKXi1JuWVAqllokCqk1PAy1QITdsNh3B++agXc6DFDAT6BthXnN9IjI5y5pmDXmWpPA1PV3mnG1FKn BXy7maWYy+cxRrvqXSIFifZK86SpVwP7KaKU0iD44g8B2tISIjvoZPoKpKFON/81ECWqwJOKFnrsI9sGQ7s0G Atqe5dUsD+ja8vEa8EaX1PunuLWAh+1IFutCIZYIeXd4aDYbQmk1YyCBDQrZDLwonZC4DA7vYHsat0q5tHSEf 8YpPA6s7PcX4GQNrxY4ZHVhpYbSTAzo8aG1HTHxCwDm5QYTH8KxQZmJkNVZCpXcrkpcmbIDpv9x73C1h4LqrM2 HB86Z6WhrE59Qxa21A74CM9EV41oeDzUjaVhF12u5+im0jkdT7L5KGhsrBTH30dhE8njt+8Bw32/Infq0ITKsF wFgP0Rh/0Sqrea+GGDi0qqQhLsd5mCb0U7pIB1bIVU7Pufs7I0QpWP5VYf+k0cX42rGgSD0oPW1k+1DU3mdh19 VYajtcW0oCPU0LoxytPOHL6K/m1X5NZBqXW7iK46+XebiEYp5IeRXrdT+dPVEcsNjt+44KW+YYyifQHIQ/zE hDi3Z+2ty9Yd16BEkYW3cDCmMLMKMXkygLG62mkbdSVh5bFrgz7QKRUQRAjtaayJcSf1d164V1KAxteNPD/v z3cIk1LMmLrGBHha+e0eX5GdBeqjp1EF0EpZ6XPVWjwDPaUWZ8tjG3m10cij5sGxb0KnX4w21MjtGLypbKwvMM av90xAi11pbVaWKGf0WnCfpn0NrGW3JMG1YpmYkuw/5cDsdyFoRa21Z2ksz2jrk1aBcyLZFWzjsjEU5sIorbL aI2HuJQ1Kng1DrDG6HPDVkekJHdYn0RRGunu0usYGrh1q8o14D0rpXM3ee+goUeJxRv0TEwohircp4SVA3txZl M8RiYYk1Mt6YddcZhL3VNipguc1VJWz51KFyyYw1bms5jnXfccZ7LN0E5+4hJpLHPk+Uq4td0fX8j6duuNn4s BFzZP1dQUoN5k1Knu4nozeCmcrg3qkv2RjbSnSPSPBpn8SJWJ11RCacaP9Zzs3WFtz7963DP6Hwjkoq3pkvj f/zv8fU7edoSD8sNXZnq9XFho+w7gu4asinukTzMLaEuuJo4kR6o6cwlyVvZR9mJj0CVV5S6g001n0UY9osANh vHdAqEnfFtc/I/Y1ZwZ4P2wEmRE12G3nVxf6GaWuGibmhtnsixCubWTP+n4kL9pQoMmxP5RoBuiZ//HwJsbZx A3BXc0IzIpzhcsD2BvNVLx3Z0mAXV/8iS2t0ISjmJXWQogU3KNrj8rcmyFmf+oqT7SezcuPSeSfgYprsIe8rv DQ3M2kEodSSb0LQvcPbaufSu6QzGIHRB1cQeo2ZwVZXYM1IwnUZ5G8AzB/Gv07+WEYJUzQrVKs0QilL6ykn13f +5EWs81qEQcWXE0rTmnyQ/uHMNoS+4N55XwrBaRir0cvaCPQk1G9o9phkrPn2k0Eg9U5009sKAJAeWSC9y9am h4Wb40J/22BQDkVA5nDxF/Sgoi0Q8e8SzC+Q+5zeM5VKm6qfqF0KbKZOUfDqdA32Hqr2x8gJoUfh4n4o1462NE BQ5itLj0i6+5gIBhHiz1dNY76c3Et609EP8i9Euu2i66Elbw3oUlniAuRx6DHr0JLijWUQMW7+nomArQje8/X/ nTp1D9+k3Njwz02EYnoY6jqwskb3RgN/3obeu0finPHS9/StBpcC3EV3vNpfr306TQ3AniEooIGes/dtrkLfAu 7NiWLEz2UmfnxZGkZ7TCrx2Mx7X0wMx3uRB4EbjQKZhpcJx9nCz91tFjQkEJ56fWD8/1g4JWZHBw1AjROT6VKi woLH2sRk76iSRI4Rr0m0KX0yDSwCEPvmszRNgeaTwPqMGTlhf1kY36z4dJF/AhCo7B6tgadeot0p/8CcbVe+ JH9/CliwJdfGmrF/xUhYFFG4hyGjDIXlngo19aABiQcgcPS5FStqpT2kr1qrVJouPXE9I9gkmqqEcjbkRo hZLnmQCZPanu1YDubjeBqModqNcFx8egbhS70yyntqAw31bnHE1F1tBr/sAS7QznF2yhmD7PV0pd8r3nIToDC1 +fmczjxZtGc+ATQ5bq7njhG9p6PL9gIMJ1ulw+Ewgv0UQzxIgtxCLjJodEYteQMEHxbCYZy7bgVxW+6ZmDt/4mm DxsDwWD+KZi0yNaaW4+1ro0AQuq3o35cY436QifJpVu3SN4miyHVftdGrXFmpxmDiAW6uLG+0Pi0JaIYLCKL+b Uas5ARCGb3rg8ftTG2ddcJWE7mfstml/AZerz5KY11qXKB5v5ARINIKiXAMiw/G9Cxf3rie1FwJQjHuLpFr991 zNwIpP4NCPKVW4jCZ5TcRr6A00AMA51xrLboL1q6W/xpo0lq8bGEPOzNt5bkDZDy1fT4CGkr+TpWEb1x6M2JDJ EjvSLbaIaduLZDeRw+/1T0s91HRCdwh2a0a/7G/wql+7JYiTKF822JtdUYVI9kz/2ya4cPMDocat1A6eM1zo3T wF2wfMIMr5F3RsIj+CrEZPyzAmq9EsiE3uAMk1gXTMa0LidhIF/Sb6qYagpKudYjge0hX+8Qhp47d1g5unUxOB /un2F+nUKRwpIY4CiZ/hZD+IYule8SzvY/0wNRSqHQxOKYVXQpFs/wX0bR33jfSMa90+XDIeef3WPejWqL+pP gE4dzbS9FmHE230DUBzHTU2+iJSEyqNuxrDGBmf17UYoIY4ZcM+NqK7a9Q/kMoRUf1fQ8Ej79+7XJdhJixkUf 4mExbSprGJ/OuYzdnDs5aMShSc/rvKJXP2FXrRJzSVePTw/XKbLS51947ZoSqSyY6Fg+EbPQpHGKc1sUX0dDcF

SR9fnb0mF1v8ozeeIqxhqSa3zHUVQER3t8JCI3H1QHY5Ae09bT/AL0yhBcVaVAh603uKRpyR7TEiD9FtrKYSSN
c6spqwI5t9KmlmGVl+185vG0bB50fYQNEz5T0zt6A3vfcAzPtmn5ZkTaeLHI2sdINm5y1arec2a5R8vfQPSFSE
itjrjrEjyDLKacepCh8ukaPuhEI9+Rtmilin4LGkb0PYwDMore1AcrlzWSAakdg2j0uHnoihD04EXUXIAgGUdI
3DxW0vNGfwN1Z+pFc4b4GDrwJ+lHkgIJodDqquaJFH8MkyY/K4WTmUztc+vkMWWybn4tHiAnW457iD/CX5LU1D
VQI/nn4MK1hMeHF1BqR+Ifw0qD1b0t77s7t0AX/AOJy3qCpiWK1y9+D4GvW41JMPaH38ha7ZrKrWtz0Ym5TA1s
690mzkD/L1XvUgGKtdhFA1BRqxLKs3Uz3XR6KfcuQjtRSBdf4/xUVH2+q21z2aAjbd0pgfl+rsvDaT71DR/Ez9
0ikYkPYq0seC0Rr4cZtDxyes1CjvjY/M3g7dMOTd9utdIIF4b8yWBpy6hG46w5JaZhYNl+yJ7cXiKYb55JdPI6
IFAJeR+h14jnEtEB1YAkJHQ2iUymA11TsQzjafWbB09Mb4TzANE1KkMIcCN7D2/I2Y140G5YxeBrPV5z3h0we4
/wz19JoURPAf2XEQdpay3xiVKz3mOsMemHWnWkp0xC1svLXvbmoa2FYOF1Gt8YnCQip8cjnpnprE0R/Z1ML0RczR
2/ZtXub25cGLrnkg4YzueJNOKGyW5md8YVL8fYRruN/TctxScdr7kbT6MA1Sg4Z2CAEN0ERv8iilQwfHPE70kR
UwteT7HNNXsxhmuin+Y41Lp7h57mVpIPabrqUfMUusoifZG00IhNUQbYzjWsowApOnJ1hAvji2ent2+6Z91wGY
rZPG1MpuK06KNibSa+q92nwaJaqtib0gBcTyDevbYhIkZM9Jh/aCtsaxjGctbYjsJKWlqd9dhpjcxdcGZ1ZLQp
kASvuZCcXpNsdp2vWaZpLV1Ix69hiyH99mFW4BQAH56tKLGizhrP2objyUWEg7LzuBhU78pEHufFerpC06WePc
7yWJmgF/BWmMNrTsaqXBAdqgHdA5/uq2ZC86yy0S0gfPX0sf3q13vSS5AyUrAxuAtQeZLQMs9od80XIXWXa6HC
S1EBTYry45Nuc1A8MZe6A0RBVK1JA1RASf0GPYiTXs6BXiGIEHMyKf1B1pThGDKyDCQ3Bw5rqWejwA0SqhJeU
zqSK89uUI9FSwp1ShDAEQ2Wrunk50NPuWVERXzW6bmu/iHUVcJs6KMm11/EXfyr3elej6+jRzEAoHDPcJaenF
cZEDFb4x5DNtesi1UI4twUuLfeYOIeUWlpoMT4nAfPVGXeWd5GXgMXGGtqH0tGNWG83obHRa6ju+ySlkxg8YNd
N6GTBDXsDWwRxf0VVY+N1E8+bxFj1UUjEXXeaFudQ0jNnt+7F0tEMTSmJssVks/IBamieaNufZMyuxZdnyk307
MtQLEns5m9VixNnLHbcXRrsaKuJYRH03R5nbF/gFNnMTx11HD+ywh5jaoBGoJsXeGGOU2+0citosig8kVtCr6U
kfeDb8zuk4/lirb5s8he635J1oKSMqM2qI1QIdib27iqeMybCsh+5UuJG8tE51K0JU2IGM0NAZ7UPYeenrN1BY
OnvnTPgbxWcuLCNxAhaSFFJ/BTSuGcovkL1ow15J4qtfTG3rkNF/TPFhYHxN3m2qJcBB10yMV9e1150tH5Cz4S
2w3837bPjUfPw7YpYsmhUIHqDDVCtyT1zvJyzoYvvK2F10W5JxZitqDCweANHIm6k1wrWBLXg0SYde+SqcMN4
1Q4er3Jd8FXH+5kYqYTvx2wRUNeUGmaLb0LoNW/rRWxvLrDbKP2aFB1LLRSRSLHcQX7Ccmvm5zM1JPgZiDqbtC
y1hwmDEH6MZ3CEag6C+aXiDdgo40TYFRQ4R5h2pBBTjxYz0qY/JPYAkfdijhf9UNNbExbPZkhDVli6xrfdDTaN
4dpv9tnir1W92vds9PiB4c1PfUQMH8GMr0LvvWzr20qYQdtWQS057iTMfguTQYx59i1L8EIDYed1TVxHvXs190v
CQuKmP30kEMbdLShjoTB7ivk645K3GbNSKRGKhZ4S1c7JNjM/hTfsBe29VXyZQchQnms9J3+UG1TN6HROEr1/Q
82forLgePuhJXL9orimkeqC4J/wBojxx8sYXA3YFkwqy1ndpv9GcfwR/L8vWqV8geMrJ4f+DtfP3DNnSkPcmq
9r4eYEMftNb2/q9P6gbpdLPUYqAUqUk1PgayCHF4ZMQfD5wLfmhBp/hMxTmPpWWY/nQ88oIniIZMjrs8bTmsh
g2h7a3RIHTwLccIKVV/1k5Ejd1r2bCzE31d02QpcZt40W3dAWDJRagcZ6C260yoADpZ/zYdUFEQQ5WpDSNvE2Z
0SdHifWV4Hp2Y/Osy2G9457kHVxuK4k38zbZN4kdDeD/3QeGIMAh/5xfXNNFMYh4e1j+QtDiJw9buo/Z/LInhX
VaSWT47T1/VqrjDsWjImHcsSfhgRqSt06GG71tc001BeM4+Ti8Addhn6LxQ0S2u63BShRhgDwQ58J5ImPVY3gX
UHHS9irSMmenbIPvV9jeAfHfzEYeCF+zv5mMd0XEHyAtNporMk7FkDcg9i1iFEf96cvsdLkMcK+Nmv9B9XXH2Y
XPeho2/h94R4RDVsHmaocs0C1/GG/kGMuUsys1YqK1vFwa1Zq9Gd16AtgW9iHb7N7LLwgK26fm24Mvt4Xsw3D0
X2CrzV2cU5240iNtKwgIGW+7tpUIC/V7G2ihfPGvpQRY216Dy3z6gF+cMnmloppvU5KDyn2d4a3Hg5bTj7BBf
GQU7nixR7EkTzeI406Lpaivt/zMSkjZGgbnyiuCSnAAu0zGQ/0smAIflkm1JGbuL7N0sdqiq7mfUCbGQ101z2o
ghTF+Suf/KZjXTs+7TlQfq1gnVzJTY4KhyYDzASTFqE/SfaRo6WsWCuDUVypDwf/u4VQ0VWq4c2eM8s9KaZL+
tDYE3H4x7rQ4SgvcVEtugp168a6nxZNonSKycz+rx0rhILt8NjvYn1v5cZwML4b+66f4hNNDgFpEn4dUkcZ+PL
7wTdwJhVV3iuzXeirPI5KhysGdeRkYGFcB8gmXhd30p09fosaeoYJhhMujm34oCplsoPLk2WhtbRr7HBClzsZc
aYzWFxbMibSPDVI06V8deM4xnUVIL1GjinIn0ZJ377hTfb5tFEHFqr6HIubxazAVfqIOqZuoyNAAFTNuyw23fo
03+DfAWhD1jnB1DERU0AvPSQX8ERc1LztHDw+rqFx8EbJ284G+veCbGPSkyIMutSvbchTZP0f1brGPjMeX+i8g
gn8LtrbHQiVr9nQKrjWk8LQDvVTicdlgDNpuK1Rj/WNfdXy2a7bfa7/Ht6m1YZN8cOSI4nWE8iVdWEExNYKn5MV
DUt1NggoM1udmsm0JICsu0dALGdn10tm9VYUSJJ2QFcnUD1/bi4PCWRuFpPD1X1TOIBeIW+Bvgscw4DBa96GB
fLEd1L9p2+LszZPkamP2I/Q3exc9NcPMndbRcphtTLeZlIz1NAwxeC1WuMWvaFrKbf6rgoJ9w7mC11Q/N9Kt45
Khh0CuZCJDXYq25jks9FQE1tAH0DT8nX4bHh0FAW1xgDXc8Fd9PwZkjGT1/twW1s7SixerV502noL7AL2/jWbb0
WRqZqI3wiDcnF990PVI0as7nvh/wd+xUqr1FWRbpR38XRuqc5rzUtxxFtzJQenQ2qfK0Rs585AY1linpE4w505
djnYTVxkHlmwPOFFRKMBN/WCY4br2vPxJhtYj7iyk12A+CC8gH2CzzFM+x1As6gymgjswhS1zPbilCJWKFTYo
31t85YpH6npmmVGEr5gAxQ+14bAJTHQFu9TLrxFSd+1YFGAs+1D7K3qQxf7dfeqZUSFv6u0930/ZT0f40Bog
2ms3LizGiu82MqJWW9TRz1qKARzW54T+9zeyj84vb5o66LTt6yACfUelzKHSX/fnQLQGSFEgpFgPIz0wU35C
a1UCSVZ6CRPm8iSSzBnqlt/p6DXrHdSkt3JgSUN4u098UIjEgwjlwNUY9MgPJZIen0f1zmvpaHuVGwDGEKriPs
H8EV0fAj84RigbGrqtqT0QZej14B2Cwjsb1A2t4aL1jwkjyZBsKH/qvpQEx0qMhdZbmia6xHMLNI8/B55cZY9
6jwtfLsTyDSrTg3qC+cAkcL7KfBdFtrJUiajohC0u1zkjdnEtzqf0YngCNBmMCQTSXG7aIaN5rbqnjgsSC03

TGUjgHT+nq2s2AmNZxD7JtNM510Bvhrlw/ALmB7I2W2yKxr+z6WotbarDy20aSWLrTn93nungs04Ladj0rk+s9a
XMd0ojgkZ94g5480B2YrSAIkgwXNEhG3mV/gRktT1T4BQLjsicwL/z1EKEU7eT0gBCc0NN9CgDDEpeZpxusgUX
mh3YNR4vppgMJ9NmPz0t1IWkPpi7Y0qMLOGEQPXzHUPWrShafE6JL5gfk5d+updfYg14qxpY5v6v78anLf6g
tRQCLCvqtCsPvQ6m+6Nyubx0RxSLPHxjgi5+qH91t+fWqu09S+zJI5FItliMSG9tCcKrq0kjUDDiTf0vEv1o2v
EC+7C0v5ScKlmMuAMBUrDI72izoxmS8msrWEY7rmWWtxzSCeWI4gnaY/TtsjfCVj6ecMRiK6iDYKhbhCwfsmvu
+M1Q1lKVLjA4uvJkYZ8BMFljaHCukeQTGFMRIBeM20US026VJ2B0Dwvxs1p2RTDDgJboummRa1Xj8w/c709K
MhUv5K1rWlqsGRpl/QBLqXbdy+CmQS5z+wPPWEFvudUJn7uZRQ0+upGk3/gXBzcBZufmM0J15qazaKfpZAl
mddRd9zj7B5CpS+VXWHUI+MWVaOYujI4D51Fa9wbLzHlol+6cmaGzo3U23guLn51DiiJr4m2B6jWdtCr19aQd
4Ud8DRgWYZQLgvTgtUZVggR7gqjzUeii3FpjAh1dUJ3HTsz9Eli0bSgxFZopGZJImXhio3uTGnHhCvrprERke
Gz/QecNLx5SvNppXQ61JhwQ1u0hbqDk+l40Dki6Sb6wBvXluPx18nKjH6NFru9c2Bt1lQKfNpAhoC/ONNfyUL
0DjfJNYndliXGeRNHjtPTPrfHE6R6uWtVETACURbtkpE9fVYpUTOcvk7+ybe2fgkI0yWwA/dBDtm2+EBczLm6
DLGLJYBcB51o0oKENhgLK0gKZ3/oBmaaUSb2jWic6ov60iUHlgQbSe4+GKDHWtSk7luHJLPEpAyvrgYfZe4gG
IXRqv1ao32+cc4NMXmd91kZ919CsMCgstjK6E7QNnoHEugMJbR04M+zKCuu042TtFyL2CzJR9wne7sXj8/nNJ/
2PmeeFQ8QLBACdfgCQq/ItJPJuLh4YIGf6ASxLwBwZGhF4fNd/GhjQEohtuNvJ1D8HETIfUsuXZHBZ0zSn74Bb
kPjVQapugJ78eL3dTbxKx0/7Poqhn1ZxB9ymv2fC+Cvnk6hhnR4EVkPN1+t5HPSaQmv0D4+JX5XLHZfGDQM1z
/h7dqBy/AvHhPcxFmJ93fVP96LZRWVnQYDnHKTLR+uHz9DCsdvAP4mm/g91+derjho+HGUseR3w1PRh0lyjzh
346FNs+YHBNkP8GIckpu+8XKZrReXCh0ZkaknG1dA3n9GpdPvL+gT9SuHppyo4mwCd05aDPL3I73EfM4p19p
uaC7JTk1a3EGIVSPxboBpoH1Mng8fWlx3y0rCunvVarYqGTMBMZtNXKI4xLWILGdxKWAo9kW0aPX/c7gQNzA
sZ8NYTqJ+Dp/rGHqVTKqKAKjSK1HCVsM117y27stjwtfWQgFn0IALHWrGMj6qSpFRtbtnVu+TKbPlumCf40at
fJPWGUXU4K6UqJHplkHe2ooN6K8T1ndWLxNCjfmC+ApI1rGbxPwBoZG8R7rJ0rcgcxCEqNAMhs4BHval3WDoJ0p
qUrNCu76WwAs1rMhebXBvmKEZ/9QCaVy5lwRFXv9z2QCMmV6Wf2951DUKJ0SwB4ItvpHfaJrHvqeVbXaGzr07Q
P5S2mIZ1sXAubUvTTygL5v3rmViyHzssIgpMFRzyYsU6dnNM/g3axWTb2X6BUoqFMHMCQ1xmxzISLubBsAV0gQ
eUKz0S9vgufTqdeNXJyvvUziHLSsvHYBiN+Dz9r/7WGBKTesnX8zLdKsI7yxC6p3nzFBZQUtZGTSq68po1EyV
GL1wDLN9yMN4Ye6n5zT8h4wQQCXWzHPvsRnDHLLDGfkGQ9Ki8VnqZKSMCmpOFCr0ivxGU96nNeCs+OTH1GfFdB
1MJre/yH31di+uMrI6nH0D8EkjSJ1xpoyRzDEaIt0AtVZfMv6sRz0GAU80yv1BtzfRCsWdBqbBav6miQISaFTs
msJhXJwbD1NP1pPq/4NgyuY85FKbwAW8k0H35CCn93R10kNV6muVxDWATXn/WwjUqxHHJ2C20sP++5ZxEdvAu
wyv0BR2vGQC8VEqmR/MdAEvKNcdU3U5fUZaLpRJqBpm/gH1hi0Gp1MTWE4GwM9NEJz1hyS1NiNTqaLBuVC+YD
2hhutzHc1oZ3fWvP+4V2d+V2xpWdmsY04N9o/N6KvGbHh9+1RfdR8mu7LxAmvaLL1PSS+xCJUWE8420RMcKry0
jD1zdosPg2RkZzG5RfUwCO+rs53XndphnLsXJTR81z0gfpAprHL8WOYMUkkfhi18VMp0p3TdkrMNgFUJPTKgY
zxbxbv3UD2qWrdjbyfwBVW5BxnLuV55pYP1t2zdy1IUj48xQ6ElHb+z5R8aPVAeLkwuGBu0Jh0KcqpFn5mWzAH
Qa4eW3inmkzKq8xeYgAVzwBxXwHUJXBHTMTy7qgcyTgo1Uf471j2BbUL+LK7LIVAAu5M1sYyWm/eTHcL/XzKta
HsvNqaZfWCsLso03DgxSeroiRtepMWfVfYRmtxyahd/XCIz/81ckbiz/UFMd+mLIopJ3ZB3lcs+yX4prjpsrj1
/cUt9BD1cEjTnURiBu/QUXR5QUu0YkLLJrvN2wVJm6y10QhZnqf/bLqp/TmbpXtc9a9Xf8ALFHg41mow198cZE
kAmvYUyNUy31zBcNSGtNuHdEmrDAcUXLhIRO9fORjN5f2fL6KU/bCivJ4+PLQRmArgkUP7RKWYt4sS1TrnYXWA
XcgCGxwTclg19ii1QDvY+t9M1CVURc9hwj1VBhvDmdmoevcWDr6FbH+GZ3/UPPnX55dDGeGndKE1ZBSD5w7hNZ
coqRgxHtux3TzNNuBcT+SowPCVqMf02YjDwKugYxk2hQM1RkZI6PMn8171PTgEgPEx47vysbSiSwfVmU8kqAPF
0vM1060Naq4aCYH0tkNz4odduLqn4DYDvVuIKZrVtD1v/Vgb/DMGE0313rJLHUEXPbxdpVzH8XBpKy6VAx2QZQ
SwF1uT17R4qgC93a2aCc9gQMT35fSfuMQHhXj15mqI91JbtIRCCi2qdGWTAGRqvWcbFmFvTzDT0u30tpuLhI9y
R0Qugmu093D49UynjRW0chUJ5mY2c3DFF7fU9nix6C2adqDjy6cNuX44Lichgx5y5PpMjNQM5L8wTMTkoEz4TJ
r7fZcumJat0MlaGxw1us428Mi4nBHsC6nojh7EMEYiJv1h8v2XBtxCS04cLzZPbGT2CNM3jtRrn9Pgb1QXu
HgFPEh2J59fPbbaY4leGfijmVcZ6bxSIPUxncv/Z8CfxQGeoDnKRyj+EGvqmzF/cTleLo7t0N1vnj4yKI489Bv
JwXQI5vqNjmxnZb7ZzTdiUMgKxj7Njs2Nix7aRj27Zt27bdsZ00bdu23dH867zCuZs19QB1WVxfSm+SrjFsD3
s7W9bHbp+4DvTdTJh+0htvo7uy7V77dqydcWRICc0LLz1pbvquYEcya2TJbNaWpzsh1TmQilIFphjpl5dSg3j+M
aadIuabUkLWYitp1AU9q+CCFaEo5tW0VnTjjk5JN8nje2DMnzFMk5EM28yR+/wmidVLS+r5VZWWtJURZXkMa/l
uhavnQgbmmDTbBZt52zbaW+yN8TDOvQNYBH7MkA7ZPw8IHpSRMZhZoh1LLt57K47JmcGd7e1Q0ZK7XK0SUP4t
LdKF/shXc/gK3HsKEdZQ21gH6J/9EmnkfOUztsAZG0rVL1hn34/UuEyVSf0T67xoXreNWRU+B/1IYSYbw0IfLS
1s0p3YAcyx1znyACUCn26470wvjuvQV/AnXnVvD/JukjbExhvHtjwMIgoA1CoZUv1I5SwuUtKnY1nk5nrFOZmB
YXtwgRxRqhQTv80i44cjxiu0tJNkQreumFVgtCA2WU61blamQ+dLf9m4EahKLY+Lw9yIB0zuKBG4UsktAjDxjn
F6tcrbmFpziql0tfCgAqPQSh8e6KqTb78SC2+MbjEC61wBWxtaUsFl+Hxv6Ip/G8XZG89QkcLrt51bGeFuJV+1
R6/vpkQvDg3JH0akc0+Mw0ecUwSz1WEDEb6UeNGxrPbyhRqnPua/zY0716emwynGocgoopzksa04YhjBWTMud
gerP7oav0EKbtQbytBWuXLGacBmLaust63cgvrnRidkXGDZiWke4++AwMU8EGw88F32YIhs0MW/vqb64arKLFP

BvJUVWoL4W2Y50ZK4FNLvVLG2sqbqI0NK2d0BF6qsQD9Hc4bNjoPZP8aanrb3VR/Dxf3cEg45VNA3TRC5s7EZw
pKRyTgnAR+/7CvRURz99PvaYfe+1Gx7jy0hTjdgiwOP8Rb17UfGy7i5JAsPX8Xn4F6HHP7X0/b9z1Qcy+JGgzn
jpo9JiJh0HzK3CX73zEJZz0sZWTzn8ZADRAhylM7gCMXL80enRJtcQbgUho1ywAbcnkzu8yTehOfBHKCKgD9W
qLVk4BqMUxf9ZbtGzYAZdRkWwfjNQ1J9UP8Azbtx5Lz6B0MPlpo86FRNRZIH2m803y/4rq/ZcMA9f/LcIeYc8
v3oLNWS1U0AtdbyqbkzuWCZNMrATrZqEZ1pbxI7nshUnotMwzJDV+qi6UTAAcnz1scduIzzBNo8ISVPHqfhShi
Qm0zdp1qPQqcweHRbLZMm8MBazeVPSAgA1FWp0QVSojJK3/jlwrrgNAidwk/TQuF9B962mjTapFY2XBjk9os5
T3Vtk0gEr+xbgN6AcikhkFQE7bgMF00BC0Cm5V3wYgc0G1x45x+VJnYZW+657aNi721gU80fkyp05ujf4t80V
TGgR7Gyleb60iAzEwa6oDeh1TH7ICTZnmjcr+GFAbqSKU/WawgfrT7I4wiM38vt4POZXLr1B/3dCrXR8g9WgZl
SvebUGUW6dcSQc9q/H62WkCPjjmppQASGubyFIIVfKhv1Ts16cmTDKISSvaY3Ea3QZ58Na9ZyhG8tfM3WkYOGU
1/KhXg75D1aebbv9Rp17SmLXcZN0v+tsplJYwpfM0zif+Krd1TzIxUuft6cuMDiBLsjM8mR15Yedh70ChhXq
1pJ4K3ynaJ7wEb1rgovUM3L0ySCd8p42S/ls5HMuW/Q8qwWhtzIWkNFajtk1HGi+iEyu2BeeMyrH0ErHUQr1K
SIcMcYv24Fa9mRCrt7Guji3fkWBpx/PIE0b1Y8uhylp+D0AZV9mnQYbKPzDiTuait0JQnPiXHiqP0fbu2w6Jxo
JETCNPGHzI26kLQxK6GBT/+EpU2CarkCieMTsyswjNbeTw89ffdKeTiKc8voWGKv06UfZeR3MFnn1fakcSorhW
jfP5VGHYSxu42/du6CI7vv9pG5roscn7gE3XG0vnn0M4u+m99e0SNPlXSkYhpNISxoTJvriU1ZFTPQtJYX8aF1
7fEaWug1kXBy+K1R79Y0yAtWoHrtxdvsPDHfga14fz8jb8Aa+RP6u/cA51cf4uQQ20D+bAymN0Zk7RroYA75e
yoDkUE01A2Ninhr2pwCrLJJ8k7+1qZSZGulaJtP00Drm+D1dPSFuW97qeEsoikbwPiYfoGhN24vrrNI2yvH01K
zEoNaVm+GVzaMSH/czzvMWCeSQ71aPrZ34f0jCbAZxBm4uRLsebE5nyn8w65XTSPTGfogw39GA7M1W9+prtHkR
NvTbQhjpDrQUIEMI/UMdHh+8nK5oE8lshlap91Ouy/XKmPbxWbeBw5Ht4H1hoU4cfyeT82BM0N1xwQ+0ymqkJ
OFYyu3xTj0A/AxQ9nac3kKktdhAj7HtIHq94xBmhb2k4V9SP5kf0gZRzkcou6Zf1cmgilyYx1RIbEoX5sk4prS
rUX0qnYfMLk1fkCzJtkJCoSDn+IeKvBkr4wgNz+F2DsRkCtmEvIjknAF72GkPdFE+ziynUiQD0vq20mW9c94z
HeQMjojaI1CTa+NDmwLTZn0M4WtVGolVWGIeMB74XKEvELGrg0LC16F4LtF5nxW2K0W5T3oJjNrzSoSANG2Sks
CjbRinh1Wr6c6Xi0nFLy4EPTf4FQqQAXNASEXoC8FnAmwH4jDux4k1IwXxXk9p/3svY3NETp0ABVaVh2GZroBw
ZFBX8iJdaSBw7r7nExxf4WbT16wTuk61T9y0Hz9VNv7RruIOPc1jCOYZQ6Wn5vnTvQhsjemDcRe/36we6QkUX/
vjZ9FWtUFOfLCZ4cK1UKH9V9LEDrfKaDt9obpsLmHBbctcea0KnXi3PCsKGR02Ue9kAbhRmIyB4huaDzHNN6RG
MxhTs61HGngmcwnwn183oUv3b4eQZRLb85tsiZPsxuqmqJVsZ2zqGvFF81VEx0IfMiParg7JGLs8EuHik2B5k2u
ycR8CvDXChwNOKnY+XNYpo5eMVsfrfrofFtee/ rjIPX3k8ss8n0jHXx3j+AVxoMw415+J3vvDpE4mAA59B5E1lw
y8B6LRS7Cke3uavfqLerXfuKchv6DgHYXG1n8k8bPtR8x7YoP8eFYpKB5trg25fw3CFYT9jeMx0zPKT1LhCmYz
xuQ1GLtm4yB4d1EzeVai0pHkdzed5o5T/hjeLGM7yte6Z3K1ZC9QGwCy+Iu093FZg+8arayIrnHwiMC4y5NODO
m323M8jH7juj1LwvgV8cPFBwwK+bkzwMd96p8HSXQfbPTUU29rFfaJnIIPQnu7wbX93pKDkcrj9guJnvxZSVnm
F6dkTv3z5iSRghha20CuDk+gSB/n7923/l1mTwW+2XBhVG/zE4ed7estYFMDs/BbRdUE9yKN0MDmszJmTUL2G0
/YgF+0idro+1UCbc9x2W7c3eAqDz17g0KhYj91cYdg0uH8zbGzufzunu0jcEdnfZVgDH8m4bC5+1/Wo+HzKeae
2sWg/Y7oRy5ftFgQVoiV/Iu8fTV2gI13rZN7G6L+F2hclt1IPMVdckMb096YNe8yKUAs14eGs2bYtUhtPPUrTI
eCnypGSFrWxYaldeoivL/kRp4TnaUgewJW5KJ416mWZGr1NPIsLj1nw1b2ZqpEj1DKros4AVUvKw41Fz1qWr0S
HcLgM91M5ZkTVBUEfEguLaEWWhIkDPsVTHUviVQXp4Jxsilxub1+Lip72Led6czBvP1qUMDMY5QbaSSrKjfwgI
F9jVBXySop1D4mU6MbPqcC9phk1mJv6jeuhwFIRiMtzhvubXjKzNYcP5o4MJ2LU5ma0mt4R6dp/tEZkHHq3hph
F943wubIepD6FN/rDJ9PZ8QsuUPI0BxKyv6r0zAG1PEimTkBu2bDzHdeDwW3hUA0GyRMEibzy0xKS6ijkFbB7u
QB5a9X/OPB4JCKa+Kj8NLbwhEcys1zVtgshj69wyh07hPJ6Pj0W5uKLfztkZJ1T1lchhMt84+SznaLZYwMddV
W2kxzFG4SKwdtnEzyJab7+dQDr7CSsNyRDPqwHX9MbrH3nuUkUnjcl3YQqYNTgVal5+WWR/FdFc0cK9ku3XP6h
0h5/qFBgXJj0F2N1u2R0M54sd/o3R8paFjbARuNiZCjFsh9186qu3TauaHdsbapy8Yi071HhXP7a0h0qYqCJ1o
EXG5StxfC1zJBVe8B34wyh1b3Rjph/NT0E9GVnDeq/sAn/BMwYrr707R1UKMpLzdxNwDwiupzT0rX09+mrh1j
Q58v9R683eEJwtJUj+3s0761eQ+TrA3Vqym0hwggd5yZDpTSMg787WPcvJHIMTEWb7QusG0DZ5vkm1sQXjq2
Dh1bSsPlUisaru99aaWwIMeq8oFRJWRV7X1XWtukxjsbkAysnJH81c0ZMKP9FbtI+P+P+NCM0sdZ1qsSjUEQ+3NT
e1YPBWLnt0q/SqylC6FTG6LasbmoCCgkp6U0mr6d6PTGBT2zrAr1mQ0xtWXqzUctx09uoF0ve6PiV5d0ww5ueax
BQsvqc0ron7I0dWSg9WsgEBHqk+gZPo+WjQ3ma/Pxmu2vnRmjkk/yZjdtnlSijwy6XZkUkinScAnctXXDdfLZ+
uXB8Zu3CBRGHbYzcTdpQVI6RrqN0JeMNsHIQmRfBB07nMByI4eUcf1jwz0X21j28UnkMFPs/f/IP8E3T974E4
BmYT1ApG+crVXV/ZwyED00MmZxAnN/BTkoNxLc4E4aNrZ+IraqYJBNSSNIMwWNQCi za0FRCY5FW7gDFimyZvni
NJKUMEkYv4j0mQX/PUR80jkwlYyEw44W08euMOPMNJBmWJ0veFxEmwdW4xIjHr1hNSbn6FC8i81AT5+0geORN
YEK3ABwovX4LegneUY4FgMEH4wZX26p/kgDm0LeMkJzfiSoglxU36Fy1uxRKzN+f2pYyyqNpV4sEhC1/oZoZww
2vWgUnhc0tzWqdDJGQW00u36ZmeS3d7m7JGFL+K/D+uuhAdt6gye9qR57Smx1GtvfWEq0F3htZCxFZhdQ/n7
1Wtj/B8RMON0WK/vXEtmtQwY0G2z4sdGzzbLW7GrBq65Ycquo3ZnSrTJ0g10740qA5fL9bFnc5LKqJz77yBFk6
xyP8ceKiPv0CLfy6NpW4Hz+JA134hIx10/m5a+zFNiQVA0dMRDYmk8jMIYxE+QiCpbi9e3p1cYMcfxdL41ZMKH

0gJAtNHTCLnj9YQiXxKXF1WxrVSm04LJuksKFS3N/6v3ciJJIf7yB7jEGtjJFSV8sXAEs2DBI+5z15QdXPrT3z
bm6kFCmxVx9sMT6q7e84baGtXzumtUU7d8KbBdPRLjwHD64NjiRM6nIoCtDAzozJmV4V+EzAcxafOUfoa9q+ai
pdfTxs7G5h1zQTA3SSrCGQhP+auRhxi/q0iUQ0eCnPwcl+Kdf3PUvfB7Magfcj864IbvlyIdwhJyF59ihLCEJ
zn7qxWjipkQ6RNR0HHNeU5SMVgBK6Jkc8Vra2N2dhSSIfmFd1R0wTE101aQtUIiat7uKsuu5PkSVYjoqv9g3Ak
ikz9rT2/KQovrQpan83LhDN0oG5KodpDycaApvzzQRDpPx9b7R03ZV2Cm0LLsF/sPjultJmtsCL7xaX2V0oSf
w06etu0RrThFU8Avne91oltHS26zHsSCFQaIqRZSfswxTC2U1GdXK3Wu4VU/G/J97cq1Ti4+Vw8U5JYPha8Zi6
b+k4wGH2k0uBT9vlxCWwxsUQUsmesMh1P7rDG19zAcP2+FzQJxjGhpC8CaYYoRf5D8/XHFN1RmjqBCpqqFF9dB9
ucDTozCYnz0qfcxMTB5MMb9BpWACInuxek7x1WztAjmBCKjJz9aGIYe6ZZvyN3QDCzscQuwkjrfv9y1qarYtmd
/Xqv1EMrYslru3bCl+icKv3klsnn2Mk3Z7uBkxSVfp6+uvkgyJ9RfZD83Ki3S5haHiibMrUvBD6iIycLHn9FfS
S1Kb0GGW/9gzVrskJxqc618zF6NPI2B91QtzuCz5o+1wBPyM6d5y82DLI50d3zDweR0ig7oey6bJolwLIR11DT
XZ0sm0/gJODH+VZIcsz6b0Wx0d/EW1n9evL7UNYUgTJHFwaE27zy/CoLIUxWAsLsasWtIosAzfCsUy5vLw19h8
3R/KrTH1d/dBg7aZKVDxoopfGLDpdgJs/27Fs0myCbWRq6spUxJf+mbLTAs2F/hG+dq1/Q885DVupB128ppXJR
531PSNxji45v0rwwRp9K1t5kQntL0H9Yyat6B1Z/pQqYulu/6o20Rc7zRrU3Auuq5Fq0k/y0qQf3dJzVI0nUv
DSS7Y5L/RBOIWVQEUsDwLLQZHOAR8a1EzzCThfDDjCCQBuYzpHDBESJfymYOhwtVYdaQ5F1L2c0DhKyde+fgIm
jz7hKX1DsLyC8ry01C5aPHzrHxzYy0ZVsLPdUttRae4vTKcQ2z2J5TALIm2ZMKjap06U9JwfN1i6a7KNPd/Wwe
JULs7eZby+EHCXV1G1MCLvA4LH0ZCtRZC5IMD2J/C1MoKgE4u0XY4mT3MrIRGQGvkzwPwpUcY/HMOUawX98cLI
4km8cNBU7EQI5CsczpNE7AkD9TSujoje/kowtFicTX93JbN9chjWJnm56TJu27NMFShb4RiIZxm60n4prfbr0V
NI/9wULyp5wLFYIiupVMXpjtv60xM/Ro1I1Mxcd1VzFV49L2UhxW1+bulk9cDw+Tu5Cwys7kNfxWHewpTjoJ0+
/rx12rXJ6/dsZLNKT5jyFjqQJGSZKoU13JrzeuyWB/zW9wJTPvdfl+sdJh0zKHjXN1XbBXSC0Wzq/Dfnyt5ef
6jkj7tLVZQ6ExDGYbhCUBFEypgP8v2lgKocbTBrqHz9dFOQYA76qsH+pWwbLkKCQcfWFA3J28UvbL3aVu1BCUH
dSRmTHFm8eRRn+F5ozp1i3ztEiHj3p0dpUq59wd8J5msgF9bEf116ivoTwmMVOQI5geZtXuzPspB1HDd9XYJDY
v6HqkJyz80JtV7Dqn6Z6xuXa6YxS+QWtrdibeocoX74hdA1H73Kn8GIP2ipkZCxkaNjegNWI4XijST0/fbgt
LWXmMJy5Nf4jbTYT7U8Ii1GftDjhj4mAWbuR1KasoD1E8fV5UHEfp0+CREqL/qjhB1PMk13NoLdhsg7MIxBE
8f+74g/9hA06zG5Nn1x651Dr06c51wFQDHx/QVgUt56vt9DEjtS1U3/HZtc5gK4mG1HWtw1E2fc08sJctLFU9u
X8ZIYcvpaxp2DXJPOq05ow5erw+4fE9ifaEV43r/6A58M3x+E7JbJVHW0e4xI3Hk9mPQw1ASgVRQLvneQifGwt
PyhKm50CF0a0t40nA/WJX/N90X0tZ7SkqgCYTY196o9QvnpjgXA95Pq0S501gJEB0J2WAv9pIPkaqI7uzRJEaU
/iKuJ7IMc1NNJEFg4iReD11PxcM2RCMsIDmCDT9jm1g/UEkeWPBhtk2Nyxo6kaVT1Y0ifZSkQUm1F1eTkD2pm
szaEYZXeVXatcu90PEkT6HVbxuVX5iQ1fK8ErKZa7FsKNES7pP52NGb3K0ZsKDdNbflv2jCM58Ye6P0M+YJGe
DJ4VGI1MxzFt2RUY65YvEFNjVhD6aWWJtSScFKM3IrxuzwcI961tde4QU0DXmi2hc2u2k+ea0iB5zxpDt08voa
WVtLoUadG7FebRHma+vj6CQakVPw/RN71bso5swcF4WUEhxj5WxV4G2tovt5RTwyh0Y0vwmB6+1YokAsXA/wxe
IAyB+BXsIJtTLEt23cVyiBTwAGGmSW2GVMAiq0sBGHK7ru9pfkea1TC/r1h0K5od4udS2+hshiE6CvFGH7D+oA
XJShBW02VbB67MdvX0mmsAS3FezWnYDd1YzWmkC6ai+ln4L9b3LQ3UV0q3EDuuuyXdfIezF63ZR/xaEYhptrPDB
tmysCFILTbPQaP9s089pc1RxaecmvS9d9KGu2UbQ11Yn1xPjmnY03H1S0mNNI77i9oHk1eeQR1A7m41LWIQo9R
+uBj9RFGft30VX4022POPZ8ree1/c3bcW7CSsz8QIVtUQS1WX00hQbGnPx+WnRKruvwxG5YiHY88H3n1wi83Gs
qep861IvBNhPYvMC0mo9DcPgcZI/k6ehIsTG3msKczlsqk+jM91pmcDJNNhC+XQ/VIFEqAN2tt404Y8/2+IxC
Le0Xg9YHG3zoam9oTyrmyTU7Uxybp1cdfkCvGY6CmdIpdm8M12t34sRCDXwUhM2q7dEaimPHoB99WXjb/V4y3J
CnB11Uv2JIY8jytD/WFs0KDLLE6QEL1184gp2ae1Bsh5f4hDXnKlvIVE+ordukeyfrm8N4h1omKmTmCVI81Zvu
FEF/1mpKyZ+XLF1Upm47iFIBLq0aup01i6ZEGd3qGsb2JSqHP7hp6ohLS9wSR5DLk16Uhp0MHRfXc70h9g4HeU
NcB0N1iNQZSGs1KKC+mQE1JGZnYdv0o/0QVYv31ufk6triP3k3WnroTkn04w+gyPSiZuiFigR+iP4zUJtkoHF
6aYYDZNe+47vL42kqwJ7cdLS84rjqSpg49NNbyfdsPxAV6E1brZ2BWNV+vcnh0n4DGOKVeZ+XHl+ifAZJPKFmz
GJ/EzvWJyd05eVTrnfUfEEGU5p89XqsR0bov/CvVXV6r92Kv99qIN/U7hJR+9iXUvr7jGIuCHMDkkUhmDZKX
eytxbT/ZGAdvZJpfkKBxWuIn58zxbBGMVHAjnkqhmrxG87NpQjExd1GsrRvddfbEwvEzTahk6PyN+HLC0Hn3Te
HP4mYtomeawq/n7WX3c5AEf5edNxt15C1D3Uqn0qAcxE0S2Bx0I+qbAhvo62AOYlqWAcI7ayuyZBY72QPZLE
16ano9+nBP401YyiE1dPTdhUDTJo5MKWB/XELH+nYBoNTRCasowYPFCeY1oRRj4pNhLKUSGkIfuQj9H0QYHqhn
FAuiiX26QjKeDz/TxmsVQLq4unTtmQL7+vWSgkBj6Z4DC+k03RN/CjTNu2viSeossJPEwpRraXk08k/20WNuQ
cZIwKMi2vqc3T5fMgN/exfG2Pa1/T1CcJy9ZA5k0BAatdc4xPyewGsIFJ0Z9cjqqsbd7P8gqMD0rtyFnxXm8Pf
vjszKU9DcY4Q5QazCwaZXZRwgPPtTCXuF2+/hzV0dFVLfACoMQkzcg1er/TxN502K2uQY4IcCb0M6KzHf5etmW
oYVGS51/K7zbVC9MKE+a9c9WrtZh23yq00UGcTZAmoyUvrQ4sEW1ptmfddPnYDGGvU8Tzw7ssUdLjSighHu656
1cfnh2RAv151m7L7na7gbyxEitJSVm9MR6h+j0Anc38UnWXuiyvCjv2EAM5eAuW+DYgbmXZUAeHnFbSExQffkW
aUIOT+dL/em47sLCm7M65s5dxkbpzCIf2eXcfyJ/gDGibT7Sxotp4PcQQxvJVKchrjctRoL4UWfetLUioCyzkF
zgsMIwUUxts9s3ysu160Tg31Fcy2DPZ8cuydpPrfVQdBOLW08u1ExLarVVNjhoux/E9CWa5Fe2LvgEIzRC9sE2

LghcCzxF9xjq/dSgsnr8xnz5T8kUT11/WH4gw/ptrFSc1aos/HHuwNQt8wL7SvbdBON5w/pwC7oyN9gTAm1Sv7
bJQ84o/ZqL7vV/pGRoo1V0wuQv09tR0F0GCV9WT4g33mwI+j9nT2UyarM+/AxSNaIz6KzSv6j6rKDOBpQV/V0W
plpo8fbMDW+toBWH0iw16kYE+5jKzn++GCxa1Ah4CTtHxDPCLLeGdiNpJm03NPqHD+PhG4pW7TiayHJ8Y8Lt8A
fVqlDnqVwHqKnBa4d++WdBEwf2FG+r4gAbmJpfi8RhaEDd2n+PFp3mf0xL8VdHq+XFUyZq1daX9Amzev1rrdX3
vnv3MDcYMMnoyesLWC4Vv2csD8Kc09qGKzQVGAYL5H29V12/pc1EERYbJemq0R4euBTYgzwPDNgEQX4VXAlty
4hc7f0aIPr1JbK0Nn2WdUtADM6cqx6+1g71dJxdZ+e0SKG3JGK+3W0q23Ct/IAvW/lbcPXRYD7XbnApPPRZMGK
50nqjTc6+8KZBQs1/HN1xCe/cc7n1SuXYWhJM+/uDwCBgubXKWQ500BG1e0RBWoGVxH8kjfsJSA1Q1wQzG1kE
TOGKo/+1T0zJcaVYDSCKccX7173eYGiKaMKNW/Z1QGjpT/38SJvA3FMGQ8U051E9IKj2Sgxa0PRRQjKG+NnNdI
ymH9g0G9UGwsDymtqjjXWeq1Eq2p4Kiza07v0mm1NK2nhwzouoc1cs/K7zkuKfp7j2c2U5iAkxQxN59anAWH8Z
bJdyLVzweuQpyfa1zpPFG1RkbJyGjYdQDDv9wmXdcVA1A01t5VRVvLXF9qJv+wK7ogruXPDGjSC7mXT7ThCt3/
9ZP04Ru8qUvQsY9rMJW3b38Ch/F702EED1K/HibSYu4RPnn8YQwwVcv0eCSSb9CjV0utQWKJpMEzxAzBFQ77v7
E0PosZwDRuxu81ZDg42aW0Jw3BfUmTib2J3itvbaEoeCfFrUTWaZ7D8mVaz2M0rExq0kYp3t8T+4uFqkn2mLL
WGk5/N3Vn101NGgwBdoBIQAIqLeAtSX4cG9FRHHwpeBkzQRZgQ1wVPAZ4ILNDsP3TcVMr8er2dfNVB0pc1xXwc
hEJd7wn9bUD0CLpAkGJmAIphXsNoboGyjxFc6Ndn2Cn/0piNnt/+W5sYolgWsWTyxIhfE/wjrt+QhQiDtRBkW+
zifLR7Ca0XnRrKApoSmMjy932WRmACm4Ct4ocGfSzrWyBdaDIaNPnKTLiw1AmQwGg9xE+8aVFBE2Rpr71IRuE
rbLyP381vCfScv0b0laP/7INX+449nCL0IEvelrtJ8M+RftDGUMis95yL3Kj1+tk3WwiaxJZMiQpMk6DCq3Hu6
JCA5+nGo8YJv6cKPNUkgEc/CJV8ZU9nNzSuLqexLDPMe2y5roufDskz3hmpRTKHZUAzAfhvTfgkljHz4ksp
yrN8ExjyFCFS2KoCeSLfYi7z02Sh8Mq0udVzF/UoM6xbjSukHU5RUVWvh708tM1kQf6cMmdyFlNCc1hN3WX5v9
S6bhapjL3dG5gsLqpRiDXY+0F/GXsENxfkPO/GPfcWsnp1hH5P0ZXDmBGZH/4ZMwrxPA57dDdZbmYquVLK8N5h
vItQCk4cXkh1mmnRMqY8nmfS0cCyQEt3csvpvPuP28yXqSzUQ1R4ekJtzq5M2+7I5NjxGSorw6QgK01tBa8UbR
zda+RzliQT60dPHIuVXm32s1oWW0glnymkikhnd3KnMI/K5jSolQX5BLGsgVqNgcnmnIxwxgflJuFvYPbiXKFA
YkWozpbDSSqmx8e81GsvjweicR8nrFpbvx1SKNqda/ZiJci3tZFa/nLU4+ZHvDwDuxgjhFPB2G7kKHVmZvJ4f
nLxkyk0I1LXkq402kZ1Y8vtNYFqUp+fVWYsXq0SY0ofAdWIPqgEYhAVTxop16UQoaF06va64tgzkc9PxPipLpU
kPtZGxtdZbQFWTapaiN78x9pVEa4H8vldTq2nvJqsLD/6PT4yFoVrHYUJjqxM6VSTDu81LvnneIcxrf1jHR6sR
m4DtqCa7Mi544bUUwi3yrb0fVR50Na2I920rvGC4S6uFmb9S5/QF+gM0fpoT7Qs580c2zUVS7YtH2HpjMT9NiC
F+6bLfLc4Ub5QUmrZFBtkmu0yrufTcKS5qwYDjZ8t98106rDLnLRclmj1o0fxe23o4I20jp0fS+GpTorqRSeOG
k634xpVqwzoQty10RCpWfYCY6UBDt10yeIvt0p400Y0b2UpCbJNoW/0A1dCm7zrzmakM8Xxwow1R2rxZqa90S
MB+1/TL7qqBZp7zBJMGa9p2wGM/zN0FDnC4KG0z/NrLk1tJjCSTz2J3ZrpkRL9Yw0mlkaCmvfDDNp0zpXZdGrw
1/ciNY7+t44v2Z0c6VdjCpkAkTQxhiRtx0y8S6Iz/Q7owc10UxGaHK4YnAjssCKnYbU7So3Z8DPG1KXK0rMOGK
hg0Ixvbsq4M2nUghxXiDa9CmQqqv5fvtYvbM4NRwA/AeXWQdEFKd4215nYma0nyUM4CqGP1GUzM0rB4kyrKIJM
PwhTJ61+x/wBESAujoFNQ1WLwYF0nGGxsf1E13M5bHrj1AvQZAq2C1yoXcxvMIYwd56IM+j8Wx0I9IfgwUKVH6
DU0y+VvKb43FPMxZxVz0x0B1Jr54TJQqL8KSQEIMG30vzEneFabnVdqqNrNEqkDvGpuNBxChWww98c5Y0rk3DX
1AjdytZ9NgNgHOEV5KmAWpTRoyE6yRZ/jqy+eH8k785r14s1w31UmfaQfMUGUFGgtNPPcueA/rWK3QXXnbutydv
Yis0mWi+7wNT3HWYFoyhMZh0pkwb4Rqo8G9fj14Lq19EgcspH2t7IDL1k9j7H91PksxIUzSX8x+dzjEBGCyL
tn30bKekA7r1fXy6bFS+5kYecZNcG+heJzjd0ZzwKvch01oGMgB2Sj819z9KknP7yxZIxtqsfyoq0AE0+/PfN
bHN87ZScN7r6JZDgVQZGtYFQNHTIQShphK1bqoy2i/kM+P1+DjC37kwCtWZI1d3JMMjwUG+V1/eB+Zo7g3XIU+
rYx2g4o3/BWF+r1fj/jkCzVkjxLwte6xW/9M/LWu1aHq6TTvndtDeUQ5u1iNECpkbfRECMFnzBgdWQ1jAEudkgT
1n2T6Z1tVmCPxUmVkJW2zdfB+WebRs+fNL1UJ3mAf6fXv9BwvHfCM7cLuX6nH3T0fPVZ5Wnnrp60Kpn+p7fLcc
vrkVsAgJ1k8x8zFc2Ixp685qWkTsou0ydhWt5IiXo+UX8ASHKPI5X8eCQyWV+9GSuh5gJeqZ1cXbevmCQ68SM
TZibWKbSGpdT0wzzgIkh0ZJTAE6++OrWw1I5c+CBC8jsacomMa1auRWN7HBvrtn/n2v7+p13ifItCRZ6Sb2X1R
xwttDxBjn3sCMdiezc4y4EkaVeeUsPh2rfZtmR5o36XM60Q8uQWgsZtK+XrjGqXWH3KshNKKR16S9PdK7u3Fg
jrz16NMh8WFw50mEy0tgl+afIZA1DbsB/mUEYxCA98p40ogosQJjgJTyPmzNRV10hFkNQfV4WtTeJ+I1Pwwmk
Qsu3XzqFxrVMh2P8UhzNW+o9H2/6By4UKSwiyEtfGcexcmgGb0YMFpq31TPOyHlrxTXYLCCP1YpSBT9DUhBavCe
APtpPCrT55dcS8QUm8suxozvppusA+5+Rq2bT49Kb+5Pzx5/Pz7o0cXN9RrSAQWtbIMCFJffxJV56ajbX2Nb5x
+iaM2jhjcjkBXYL3IeR3g9UJMwaILITs1d2EBVfBtoLtGPfAP4xAVQ6Gj1a10tyqI4SkK6Tq6asqtAoT50Unz1
54ZA5ZnCNiikrfvWrokAdcr7QitMoQLpSids7S11DaRFK19EePuQqpn4FcL1bzdxivfUckzRuyPoyTHQtcOs
MDVuAwtSchWYtksdPiHWwP2jhqb1irHHW++xaKaFApuyavp3kp6GRT163kbr7zaXndKFtQ0ZzIiDtWMR38v8ua
1bT+btQRKpZ8JzCe5+PcTwq3qoetSsBen5CK/zz9TePTt154qVDegVo0DoT0Ez1L75Ga4VVx0L0TuQLwvkn/FE
0feVrTpV40/uUVnbJuaVLWkA6kPDG8xJ7C4zLgLrJ1+cJgB0iT/nq8LwlvIPqnrBvUtoisyKdmxaueYjMA1A
qP1E60zZjU/YFxkq0L1y28S3TU40/BmxffNBy9R69z15NffCw5Cwz/bF2iPFC1auQwsRoB2em+PhG04X4jTzX0
/epaP7YCiRedP9VDXwKurfaamqGpcnY618UOEYbf9ue3GYhHHdRPEr59Km6craz0n5nY3z1cUrJxUKbFuL3HO

0QEFLycLkHEoWyyd5qPkGIV04FK80WFpkkcFZGxQL7NPDmlMtqJS1Qw1zwwGU4SUiEgQLSDksWsAEf1E/YgSKD9Tx85L0jkJLvEsBY8oL3cgziQNYzGsZLnT2vNptMqtmqMNFKCQd7hImKt3bZThyqhK6D6SKdLzdXH7ik1HS AEj7jHiPDwCR2JQ0YKfSIZhfvgyia4Ba6g/0SKEv3H7m0dzRxRFx0er17wxk/7+F1qm77qRmMkXaU50EOR0 osjMJJ0Y+nyd7jCmMmBDz/Ge4m7IPhL397a5fWcuk8aE6a5JtnX0GTtnIsHuILZsmAVK729Nc9YMNNUhEgX+0f uUEHYN7Y/FjVv+y1I5/3FKoyVs1Vd/ehwW6fLJYnRCc+fxtNPyKcVURcg9z0LJntnDw32D4RynfIweH8aItPG0 0Pg1ah91BYd20Bq+Gtm+L0f4nKV+mo7wQeDLLPjjIyyfso+5ViD4Pj7RFzdiazJ6PPMsz1PC81ImVY0/4absBG n/GH5qqgXvq96zzpVzyF8v0vDGrc9a/baW132HecvWYG9teums88AUw8PS1ddmeUjoICV8ApTqDsEQVKzRfdIt ijhE5GBNWILTE5XrzFVwbIW9ArsXRwSP58XIb05FK2doCD8B1P5be6+WlcjQYXmE3yx3WG4CKfRuILD8vqFN88 XrxJsq4/qZl603S6F+vj7Mho9RB1SPE1H7rb618AeeH/b03R8G/R0r/2leChokfR6NqhGpgcZPMKH8WmTomhwq TsXw8f84vKWL3srTEnM971Vt8D5t+p/geo07X0WtTbgyKtSG3sC4p7i30qQ0hUctmsbL0/9ux0w+r1k1noXW+TC Qfx90JNCRWoZ9oX7YYWGVb9MJ3DzQn0gLxg850Xbrbn0Tf1v3d7GhVjAh2kC3P1MkI0qYDzot4c9m/ZNvVJF8A paK1V4T0Z49idBPAQXrlnyhEkCiOKAJeRqAKIPDuHeTduFTGjRdE8vaX+U70yBsYKGF/27SbFMsCiaumVS7KsL dI1rR1rAkjPWLqrTFG01BeRVazU8rkA5adqpo8xdF9zPmcfHSPuLKK+456CVNb68sLMj0hsa8zzGBrd6tbLcx4 gnYnrUkUunnCuNK/50+7y6nfZwCa50CFanAV0QKUorr8v66dSfN7Ur1UWohyxZ05NBnkp8Rt4aip8wKnDWFjbA zSjhgXJBgMLv68tHENCRV/e73H5PbcHqsjKFFJU7kd5hNFj4RaE0zXAARYSNOkzdD0c5cSDcBIWvFqfX10rVg QeIDA0yqBtETtbVows3Sg3WW+951b9qw4WAGvKntWW++XKKK/uQ4AudwvYjNWYgx1qYqsXcNVLYLAss9IaIevY zLZXXXvAHVgY3Kcj9WXdj0pQQ93oduNKcVd9EvzFXEptDfP/VkzzYNWHC8/VFFKSK3n5xVz9p+KiA5Ywv//0Br uptrnXT0q0ubgAPvUFswQKs12V/cjMhR7FBulTWIw5rnk9mvzKvf4mT+Zp4C8ia9JSXyZN4SglwwJ61UQjqIN EXUbjr7g/NzDUn8SEiVQcSZPZQ8YEvOMBosvRBXkiKMdoM0hMkuPsw6psqCJyYp1/0XfRvPU9jPVPGNtXEzwBs zVgTiYQbXbycaY7Sp+OnkKaUEchIn8xARSyv/Vzt0fI5hn8Cx9K4k56305+KzKyUbsac3RVXSxsCHJb0nFQiK5 7Zcv1+CY7CFMQAgsW+rEaiy2vDT9Y1atb0yqqtG1Z2+W/qWZ6+ehGfoVoHcauEPqPAoP4GCv8ur//35unyduRC i8ZjVg2+xUWS0g4ncEFnHgoIhJDWGJgCjW2ZRL97EawzTxENvoQontx76rF4wHw08qKddLw4i/0852cdVQtdFO BHI9jdIL9wR30p5hMzC7pnkYMg5iNirvRMqw4AhVPpzG3XLPQw5z1R7zy/aJN1ojoRTf4jdmyRlw7Iro9bfr9 9nYNHDo/nVkdCBFP8QX+34ojvF226hgaSK69HBFUY0s0RYkxL1T4x9RMRH3V0qK19Q3rVQ3KJrLaA22KCp3GMK VZNNwtJfn12oy2y/ioHC+y9b889Yq75ykYapr9B+gkxCaW8b2LOVQCGEDoxojGurYVXsdNpTjC1xkmx+xY2ux1 Eom+d7Jka04HQpqEK4BhUztZYoHu5Mavpk0c4KqVoDcZuYznK6abCYBK/kc1CBo+hiommWuJe8/KYZ1+n8EEDs Te41cJA8ro3+zT7hZv4/MK+ItY3BgHK+Rupmj1m17hJjg01g7Eqj4uRi+vCCxeg3kzpANoMk8cHNMyPx4V/XRA kaw9iEpkh6JUpqSF0nzFbrmepkJH7qqRYNXBcwazHGvq60RiUeNaIioWJWKUxWe50IykXoEn0kh5B6k+y6T3y AOHotFFMuz3qRSzNE1Nv0V0UGP0j4dyiLzpD4iQwAxv9P/Z/B/w4DYxtTQycXe1tDj2uAt50ps4u9k6mvprngf wA0vbZhCmVuZHN0cmVhbQp1bmRvYmoKcjmzIDAgb2JqCjc4MjEzCmVuZG9iagoKMzQgMCBvYmoKPDwvVHlwZS9 Gb250RGVzY3JpcHRvc19Gb250TmFtZS90aW1idXNTYW5MLVJ1Z3UKL0ZsYwDzIDQKL0ZvbnRCQm94WY0xNzQgL TI4NSAxMDIxIDk1M10vSXRhbG1jQW5nbGUgMAovQXnjZw50IDk1MwovRGVzY2VudCATMjg1C19DYXBIZWlnaHQ gOTUzC19TdGVtViA4MAovRm9udEZpbGUgMzIgMCBSCj4+CmVuZG9iagoKMzUgMCBvYmoKPDwvTGVuZ3RoIDg3M i9GaWx0ZXIvRmxhdGVEZWNvZGU+PgpzdHJ1YW0KeJxd1s1u2zoQBeC9n0LLd1FYEmc4MWAEIGUJy0LeFk37AI7 NpAYa2VCCcRd6+OnPU30XiY1oafhzJ1Nfd3e5uPF3Xn6bz4b5cqfTeJzKy/110pTqoTydx1XTVsft4bq88/+H5 /1ltz7PvX97uZbnu/HxvN2u1p/nz16u01v1Lh3PD+X9av1x0pbpND5V77529/P7+9fL5xt5Lu01qle3t9WxPM5 1/ttf/t8/17Wf9eHu0H98ur59mE/5fcCxt0upWn/fkHI4H8vLZx8o0358KqttXd9W22G4XZxx+M9nTV3znIfHw 7f9tNq20Lau55c5N8wNcsvcIgfmgCzMgqzMihiyZ1LNua2bDfINx2+QN8w+npgTcmb0yB1zh7xj3iH3zD3ywDy vcBvoD/AH+gP8gf4Af6A/wB/oD/AH+gP8gf4AfzBmQ6Y/wB/oD/AH+gP8gf4Af6A/wB/oD/AH+gP8gf4Av9Av8 Av9Ar/QL/AL/QK/0C/wC/0Cv9Av8Av9Ar/QL/AL/QK/0C/wC/0Cv9Av8Av9Ar/QL/AL/QK/0q/wK/0Kv9Kv8Cv 9Cr/Sr/Ar/Qq/0q/wK/0Kv9Kv8Cv9Cr/Sr/Ar/Qq/0q/wK/0Kv9Kv8Cv9Cn+sea+ifqQ/wh/pj/BH+iP8kf4If 6Q/wh/pj/BH+iP8kf7o9emP8Ef6I/yR/gh/pD/CH+mP8Ef6I/yR/gi/sf+G/hv9Br/Rb/Ab/Qa/0W/wG/0Gv9F v8Bv9Br/Rb/Ab/Qa/0W/wG/0Gv9Fv8Bv9Br/Rb/An0h0cic4EZ6IzwZnEr4vf54n0+QU72rJzyebvjSz5ApqNF +ICeHaQuIDkhZYNyMcTLzygiQtIWEDqfNy/5GnHjEumLGDItaMHZsyVG9YB0reefTPNgRnHzzY9Y97MRmc00n0 j9E0237A05sobZjQrL040Md0Z4cx0+oabF6fPy5sj4+bolhsa4x0bndGHzhvddj13c1u7Qx+6pdF+buQ4Lnxz 0hJt/TTxxPHYe4Wm4/v0A5z1zPD3A2cF33YLV88eHo63dMvNwfq9j5X4xtiz2vnnt7nbXxz6Zd5cfzA0j3qDL6 Wxh8Gw1IHxw+sM6D0sNTBtRtYzzf8eZPhAYon/M8Hc3V4nab5oew/A/xpj0fwaSzVr58K1/MFp/nfD5Io2CIKZ W5kc3RyZWFtCmVuZG9iagoKMzYgMCBvYmoKPDwvVHlwZS9Gb250L1N1YnR5cGUvVHlwZTEvQmFzZUvbnQvTm1 tYnVzU2FuTC1SZWd1C19Ub1VuaWNvZGUgMzUgMCBSCi9GaXJzdENoYXIGMCAvTGFzdENoYXIGMjU1C19XaWR0a HNbMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCA wCjI30CAyNzggMzU1IDU1NiA1NTYg0Dg5IDY2NyAyMjEgMzMzIDMzMyAzODkgNTg0IDI30CAzMzMgMjc4IDI30

```

Ao1NTYgNTU2IDU1NiA1NTYgNTU2IDU1NiA1NTYgMjc4IDI30CA10DQgNTg0IDU4NCA1NTY
KMTAxNSA2NjcgNjY3IDcyMia3MjIgNjY3IDYxMSA3NzggNzIyIDI30CA1MDAgNjY3IDU1NiA4MzMgNzIyIDc30
Ao2NjcgNzc4IDcyMia2NjcgNjExIDcyMia2NjcgOTQ0IDY2NyA2NjcgNjExIDI30CAyNzggMjc4IDQ20SA1NTY
KMjIyIDU1NiA1NTYgNTAwIDU1NiA1NTYgMjc4IDU1NiA1NTYgMjIyIDIyMia1MDAgMjIyIDgzMyA1NTYgNTU2C
jU1NiA1NTYgMzMzIDUwMCAyNzggNTU2IDUwMCA3MjIgNTAwIDUwMCA1MDAgMz0IDI2MCAzMzQgNTg0IDAKMCA
wIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAowIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwCjAgM
zMzIDU1NiA1NTYgMTY3IDU1NiA1NTYgNTU2IDU1NiAxOTEgMzMzIDU1NiAzMzMgMzMzIDUwMCA1MDAKMCA1NTY
gNTU2IDU1NiAyNzggMCA1MzcgMzUwIDIyMiaZMzMgMzMzIDU1NiAxMDAwIDEwMDAgMCA2MTEKMCazMzMgMzMzI
DMzMMyAzMzMgMzMzIDMzMMyAzMzMgMzMzIDAgMzMzIDMzMMyAwIDMzMMyAzMzMgMzMzCjEwMDAgMCAwIDAgMCAwIDA
gMCAwIDAgMCAwIDAKMCAxMDAwIDAgMzcgIDAgMCAwIDAgNTU2IDc30CAxMDAwIDM2NSAwIDAgMCAwC
jAg0Dg5IDAgMCAwIDI30CAwIDAgMjIyIDYxMSA5NDQgNjExIDAgMCAwIDAKXQovRm9udER1c2NyaXB0b3IgMzQ
gMCBSPj4KZW5kb2JqCgozNyAwIG9iago8PC9GMSAznIAwIFIvRjIgMjYgMCBSL0YzIDIXIDAgUi9GNCAzMSAwI
FIvRjUgMTEgMCBSL0Y2IDE2IDAgUgo+Pgp1bmRvYmoKCjM4IDAgb2JqCjw8L0ZvbnQgMzcgMCBSCi9Qcm9jU2V
0Wy9QREYvVGv4dF0KPj4KZW5kb2JqCgoxIDAgb2JqCjw8L1R5cGUvUGFnZS9QYXJ1bnQgNiAwIFIvUmVzb3VY
2VzIDM4IDAgUi9NZWRpYUJveFswIDAgNTk1IDg0M10vR3JvdXA8PC9TL1RyYW5zcGFyZW5jeS9DUy9EZXZpY2V
SR0IvSSB0cnV1Pj4vQ29udGVudHMgMiaWIFI+Pgp1bmRvYmoKCjYgMCBvYmoKPDwvVH1wZS9QYWd1cwoVUmVzb
3VY2VzIDM4IDAgUgovTWVkaWFcb3hbIDAgMCA10Tug0DQyIF0KL0tpZHNbIDEgMCBSIF0KL0NvdW50IDE+Pgp
1bmRvYmoKCjM5IDAgb2JqCjw8L1R5cGUvQ2F0YXvzY9QYWd1cya2IDAgUgovUGFnZU1vZGUvVXN1T3V0bGluZ
XMKL09wZW5BY3Rpb25bMSAwIFIg1hZWibudWxsIG51bGwgMF0KL0xhbmcoaHUtSFUpCj4+CmVuZG9iagoKNDA
gMCBvYmoKPDwvQXV0aG9yPEZFRkYwMDQ2MDA2MTAwNzQwMEU5MDA3MjAwMjAwMDVBMDA3MzAwNkYwMDZMDA3N
D4KL0NyzWF0b3I8RkVGRjAwNTcwMDcyMDA20TAwNzQwMDY1MDA3Mj4KL1Byb2R1Y2VpPEZFRkYwMDRMDA20TA
wNjIwMDcyMDA2NTAwNEYwMDY2MDA2NjAwNjkwMDYzMDA2NTAwMjAwMDM1MDAyRTAwMzI+Ci9DcmVhdG1vbkRhd
GUoRDoyMDE3MDUyNTE0NTcxMyswMicwMCCpJ4KZW5kb2JqCgp4cmVmCjAgNDEKMDAwMDAwMDAwMCA2NTUzNSB
mIAowMDAwMTM1MjE1IDAwMDAwIG4gCjAwMDAwMDAwMTkgMDAwMDAgbiAKMDAwMDAwMDU0OSAwMDAwMCBuIAowMDAwMDE
wMDAwMDAwNTY5IDAwMDAwIG4gCjAwMDAwMDAwMTgx0DggMDAwMDAgbiAKMDAwMDAxODIwOSAwMDAwMCBuIAowMDAwMDE4NDA0I
DAwMDAwIG4gCjAwMDAwMTg30DkgMDAwMDAgbiAKMDAwMDAxOTAzMSAwMDAwMCBuIAowMDAwMDI4MDQyIDAwMDAw
wIG4gCjAwMDAwMjgwNjQgMDAwMDAgbiAKMDAwMDAyODI1NSAwMDAwMCBuIAowMDAwMDI4NjIxIDAwMDAwIG4gC
jAwMDAwMjg4NDIgMDAwMDAgbiAKMDAwMDAzNDQyMCAwMDAwMCBuIAowMDAwMDM0NDQyIDAwMDAwIG4gCjAwMDAw
wMzQ2MzUgMDAwMDAgbiAKMDAwMDAzNTAwNyAwMDAwMCBuIAowMDAwMDM1MjMxIDAwMDAwIG4gCjAwMDAwNDQzN
DEgMDAwMDAgbiAKMDAwMDA0NDM2MyAwMDAwMCBuIAowMDAwMDQ0NTU4IDAwMDAwIG4gCjAwMDAwNDQ5NDQgMDAw
wMDAgbiAKMDAwMDA0NTE4MiAwMDAwMCBuIAowMDAwMDUzNzg5IDAwMDAwIG4gCjAwMDAwNTM4MTEgMDAwMDAgbi
AKMDAwMDA1NDAwNjAwMDAwMCBuIAowMDAwMDU0MzkxIDAwMDAwIG4gCjAwMDAwNTQ2MzEgMDAwMDAgbiAKMDAw
wMDEzMjk10CAwMDAwMCBuIAowMDAwMTMyOTgxIDAwMDAwIG4gCjAwMDAxMzMxNzAgMDAwMDAgbiAKMDAwMDEzN
DExmIAwMDAwMCBuIAowMDAwMTM1Mdc3IDAwMDAwIG4gCjAwMDAxMzUxNjAgMDAwMDAgbiAKMDAwMDEzNTQ1NyA
wMDAwMCBuIAowMDAwMTM1Ntc2IDAwMDAwIG4gCnRyYWlsZXIKPDwvU216ZSA0MS9Sb290IDM5IDAgnUgovSW5mb
yA0MCAwIFIKL01EIFsgPEM1NDcyMzRBQ0IyMTJCN0NENjVCQTK2MUY5RT1BQ0FFPgo8QzU0NzIzNEFDQjIxMkI
3Q0Q2NUJBOTYxRj1FOUFDQUU+IF0KL0RvY0NoZWNrc3VtIC80NkY1M0U2RTZFTNTBEREJFNDg5QjBGQkJBN0FGQ
TQ3NAovQWRkaXRpb25hbFN0cmVhbXMgWy9hcHBsaWNhdG1vbimyRnZuZCMyRW9hc21zIzJFb3B1bmRvY3VtZW5
0IzJFdGV4dCA0IDAgUgpdcj4+CnN0YXJ0eHJ1ZgoxMzU4MDkKJSVFT0YK",
    "digestAlgorithm" : null,
    "name" : "sample.pdf"
}
}

```

Response

```

HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:39 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block

```

Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 209187

{
 "bytes" :
 "JVBERi0xLjQKJc0kw7zDts0fCjIgMCBvYmoKPDwvTGVuZ3RoIDMgMCBSL0ZpbHR1ci9GbGF0ZUR1Y29kZT4+c
 nN0cmVhbQp4nGwTPWvEMAY69/wKz4Wkk13DhhD7y43dCscdCjd+rEV2qV/v5LsHlmUgHAkRXr0WoEJ3e/w7cC
 BnHjmybtMOGX38z4837kviaHEvGT8fA6Hy+DzxC6DF9f1zd2f0aGcP14K5colch2xxGMdvVpxnXhoY5UOFxh
 cnXvKZRzwgloB6Dtwwmmy1Ai5aalf0UW9w6pWBxDtE+jcvGScf6enkclsVwpBNOJDPgdoqU0xTbFH6dArBiAes
 AQXcbEUQuwBAr76FAkmNBXMdkwZn9Tyo0WngNGS6r+6SeVnDRdJ3fF4SaYkEERq/ZZxUIA9INs1BHU39Lnbcj
 1GHDtvJE+EZDXk0ZmzI1M+pB3ILzDUqrnJ1XnOfJM/LswgSKqZg4a2QvokpaFssnteVoA0Wdyz+p2RK69uN1PN
 eyrFpGZqWej1XHYPyRQMUB+s9iJpeU7Qmas2dogYfd3tAvKLzB121srVZ7IE0QeG0NeseSBe944au2PAABzgaq
 F+vu9WwTKHFDmsxE1rbHSEdSxDLfttj0/+fMIWwrm/cra8gUWmGuqRtD5QDeyTb8Sq27YHKbH+lzUDKGfSzU/s
 6tDU7yyBXuif3B+aa6fYKZW5kc3RyZWftCmVuZG9iagoKMyAwIG9iago0NTkZW5kb2JqCgo0IDA6b2JqCjw8L
 0x1bmd0aCA1IDA6gUj4+CnN0cmVhbQpQSwMEFAAACAAJme5S17GMgwnAAAAJwAAAAGAAABtaW1ldHlwZWfwcGx
 pY2F0aW9uL3zuZC5vYXNpcy5vcGVuZG9jdW1lbnQudGV4dFBLawQUAAIAAAmZ71KScopFokAACJBgAAGAAA
 FRodW1ibmFpbHMvdGh1bWJuYWlsLnBuZ41QTkcNChoKAAAADU1IRFIAAAC1AAABAAGDAAAawv3H6QAAAb1QTFR
 FeEsef1MoWrx6ZcGDbsWLdMeQd8iSesmUgFYshV00imM9jmLE1HF013RSmHZUm3pYn4BgoYJjpolsqIxvqo5xr
 JB1rpR5sJZ9sph//5F1/5N5s5qBuKGJvKaQv6mVs4H/u4//vJH/pLrRqL7Thc2dis+hjdCjkt0n1NSqmdWtnede
 wo9m1ptu4qty7msz/sMPXtMbZtsjau8zcsd/Aoc//o9D/rtb/sNf/s9j/vN3/t0HDueLGveTKw7CdxrShyLakz
 Lur/5uD/5+I/66K/6qV/62Z/7Ge/7Si/7yrw5z/xqD/yab/za3/z7D/0bT/1rv/2L7/1ci6/8Ky/8e5/8q9wM/
 fwNdf2s7C3dLH3tPIxNLhydbjzNj1wN7/2sP/3sn/0Nvn093ow0XmxujSy0jS00vZ20/fx0H/yuT/1uDq2+Ts0
 0b+10n+2uz/3fHk/83B/9XL/9j0/93V4c7/5NP/6dv/6uPc/+Pc40fv7+r150vx40//7eH/603z6/bv4/D+7PH
 06vT+8e3o/+fh/+nk/+3p80b/8+z/9PHu9PTz8/X69Pr29fn9+/b0+PT///j3/v7+AAAA///LxzPJgAABItJR
 EFUeNrt2I1T2mYcB3D30rPacU5bWZ3arU5Bypg10z2x1L4ggpuXZJReXWBxTDdCTBQ1VJcLBVo1F5Lq70/uA1y
 3219A3H3z5HnJ7w13nzs5wpPLAF3HbQBqqKGGGuprpj6uEFVYpmzuKEt0dkYV0nqf0e/0nWSPz1hV6Z2ZybDi6
 t/jPqrfJ5PHEx0HL9KUaeePyvJDM01zyuZVPrZ+XEeqk0snN551cr1U1jLZykQ61U1Rrv/qyuHJxvphLk3H68k
 X6eNcNp01w6NnuVQm+e7k3deHbGNXsp5LUXoj+zzDuohdgU/n9dm5r+e1zRLZdrfV71VsNy4to9dvVD/Gup2yL
 9SXoi3yF6IgGiTLoshXt3ib52nHqskCb9cMEm1Br1YtWaZNqtFWzRIMoSbU+qzeF0/zhsjL1lk/b4nVTUOW2zz
 VLFvg2/LmDokk3BdE8dva1ualyMb6J/4bEnijzzPkgnqyZVkyG72qcSkbtUp1lhTFrdE26qx02AbVcHgSb4QL
 doRa4JM1mz5dpVp/906tH32NHqJvTfubsylg71wou1S4s0fdBBLRdyXdyZ2ySWPVmN/uuSQ6x+10zgbGlydC0f
 c0GwkEg5F5mZdirqRUCQco8Hd0KfRz79/FH1wIzQfmXf9o15NRGM/hsNRGjwIReajj0JMHXsQm4vMx4i1SGI+s
 Rr7LDS7t3rD8f3b01/+fnvKs0T1YrdZdn5nVz1IpbJTzPdOKBRMoiL1YusNAtqJ9gsqn1VLxQkiSsU61KR8py
 0KEkPOVOSSDXb5TxXeC1RgaQCJ6mLXkHK56mc13P5hWJ+xuurrWiJpgVnLp8xXmJFMTpWKK10fqpxaZNdDU1PiH
 krcjCMVpEU24tJ3HPvNy/70EJNmp1k3PbPzN+jRad0sU3evM3S7zrpVaktc22y3qWmSp1K9rXaCfVUzaTezRaV
 Teh9jvWXmP/OgF/b88DSurLT0Ukn3WooS11paSY9TvBVv6A2dVkhbbumNh1ZSPF1r0UpDa2j6D/5Q7y8rbKNSv
 BRXtE5LIVZprPYUpbS0oi3FWa100kvLLOQH9b7e0jRWxDXqrF0WG0udSg31N03TidHZ60sNRVdaue4oWkNxfo
 DeuSxQ60v3zrB7bEr9+3a07VutM5ykJwrch7T9pXvVpnXAzenp28FpmhoeHQoMH5n6Ndxt16NTAbp176bd4bvj
 g7fDgw9CTh+U9+dDgaDX2zT9NTU15NPJ8fo70++GhqpnsjwbHx8cDoZPD2k8C9a/AVZ+1afntyrgua3/mghhp
 qqKGGGmqqoYYaaqihhhpqqKGGGmqqoYYaaqihhhpqqKGGGmqqoYYaaqihhhpqqKGGGmqqoYYaaqihhhpqqKGGG
 mqqoYYaaqihhhpqqKGGGmqqoYYaaqihhhpqqKGGGmqqoYYaaqihhhpqqKGGGmqqoYYaaqihhhpqqKGGGmqqoYY
 aaqihhhpqqKGGGmqqoYYaaqihhhpqqKGGGmqqoYYaaqihhhpqqKGGGmqqoYYaaqihhhpqqKGGGmqqoYYaaqihhhpqq
 Di3IN/f+8NVAAAAAE1FTkSuQmCCUEsDBBQACAgIACZnuUoAAAAAAAAAAAAAALAAAAY29udGVudC54bWzNWVt

zmzgUft9fwdCZfSPYJnEDG6cPm+10Z9L0Tp09vCogshafErCdv59j8BqQY6ocezzFyfof0eqc80++bDLqbPBQhLOVu78YuY6mMU8ISxbuX88fvSu3Q+3P93wNCUxjhIelzlmyos5U/DXAW4mo5q6ckvBIo4kkRFDOZaRiNeYNZwRSY6qnTVJ1I9U2v2CmxyK7xTtswa2+FFT/aaK7DjnQi0tWXWWAiqyZ5yW+adpF7K1Iep5gRTpWbGjhP27ctdKFZhvb7fb21wwUXmz8Mw9Ctqa3Dc4opS0AqVxD6mWCuT/vxi7jfYHCtka5/GmjaxMn/Cwjo0SKHvb1VuMuuM2GQjoYnXSFjnRgXuXm+Q2F9vkJi80VLrkTu59j8Dsfr4fH/IBZHb6tLYTqhiQQprN2u0yc85b03VDHWBvUyuZrNLv3420NtJ+FYQhYUBjyfhMaJxG3GeDwUNcHMfEB7e6DRTe18HQo4wLPya3IJlMir678/3D/Ea5+gAJi+DPcKkQuwQGaEvYdTtk1/gggvBbia1b5hwW4vWtrXK6Xi5a2oDzUSSDELBNMCH0ofC8zYEb9+5nU4+nRBhLyGqtvgSSwUy++Ykw3zma0xbxpAihyYvsnY0pbxk4ATMn0A8a7AgmgSohVb1JFgZj3lrxC5n12GhE5Dj5g2HaN1aVAM514uIYGgUHgRGdzd+SDynZ04XRw8SfsSe40iljJQQ/nw+NXXNE+PRxgAe03GWrBwb5sdo04f0m8PUtgFvBTF2EtTwTOxtTd3L220nftZ2r9x7vibKucMbxFCGBJm7DrTvBpwT+rxyf0Yf17/0kfWp60/Lv8P/oD9L5wEx0SrZw0yF0obIgqqYWvcGVFY5a6kQb1n9UaMGHa/yo8B4xEGD9GrxX0j+VMoqPM79qEsd1PE+xWsecwpxIBCk/ZqBfcENpKqrl+4qz5wwLeawcswAwBMMN8FzxN7KuKnM7eEsTJNbIo/Khn4Z/mi9Thr3LBX0j7HugZcCjj6h0iD0r1+crzgrKRKjdo6zvJnF/1hn3J+jukHGKBj7lZy2ZVafHf9+n7e69k4USKBMoGLdE0BAv5FVD94+KrcWJEGkb1NY93WvgMGAhSJY0u2Yj6DDw8SCvQzdPs1S16C0irweRjto2Dth/0Ikxhux7/fDM/kVnMevXiM+k30XJ3Eu5RH0cA7vh0/SNAhmM/f/4u/Vqf1dLmezNB3wt70rnMnb5am9hWU/DIIBb89epe9P7WoQLJdh0HGx9Z5xJnevT+9uGKbn8yc8fV209Cfj33xbX/3R+b8nPPhkuX3QphsvWSLX1sML3YE7hXdWuXJTRCW8W9fEeoB9JcGSZMzLeY1bMhhTffkp8X81ZoeV5PtDpzpKiCwoevZ4qShh2KN4gyk459bk01K/UVpKVW+kwxp+QNhjsz4dJwX+PVrIXf11bXWD41ErahYz/2B63n7C1HLnLy5o4swXxc7pzLW9wGJcxWADGNWWHeYDkho+q8F+/sJA3RPsxBxPSCi/2Z1ISYcENOWpwV/YK7jdUH5nVprnvq/bNx+A1BLBwi3Yk211AQAABoZAABQSwMEFAAICAgAJme5SgAAAAAAAAAAAAAAoAAABzdH1sZXMueG1s7Vpfj9s2En+/T2Go6L3Rsry7zdoXpw8pihTY5IA2vdeC1iLDSUKJGwv8+k7JEWJkivbWW+B05zzsIE4fzjz48xwSPrtj885m+2JkjqXmyCaL4IZKWKe0GK3CX7//DN6DH5894+3PE1pTNYJj6ucFApJdWREzkc4kGtL3ASVKNYcSyrXBc6JXKt4zUtsOKG1z702U9kRo2yqUH2pRV5V10FNW9HFm+nz2yYfe1e4MNUYc0LmPrkZ8q/CwZsjmKeV5iRXtWPDNafNkEmVL10gwPh8P8cDfnYhdGq9UqNNTG4LjhKyvBDFcSh4QRpzKMo3kU0t6cKDzVPs3rm1RU+ZaIydBghU9WVe53kyNivxuBjs6wmBwbhmr7vHfJ90W9S3zHKtsZE0ew49ANH8+PrWxIPKpc2neD1Sxo0VKnY23L885b0zVAjZBjbnLxeI+tN8e9+Es+0FQRYTHHpr91jzGLG8R5PgQa8EUhcCcY12HquIV2e1TzQyhIyYVqDEmnFyhAZ9mkV6ZyNp5emupYdyJJBlnBnLsQUG0CHe0p0XwXdCrn+QVY9RbA1KFLIobJr1NnBaJFqHmatIE1aYuq2DV1P+VVAU7AV1EDSJ5LIqgmYWbE1h0NfpQx/gKV9V7haegUUEqYy9DGpUE1nKnc1lPAPJy7U13qoWUd2po+T7/Gmoa0rsH1Md6Hm/TXAbv3A6ZctgdUxwT1JCYyXdvbWVrhmF2Wxu5CZ54RtXsJ7LHBd5hQaNgBsXMMeeUHTfBP3HJ5b/6nHY0CM/r/4n8if9TzX7DhRzV7PHUSmeeypKqGAzHqY0ETVxQoA+vTijZrp+yp8FISM0eqQXq/9E8201DTyzp1GX01zX+xRnPOYMNtWBCX3aiy4orA/m6y7sFZ9xgHfrADakQIYoNYLnuPitYw7F7k9vgmmyQ0VV0VDPw2/NV/PGneUiuTXWPcbrwQMfcAwkNn7T7Nfy5iWIza0S7yahaHY5WxHrdHCedZQ1JcsfqA4TTXJu8ELjMaB463/kY1FHUiFIUDiXZTKsG/EGgJGYce9Lu7+x8e8H0w0zscFH3GGsqb5SqnIfZTvj6AKsRLZQp6wZH+rkvkhN+QGctJAo9b4LF/C70B4nHH1FBR4mgASdI1j1g9h91XNCvXG9KmnX5eJZ5r52KT1mhY5mq9YR1QGcNMgM/D1R1yB6oUsykF3E1Ftjg3UHbkDQ/wpXieg4IQ5oQblkxK70mThkztoJgOKzAEtFY0YruGLVtOU9AnAmktp2gokVcdJumD56+M85IZyM0PRA3vJQ66sbNbt13SfeVJIApVeVTN5HSxKVkrj1FghhWAdEm/Aj1alsqMMVzsKryDoawyAzE0PkpAxHz4vUGIKGhQ0BciCuPd4JwI+ldcXEj2Vksb4UsixfyhbFB3Fjnq18xRatMc4f2nUyP0eYuR58GK2Ju8YbVY9CZvqBntT9+QfvkUtvCvaKRBTqkYTAshZWIY14QLCtZCbYGlraoUgaL6yY5mRwiw1YjhJYKGmoaZcMJrTxreJ+VBWRawqq1CXG4AAQIGFv5wWltBRQqFsfHoSaEIfojabuy1VAtJtKt/i/n8n7v3II67e9cNRkBzTAukLDBeTpzFbVjLrsVvRVPZ86RVeRvyQsld1WY50mugYhP0GAorhUuoAv3ZiJPihNzmM9LL5CyE1UnxHVkbvOnRCXprYn7BuqSC9EiySYLSouMVjWEKt0LnVZtqpgv8EJ16Kj6qDgeaaFQ2bUug09hk+w8AfY8UfW54ch8y6VP5yLKD8AGS1bg7u16Y5aMe3XC19PwN9Q7TUJHsejyHg4f8Ks/623Cyc6SoK01VgdsBHeakWjRQacxhwdebvPwv0S9p9m8YvrUq1EbXQ1S24UzTWiDt669jYhRz0bVqXq01qmF94AGoVjHrTcIz70xZd7Vgb7hfSfCh1Yg8tGT56yTXzydek7ouzjcjQ799MzUjTVWSE7jK1t6bF99NReoJu4yXuny1MTKucUAw6wbr8pmCNXh6tk5F5j83R8RVjA6AQ+BtLd12Xrynz5oHMPt+Yo5bsUkzwuNedXxgZH2Kg7z1WgK4YTqtwlp+8yBazmjog/LcFwbTGzr0xoddWTo6iX+CI+/yKMUSNvrMx9Er/r/d+XqcZ8X1l7X3AC9b8tJegxMrInrGa3A0gBcKRpuqoc6fc3DMWzQVYXsVq0D61P4pKaThzWyuQdcRceeAtkCJVdod8rhtT2WGrdZjCFtpofSIK2R1t0oR/24Bi0Dahjbk8EZHkDpAvI3Q2Lid3N0C6gDzcA0kC8sMNkC4gb26AdAF5vAHSBWR1A6TXmC3+jxDpknyYCq6IhFa5S0muqu/BGgKqjwMp50p/DyEW1Q2/fYXcY1YR3fDbQSc0G6TqJwVfwp4S9JuD1ud+3qP9nW4hKZIxAmwugU69RqS1YGi0T00fbs1V56rVfuCMoR0raRFgZFU1TRaxML8jFLv9N6zt9HWvnbra2jQSWPkC050toPVxkdY3c5B5W0ZR8EAU++0bSgHmuhfHS6j+WIR1TcBhuCue5ar+ZtRD+spAEFuKD61P1Un0hBKYq6N84RM03Db1hjdHJoKjvn8ZerHqPwDb+UI6fG1/0XVj7c4iaQZLSqbNYL0aL6LGdxL2XoS0Bzw2/5okW0QAPTvWr1BBLm3+bQHJGm4M3Tv6spLJRYGPdJgtIYrc6D9+3Dz72dwgL8y/wX62H1t15mxGsn1XMR

+hD4A2eKmoj8jQEa0K0Za0jma0e1Jr0voz4Nnuh62VCT304/Jvyd38BUEsHCLzDyaM6CAAAky4AAFBLaWQUAAg
ICAAmZ71KAAAAAAAAAAAAACAAAG1ldGEueG1sjZPNjpswEMfvfQpE9wrM4BFWKmHPW3VSk2lqpfIsWeJW
2Ij2yzpI/U5+mI1XynZ5NCj//Obmf8MQ/14PjXOKyjNpd16oR+4DggqGRf11v26e/Jy97F6V8qXF04BM07Ewj
jncAQx6YKjafQ1u2UwJJorrEgJ9DYUCxbEEsKxtN4bDQp54aLn1v3aEyLEer73u9jX6oahUVRoDG6oIxeuLZTz
UgxiqCBoYNGOr+ihR0c/q+pgV1bk1JeGg34ZHpsFwVBgqb3QteKsebeAJaNkXVIDPFeOfTvXWcef7XwyK2W7Q4
2qnI0wwU3nDQeVUCMVNUTMX9+K+e7l0p0V1kShxftrRnu0IVBWHmBakXpbswwUmBg9zfRHEQREkSz3WuM0pG8
Z3UNMNh7GdFkWabII9KtGADf9/kkjBZA2YNI9pjnRobVp934cc4/TL7uA1fZ9FftAFdRW/oWZ7Yy3lqY0tow6k
z6oYcGvCo7ITZuvYrT/s7kfpGLIcfQM1btV2R4UVTpFakPS6BzRzopWKLfhWzSI8WpwbUpUy2FBJ29f2RG9Ato
bbNDZnkLpoHrEGAG1f6zA8KPo13g1I/8jM/enjmojvvv+Wb/SZxVsC+VXIYyx7kw4eON8xbtvivXomujhDd++G
rv1BLBwh4qlPx5wEAAC4EAABQSwMEFAAICAgAJme5SgAAAAAAAAAAAAAAwAAABzXZROaW5ncy54bW9yW1t32
jgQft9fkCn7SkJC2nASegwpbRoSOECbs30T9gBaZI2PJAf49zuSTZoSvE1ttE8hvsxIM983N/nq4zoWR0+gNEd
5XTt9d1I7AhlixB0X8uvZt0jv+UPvY/usKZzMeQivCMI1BmmMNxtAj+ohe17qV3b6upUq2kGmuW5LFoFsmgGEcc
vta6+XTLacsu7IWXC6vawtjk1a9vlqt3q303qGa108vLy/r7u720RDLjM/fqip7+qUqRHxWZF/IFuOUNU50zuv
Z/7WjfJEvTN0otbd22G6/fZUryP4ccw0xtc1Rftku7bpGK1tPHFbPVqvte+/Xd77T84ECNsGktr1jNgndESjnt
fbJvF21iLeL7cPM+JD7yC0z2Cu40Tg5b1YT/gX4fLF/2afn56en5aSPF7gaQUQYg+6CyTnoHQ1TRAfM1tpGpVB
0x63sKFxpuMcIiqtPmNBvFn8cs+SYwjWEL021n6AuXeIGmrzNpFrjtL1UZxa2yL5UZ5VxZi7/1Z87K82CKmn
DUbJZHxnWs+FXB4rjixB2e2kzoqpIgl4Fk12R00BuMi/p2U9N0PxHhConbBtkBlyHsVcMY2mJouijSWu5z0pZc
Gx1Z6B3F5MFk/tkuPhQZVwdqb70sGIz0GAaGBoKf0Qom177n4MrgU3c7j1f4HKDe+PZtmF1LFD0XmP0mrQ4pgZ
qhQJ2QAD05z8m3AGDIviSRbP8nusHA5V5jK3SB9KCVdJGei8LWHQEo0zn176JMjvGQkdPI/K5Yse0hr+S7M+vR
xj60fXpMzz8D1NwM1AdVTG1/BpLvZ5mBwn8DaDAULYYEigt1odQg1QZK1zZApNrdevWdq2UMVmzPBhzSegr0Jp
82NwHYAUmiyCvInbDog4bSd0Zce5H8i00Q9onEhRisIH6fKRR/r/EfyjhmQnjQ8zPOWF9be90otnLR/vDKuoI
nCURDHppUFV07goaeQGYMo2LqAWPneB9avskptUtLp2ZLEB96HMLGG003f0EsJ81M4MpTpJ1YM/kK420yvICv0
PXCQcu5Lkv0EFRItRelo4sLH7RAIUgLfIoTs+mCEJaKhzfYhDCbGywr2jsWxIF25Y796WFr98ws0kyD7f4Dwef
Szo40v7UgymrtwWxGxaoPLNzqPpuCuMnHX77SIahKVckgNdbSfUqq4m80ItI+E/jtXKKCH1fa9EnrLbUf0txKn
yq/JREzYEuuCcSJ0n+eAkuHCq61x6ArYISr0wAfVaNLGxFEN4wyIZGvEqQyJ90KdSAjsopcaio0rb+7TISpcD2
GD09rGIgoJ8U90HqLm40q6JaU/UwHFVXW44SFmA7Zk/wWeCUPQcQMuDSE7T0CKxXnmCCGSUPT5GR5rsVO/mp1
j790GxeND5U6b3z2cFo0vcVY12a/c/esmorRdhVlnyeYtMXTuyWtjPwsYHIdQIuROWECMzYMOUBRj2booLUYBb
WD68gCyI+E9Iecx1+G90FqQgpDnYtTqhxsidmxSPdkmMdG60+Z6dxA9kVqH0MXe10jE+P5wnDbj57WmkjrRskF
xy1LPh8cUwWpXrY5rGyteEdKBlozuQwldTbesyJdjADajD9h2LkEDW3mrwhm+BgeMiE59p6nE6NxXdPbKjICAn
FEAXuP3+Ezbk6Qcdc23R5aFFJ16c1oVEy0QcW+YkNqVKE8m2tZ/+OMVvH+yovEAJXjkfUV3eZDKFw8FSFp2gIw
hbOPoli61R/ZZ0z0u/mZZWyV7aL7RAw8dRUyTXpCeEHKCS8/gaq1YrTwJ3eZHUpKX0e0utb2efaR0Gtf3uQA47
C5R1sdtUQcS7001wytXnb8B+TF8gaL5St5p8bYQ8kR0kV908gDCR1AAr/PsxFikaIu57Iyvv3jbPTZtnvQmXgg
H7AG5ixVPjw02psrqi+Myr50Ld7Irg0zaYbNPu4d18gw9ovqba8NnGakk/cr04ZzJlwteAzqkcAWV2S8XAGGX
be6qY7ezDRwH4a7qigjZmMtpTzDrA1Sx1+8iiEUuXl0IV1Q9lsy5L7CGGjf7ERT2QtqP83wxW7dwS1N4Dyz+Mg
vY0J9DPvcm2cntU9Kii6s1VjWVCVOHnC/VXnwfwiz6cbP8LUEsHCCh81F80BgAAeikAAFBLaWQUAAAIAAmZ71
KAAAAAAAAAAAAAEnvbmZpZ3VYXRpB25zMi9hY2N1bGVyYXRvc19QSwMEFAAACAAJme5SgAAAAAAA
AAAAAAABoAAABDb25maWd1cmF0aW9uczIvdG9vbHBbmVs1BLAwQUAAAIAAmZ71KAAAAAAAAAAAAAGAAA
AAEnvbmZpZ3VYXRpB25zMi9mbG9hdGVyL1BLAwQUAAAIAAmZ71KAAAAAAAAAAAAAGAAAEnvbmZpZ3VY
XRpB25zMi9tZW51YmFyL1BLAwQUAAAIAAmZ71KAAAAAAAAAAAAHwAAAEnvbmZpZ3VYXRpB25zMi9pbWF
nZXMvQm10bWFwcy9QSwMEFAAACAAJme5SgAAAAAAAABoAAABDb25maWd1cmF0aW9uczIvc3RhdHVzYmFyL1BLAwQUAAA
IAAmZ71KAAAAAAAAAAAAHAAAEnvbmZpZ3VYXRpB25zMi9wcm9ncmVzc2Jhc19QSwM
EFAAACAAJme5SgAAAAAAAABgAAABDb25maWd1cmF0aW9uczIvdG9vbGJhc19QSwMEFAAACAAJme5S
gAAAAAAAABoAAABDb25maWd1cmF0aW9uczIvc3RhdHVzYmFyL1BLAwQUAAA
IAAmZ71KAAAAAAAAAAAA
AAAAADAAAAG1hbmlmZXN0LnJkZs2TzW6DMBCE7zyFzC7YQC8FBXIoyrlqn8A1hlgFL/KaEt6+jpNWUaSq6p/U4
65GM9+0tJvtYRzIi7KowVQ0Yyklykhotekr0rsuuaXb0trYtisfmh3xao0lnyq6d24q0V+WhS03DGzPs6IoeJr
zPE+8IsHVOHFDMa0jggJHo1CafXkfBo5zuIJZldRd0ugkHn3ID2L3TqpoLIKYbZSvYe2IJGBQI0JTMqEdIMcu
k5LxT0W81E5waHt4sdgv0Dojxg8Cu0z9jeiAym5V7gvbDuXIPffJvoeu5jenXTxfHfI5RgnDLuT+q703n/5/4
uz/8Z4q+0dkRsQM6jZ/qQ57TyH1Vhr1BLBwi092jSBQEAAIMDAABQSwMEFAAACAgAJme5SgAAAAAAA
BUAAABNRVRBLU10Ri9tYW5pZmVzdC54bWytk0tqwzAQhvc5hdG2WGqzKsJ0FoGeID2AKo8dgTQy1ijEt69skti
1GGLwbp7/fBpGxfHmbHaFLhiPJfvgywD1L4y2JTs+/yVf7LjYVc4haaGQPjhzKkPw9MtWexQehVMkKgcBEla+
haw8jo6QJJ/6+U46enNAPbssMumebWxkKf+rp+q62ht3iq61Ews iUxhB5VR0fUtlEy1rTVaUSoTV6z4CMznnJz

gRkysYThfovBZWwQ9DB5i80Cg3GqATHkV03RHmngS3tcEB7IxZBepRuotxA213VAantWIEqHuT3tyWNtmtiN1xH24sXzCREHFB4N130F1au6x3hX1S8MT1Vv9wcW4t+/PPwCUEsHCJu20BwMAQAA0gMAAFBLAQIUABQAAgAACZnuUpexjIMJwAACcAAAAIAAAAAAAAAAAAAAAABtaW1ldHlwZVBLAQIUABQAAgAACZnuUpJyikWiQYAAIkGAAAYAAAAAAAAAAAAAE0AAABUaHvtYm5haWxzL3RodW1ibmFpbC5wbmdQSweCFAAUAAgICAAmZ71Kt2JNpZQEEAAgQAAcWAAAAAAAAAMBwAAY29udGVudC54bWxQSwECFAAUAAgICAAmZ71KvMPJozoIAACTLgAACgAAAAAAAAAAADZCwAAC3R5bGVzLnhtbFBLAQIUABQACAgIACZnuUp4q1Px5wEAAC4EAAAIAAAAAAAAAAEEsUAAbtZXRhLnhtbFBLAQIUABQACAgIACZnuUoofJRFNAYAHopAAAMAAAAAAAGgWAABzZXRoaW5ncy54bWxQSwECFAAUAAIAAmZ71KAAAAAAAAAAHAAAAAAAAAAADWHAQQ29uZmlndXJhdG1vbnMyL2FjY2VsZXJhdG9yL1BLAQIUABQAAgAACZnuUoAAAAAAAAAAaAAAAAAAAAAABAdAABDb25maWd1cmF0aW9uczIvdG9vbHBbmVsL1BLAQIUABQAAgAACZnuUoAAAAAAAAAAAYAAAAAAAAAEGdAABDb25maWd1cmF0aW9uczIvZmxvYXR1ci9QSwECFAAUAAIAAmZ71KAAAAAAAAAAAGAAAAAAAB+HQAAQ29uZmlndXJhdG1vbnMyL211bnViYXIVUEsBAhQFAAACAAJme5SgAAAAAAAAAAAB8AAAAAAAAAAAtB0AAENvbmZpZ3VYXRpb25zMi9pbWFnZXmVQm10bWFwcy9QSwECFAAUAAIAAAmZ71KAAAAAAAAAAAGgAAAAAAAAAAADxHQAAQ29uZmlndXJhdG1vbnMyL3BvcHVwbWVudS9QSwECFAAUAAIAAmZ71KAAAAAAAAAAHAAAAAAAAAAApHgAAQ29uZmlndXJhdG1vbnMyL3Byb2dyZXNzYmFyL1BLAQIUABQAAgAACZnuUoAAAAAAAAAAAYAAAAAAAAAAAGMeAABDb25maWd1cmF0aW9uczIvdG9vbGJhcI9QSwECFAAUAAIAAmZ71KAAAAAAAAAAAGgAAAAAAAAAAACZHgAAQ29uZmlndXJhdG1vbnMyL3N0YXR1c2JhcI9QSwECFAAUAAgICAAmZ71KtPdo0gUBAACDAwAADAAAAAAADRHgAAbWFuaWZlc3QucmRmUEsBAhQFAAACAgAJme5Spu20BwMAQAA0gMAABUAAAAAAAAAAECAAAE1FVEEtSU5GL21hbmlmZxN0LnhbtFBLBQYAAAAEQARAGUEABfIQAAAKZw5kc3RyZWFtCmVuZG9iagoKNSAwIG9iago5NjkwCmVuZG9iagoKNyAwIG9iago8PC9MZW5ndGggOCAwIFIvRmlsdGVyL0ZsYXR1RGVjb2R1L0x1bmd0aDEgMTE3NzY+PgpzdhJ1Yw0KeJzlem1UXNeR4K37Xn833a+/Hh9N06/1aL66oRENEkh8PBA0LSFLLQQKICMaQSPQByBAKLBjCNv6sJC1ZmLF1iNNonidbGJr143kZOU4G5NznMzJcZw4ux5PMrEdkvFMJicmVhwnm7UttHVfn1jy500cnT1nf2yr37t161bVvbeqb1XdRlMTxLETGYIR5SBQ/3jS11BSHk+4SAfWB6SrrZpEgILxJCxaHxfYeKwj/9DSHcHwnRafYdvGuoZM/9DYSYkCUzfjzRP9j44b1qTbK12LdMCJ2L9+lw/6d2M8fPjT1ya9pn41ifwb7HQfHBvq/XrutHPuvYj94qP+T41/Rzv6EyNg10mj/ocS2xBPnsI9r0G8dH5ucGiT5NwkJHGTj4x0J8d/3PKbH/k04vinEaf5jHzOCWtanHK/R6vQGo8mcYbEKNrvDSf5/+mj0EheJauqI1Yyr79s+3GWSzdqbb9/+Xt568/3/m6vQp5rz5MvkWXXW/IT0pgciJEZGyBHE3Pr5NvkRytknRnrIU2T2z4i9TK7heIouTh4mj/8Zuhh5jFwlf3fbLDFyiNyDa/ka+QmSJd9DVxkj74Ke3Ee+g1LfRwdfo0oUteBrSAWHbsH+1FygZ8gW+hZ2HmcjNEQF8iK5CHtQ8hTu8+zqjmv/jdBT5F587yTDZBph9a0p+/AfieHm73BX95It5H7SSA7ewvFN+AJnRpt1kC+gTr+t4kIrg7oot59+ndIbj2Dnb8g+fPoB907Pco2kWWODZwLRWrq70jt2tu+Ibd92x9a2LZujrZGW5k1NjUpDfV3txg011evXVa0tD5WBosKC/z58hqfN8tpE6yWDJPRoNdpNTxHgQRb5EhcShbEk3yBHI2Wsr7cj4j+WxDxpISoy000SSmukkm3UyPI0fQxSiVFqaxSgiDVktrSoNQis8mXm2XpGvTs6EL4bLPcLSWXVPg0FeYL1E4Gdnw+5JBasoabpSTEpZzkZHp4tXejPLmTcZN8qaEsTRI5o0mBE0IJYvk8XkoqgcVoEUtG+Yp0WewaZ0cv6V/MBnb0dXS7Pb5ukuDm5MWuVkdIptUkUntpqROFSmNsKWTM9J8cGH2oWsc2RsPmAf1wf47u5JcP/L0ci2zs6eStkCyWG50Ft/9VhbuPJEMys0tyQCT2ta+0k/bR1NCUuMXZGn29wS3Iy+9fTumP43R+oXfEwZGUL2zsxFZiszGZ/uv3ZzK0uCPDtvsNs+0t6CGSawLua7d/MYzdzLyUHdSiADhvRmI+1tSce03V1J6o9Iw/2IwW+D7Kt2+2zdKzSxPzdMUBGoDtSpz8c2fuaaQvZiJzmzoyvVl8he9xWiALdSRpnIwsrI650NjKzMrLKhpfRmm07u2aTvH/zoNyC0j7Tn5zZi/60n51CFpkWP7h98qzdJtWEu1VaCve1eXBESmoKUC3IdSsDegpjmRXUjuUPqWbJjRMU20xSjYximJwWuSWe/k4PZ6EAqTSYjAZSpu/oSirNCCj9aRu1zJeHkKM/jiYaaVbNlwzJ40mn3LrqT7as1pGdXSpLmi3p3JQk8YE0VzLU0sxmllpm482pJTBZ8o6u50j45uJ8peS+GiaVpLuZEYub0K8KWma7BoeS3rh7EE/akNTl9iWVbjRwt9yV6Ga0hhoqXsTpf0qMSbqpo6ttpp9y206er0r2Q1AATx/tbPjZG7nKnxKDLJfV+vDRF3Vw3EgqIkCIIyE21+E7q/Hp8BFS4imWu21QrdYGBrFDjMpfUkuioU3H+rCJ1TB32hRdkaZ1XZSzKer2dftSn9IgxWEpPTFy6J1SoytDnB8jAeIoiLFTJdZz0e1Ljkhd8vDU1KJdbG9MfWoWk4rQ9V521Ydt/VuURaqifhweKXD1JmMBNy3KjfZqvZXu9GPDW9eGZZm9XLbz1kmXE4LJLjyzUnCXFiptrnV08/0sxzpx00MJ1o9z7PzisL08jA7trPy5sFZeWdXrUqNEeRe991sLjtpg7a0ptIgBr0meRke3DGvWIM7e7qeE7CkerCj6woFuin1D2fj2Ndz0mYK1QsZViGZB2JdZikduzoVxr3cwohM+ooryLU/sA1ICp0v4IDMnCNpnDCCo4ijk/hFBXHPmi1rGHUMcbvFmmQ2edT3c0z8W7m40REjeAXkidiXo3bk+nmngWnPSKCeakia5ieEbGL4hdhcyvA49A0QoDd49K7TIV88qVVM3acbXoKYTK2AdKZsHEqq9ouP1SxXzW3rtVc4iiCZ5xhaw9BXdFrDh7VXg0HDNp/N77P5mqm0nA/n14c1ne8/3cy/TFg1uhGz+BmsudxkTinWu0DvBH0G6M2gN4o9u21QaWu2UZun2TPoedHD5XsqEeQ8124uXN1QF2wtU1ZYE130AI15xj2XPK94eEN2T4wuoimy4gYDFUx6ri9Dw2WLntnJhbSULFUUVHREGZve01oCUK9SxW9fb2BwOEJ4fVe7Kwt390Ln3DVelthlc+lw/Wn2kyXD3E+bDdevkyzL90j12nW5cs3/vXyjd0Xq9z8530qqnJu/JZaWfvBX

ndV1ZtW3XjJXZXSYezm21yE+w7xkhIypZSedsJ5B5gcZxxUdBe4qSEr06s46/EsX18Q9ZpM3iAJQv1M8FLwepALsh1v2hJ1rJZUhb1Q/RBEUQS8/u1Uixb006wiaRhqWFpdUu4nz29wg8qQsJSek094HLm0XBFPV3vsnDymjJaVbkuXJFHPQBrLNTlqyyjwLsaJnrzmpqczIbt3WVHvn1YPAHL7Q9sLdm+bHqHVXZ8B1bIAo/sW8+ua90ozdq61uMUP59Dfu+s07RXs+P900F0077tm69Z5daqEFpBq3LmB9bSCn1CJNa4CAicCG3eQA1pUXC08mu8nz5CXCs97T6F2mF01gYpttiERZq+RWb4z0mYASK2CKmS6ZkqYFk3Y0gesmTiUsLU8Rmj0EqDZmQH+KcS5URUXDEgQCvYf7JgIBZtjDE3t6D6Mq/CkrokVh7IvL2ZcuQSSSXVqaraErvr4N7XQF7WRCSz2k1B43njNSjRH06C/oqVEPZ/gLPDXwcJyeo1RL0VmjxEd8ko8KvnJfzLfo41P8XEbfWxhYuOW6Bd8M04DxRf3zfgu+fi4D9Qhi78sKka1tphBcMe41AEhtISrnpqI4HInhDdXjcf56TymgLvaK10jSUzq6VNyl157Rev/vjHr7/2j8/m1A1u3hKvFsXq+JbNg3U58NN3bpL13/7mw//5u/7HR9avH3m8f+/nDtTUHPhcykb05s91p/EMOukP1Ss6K2Ra4J4M6MqAQAYMayCLggZAz2mg3WQ29Wg1Tq1WkwHtPMffac5wms0ZFNotVksPR50c1rvQL1iFOwngHRJ0mkFnLjSvNx8184ZPWPdZ6TpxEp11kxroZUzmjmrlTNzWiJ+T4TnRfiqCBdE0C1CgbhLnBa518R/Eemz4osi/bKKvkeEIRE6RIiIYBCzxWKR+4XIFHnwJ29Gz41PivSkC00i5IuVYrPIiSJQEd4V4S0RXhPhSZGJ4u4ST4sUBRTj+RELcJqT4tdFjUGEp34l/i+RisqXvhJ9Sfwpgz73t1GccoTNT0ukBeI6p0TYhFfnHomqE/tPnYk6RdCK8AcRXhWBSfuuyHwl0MawI1JwEyrHgbEUR2hoJPqcCPeLA0MiDDKut0R6RrwgPi1y42x3gyJVRHCKbBfqbEpwUyR6RgRFjImURyyteY/p6yXWc4rnRM4pTqmb1XUqAxVzsyowwyIBsKn5RS5KBriGrjth0fmIIMDTjFnRYuxzcwWAQ70fbZ9ZzVbGacdiSwz0Vmxcw5zUDZV/VUS1a00txspmYe3cFBGsL46Q30HT580GALp6JpRSgcQhxDYQTcCDoxwVyXIfAzMdGHTh1I0zPSqAQB/LeHddM07BNydfg0XviuLb0mLhQKsQF8TmUjgVOBF/9ks7Yc+l1b5DSDjF9TobYcwIQU517dULT+5fKFqufkIJd+BbhgphTuh/EfwTf437/+0u/jhoCaDhfApT3NPfLihm0d4JZadx7iQRQ5c5Y1Ar938sRIyWKMOL4zBMeAADK3E1lgky41FcuiRau3eHP6cqisA7scQw7q4LIo06VgMEcxOwnWmN1qsMTM6WjVEA4Fh0+Fe+HwBG50qSLUi5kogEFbLsRzvpJ9cAeZ9VyYng9syFWUjeIX15u0HgW7ITPW25vPfWd5VJ9hN950YhGNk7JLjzjWBvMwpm3BmPZLjMM5uItPKVVDrdNF9LweDPrTenqRh7M8mHnQ24ncmhkgAwjFRwnMBBYCnBSIqwAfUMNx0Czqjm7XgCYzluNyxERSGDMK1ZF09T4Frax+4JYg5qDAmrQFV5H51iNYn4LyJKtsh6D1phZxtUxk7CtgRrH1q231Yg8Rkv/R/CB7uVj4f1PjIUnqygGlc9D89TyH5e9/ub4xtr9/pLR81ljEXk9/PzI8w+0mE2mwNpy63tZpe8/l10KL4/MdRdmCvSxesNraDe8yXMnNFFiJPVKiWCOmWnMPG50mq+beTKqCBogGkUT01zSJDUavUY3Y9QszSSXpVo152VAYxz0Weo9jK1Dgybw2zRV/jCdBpuneDiWfwunKmMs0/0R/qo31Xjag/M9rN1k1pBjSqc/DzTzN82merM9115r32rnT1uhhAenkxrGctasyckn+Uo+VfLj+ZfyF/P5S1+zj/rKscSj1JDpe9g0xC7Yqd1unjyVCZ1a6puG1Nj618Kh316bvQZVHcakjksr1tprQ2nKCuYIdlGKoqtdUVRawRGEF2VYPqHWX08LrfNzDH76U+0qpxCbf9Mm8jetCdr1p6y07Xn8jEjs6Nz9Irzyy57P3Tc+c673/QYPVYXwSqd37v/6n9ofuu/fkY7vRn3zLW7kkngUfKcfKLTESuitEtR44bjtno1obHDedM1HOBHotHoo1UUuFUgGkYqaC1iAQqxivmKt4peJ6hSYFcNsriCDyua3EJ2DmfMXH632+3FieuyzmEEsKd/AGgcQ4K0uPYfWkLFWAekQqKnpVF1vJk2kfc6QPCZY2uFmWmwz0A+oBwjUvKnV2fJQF+v0HwA9ddU0bSnoemhvuHL4b0fCh8PsJD+5rBylg2sa92wsP1RQmHq+/kluKLt0vd3jMtf87XpyeceiJhMzq8v17CcFQplcdug5naX2Iqbnr3hp8wH70gDv+QuEzd8S1kz7XzUSTXuE266P+funM/lcJocyt0GVGRvY44wH7t5uJVBG3YKhkIGDNmM6gBq2BDeKTPRsoR4BkLx15uB9FaNLocndNBMiwas8vsRsihRdjS4oATDnBcu/mvykMFxdEtGpjVAKfJQtgF7cjtZnZh5nZhTkeWVovGabFoXftyICfHiXizULCGJ05CjctjUfoHo7zH6aGp8nvQc9zzJFbjr3re8hgYPh+RDPUsIt/yv0cx1jBspWfKc07F6irx9So08Grh7vG1CvdKa2Y0hoW74FE81LHXAhLjpgFs2Tm9GaHK4ezxLQZbidvzBSsRKPjzDHOiKEyvJSJ1bs9swzb5gXhc0jwSv30SqaA8P09vRV4LGrYg4PhU7fmBwj09R7GwCScW1hIPfoFFFqNuSMQUMUf0747e01valy0ZRynnQccaca404aXd5170+W71v+L4egavn6GDx179deuQ/aDy7/sclVWpoJdyzPYyvAefgbjMju5XdbwNa1/FT67sMPY901kEXFqNG5dNTA6e3oCzNKeXwwqrWL9gJ7xL7LPmTXvmeHf7HDq3YYsk/bv2z/uv27dk2HHdg4fc00K1NNY0uUDdNme4edVtgBJDs4WQCBGqe90j5oP2d/1f6W/T27D1f03A6KPWYft3N2ZhZ/UVRtczxqq+Q6Mq00J2ZexJwPGX1Yp/UZY3rQ67m40c4ifahXSJ1EDPSpQ1tN930s5actTNxrjB2vLMR8DXqIwuDvCXDqH0aM/BP29n8+rb7R/8M94marV6HM+GLzoIXP5W0omlarFA/iyYyx1EA+5q0wkW0zGC8anjdyvjR8Y6XEjGLnbTc6Ak7Y5dzsv0D9w8qy30fm083nnr51awanU1EWdXt7r9GLN5Iu5L14UvZe8Se+C159DgHrTdw1zXKrrSLgXUKz08rnxDxWZ3YscyVDSxvFyk6F19UK/carveq1AtRAw65ULLXlUe6ja8Y9trwiUSzMs9nyCkWxKM9mZPe0ExDgF3YrFqk+2LGSt9neCbuHYM4W8L74sCKYZFS+NctKLVyeI4/msfJjg94UJVGd1C1RQqoqQSDBmSctEYJzQaoE49iZCyaD8HFoE5SuwtBPsFu+rMSKGH7d0DdpCQS84s5RqNmh0ewxbBoV+MrS+MQCrCrcTrAYrzFxIL1yJ5eggZNx/ScVTN4qgB3UqGUXVRSJf9Lf0bs9avq7CXHAzPfurGmdMQAoyvpfdtW3i58uCXDpcPxHsK4PrQmV1+3mDW38ju63/M12WVLicda6uqsuTA94++sLJqmmebVX9IYo60cx9GzWyjjym+A4UgDszkEktYr1I7ZLJGvXYS+3Ub1cMGwAPHAubHoMtInPqc43rWrXVM9XQVw1KNSCwttVZyPTgNVqihYXbneAsKFgTi0XmknXhHUariPdJ15oYgkCtaTD3YHZ1N2yWcbFYW2L3tNXyhsUL9QUs77CapnD1113PN0BVKvFgGq6qB4f0wrmcrN6BHymjsdIjy8s0azjat6G5tzorb93mzr7ysxZfdUn5Xv+a6sYzrz2wcVd17sPNAXct7M2DLTd0JFdusdaJGeVt02rrd9dXyjgqX+kpKUin8d15GWLazmPp46yWH3Sm4VxxnnzbVrK30eyySeU6lqtt0V5LTzjhgU3NLi3u6nR0srFnHEndTp1hBM4ieP0HG+0GRSDJWrQmawu2w6iFnc

N4R8EllaiLh5/TL8Ta8t7Axq81drkqgYIu8Iu2eYU2c5cFg62xfvuutFr8A//sLHcv91rXbuxyTmxjz5SWvj3f
99x41hjk1HbaHRaja17aAhz5s+xppDJ158jejRciUGI8hhyrKxitqqtJt0vTVQ0+Z/0A+93+l/0v+Xna570v+e
nfmbFrEB5NOAHpx8W/HjM/TDun/HP+Tk2eNW7JqoSLTrEKMlpncEAQI1X8EoIv0Jd9Gr1XikmW3MEkzdm8biyC
XG2a5jhwyvXG3Yg2E8IFWoIYDd0l+m+E11kheLAdANI1R7pUT9VdtxTu0Jx7R0dH/rqeRv/E8oF7d3Tm1tdW2Y8
t7z98GMxcXMjHc920v+36oywK0K6dfVq9ib/h+CgmADmN/v8exs0N5J+VJ5zeFG+llzM5IAAbgeYAGCtzKunxE
JwsA1dZVR1tLAE+y51F9cUOMNrBZLKAWWsUjdSa582jQ16euWdtHamD6sW663WU1L1SR8vrFGy4oh4XGsVV7J
cMRevdw0VQuCRPBCE6SB0BAeD1B8EMQinBdgj3C1QsxAs4nN0fesA1jnkvhziBa83h8d0raaJMFammLJX73WqC
4XUA4SR9E1MJnt6WdmqlnDpKg7UU16cmce3zfw+Gir0tcGm1zIbgnqjx/a1WBzplF2N0+8M7h1fHN+3cCnT3x
6oK526qsHB660NcrBmdiWA5E1dQPHThwbqKuZ/M9H6j+5v8cHI1/LCvgcxdG9G6J7GkvLqncd6912rLs8x7b8q
yelEm19W6BxV10wtHH3A/HeRw/UmJ05GS1fbcB9J7mCeKDZ5QmgzBw6z19Bihb8jMyfb88YPoGRkq5WZ5U0a
0y6/Kb8nvyfy4DE5EdSCSZ68p+V11QGuS3TL9/nUZX1RJ0ZWXjXNPrvCm6BmoUacwJq9GVbaLatd8/kL0ggxT8
nGZqoi1p89Gn5aBsR2X0bcMvAzvyfC8DEy0igrIFJEHGME5mV055hLD0bYV2qf152V6ToaAvJtR0mXKMC/JHIP
ZNqZkYYPPZHgW10gvyZAvsw1PqeK0ggyUyCDJ5XJMnpHn5KS8KF+X9YIsYXdB5rMyMnJbuVTP8Mq/lxfz0si0
TEu22qPGfpYPWiAVQdi+VeNPBV4M2ZJqu/9rWDlt4PUaVz5ySCg/oig/oy2QqJi2C11yFXrVxzJteJIHmBxSz2
oP3/iicCOI5tLI71rS4WCXdmYY3z//ZeW+TNc19rCpv1fPFRT0r98j9HkbRyMX0z48A++01Jfyifwsf16yy/Wn
++z1v6eeFN/z//vP/6P+z/6k/TyVt1p9Br2x36aRiGfzrfcQj6xSgQf+8u3i75NmjV/RzZyZ0mM1pBinpBtjF1
311D61Nrgfs+z+PTg48PHzv8T2Yi0AYQJ8kURdiIc4ifJaeRB/yVB8gTkwjT8Bn7D3c8t8nv4R/H+e1771K5W9
w19s/5NQ4XhRe0Dps+oK3KR2vSaKsBEElv18iZtDeFUbC7sWl13fHUPgJTxEyJjoynYY64yde0zCPNz9Kwhlj
IE21YS6wkmYZ15G7yQhrWEyfUpGEDscAdadiEa9i9+j9bymBFFgYZgy+1YQupp06cHXgD9hZoexoGInH2NEyJh
atIwxzXy1pmEea6TSsIbnco21YS/K4K21YR97jXknDellev5iGDSSXfzsNm0i1Rp+GzeR0zYr8DPKm5mIatpB
Pae/eNDZ+18TIvuEpqWigWKooL18vtScGpWj/VFDaPDpQJjUePCipBjPSRGiyMTGdGcyTtm5uam1v7Ni8fZs0M
in1S1MT/Y0JQ/0TB6Sxodv5t47sTUz0T42MjUo7ExMjQ+2JfUc09k80Tg4kRgcTE1Kp9HGKj/d3JSYmWWdtWfn
6ssqPRj90/FcWgqvfnzI51ZhA5Mio1Fm2s0yK9U81Rqek/tFBqW0VcfvQ0MhAQkU0JCam+pF4bGoYl7r/yMTI5
0DIAJttsmx1B5vGJsbH0kuSkwnpDv6p6YSk20jw1NT4xtCoaNHj5b1p4kHKLzsY0xQ6C+NTd01nhhMTI7sG8W
d1w1PHTq4FRc00okLP6LoiKu5VWuRsVE0zsEUTVCaTCQkJn4S5Q81BnFp4xNj+xMDU2VjE/tCR0c0jIRS8kZG9
4U+Es0kp0f593GTTWQMz+BdZIKMkh1kmEwRiRSRAVKMbQUpx3/rEWonCTKIBZT0I0UQoc1kFKnKEGL/w+Ygth9
JmFR7CwWt2E6rvIxyK3I1kRaU1kg6EN50tif2RKXvx2cKqfuRNkE0YtBDibujAz9xfm3Iv9edR42MoL0oz16U
8WMIC/j3Ee04AqZxEacawAxo+osE0hZqq7rL8v4a+07VGhydWQtr0vprYxU/knevyb536eR1073qVKmVNkpyhF
VdidS7FSpYion08WU0tuoStXxJ2bcjjMOIT/T3EeUA6rsKeynJI8hPJzW6n7U+IS6gkGVb2Vvkzjzv7UB88EJ9
MKxj2mJrW5anfMOFT+1+hQbG1Z742QDZp0Q5g32rwxpbpc8kJZbpkKHkPL/lG8KT8i4qseEaud9SJuyeZkq8xD
619a0hkZVv2ca0nLLh106+X0+F1Hb1Mk5eJscZlnWMt6V1U+m1z+kzpPS2ji+x1DvCVxbZSp2n7rHEbThCEK3r
o9ZbF8a9/HVrKz19v38v5ybSxURN30445/4zBvi3wIdq8DV9wvAK92weAN+eA0kG3DsA4h9ADPvzr1Lf3u92Pv
M9Reu0+3v9L3zzDtc+TtgfQf0ZE1Yi3f18aXLi1pjda3wUx+DbZ/Wqz2/iz8Rueb4dc7yRtQG3tj5o3kGxwri
Xve0JsbwDX+ToneoUFaaF8YXxhZuGVhcWF6wv6mW/NfYv+t2+GvNZver9JvVe3Xz12LYt/Baxf8X6Fxi7EL9C
5i2C96L0Yush97vEy7+0ted7HHi30Lj56/VH1j0FVj2bYIn2fhW0fefgzbPzkzMm5k9zMibkT9JnpF6bpZkY0
zYa8I62lnizw1mdujDXqeVuqj9SNe/1F0XifYq3D41295R7e1qlvY6wvVODi+WR0Mp51b/6jXEPcy9w0n17LM+
7A5/F2PUYtW73bg9tV38f6W/zoaAt41tmtncbI8Xea6u119rqbQ21/rD1Z63vtGr7WuEL+I08E3khwimR41BEi
eT5Ir1Rd6cYdnUKYwsnBdIjYdIZst60Uqu1z3rMyn6wJ3RGBA1cg7n5jp2BQNs13c32tq0+tsJdyb9091b2dG
T1D6YJJ09u7vmAf5D94mzZ0mTpy1ZsbMrGfd0tyUHEVAYMIOA4JkXSVP350QU+6k2AIEAgkfwTQJHELVnMoUkg
ZVhEpiEyUky0QkBNqaCiCGTAYZmGMYDyLlnkrAXGw2oVayanMza878BOUzSmgp1bmRzdHj1Yw0KZW5kb2JqCgo
4IDA9b2JqCjc3NzEKZW5kb2JqCgo5IDA9b2JqCjw8L1R5cGUvRm9udER1c2NyaXB0b3IvRm9udE5hbWUvRkFBQ
UFBK0xpYmVvYXRpb25TZJpZgovRmxhZ3MgNAovRm9udEJCb3hbLTE3NiAtMzAzIDEwMDUg0TgxXS9JdGFsaWN
BbmdsZSAwC19Bc2N1bnQg0Dkxci9EZXNjZW50IC0yMTYKL0NhcEhlaWdodCA50DEKL1N0ZW1WIDgwCi9Gb250R
m1sZT1gNyAwIFIKPj4KZW5kb2JqCgoxMCAwIG9iago8PC9MZW5ndGggMzE1L0ZpbHrlci9GbGF0ZUR1Y29kZT4
+Cn0cmVhbQp4nF2STW6DMBCF95zCyz3QRYZMQEgkhpSRILPqj0h4A7CG1Vix1yILb1z0mrqdQF6BvNm/HjmbisL
7XRc/zqRtnAzHpt1INpvDsJrI0bNpFImNjyxit6y6G1Uexnm2WaYahNP+Z5FL/53jS7hW30auzgIYpfnAKnzY1
tPsrG183d2i8YwMyMR0XBFPPr+z1Nrn9sBYpra1s39bxs/cif4H2xwBKqRbAiRwWTbSW41twgyjkvWF5VRQRG/
es1PIx0vfxsnZcKL+V8fyw8J8SHFHkXuETeB66QU+KEIx+I0ww5I84S5GPQ75FPxDuBfA5Mmsegpz11YPJwCX7
o3GvYc0KuAuOs4IFxpwj+MzxLrP6vyMF/ukNe/R8okPXLMRq8u5/Imbw75+0mC6acMWfGp3+BHa00EbpN3Qjm
gkZw5kc3RyZWftCmVuZG9iagoKMTegMCBvYmoKPDwvVH1wZS9Gb250L1N1YnR5cGUvVHJ1ZVR5cGUvQmFzZUZ
vbnQvRkFBQUFBK0xpYmVvYXRpb25TZJpZgovRm1yc3RDaGFyIDAkL0xhc3RDaGFyIDIwCi9XaWR0aHNBmzY1I

DcyMiA0NDMgMjc3IDUwMCAyNTAg0TQzIDMzMyA1MDAgNTAwIDUwMCA1MDAgNTAwIDYxMCAyNzcgNTAwCjQ0MyA
yNzcgNTAwIDU1NiAzMzMgXQovRm9udERlc2NyaXB0b3IgOSAwIFIKL1RvVW5pY29kZSAxMCAwIFIKPj4KZW5kb
2JqCgoxMiAwIG9iago8PC9MZW5ndGggMTMgMCBSL0ZpbHRlc19Gbf0ZUR1Y29kZS9MZW5ndGgxIDEz0TkyPj4
Kc3RyZWftCnic5Xt5lGRXed/d71uqXq3vveq1qqum603P01t1Vc2MRt3Sj6a0gxCSLCAgekYtzcAsYhaEIITNY
kfIZjGWg0jYsTfGwIBTCYgBEkQMD2AXwi4pjhNHE6BiR15/fdV9IBzDn2Sc7JP+nuqXrLfffd5ft+3+/33Tv
nzpzfYD57Pp0sfPTk+m0Lc7Mttxi/ZYxXbz1xxxy0/ufxFn8bx+xibv//YxvrNb/j3r1libPFulBkew4XG1vcFz
j+D823HTp571o/1dBXn38P5JSdOH10/cIo9ibG1Rbp/cv1Zt+0TV+jw6Tp8tE6tn9w496wv0/1zj0nKbafPnlt
mH9pC1S+l+/jH8Us/BRwa0hdSaWM9PwgLxahUzv/f/RQZ29oSr5M+22BPy6fYwfZM9mzM30vYK9nd7A3sXvZmD
FPELf6MnefZPF/lQzoX09xYMy+yeTEY4hrutfN0uLIq6GxV0hn9Sfq0q7yHq4N5MexNSxu5i3XLB2bYsYN6RjX
HwytdbM4Tdj53k+7y1nSTYedum1yKoHH03gvyqeDYQ1XxKsqs4VCUgxUOp+0V3hR737i7sAKwaWpT0+MRd3Zg
meFEqZorA2ltr6SVtpGVRYL2qtuqxbjehxpXy/vuG7RFoyUWkq5dHguDIU1Qizjodi83Qh1PCHeL7goezIU6kK
j0W8o7uHvISn0E4SIjJae0GLMk3j0oAkDabzDz71EcK9oC/s0XTqG13hZb99gviFF+rCvsc8cVd9e92reBrly
/Wo6I8tNiz2715u6aIxodFRKVJ+1b9fm9nLZmvddqvoTyxcsafQmmkV5VjRePxyvP1mJdQNwsjwRult3i2M91a
ptdWJl6Xn+fCaCXVKSXQcTmBDgqYvFAsxrw/xH/Kv8BugEcdYcfZbew09jz2YvYK9tvsd9mb2FvY77M/Yh9gH
2IfZz91X2BfYWy2HgnMXUZm4D6TFBNNc91PM02DZLhMV/gwn/pi2kWL7yQ3idgkzorqmqGy9yjMzHKzQ9A5XMoM
DzbIXVGnN0qL9qftzNAdm5uVpaJZn0vqTU5vdE061p1J1R1z5mmK7qynNk0kzAe0kX7cD1yDZrnw6xt0vxzp
WQiaKxdce2uo0pXC0r61khP+SYs1isqtFyXa6ViGcbheFaWXHBdMKau4sU9B+cExnH52sNrrWiqpmFsRQy3Vpv
/WmNUz3MuxKtQnifaN9rTAsMNKxHK8lhyDR0yVERJzvUMZkmFQpCByga+xnGRozBXnGPxLzVZTyBsjxsTifSV
IvlnUfo/eY+qubd5cm0ZquzDVXxil5hfGqmJpQWwnR607LURFEew/KLkVTSRAGHeepQ18aqc4f3LycKXSrVdvR
rfqp1WAw9rWGKuk0KoU0bZaX5NIylwIc6ULDGQ9IYsQbT4eIo+oe0XQUrLxpYoBCzaCKs7APoiuDzirqExVJDc
Ce4Q8zFPmE2PyGcPXIbj01Gflk0Lvwunpb6Fk6YvBf2uMmb4n5mgc5/zf6G/S37DvsB+wn70dvEuPi8wqd5h2c
ck8jX+CF+BX8sv54/hR/ht/AT/Bn83/AX8ZfzV/PX8zfyt/P38D/mH+Yf45/k9/Gv8K/xb/Bv8x/wTWFEKKoiE
ZoiLbaLedEXe4BvK4Nhr25iCStNarCrJM0Rj/4k2Z6tmymepINev7u0mCD1I/zm10YWcTj5YTtwLbZdmxh7AWF
KD/cHKNwxFl4xydNON3NVE4r1BkM8nybLae4UhJz0DF2EubZtAk8zXbr1rJc+eyjU6zv8paL9HIf7+SnMvJcmd
Wo3NUQmabc+7K4AQocdONcaBi9d6fYtYRruaM1gja8MM/g0ILWXLQ96QFyTrfL1ZnjuZg5rcaVOTwD6exiFrI5
qUSJzz6fdFTggudZ+XJHJ0mKT2taJaQRTaoVNuk0U7NboMiqqp9koUpg1NMJI8uQUV9CygGXQ7xAiwEHTbpvwH
mOT0fA20Xg0dDkbJkMzdK+0PQSL3rDfsTE93+52suGKG9562p02s7KGwe2ihTQ8ie1Q9EhM3M2/LXvImNTtIj0
SpzVc6tAcx01Mp124GpWrubLDQb8HJMk/Tt+2484AVzuMj8Hi+evIz0+CB3A1CnA8EVUCLbUXcG44XB9eYs101
aeUJu83cGYJ9wEoFxBF1EIxeIjwDD3gWQln4kYqbgSeAlgAXJw34bEQHwI1aWEKCEkKJ9IvKFuiwIPoAt/3atJ
XJSsCpUJdb9ccbkn1160Uyh8bTxEBne8SMH0ffupzAjeLs20E0wlhEzm0kJK7MMYjRf3CDwq0SxAlraucLq3ik
zoe0Big9wodk9RnuDSxysNXSaM0vBBDFaMUHYm6Ky5odLgk+kUhyXIrNL1SKKwM0Bo+ASzUVNLS41wg1qIo6pS
G02u0UUbHVJLAFnja1ErTMX4sItaacSNGU9JoU4+55e7VUlnqXAFwjza1TdSwQ+401TLX4UPKgjRykQbCLtN8G
WA7P0d134WCH8Tgo+zDn1ZW++KwDrQf21BH1FyYaM4NdkgoE511bRRKKnRZ8jk1NbWMuF73uex6XVpoR21CJ
btP4EYgVmnsYSRmE9rmAKg2ECAxERVgvREMVjEYExApgMd0tSYXKEQ3QeAQf2wz92Kja0kovvuat18DFE2GUq
s3VPFgE7Ad1MGNAcC0ffhW6CxCl2Z/HtNxJwyj4+zEnZzG1Wj4GQ9jKh6RqDsISZc+4GUTxv40hmEaF5zEfXF1
qIFFgDv+Kghum8Vp6CV80FDsOS/Fuah1/CoU5zMYt0PMx0JvfgtmpEG8x07gcTQ0qq9EcP9VNH5Bd8AM05u8mb
+Krzna02ayLnw3f45e50Cwvko3cXNI3QG/ULmBbX6U7LZGvskjFPTNMzEvEgNQhq3pFLHbgyeJqntM8BUe5yY
kNt9GVv0jWoCwuvnv4Fqy+HfS8+RQemSUUuxzIYvNb1t/SP/mhiwC9jF7FL2GHCPj7J1doydYM9gt7Pn0hb92
+x3wKbeCjb1fvaf2WfYf2VfQnz7GruffYt9j/0Y0c1DABnnLd710/ki2rGfH2Bs0E/EathdFYTXFxB+E+pP8zz
smCS1EEGY5eKUoz8cgD8Nb0w7HjVIQYocBAmZk0Qou4CRRpHo9Ke7aRDh6RxTqSS5V7xsWyAP06ArfUOBZgL+
LCbZGlm4h4gesVSEKSokaS2nVHcQ+SQCAJ4YpDYYT8bcTabhyI57FBQGgwETjEQ7o+HCwjkKRDUwguD9iHH1g
VeG0G4K8jZqxHExzVGtRuwsZw0wEjTzLKV7xCBvs0omA19WTOHVBNNtbaFQ0jMCv1sXE305KFAZyFt43d8mu
oMgKehKtSxNmfcwFBUBjpUyWanfWn53WRPnRpmvE530AexgamJaYCFr2HSyvPosjC+VbBDThamVKDIxj7ijNE
ZnxC30hveTx+JIFsXE1VqUxbGoZRIJpk7fMEI66MAqigk0CrRuqgs8LuvREk4wku04gHZJ6nJF5XHKj6f7JSEt
y5cY/AoYIBiC9m3yK2Z0BJER6gazkLqTBX3I3gLYERbYOQwShrNRn15ftZWhabUtYMosv0xduTfr9X88Kg6Jd
roY3jqgIzFGtPj1s3IIoV/jqCUDNPL5rRAsj19xgWIFgHrX2sgZ4W+s/RKCnv0/x9aM97iBeK/fg44zqvIPvEf
RiuLrzJC00Q9w9x+zPUAfUPIqBCpmBFpIoUHszzFQH0c50+4xv6adoqkFZ7qxZ1wh79JMF9sReYJnZBZ01p8G3
91fH3Y3oVYUWRlgCN+YWs6fiBeqb+CWuBo3mjSKhG67aKDxd40mcAn/n1YnFw9AGxsWsNe01PA72B+w90IV/
Sf2Z+y/sb9kX2X/g32bPcD+F3uYmx7wGL68Df7Ygx9fxC+DaoJddub5bL+ekheDa+TMMldIzldyghdDYOeyCp7
dNtY9J0dSi1R2Mi5baYJEiLQInDAR2S2092QzQP4+aDvRNmw16R0i2n47zCBg2Zw0vgeKBrIT2rJrd1b8Gc
6c0Eb052gRB93rDSCjYhLdI4G/ln081JR7zmWJKo20Gzz6kWdtjSRe3TrK08CFbFW6MiQ0M1ekNqB6iQ12qBE

4Bgs10mrEcYezfPMawX9SnrG6JAt+VNwGKaICq2wAJR0oMI4ai2NczeYqHralsx4twvti7ipZG1mMkWgxUwX0
jLRK10fj72kMVbQ3PjCVY9g2qiHcvnQtG750oiK0PBA+AnhBVbLsJFWFfQTImzaffjl0okbEtou0gnobT01sbU
pKPgg1DWEJ1RZVcbYMNQac0tBD/gxkLBis1yABtcyuAOPe+Azwh0HA6Jbnj4qKQcgykYf4i6aglhdEFTBfTSam
9ZAWiD3Dfih8uMqBSQ0ivMmUZiecjSMgEfqAF0Hwfk4op3yvuoBbA5WzfrE4XsYKf12Dbe5o1PTRXwhnC4d0F
5ZU97YTS9vT/b2B2NHbr8EtBauq19cqW0c7mXXP/B68uD4Tz4i05q5VJQ3NbRs3V/Xsa0sRrhw40xHjz808/Q
Sbt5gyQC+I1B0PUk5120L3yisu3cfipFs1vobt/hC6aTyiubxDGe6rh6qm4+CDAz4NULvcDxGQQJ1Us7qF4j76
0Nyd13HxhGE6H8mUgxxgDfZcpMMyrJDfmneT4NCVgWxiNa415AZ4HcBgVbn5WhSHmAmc6VAX1XaK1SP1CNP4CI
1V2xIfppIz3KQwlJWzOC/M4kt+CsoJVaNEH+Zf459117Cp2LbuRPZkdZbc1qt/OnsNeyF7M7mKvARq8kb2FvdN
F6owczhk+PpeHiJ7k3E7mUdZrWsAN4MoUuB9xebL77g7EfMQ/eqq9wEnruDiL4E1+ZEkrkSwiH0UcxBMd0zCZJ
F1Xp5yaC6YQmUPtJBBmM0ooNIDiLjkX+0UkPIxxkZ7kUc9B8n0qvVK2C5QpLImnh2fu3KpGhtVL6zUkvL4fDY
dQmcUECY9EidK+/V6jt/DmD7lpcx2RYqEBvgEUxhvxRAHOY8JPB3XJtJFyIu5wMMVipeiAPN1IN6DdvFJz/DJU
qjjCeuDEffmd025q7ZEczhrjaiWhHTbayXGD9ZvKfFqqam6yrcfefCidrC/HaoJ4EoCY+DILhMqBLFwbokz0S
8Vc8jeSG+pcQ3UNcppzW0IFLLt00cYAQuw0camY85r8Vr0YLLzzjVhlar/PAZ9Wtza2fwyzuZxeyQ+wKdg27n
j2dnWbn2bPZb7KX0gu4B/GAsmTvY4wD14Haj4C7dIwMHyXuMqQc9oE51CnLEVuPaJnskLw1LDVpRgC46vCeKFI
WQR4PB/UFThre1Aith4Nehu/Y3chq0dGjRJftgT1LOCb4pmknQAXXImHb5z+Mu60JX9sw1CeoL7e1MYFfGkuSY
nGseN3E3v60o0FXs97eiaTXW6hpgCymqQwZ5Fmz+cmcZyCuCz1F4k7cS6Tj1ZV9Ky1fjk1PGRCEgpje0eYVZj7
kWDpc8wISeRc69YqBBxW5k55XbvRzhh5wC8SAQE1IW4AR0M99sJlaUphpty0v4j/2nkNhc3rMBqT5gnqwzSt0b
itve9L6TY/bf4alqBpZf7I5FdRazU6sIv1TTbp0b740tlaJbniY0TV/176LJk1jvC4o66XkzgsP7d/+EejNP6A
im88h4jZGsoXk5gtwna8T16jQ9XXgK5fPKZqgh8ByF2kioBSMhbGU2a1NRPwvsgXWZ/vY1Q4p1tnNQIozQIrnS
ZeD+9/L3ua4/4fYR9gnwP+/CsSgFETE2x1xfE6kN3LZS/zlgLHiPpPUsegMs79s046xvRRokBE2pHUTgxHHeFk
HonAKSu1SKKDEFgHe1F2yKHZIU0PQX+n20yU+6K6RVrAgzMmyyZm1u/Q4TmoDSrxn40YEZGmHEmKDOuh2d5CZG
mrLEsropF3K/GTUmjz49/qOUeCk/ZDy66Zcq9oqHjmTpSgR1mumYgPgSbHSqWgvqHqtPXXEAC2/4BKFVcV1C
aBPwMAwugqhFAUCqBcxx2gCt1/CoH25iUwEBjHcdntYOSmBNZYhksFvSw4iv2iCIGHlfcPwiqXHS15CcXASUq
SDCTKKSy4p2BgtT90LJfmZuWKUrYXdxMFGYAVRBaIaBH7WqyqYLDa+kUTkvhDu0LTavej2eLgvedZKKYubkrZQ
UEa/L45PmJ9HFcBzx03qVeifU9rxnCxeE9LuP+yPCuCXj0j4E51c77541wT08zmR6jsb5Cw1kBfs/kpz3uj1
qkiQvzHm1TtPkj2dCsogtmEGnFsG7Nsij3Av8B2sxVo1EOIZb/B/hU7AoV6ip1jd7DnI5bdDWZ7j4t172F/wj4
IXvsp9jkw26+zv2XfZT9kP0WDA14FrwUfHuZKU1LE6jrgM8IAd+WKcpJ64yW0G1pkYwU0EYSDiZu21bmhkbpY
H7m0quou2gdjI3SknmaftDGm2DydtVsDsi+02wncU9U5TKtVsXzGTFs0pX5QZrn0LsIpRQJKfx10KdwnzqFTxT
pu4IZ2oVnY5dGRcuku9rk1IVMuG5I79yuhtfVH1+g67bm0jquHuM/fvS7uXFKsKRoVwcBGEggFW+FZ5RtST1Z
/ozYWDqcV3vnutloSOQZLVFaWSuF19KJuh+fRFQtcIroFOXW5TcJeUoe+hyIQgrhKviJRKUsb0WXJ15VKGUoR
aFp2oDCQpHh1oSvi7AwSS6CunNQmX/aM1L+05+cNdVu6AVk6Gc1/JCGYEcyyRF8FNdGK0CNWJTNBA1i7zBOIA
AcjdT1KdXESmXpB0dIV3hJNT02GIICFhd586IV4ptk/0K530u2arJUm3eJH0XVbF13qj/MEVv4HJcbZPd7YeA
mdYr5RtewZbc10qTwfi0unoRba/VCXeC7XVwfx/Wh8301zTht0/10dcKaKDv5fJJ5HhdGv8TkVjLZA40XUoXo
tFbDIQSQHI1dcjomESn119041MNkyQ0gyRpFPZpEr1fRkltxZvkEk2o5Gxia2vzo/w7iAnPA094BXu14wlvBk9
8L/sA+4/sk+w+9gX2ZfbX7L+z/wn/eoBtYkaLgLkx3uYzg0ojD5awtZ0twPZi53Lwgm6+jIYT0nqJcaa5SrqKG
GYK9HYW3c/X39q0QqKChm1GuFTiyImrnLzXcb+uW2XgdQvvcCn9nluMTdqphY8QBV31PVq2oKWF7vKQvHUFYZ
cIU1W8ny/k7L5atwwk04doro4Ba/FUzkZGfZ1ngTqrWQu24M2d/FKuQPeuJInotA5uUwEBj3I2hbRbJhTaAprb
4vqpaJvkF9E+2idrBiFBZpSbQplQ7rHVyD0kPseEBcYQsqJmbqv07FyMewc7J1cAJkBiYg8MQWbZ/2kpqrKiai
DLtanaqV0grLhBQgZiM0x1Pen+ATCxCR1YjSgnqghZ6T4HMhLYLiyJdCFxhC8GPoQcyFaRL1WDUuWJzSJQyN3k
9iEtRtdIS818BvbFJQ80SHE9dFssy4KWBd1bAEC9sW0mBph1Cf1Myh1KSU82kQ5uxb4jtK9q2uT6IJfbpU1qoU
vSQ/9s/W9q/uTykwFw2SbM+MABBMT66G1VrZ7t67NsYL8XihIiFGSyD24NhTIXQpuG8F6rVaK/FQ6V3d+kJta
q45XiqMucyScMGThLPw+DJI2wxASztx4caT3SIn/YyrPEMLFaVL1BHS3m20SiGwceUkrTwX1hZJ3KIY0HBBf13
oq7mmHL9S19PaCD+iUAfolil66frs3fcL1Uo0CaTafBB4pvXLTWSVF2wewiP2MloxeLw0jYVpXIp2MDa59eDWN
6HdHoJi044o9yz2XPYC8LB8Lw0x9fey/8A+zD7GPs/+kn2NfYP9HbzWb+zH7GfsIfJExnR02dtkn4B+0WL0iEU
d17biHg18FAjJ9WdZF0t6cLR41NuBHKbTewy05R0TV30krwDXkg231/M3LqYJWbVzYxb18vg1m51f0jckdjWc
tqhZbTcvyztirBmCX6S1XhviJopIZP1y4aJW3Uj5yYpCQKZ1YcdtGONUw2IyG4ZsAsaN8wQuJLUZjEhSAex0U7
z1sT83TM7m42SgvjSNK6qGNEikFCTM90YKQsTDQLPm15YKYWNkBakypWS6Ax2duud3XEB7jLWnDJhozC1NF6q0
soyuF+1HFkkoSXwgHA8Ihtockl+1qGAVDd8yrEjWu12q+CbHxypyRlaEgM/w1zP0wKQFYsUFE03dEScsKo8E/L
QbcHwSN9J06dxBjtV8FjRCyWtq1B4k2qXA/JVyrfSYhvCB4c0hreoxZJWxFLZXBzUSQt4Vd8Lg8DCTs10qwFWJ
nE09+8eXtvTKE1TXDStbTNWx+3txYlrdpDxF9I4CYxf9Sf3N3m1MYb26PLsj10Ngi0irzcAoTTR3sd7tDUDTk6
MVixZsdvTysX1vZYW36bz5Sx1FbyHVnz47QrcVtsewvoC8d+itU/WRhS9MRsFFEZpX8o0QWIW5cAN1NhG/reKv

pyVmNxyEukfBdKnFL2IHeEWmze4xbFwBfjrQe2HuIPgy9eCab4hBFPPAs183x2J2Lzb0HNEE98F/tDp2c+xv6
csh88T35QACKzrZyHrFKH5FI0jwb+UjGMxc1tFAbqe4sBZLBMnBr0Fpx7dYhLWI428D1EhFEEB/c0i6dkQN1V
E+fUv6Dvst/dCwq6HdscgE1SFKQvAV6NYJZYnJDX3aszq0890PBkLj0YK6p24vw292xA5AiLaZ55QLkZkMWfOG
bdNovxY2Z8bQ4tu+iAy3iAS5fg1GD0o4BV978gWtuuJEfR/TB0/cBkTDCoSUMpCUqQcpD6E1H3mhPDjEKn1KRX
PRcImYWL1Emg0KwLFoYKAJrjd2uXQAO10aCsUBwyFjRYgWfGLv82uuu3FcqwM41oqFXjCu1XTs6PohmpcSrSV0
W9x++JOZRrREmBw70ffSN6EshKuHDejWPcvwHL1hM7VdhJT6/Rwe0asGvEf1Sh7zDmQnJKtEHB5U1B+9cHCCWd
if15Plu0SFwpGrzfqV0V0A7kbW+mvjrnSh3E0zrFlzqws18se62tDDWZGzry/y74v7R7rLXw5be7rLpf8I+zj7
L/gv7iuNF1E//EftHMCMJI1gCN5rg2/gc7/E9/GJ+mLItox1pGYRo23Ty9DmtSJnuSEcDo7NI1rjb0QHoBeo54
dB5dDtF37qNAI5CdVYGGQERwXp0Q4XL6Siyu0jGTpA7/LaQHtkQatqC7KRZbt fzDsk7J16jbI6FfqGo4HZ8NH1
3kLq8Nm02IMGfkqJ2+y6cxbezGuhSXg10Ssf1BcCdKxxKpbvcT3d1ASK1Qytv9Mywu0QSHceojNyJpIxZI+gd0
jHdpcUF2tniGjGkpTL+ru6ErPs2KEtnfqJAbIaAopIVRZ1W4KUNrJzqTibQj05bS7J2CASAD+VzS20oUeP2wC
GYdQ4UNYzpaqG0Kj0TNGep3KceIXpydRA81cqYFYQ32HoVWeTdHCOjGcmX0hT39PGgqYo2uWgp5QoaWF7tPqPQ
mW3cg71IsDYKrn2F1P4LLp1u5CYvZVubcy5CjyWaB1ZVHxpdM76LyTQvw56QjXQBcrGwx5Re0nmq8igcm4hC7e
4nIBnotMEvqGTBC+KVq9+3MzESs11CDzQSgJvxCbfKyXFYhJ6szvmivHicG12u7AeGirdQjdiR9mzRUgD1IQ8o
LiH+oLYhzIUxFFG4peny1fedVF1t1RdWt0Z6bBcL/jZ8nA58yjMVktV368a7b8FbVtDr66miPHwzyhPAW9Flco
uyMzkronAM5nvuaLBmqt1RL4Jqwe5VqAMm1XdAv9TF1E2v2Xt1W5p//v5UuEefD41X/vavIEk010Uduw3SWiV0
Fc39NaC09PrHkfPPJ6KE1ogGCUerYwa4zVIyNGSWeL2cn1f3Adm9RLEjdey33Mr409030tP2afcfsK/Yn/Dvun
WzX7GtrgPr85zXm7Nt81HuXK3q8o5e76XFJdWnEHTTkJ30ux2pXadVtDhJ/I2SsULVzBPVriE+CAZPqLvnZpKE
GNcoKBsalrvEkeb5/GAGFrPpV7JS2sgSq0EPX1XVjdu8Qtf06NdrzVoHdvJo5KTTV19aTHtjj1JA9fcJF0e0F1
KREiQQCJiy1m+T6zEM0C/1eNTRRPhSt3cF5vTo6XyuNJDdbje8rCUI0xxJ+YHDPLxSAzFqrSiEMOyFQULsTk
pa1TYikT+r1sts2U9CU41Je0a92psdcYIIv2m18TB4CGtYqa10ZK0GcfXMytgLhDjkIg0PwYqIM5AktgXpF6R
jK7QPBRJL0y6fGVhiSHm7Y9bqijZejawikzB1KBaKJwHttcED0y641CQPXW7hHnguLb+hsZfSY1eh1D3YIqmnG
pNTgR8W61UNz1Y9NdHb0bS07GZkFIxcgVH1keK6Bxt5oC1G/AoUvxw3KhWU7boeV0ene/toN0/cbEw0ZtvLgk
1ZFKEUCdpPxa4ww4fMuzrFLJCNPHVhrsYxg+Hvme9q3D9x5TG3nyK1Q3u1oZ00JdQvycJD1uk4zyFkH9e2D0k0
1u5070dcqxN1ZSyCQdpKTkQe6j76uHvqHcQps171SCoeoqcx/6+dbD4jj/IptnffCux41Wm57GbmMMgFk3iGG
inS+YunTxKDUCW0rtmm1FcUYjNAH1h3mBTr690t/50s3jfMcc7DCDHE8yBCSnD1La15HE4smeqhnzV94qFmZi
KauuvrS5XqxeuRQpV1pHyiVwoKHTvv1RmXu4Pjm7bSdmAPtKntFu2LnWhUpC43iRBy0HexQrsiIMoxoD+Ws1Ju
N1dJe0/2vXzdiop3Mpp16a0pea1q09t8wfMyzKAPDj2qGKLqu7V1bW5kR6XL3Mw52t/xareJBnLZ39tey11Rod
/b3PJFEJm9i2mFK7yRbeIyktGkzT+beQjyZKK3b749/9Q9c/2c31fb/A/Plt+nyF3/3pu89sgV+64HNj7pVfsw
GE60LnFb+t8wv7JTv7JzvsLvRhn6nw2vxhntkv+drS137Uc4fge+JVAwYhba0vIf4N79Ww/yF209wL+J+3+Bf
w9u/dzV9JC4RN9r77J3+S8PXhneW3ggemf5WNXUBvG050Ppb7NFbGX9sj+mnBQzxQ/8uj13qN10PrTGx0LtGP
/6FiyBjs401Yo86TRscHxzaNjy7axk6NjD2e/NTr2cfyG0XGA2t800g7B7r706P+uaLIfjo6LaFt5dByxJt890
q4yn1+FlsCycHYL350zFmVf2R0LFjEPz861myRf210rFhVhKNjg+Op0bF114mF0bHHInH76NjH8Z2j44DdI14
z0g7ZqvjZ6LjA4C6j4yLbJ8+Pj iMM13sPnr7tjjPHbz12rrX96FxreXGp3zpyR+vomfPPPn5qY7518YkTLXf7b
0vMxtmNM8/cuHn+6LHTR0+f2D13fu3GredPrJ8ZFd7XevTGP1fRo6f/9KB1/GxrvXxuzPrNGyfXzzy9dfqWR18
5+h59/R8079i5c7ed3bewcPvtt88f2Ti2furoxvypjXMLo4f+mdvXbjw5e/rU+onW9Wc3Wt3WwdMnT26c0Xo8v
/DPPPto92CCp91/0jjDjrNbITXPsRbbDuibw/cyW2RLAMQWR0gd+DyKUrT0SimdDUBl13MTuC39QtPn3VnG/j
ewPcz8XkzSh7FvdP4PI3SGy1Dmm7gSf043wd5X655n3uXb/6xL5faNE/vfsvudJc/dQ+0jqHN9Ii4QYcjFrwd
Fw7zW75Nb385fNfPvt/M3p05xzecxZjsIDf293vP0rdwL111HfU1XjK9XPhV970f/f0Na41Z9HPUyhLrb3eta7
FuvhH/T+J3w03h8d/qYTDuBzbP4DB+DU//xvo2dLfCmVuZHN0cmVhbQplbmRvYmoKCjEzIDAgb2JqCjg5MjQKZ
W5kb2JqCgoxNCawIG9iago8PC9UeXB1L0ZvbnREZXNjcm1wdG9yL0ZvbnROYW11L0dBQUFBQStjaG9jb2x1dGE
KL0ZsYWdzIDQKL0ZvbnRCQm94Wy0xNDkgLTQ5MiAxMTMzIDc40V0vSXRhbG1jQW5nbGUgMAovQXNjZW50IDc1M
AovRGVzY2VudCATMjUwC19DYXBIZWlnaHqgNzg5C19TdGvtViA4MAovRm9udEZpbGUyIDEyIDAgUgo+Pgp1bmR
vYmoKCjE1IDAgb2JqCjw8L0x1bmd0aCAyOTYvRmlsdGvL0ZsYXR1RGVjb2R1Pj4Kc3RyZWftCnicXZHLbsQgD
EX3fAXL6WIUyDwrRZGmmUbKog817Qck4EyRGoII8jff5tpK3UB0sb3Wtj0qubcWB0yVz+pFgIfjNUe5unqFfA
eLsYymXntVLhFdKuxcyyL3naZA4yNHaa1YN1bzM3BL3x101MPdyx78Rq8sRe++qjaGLdX575gBBu4YGXJNQyxz
1PnnrsRMnKtGx3TJizraPKTvC80eE6xTF9Rk4bZdQp8Zy/ACiFKXtR1ycDqfzm5T5Z+Uj+dj1IZpUjs2XknHi
/Q94krpC3iWvkHxEukPfEuwPygfiQIx+Tfot8T7yRyKfEpH1IeqpTJaY/nJN3g/yYmN7rpMGAuqR3SQ3e0sFWc
Rc/I+Tq6n0cHy2M5oYTMxb471Ld5NBG5xugzJDCCmVuZHN0cmVhbQplbmRvYmoKCjE2IDAgb2JqCjw8L1R5cGU
vRm9udC9TdWJ0eXB1L1RydWVUeXB1L0Jhc2VGb250L0dBQUFBQStjaG9jb2x1dGEKL0ZpcnN0Q2hhciAwC19MY
XN0Q2hhciAxNgovV21kdGhzWzQwMCA2NjggMzk5IDE2Mia0MDcgNDAwIDc1NCA0MjMgNTE1IDI3MCAYNDMgNDk

2IDQ4MCAz0DcgNDg2IDQ1Nwo1MDcgXQovRm9udER1c2NyaXB0b3IgMTQgMCBSCi9Ub1VuawNvZGUgMTUgMCBSCj4+CmVuZG9iagoKMTcgMCBvYmoKPDwvTGVuZ3RoIDE4IDAgi9GaWx0ZXIvRmxhdGVEZWNvZGUvTGVuZ3RoMSAxMDAzNj4+CnN0cmVhbQp4n0VaeXAb131/b3dxEgBxkARJkOKDwEMUQIAExJsySVGiRPMQw0MCFMkySCwI2LiKgzQzie3USetwmsRJ0ybjHH8knazJtdiG5zdDJKHGeaiVtPnQwbe5pJ3ZmmVdLU7SSZ1G1F9XtvFyApKUebyV8FjN3vffu9933f7zveW8rFfE1EJvQk4tHeejqam+juIAihlxDC9vXNIn1B/xgH9N8D72w8t5H+9onyPyHETcNP3Ehtxz+Rbv5LhITfQ8g01xCjMffeA4MIWb8D8gMJYMzeucUjZLPCuD2RLj52TLfTBuNTML6Qyq5Hv96YcML47TDuTEcfy/2tZgDD+I9gTDLRtPjXn32Jzv8aQnVv5rKFYhAt3wHRm/R5L1/mEukPR2BM7VuCH4Yv/ZiA1NIxxwsarU5vMNaYzJZaqw39v/twJu4vuFcBnWvwfRCdQm9FK2gRiagNuVATmkQL6DKaQ0E0hiJoCE2haXQBjaAGNID8qBc50MMS8knIMSudXAxD25GJ0QZb5S03vDpC009I0K+LWGHv7FHw7yymmTy9kicb3YpfM4TcfIvC/ZSKSJxbBbm0j0SIKPTnV73G8Lf9f1VxExYIVvu34UcXncksYb1qY3I+xBJALraXzmq1d6JK1PPo6fBu3k6atXXRKCZXQ+u22xJqosvc9uI80BHsngI++gSr4GyxCJ75jxEEnoffBCi+EdcSdKKDHkcrsjsrh02W1JGVKFRsc7qsrpfxRofeYW5Y/KRgKTzXg0Tct4zHX2EhElstVmCypmpZ1BNdsj5nemoZ4fseJg6D11cmgBJ8I8ypAmRDmC0hWk6vdodrvI3g7AAJNmwJpV1TY3E6v1ecieqtxDwrPLLreEI+EdcGjGs+Mh0zM7nidoEyhtx7JSSNgB7tt1AFK209yYIfePNFHHj7sCZ3q8IET079LYXsw5tnRSWQxPOa6CU/qfDfQBJ44cwbPft6K1hG7UuHVML0uhT1rYL3njAtu2HMGkJ9YCpcRQVPrZ8qYYLhJZf1qElsquup9EnABF7j00FQV0d0ibvNE+RBXciLAuj9qBzgkXe3W0CDg1c6Edj1KVRnYLdLQK1AOQMyEW5hqTcgndjb9WnQGDB9VrkFe2WX5pbkssqd2CsZg3IHjPRBqcMq0+AZghGyyj3Yu+t1k+Q+4HZ02uwSGZYcNlmoHR5GsrPLZr8hGFwt7c7h3j5cp9PSbyv29J8aHKdf/lNdNz7j0m29Bxih0LBr/f0a2xuW4fb5uYePVtTP2CpuTJhcIxYzaePdTxqF1Z3HXWPD7XduLL/Ivfa7W3u102XOP3S1NHR1+psDAX0jBvs/W5Xy/Qxi9h07Fv13HJgX5NI7PGf/e/VaR66J/S4Rajrd0JV16BxVIYGCf6FygIgJelDu1rMC2avrKWomAISvydz4KsmKHFwQag0AVlo+aWbMZeBB65bR5bCMwN2Txc+/57rl3DW/Cb29/1Xt1/EY/e7gJV6NidW/hnnAs5I5zqAzt1StrhFtjcnovTGh0Rm85ToMvFbh1m4jNtWZvbuNAvIKXiy5A5J1T8ZgQW0wjK1UH10DFyqaaqu9wNzgbQg0DIQDVc7yrs18FscoTwop/tv84N+v2xT0rDQnfwNPjXQ+dXv1WHbf/J04a6wysdtf9o0RzcbvBvXnv3Pon5vv+q/nEJX/LbAfDaQyuzwBOPHKjMmxk3jIPIGFJCEgcM4r9NmWUBwDxzAr36u0umKD6/B/gcy10yiFU1l0fG8FnRB21UnChpkx7ch24VqfkmwCLuWEx2doIaWSiCQ0+1Ncd+OakmVIHLgeVzMHGjwy3xocXnh1zJwfnr3U6FsijK131S22c651nrnz6fNfW1PU/ne6Yv9o78Z5eR7h38gN+KxhQixD/LJhpQiVUFqhlmA+Fyhpwrqw31IRCISyZA7IFoj3Q/MXTX3kj0q9Rknvt0i6mzInvKmR+Jtf/Gr0Xz/P+Ngv85xe4m5aZKMRnhlu8gCXzuj3+/GfwR5KN1C/8pEmmgErt6cGe/iQC9P/uNfjf/Kp+Jcs7/lk7I9/J8Fdvv0ZAFHiFulPiQfEYjj2o7KwkhAHCJbNFEZEYbQGJ02erNHc2jVoUCcUoo1ciLQAoYFCCk1RT500foD1Bh7aaG9dmR8ZMVtMdfgFr/71v1/40zXr1kgtmrcfgzY2NFxdF2Nuo5C5ILg2S1CtVS1g6r2BKSApdkJEXRa5TZAyxbcxUoLwdZdToNcQHEB2Gu8cpsTDNjBS3C4bHbZxg9T22ye/s0Z2+C08XdH+McPj1yqMf32WGdk+0KjA7+/2t083eFf7mq84Ka5dm5zj3PnN56dDeBFXLv/k6XQ9NsGXEuhs0+FniJ1cBwub+VehFqfQuUa6o2eg5bI4Ro91BknoDZaZibmCdR1ucZia6wGG6ApWHf1GmQDH/SBSr17+lnbGoR6x/zH48e01XPcSBi/0W52NpzXam+PUp3H4XIhdLahLCpbKXhmAI/jVBrdHFxhgjJmRYD25DaAsM0q0wEnC5DaYn1poTJ0K5S6m6KhbPYyNrc0Dw9LT1tZb6yj1MUu1QKjLjMgqq+jid0GBmnJaGtp2HXuejfY0o4HBpV2UFcf1P/D9sRJ/Aexa416Hpf2T+iC3vWr1/GYp+cpE375p3hxc0ZdSwPBIXvD5SenhoZ0Da1c6Gr4FE0NyI2x052cDXLRCn0sjMpm61kT0GRuosaawaEypsmoR1ioxniZu5Zwaf64C7P810yWuVwCmkA1U67mNwE50Mfks0m8eCCk6VqpfA7u2y0u1JibH7s0SdoPiysjX8QMuHcce/Kyebznte471+8hrMXWDbM77/Bfxo1e04dp5rfEjr7RH8zjYsRLq9wJsiFT1TW0L6PQ6ycoP/THGBB1nUagJM3sAR1eLp0Hgcfcg6Gc0vz0afeff0Ln7v++OMPQa79+ze+gWsw/vKXUXVPGYa6MaCJQ72Sbi070sTxSKPoKCDGSuuQcYsEbdGQK8GCHVntZVC60D73DV8a7+IryobytKtJUVX751bXJr7Rzh1PofK3UjpBGXUTdDPRCFW17tto211NfooKkWYjneAqp7yJ5NDgDRYpUFum3THkx5DojJKaXr/csnX0C0u1kkg1XS3pSJ4U3JfRMGN/QGrYP2SS09uHi0Fpt9G01q9Qbzi00H6Kh3WEkN0LySkaAs00NYAB3dqk71RMirBwAIMjtu0Gg33eFKAtf+kSjkDGZnn8fTkxekcZd0IzT7q5Et32gqfvPH2vRzcv9/dyFbi68HByNWa1zHoHv7Xrq7UPHs31nsq36B1zHh1rMNX3uaLn7YHxx0krmp3Se1zXA3kcgA9A09vZQvNhrQrkE5IPUGyxxNYi9g6GVR8rJyHQ9IjXvyKO1apUCFDJ2EAowNEetFDzZ3hkM7uqUQ9EEjEep+3hYPhWAGj7W0zRMD0ZeAxyMtPaWfnYwUsvW1uAMDtYr3mt1alccHMWDoxzU8Wmsbuo6R5C02FGKYtSv+6bk4DivUe9p0G/G+PjBm17Hgu9dmH1mavij32BwbNg4vh/305msPtTimrPi3Po0LZsv+SycwN24Y1+nm8KynZ629zoYNrXpTw4SjzUY9r8y3Nrl6kM1s21ZPjmuFypkAf5edZWZQ2U1RqwmxTb2shcuuTUBTtKE2B1TnntwAuxKnVH2DVTbBYbFGwaWJFpqNbkcdb+y0TXQnapyIrr5xuYH5uqdyZ005dEG10pH/QpnCq9ahGnSDUXx6HSjYd47vP89pSa0w771TbCrD30J1XtoTQgQ06GHxk7og5qop/FsAV5LPeW1NNN4B0SbU8mELxqMI1V1kPuuHstkpN1N0MdHNAb0LqCCnV8caHv9quVIffKpluyu3Cm1LnTRjcgG3V4ZXarTc62juhT0B6UCZleEir5YbJ3N7BaqVKsUrp7Rts00hztEqqJ7p6qJP7VUn/2wy9rSdXWowDZsvCVpt1JTj1AY8xrIkt0bVnaq2Lm07d1f6Zr1rwx87VmyAB/Q5BcLSGwK9e09kkutzLuK+2tqcBeC3Tdc0P+5o/adAreCL8sZHab95++cb12rGfcm38P1P2K9+JfP/g3fh0J/8svG0gpK+yYB7/rjtDKHaYc+Rjp+8n3N+gRa4eHUN7aIxrgPt/wrzPoFo8B+MA0sb5U

Bx+x4Eew3+HjNx1kP8o6uXa4FkT8H5E91j0IfQSPoGn8Yfg+wPuOLfDfY3v5Z/hv8z/RPhDTVdzulZTtcA0J31
BeaGHMQfv53DTPA0WU26Qe6hq53zVZgxP51WaQzq0qtI8akZXVVoAmU2V1oCWx1VaC/xnVFoPMz6i0jXwXvZR1
TYL30BfV2kLate8H+SwvIEuq+YGo3mgTZoXGK1h/JcZrWX87zFax+gfMlrP6J8x2gCrPqv1VRojq/bjKs0hi/Y
51ebRKe2XVFoAmTdUWoPatXdUWousOo9K65FWF1LpGnQJznMKbTZ8UPd+1bagCzY3o43UHluS0TXUfJ0SbGf
zejLYz+AK0tQBttip0o022Tz06js18gdH1QLfYXmR0A+0/zuhGrv+I0U3UR9ubjHZRvXYdo1upjN3J6DZGK36
5mUyA0ccZf5zR7XQd+yyj0xk/wuguo0vsIqNPAq21bz06h8k/wehhRr93Kpvbzic3EkUS700N+ug1xK797DrEr
sP02tfLrkFyfmGVT0dFMZ7NFMk6XPLJtVIxmy/4KXc5mksrRu1VDQPGi0kwimL2Xw6mhpha1SY18R8IZnNkN5
TfUF/MDgwVBWHuY1iMvCYCQQK0c1oJhNN+DcyJX82vxHI5b0Pi0vFQiCuWhFYSSQLYEo6VyqKecIMA0Yumi+Sb
Lxq8DTw/YTMsId0L1lk48WtaF4cIdvZE1mPZkhejCULikciSRZJNBMLZPMknY0149uUUcrEQEUxIRJQ1S7Q9em
A6jgvZsR8NEUWS2up5DqZS66LmYJ1omAJ5RQSYoysbTNxag5ZvRWT6SysGi0CEj4iJuF5nmyqyIQqCtTVfASs0
REtUoPzJJujk7rBym2Si1hYP5vmNRoZJBYoDp2IkmlWELJrI58CEBS4FXW81UiqyJpFQQ46WUj4AkuTyzcuHi6gq
ZXIiQy5NLS5MLK5FRkCwmsvBU3BSVdZLpXCoJy4In+WimuE0Nnj+3NHUB5CfPzMzNrESozdMzKwvn1pfJ9Mu1M
kkWJ5dWZqZW5yaXyOLq0uLF5XMQmGVR/GVYx1ksALKYWIwmUwXwMwKRK4BjqrhJRDdF1OC6mNwEg6KQEbtXyV
AqWxmgzkGkipoNE/iJJMt+kgBzDIS+NCEhHzc2tqqpmJKWaIQMBonC6CxkBPXk6BBfGxdzCkBTcaV5MqLECAQi
WXXS2kRwrKVS4nKOKFA7U+mnBETK+B/VUmha+UiaYgvsDOZfN0YTXzqjHOFPLPmz8ryvkMPY1nQAb7Q7EsWC2I
qDrk0itmEvFgopYpJQKBqGSwFubCe3WQK1ZT9+QD6Ccu1qs8H+hLZLciTPF13GU01YwwASNUsS3EApED92EpsH
7GcpFk/+t9ZMK0grNbpITwZN7oN1hXFjArqgaXgKSukSrmpyaJMpUVAp9MSpd5kQeVG1JYQTItloXtU9cKISmx
BkVcf+iBJU2JRVfnoQhrMEfj+Wz6iH7/fVKMJnFKjBcDG7kUPe+nhNPCA8KEMCb0CqNCv3AaLcBrfB61URS1g
FpD29iMRPQIyqBb8Dt4No+SqAhfEejn+ef5F3iZ/xwvoymQyaFtkEuiDZQACQKvbr3wDSJf1Q4dovsP0UOH60E
q3Qe/AzoI1HmwcxXu06BHH68ctGaYrnWVovrXUAkoanMB+auyy2B/BjhLQG+ARArGeXXFkUNrVuTIEUxGdt1xt
+QloKmuJL0BqNQpJueHXxANgHf3rq7oTTAsczAegZe0ANyjcNSiEhm4JmCFDaBKcKeWbIBEDu5Zi1sIHhdBPgA
YHMUigFZgZpJ5QFFJw5wSi1ge0AeIKR15hgIdZ+HzvQhPq/J+4BF4HTqYwdfLYETnFtEWW0tkWG0Dr8Qs1DJMK
D/Gzh60kchWo2tSqRjYnmWVpuFOpe0wTkWixCQUL4rgocjuIotQoWp/5UnFj/MwyjApGkWCfGdNaCSYBn9QzW
9iywiIrNCwaQiU2CrxC3xiyprD5d9X35Lt8JQ0yxNQp8JSdoFovMj4TqweZdOR06x40jtvnYczrzBFu3gnCe8
XNVTd0qltTaFJ08nz4/MsL3IE/uzor7RSrGnmYOWZhgmkWVE1Xn0+sW3FMM7zX2vMR8iL0q8zFuicldhoxaQRf
QRyjVCopzPzVGGH8S6nSJjVeAM6quWWQ61bk18+mwPumW6zRuirVKTPIMjyJDREF4Hp2DtadAr7L+JDoDdsxwW
yJvnKfZeAFk1+FLxxdhFrWRZhG1bwUkpsDy0Ta13FW4L4LcMsxSkmZZzNfLy/jh+pCytJaCUWYS3EuqPGMqDV
XUFFKMRQSRKeIag2uszzcVBGKqj0idwidX7eCu1yXNg5FTFnzaKZV+gnVSPeXIsuLgoqWkT01zHqlQyr9cYt97
+2KqSNW0K5I8ZhkGqNs3RzzPan6IKLHmHzuSIUmmTUHnYviFVW7VJRhngVuCaIgqtWyxfaZz6WVN3389ZX7XB
Ud5pVRey+kpXsK7EdIKXWryKdY88qFh/teffWcYbtqQfxvNt638+ZSeUUP5S4VHpfku03dN+Pq329pMa8oiHP5
pWY1Uk1B+7FTLFK6Qs09zYPeXi0y/5fMpDm1UFFuzf09/OPVsuv2k/yKnabTFdS7eLioa6aPdTf1QwpVONB82H
7F2BOK/jgfPSbxGdmSC4f3U/vn58Hs1EmRbErsjWPZur9MFVierAj3b27He0sh7VWdoKK9souWo1NVvVyQ41Dr
Kotxq4U+3v9VZ5V1thSd/J7Z/rUTppi3VT8pX4WWM8tshquRDT0TmTpX+C//1fsYpV0nGJ7JT3JbbD9TJmfBur
wifSAVjL1fs9eZyd25eSVZTmUZjmt+Bw7MucijN0sQ2ZYt7sCNm6jR/EZGN1fb0o4JvQjs8J5eJuYFYaPaKw5+
nNmqdiJrEPAKVcYhzeRi/AucomtoZyAYx6m9Ra7IDzbuHIW0iJYVtAm8d+CONJtsuXWDyihihDAwY//E4Y5g0
B5W+Zy19yF+Ewfp/P5/Gdd0v4vWhW0i+GZYzff5Gn6f/hI1kjs1LdEhBPRl01rfdqGEkCOSfx3rNY0iiEpCXnv
og0T9xBnFfmpze9ZyNSnReh/wGKk3dsCmVuZHN0cmVhbQplbmRvYmoKCjE4IDA9b2JqCjU0OTEKZW5kb2JqCgo
xOSAwIG9iago8PC9UeXB1L0ZvbnREZXNjcm1wdG9yL0ZvbnROYW11L0RBQUBFQStGcmVlU2FucwovRmxhZ3MgN
AovRm9udEJCb3hbLTExNjYgLTyZ0CAyMjU5IDEwNTBdL010WxpY0FuZ2x1IDAKL0FzY2VudCA5MDAKL0R1c2N
1bnQgLTiwmAovQ2FwSGVpZ2h0IDEwNTAKL1N0ZW1WIDgwCi9Gb250RmlsZT1gMTCgMCBSCj4+CmVuZ9iagoKM
jAgMCBvYmoKPDwvTGVuZ3RoIDMwMi9GaWx0ZXivRmxhdGVEZwvZGU+PgpzdHJ1YW0KeJxdkctqwzAQRff6C13
TRbBsJ3EKxpA6MXjRB3X6AY40TgW1LGR14b+vZpS20IXEGeb0ZR5J3R5bo33y5ibZgeeDnsrBPN2cBH6BqzYsz
bjS0t8j+uXYW5aE2m6ZPYtGaayZM17yM3eLXx1UNMFH1jy6hQ4ba589VF3Ie5u1n7BCMZzwaqKKxiCz3NvX/o
REqpatyqktV/WoeRPcF4s8IziNLYiJwWz7SW43lyB1UJUvGyaioFR/3JpEUsug/zsXZCmQsrEZ18Fzoh3W+Q8c
o28idwgb4kzgbwj3hbIBXGRIe+jfoP8SJnyIfIpHmKevKpI1MPx9jPDvkU/XPkJnqiTyoin5Bj/0V0w96nwrH
xLj/r5PLmXFg1HY92iNvTBvjvge1ksYzeN+4VkxgKZw5kc3RyZWFtCmVuZ9iagoKMjEgMCBvYmoKPDwvVH1wZ
S9Gb250L1N1YnR5cGUvVHJ1ZVR5cGUvQmFzZUZvbnQvREFBQUBFK0ZyZwVtYw5zCi9GaXJzdEnoYXtigMAovTGF
zdEnoYXtigMTcKL1dpZHRoc1s4MDAgNzIxIDUzMyAyMTQgNTM0IDI1MCA5MzcgMzMyIDU10SA1NTYgNTU2IDU10
SA0NzYgNTk5IDY2NyA1NDMKNTM3IDQ5MyBdCi9Gb250RGVzY3JpcHRvciaxOSAwIFIKL1RvVW5pY29kZSAyMCA
wIFIKPj4KZw5kb2JqCgoyMiAwIG9iago8PC9MZW5ndGggMjMgMCBSL0ZpbHR1ci9GbGF0ZUR1Y29kZs9MZW5nd
GgxDIyMjY4Pj4Kc3RyZWFtCnic7XsLfbTVufiZ+Wb2cTZdkMWCOQxCQY1LkkM81bZPDYQCE1M1kC0tN1kN2Q

hya67mwRED0oFxNKLVQ1qfVKLqGgtWqoBqdWahW47b9i/7W9Xmulf+u/kXJ7aW+LMNzvnJnNbKgCPi4v3uzz
0fMme9879d8QDTc6SdJZB0B4mpu94Zc184sIoQcIkRIae6KKrPhtP4I17/FX7U1tLy9u7vIRwjgPX1qedvqlsd
mvtNFiLSZkBGmVr/X17t78wxCRt6Hz6e34sZUtc0I94iPXNHaH101x/DzL+H9cbz3tAWbveqEX2cQk1qG98527
6rQ0+IiCe9DeK90eNv9nT8MV+L93YRc814oGIneQG47Q8j1ufY8FPaHWo5Lb+M90pNbcU/AD/tJwqWB3YsgyQa
jyUwtScnWETZ7yshU8j/pR5hG+shB/LxCdpGHhJ1414LbN+H0dvE5sp504s6PhYPCJnEK7u0kx8mbCLmRHIdE
hEWkKm4S8jbskh0CB6yB3HME1KFwuAeDRKQqaY9UK/VJH0iHyQwpIh2WGqWIMBUelevlnfg7C34ippDXSRbpE94
1EbIPPoSpsF8qk6zkXTgMu8jvkQraG2lsITvIGuQ1VQiSHnGNWIs7r8mHyYP4CeLzw8LDwpvI3T7hDvIWuR8kc
T55WHgL5TpI/kruAI/Yg345VWxB/19DXIfx/IMkIhH5LYESVbwS95B7pNXEvzNgivwW/xwnPUjZQ3YY+gypxvF
IhWlsp/Bjod9wL9103oQvwU3wG2G9NF56QppPtmgagEayBXE/yM4YWoTVKDv7rGHYxW6pUdhFPpQajU2I+ydMI
qS5R6xFiVrIfvztNthQpjnCetiEnLKnGeSwcYFu0cRg3EtSk1IEKaRFbhaQ54hz5Ep0Eu2ICYur2GG/Fc8+ZD
0Hsq8Rfia+FdyGMpIHmmRjqGuCfp5LyEvGA2yBKJAnIptt5hb4dvtun6p8t0G7Cn0IbeKzajsJjW7k1crfWf01
CyVxskNu+x03ZBr2i3ljn/vXA/fm+JcWLNU2X3aXaZjdTeW4V7dUlyy09zGfxCzf8aI7pZz8U9F426luVW5y3b
X+N132fyzp6DaSIvaK7XIOzAbGc1YV5L0MTF8LJjkH1EiBQe09F9FbEf6j/QXjrRn230z7dktEjkVgXGnfq/2G
q1/+3PYkIfx/zrmgjnyW8RCnK5U01bx0xK5jRqMkC7PNAvpREqyneovwj9XkYJrjuKi8LmaZEYFNnIqIp1qH4/
f419/Q/y3N944nf0G/Nbph0TfySvF107PYvzdd0Z9qQdtneRgkZDrCuIQzBtMd8q0JwV5b5Lw4pi9KX1Jm9PH0
USTw0QWiikj301I7UT/AXvKLGT+6In+o7Zj+D1xDDcKXX1zM0Iz2zN+nnE8Q55L5gpzxbm0ueNkp7HAVGB20iA
JCKEx6Ai0My+7SVgm0LizhalF02c4rMj4hdhtZGoRMeYL43MMRqnn1HNJh19Y8VpT889XqifU14s8U78TjH3iY
3c+uNcqfvmG1167+upnJjuFmQIVRgql6jsHtu155mGWMV/Br9WGVNR5zgtkmyiYSLmETKN6+gtdyTbZJdfIjXJ
IPi4buJbGv9KHgfL3/pg+01EfjKaLHCNNOxNIXuT+1I2jzGnjLgeUhzuMVx8XTTbsULX+L1pa8gaQ4+xx9Rj7
qE91jvJPck91p4RPbYe+5qU7WnH0+woaY7BkToKRZ129YSJRUzS8TnsW4zc9/RTW+99+u17jwsp6rHj/67+SbD
Dux+8/voHf/jpax8+pP5U7Vc/QuFnoYypwkyWAAwFwU0N09IckYY2rQh5rkM3ULI21zhLLVQcK4gWCzXYjSajb
Jdkk8ko2kFMQmg7lpESKotgAPJdiyNQs0mLzgsRpJs03j09CzmQv1FRaMnFaK7G23yR0abSf+V41dt2ZDzHLG
in7n+bBANskgJf1LoJPkKqtDrx0vkq2khrRQXySXURRvEFeJKeT1tpGvEHvEWuUdeR3vFrXKGkZhFExDJIBoZG
ASjZCImo5mYJUqTiHUs0CSHKS3JzLWkbFkxKEbFNN58Bc21KFbFeo04G6ZJU+VC03TzLMvcpEJr0SkXFoguyY0
2LjGUGEtMLpPLXEYXJbmsLutSsd7UkFRjbRGXg1dqkhsNjczGk8/soz5LN+kS1oiroFuKyqsNq43dppBpVVJPU
o91g7gR7pQ2yevNd1m2WldJj1i/a71xGUHHMqvsjzDeLIwvOyTMFma9z740q5tU9Sfqj1T5rY9TpGPs9+SVs3
kcW67fzjTF6B/2UmhK9WAerdBVK01z7fSA0oenM7dQBHsnoXgeOHGJBtadm5CMjRfQhHg0D0ipjuswbEaNh
gVZFc6Hht+7d/a+9SPz02FPiv3gS6efkxqfaWmWZRLz5xswFmwkncx0pY3dS6ype2XTZmufsA1GS8Qkzr0nWn
Z3KuLG0H+oyc09Ns0Hct8vjFzXeb2TEDKL11MwxjNEXmMjhJivowMwaN9fb0fveXgGXlm4C3Pnn7t8XvuuEKe
+55HJ4Xv/z3/1d8XqFMM0GnzKs6Dn7wwUH8RT28hz3WDejD1LhchjLYLorb5duMzLwz1GvB51LwGxHFu4e4Vm
6F4FdMxv6D2gBXHSCpUt0zoYcYc8IGCGJy2Zk2+VpuZj5HnmqseB9QPC/ISw4tWOXFJnfN//kW7u43rFySAu
qaTh10T08a0gzhpd1k1mWSmzfst+xvD316xJ2EMRGRYGmj7aBAXWycLfDs3D3KM+Nc3enem5EXuDMyzMbMG+
//DLPf5yfE/0DwSj/J0x0t91Hz0LuXEWLpXq53nizdLpcNW5jmhH7izRpLbaa9CjpMnS0jYyLpt90NqTdpv2c
benP0GeGGdHt8pFMaZNjz0uEzT1GqcxW0u0VlPRQLCpeeVUZUT421Tvos3f0XNVTcfWfoHIdV9Y5p6YteuXd3
C12e3b6vo7i0pPXRV0R9+9KXHQhnqH7kf7FPrpR3od8wPrn0Ni/1Bn3Wz8APYn4E+MI97QznhKIiLcMdjbmc
6z5wm8zJWFZ7oAHYgYTx+dMEDijWn4TIuiNz645RM6c0bTmWXEmesPj7Pej088Y6C6fV92v/g0/+73CH2P0w0
zB+2DHQexCKWuctm0wkp2kIzsIkuCKIBdxDR1R0hqN10BXSwUU5zJzgmuhBo1ATPHDzELaSvRZEhiDkQ9zInQf
Db2Zec2NDB/YgZla82eB+yx1HeuzDf4nnp3fipJdKzkoBPotdJvLG0xLiUttAu4WapyxilX5Nupw9Ij0jbjPf
Qu+104Unpu9Jjxm/T7TSdYguP0duC2U12mMda8mCCnGuebFGSMX3ADP1qI8tnhckVUC67zQssruQGUi80iA2wR
K43NBjrTfXmBktNcjB51dCT/A3hPuNTwg7j7uSfJb+bfCa5AJ3ILLKkxPOT5FNXCrvveVvep+94WvqeG3xbyhDy
p8fS7p18R+tT54gJx1HqTsIX5C0szdmNMmjA75bscZKv5NmGrzStaKJHTkotIullK4XnCPgt/mPej6z/XOFlgG
YLX7+zcbH7NE4R7TwjThCz1PfWgWiI8Ijwn9Kqtao3qlQs+7hbGYDjzCqN3qtvUdeqtai/z0T0/Uet53baQEaT
M1W4Rjct6Upjxo/wDsj/pWZvJJhuqkwVTEim32U6hrY704j1IP6/EGI/Yf1hsdpe9xt5oD911XttTDVcK01iuQ
pff0v/t75dffFaj565MaP978y5cf8j5gmpQhj481mCenYHxQkkv2Y4bIsow2W8mTow17rXZ1Q9a+9L3j++ybRye
R0TAm2WyyZIEp1T0BtXHoCIaLnavjwNETp5CVV7Efss+yz2L9UEdhrMfmYVahUphdmDN3oivDlenKcimubFdOT
UZNZk1WjVKTZNTMzE0cX3GxsyNWRuVjdnrc+6euH3i8YmZsa0xQ7EDjZmNWY1KY3YoM5QVUkLZ6zLXza1T1mW
PWZbQaWvrzNCT9oRpV0+fmp2YTkaJL7379G3BB/b29c3df+fTB09/LIiPb2t83uN/6Yb/0C50bVnTFH17T1716
dt2tXhfefQHL6f0fDU/f9fEiadYrNrYuy6iYg84JqI8WmQMTfBwC4S4HPrjvFZApLwXWLA7oLIsmC0HTkwWku
YwwYY9hI1JtZLNBGYDPOkefINsBbuAMx5RtEkmQ00IVUcK42VJ5MjwgQxT8qTcw2KaSaZKkwVr5GukWcY5h034
BYrpAp5ngHjxdA1BqSAfDPW9S5xtbRa7jSsM91PthnyMD6ytcotLjj96pvC28Kvf3H6NazWo6UPT16JvvAM+sJ
S9IWR2BXPcaWDmYBVMGy02jdTYX9qXxLaP9VSJWJb6WDhMEur10eLMFnaUzCB7Ak6XnZgwR6JZVMr1AP1c4LwT

N8tt2z9Tl9f6b0dr/xU+JmwT9x52vvIIy/tEND8vP3plubj8ATT8Ubk4VGsVxPJra5rkpNEq0XMzMo0mUUjFb0
yMkuoJTNLcgjE8a3U+8ZstUtbyX25X7dvnpRJLVnjjCRnXJp1ijEtNWeS7V8P9J/qP8rCY5bWsR/FaPnrMdur9
lius36ES/2ChhiBwbvvs+ay8grzqPMAeXWs7UkdlCZmCI1XKzpkwcRpv29GjCjC0p119BXqUND9y6CuPfa97582
/7/q0+oHK/60bk1/+Dv7Nz645ndvCKP/Evi1v0Mn6av62r2Z6Vd+fbzb/+2s0Dn7vI7b+24JWvM1JefevXoB
L0/knxc99NdaXHN9yXtpwJqvIrFXnkqq1GsR0H7zoDSGx0/Y0oXsDpp9SjeKEk+pvSn9+4t+V7nK6+K005/SXz
4kYdf2nF6I7ZKft+fkG7wzPvwGtKdSD7Q9V2nq7surm6yQdgkpW5wbBqz1y7tze2L67t2nM1qNKXmuCex/u3II
H3329S/HLP95VjKYH3H1E0mMr9vT6fp1vSkfHxJclqcSXPMc+gcy5wk10IU4QpxEp1kmTyyILXAMXnUpMxJWX1
KXvYVEzfQDZYNSRuSU1gFFUUDNVggCZLBCiPABmkwFsZBupRhn1iQNzfvK3k9eevy7s7bnnc8bwgW1DDW5Yf
7Zhp6Py4KtVT9ywaVPTfxMPPPafv7rhx20tr3pv3+x/yvXU/b/915Y90txnJk3yeFwV2dbJD2x66Pnx41+aNq3
h+oU1uS0u2Hr7w09navUdc2zDA3ev/+j0V0Zc8xeSZeKznF/8Kn1rbK7Ds9xp4Ens0zxYQ++6bSrGYNjnyHjI
Id0mLQYJPK61E9uEmeRV6Tf4/Wre05DUgYLyD451dwkjybv4W8rvdJ+8kcfKt+XZx15jfSB2SNvJrY5BTyDD7
biNd90pV8TpFDatAXfyVMFiaLJtE1bhYPQBncJ9VLvdIReYX8a0064X7DEm/cY4cZBLmDcIriY08wCSQHOIov
LJZ0FjhugG+7x+QQcA6d7++FhHu2/oacP9xfS3h+j19LZMk8gN9bcA69bq+ZtX6TX1tIRnkqL50TvmmeJseWsn
VIx/W1zZiGfkLfW0n0sh3kKIgmZGhwpHv6WuBjHLY9bVITI6J+hpwv0BfS7h262uZjHHcqK8NjNUR0dcmkupYq
K8tZLbjSX2dnDvb8b6+tpLWORn62kZGzVmvr+3EN0cbpcHQ6nBgeWtUmdScpxQVFk5Vm1YrJYFoJBr2e9udSkV
Hc75S3Nam1DKoiFLrj/jDXX5fPj3r6HR210Pta18R7FiulHhbz3GwzL/CW9+pNLd605b7I4o37FcCHUqos6kt0
Kz4gu3eQEeMps7bESkJB1cm3CYs6/3hSCDYoRT1T52mbScAtAQ7kGoUhWiNRk0zCwp8uN/VmR8Jdoab/S3B8HJ
/foc/Ws7BGA9MigHBlUkRv19p8rcFu/PylQvgOF+Z17Y61BpRAu2hYDjq9ykt4WC7Uhz2d+msxGhwDXVqGkokQ
2mc0krmVTTWBtRMp5z3h55tkAu2pTKEciBCvUo07PX5273h1UqwZSgWSmv84fZAhKs/EFFa/WE/01oe9nag6E6
UHcXCY6gx1LNTiQYVb8dqJYQGwwPBpihqLIAq8CrNyDRFyGirP6an5uZgewjBGUC0FbGjlv0dEdReD1dJTh4i8
yneSCTYHPAiPeoLNne2+zu13ijjpyXQhkaaxDDyA0pdsCXajerPyeOchP2hcNDX2eзнаHwBFCzQ1Bn1Mx7ooAN
ONHNzW6ePcdIdiLYG06PITHtAJ8QohDVVItr0CM1zcZxKu59JtBmDRFqdCTScjGZBMKxE/GgHhA4gq7r4Q0g5
hBtiCk6SjXVcULdrehYZx1gZmjPdhcgQT8/6AsqkaBTiXQ2rfA3R9k0k6812Ib0xgRqDnb4Aky0y6xKPYj02xT
s8nMJNC/iDAw4QucwimaIaLwMKqG4B2jP1Eirt62NNv11rSEbGCxeQXIG09Avwkp7M0wfVmwlujrkb/EioXyNq
cFP272rMVrwuC/QEmC05m2LouvhApF6fT4uuuaY6FqDeMPLV2eYNU0bI548E1ndwNpZrsYqHmId6mxFjhJ2I8RM
ZSomhpEiAK8zbNjwC/UyMjzg2ZK+jbbUSSHbzysQJ+9nfWHFYtowgRTK7xMLDjz7nD/ND3cGwL6LkDMRhDqMde
0BzWNjmcJWhZSr1eGnyYyQxrJ1oA6aTrmBggDH/qihGj0INhTC8vE1tfvZAkx0xswWNG6XVG1VavRHE608YpBP
mdXHv9imdhT6d4Tir1DOnSXg+q0aCbSyqudmYkbxKG8seGCsxwJC3eaV30QqGcdgRpMxVP51TDSKFCQtZ9Le1M
Kbmu5Xy6iqPUl1dd711SX0tWKuqUmtrq+ooyd5mSU1yH9z10ZUmFZ371Yo+CELXFVZ4GpbpcKa5qUBZWVJU5Fff
Sm1p3XR2tr1UqFtVUVrhxr6KqtHJxWUXVPKUEz1VVe5TKitUVHkTqqeZHdVQV7jqGbJG7tnQ+3haXVFRWeBqct
LzCU4U4kblapVipKa71VJQuriyuVWoW19ZU17kRRxmiraqoKq9FKu5FbhQCEZVW1zTUVsyb73HiIQ9u0qmnrj
Mvai4dqFTQWTVKHKtwkHykUvEobjr2eG6+cWV1UpJhafOU+suXsRgmXbmVVUvctPy6sVVZcWei0qpcSNohSXV
Lo13LCU0sriikVOpax4UfE8Jk6MCAPTxImrg7ID89xV7triSqdSV+Mu1WAL1GNFrBvUwyFR96iJSs5uaXVvNfv
6xbiBcDESTrpkvpvTQAGK8U8p54yLX4XiMjye61rPACtLKurcTqW4tqKOWaS8thrZZfasLucesBj1yYxXpfPLb
MT2zvYOhGKndQHL3MWV1LC0sYEbdBAsepd7VbM/FGW+rQe3lhp5GtVyp5N7rZYE0IXndWDgant8iWUJI4tXHS2
7xQs2K8d0Lfx9IHejZVIS72+Lj9mwAhLJcEwDbJk0h2I8EjHETge1GqeEvG2ITE8xaKIQ2Gu9LbhscgAm4MCi
saKYSGcwCPd4UAUk4ni7cTdc0BmvQyH9TLFJVDiEjAq8eSg8R/2R0JYpQJd/rbV+QgbZrWMcxLowF6tXRedq68
50jvWKKSV5Ry5Lxi12NH1K5TyuuSW6cL7WUvTx9EtT5IuZg+iMb7I0U1i+yB6dh+kJ/lmjikSqnDNKjxh0VeS
q+kxHol+sXolahmh0+tV6JawF5Sr0QvY69E472ScpG9Eh3UF1xEr0TP1SspF94r0YReKTF8B7VLWM8xSVyudon
q7ZJySe0SHcQuf2+83C0T7Qgq19wy0cvaM1G9ZV1uvmWiQ1sm5WJaJjpsy6R8kpaJeorrFy2oZmwXz7+o7ojGJ
b+U7ojGu1PlUrojmtgdKrfvHuyPlUroj5qyDAmWg8aHnbHyUT9D40PM3PsoFND6UNz6De4d/3NBEY/Au3jT
QfLzkn3dyVdAdWBkoCGAGWZUfag0V6G1sy0SM1J1gCZHVJEwCZD1pJVGikEmkmeThtYgU4mcqrpoQQiE1CBM1E
fwNEz/xknbixN0K0oHw+bgqJm34UUjtAK4IV/Pj1Y9nuvDbh5D0AqhOH6DqQUpdSIv9s7U0hGZ8ePHMJ6NYhqs
VeK6edCJEM8J60TY/P+H1EimIpQ0/QwjThHgDCKfg+SBS9/JnQ/HUcSwR5C1In5XneDr8bj3nMIJ4g5xqEfI51
UwbBD08hhZ+QpM1qluCyR5FzmeTAzv4dPguhM9HuCBewyinNn58Nc7nzEYcfz5QnY1vpIWA1s30njHd+r19/Ki
1I01GWGaNy6NjhmkEp1mNMK38ZACfhTjfUW5PpoEwP8E8gGhtGqKVoxLEfahzkA+dSxr2j6mGk12zmRdXiv072
5spmXIJH3pBEXL543J4e8d1DuATyldRvs08rJ3reiXuBdEC/4gXJ1kNx9f0scW9P8B5auXP/LpcyzmVDt3qTt3
umrU0apqPaf7s5HwFufU7+PmQHmEahSBijeo+FtC9wMtxaJqmOs4o52KoPzVz00aHGvYYBgt8a75sp/Hq+Z70
Q1eksMtx876+DXC+WrgM15dPsqj0Bk9tJ1jifInMf204KpNj6RJAzzGKbC8wviPov9q3s8oxnXCdkI8anxIoZm
fjnHj4xJEua814dMof6rRo0eh4NSjuRk56+RYNJ10cx9o5Vknqumne81ShSTITzIKzVu07k0nQnWYet2bk/N1
jQhg0TwtPMccjgH5CzgGUTHmLV40HAhdK00tv75pY5pTuM2NODRUc5X30viEnVzfbRfEIVYNLTwrN2hS+hPo0j

j34y6k1+ZJ1YgRDPHp8HE7Mf8uE3PbDELNPaPs5xQ0d0No90j86dFzEGeWaI2yAxF8U1cHYm6ED4qB4NkUGws
ViJaywxBySeU7jMXs455b15sK9p2tBqifc89gzyKqfotm/n13j+uBBbRHk1YpXTq0uUP0hT5zvLdLJary0adab
zFs6jT/ekNu6n4YEdjV0mU1+CzR09L1ZBvbwiBnj0aON3dEAiH+eU2asjQRvLB9VVjVIsh3q592i+G6MxVD+Rf
yhTjEuqSxD3MC+30YVzMJjOUH0Mx5tTt3cbPxc4RzanA9YJ8zzr5Xk1jje2ExnwyFi8DK0efj3P+bkUMUrdXCo
fP58zTD3MGZB76AmKz2LVNifBy7SYqRxSX5p4vAcTe03U4yDmJ134NDCMxvxkFddzhx7JIfxo1cvLM6p/4ESi3
TWeYzt02Ehp5R1e4deIzq0fe9K5/CSW64bL3T5eCTq43RP1NZxWaYLmEm14sbEa4VkzVqvj0RaLJNY5tA30HmH
9xGCMi7RK/F7uW4xrR4yf6IDWfXTzFTnlqj5GoXg9bBjQ1n7g5nWpShXeMTjXeegS7CNr+bMK3F0wj6vFJ
/V4V4a7ZdwuxfwJe57Do3EJrhGarkY49Jw10I3w92A0wy3wu/Z3UKER0Jc7KybLOU03IitDjmrxjXDvQh3K/H
q1uHYiVLcWYZ3bD2PsC5Uo1eFpzw8dtg5xovGqQf341QHc1XBKcY4W4R3tYh/vv60GHFXcHyMfyfvj9i6SudT0
1wtx850xDAnzKXIUsw/Y7uL8VqDcHVcn8VcZo3bKi5D0T7XZHFzDjRLaByV4rUGaTOIeciXh2uBUfLokE5uRyZ
PGT/PqC7kUBpn1bqV2Tq0JV/XpcYH03/9AOU6Ln81fhQuvwd3PNw2xYg/hjfm0/M4BsY35dpYz0Ur5nqo5hRKO
BzTItNn5YDH1SZYpZTr9mNcV7GKRVzjdQNK0kMW6J1hvM00kBhHpfPzTVVyaHrUI9uhK8Y2NH8sYLLWqrrWs0
p+b3mE5UJ2i31MjLLXo9U3bpPFXPdDzA2WkJ5z8uhWaBYv27NEFncetX6daN8ePh1D3DaGUJj0U3hyrmq4bi
JFyHr+LdM4XD3hYPAcs1v2zeoCzwfqNxVEM7kJyh4YrRnuwBcu4P1XqHNYNaE0DoOfBq+UuN9a1Zv6eEx3I24M
rd2LXG09GE/t0Z0KuTewEtCw8j802D4GL72pvS1rNir/rJPZuw71hx960tV4+1vXGuw8td2vvRIldr4/351oPG
BnoSoK8Dww0dCbd/Gm8pof02Ulw0Hseo+z1td85QctWi+K4tL7Sy7sFRi0yjDbPXaHoWW+GIV7vNSrdfB3V0xM
mX6c0y/ZvHvI2HJv/nG0DZVgbxGQZrnNI1H+Y2zukv0sFuIZZP5mv4w2T2HtZXcdMA9pcrX2I1ePex7DNJk0nC
kwHyxM493FdU6LN6BhNyvNVbMb1+U+dLvd9os0D6KD5kFD069Pbx5Eh50HKZ/xPIhe0DxocCffnMBTfNYRg7y
wCepwExb6uc2V1LPmSvR/50oJc6X4h0G/51yJDqqwn99ciQ7ztvZFmCvRYedKcYk+m7kSPc+84LOZK1HySedK8
b91upxzpXi8DZ4rnav6nnu6pL2fa53EF226RMng6dLw043PZrpEz6NdJUGDX+wpE+U+dnY3891PmegXeMpEh0y
Z4u+6n+WUif7DKZPymU2Z6CeYMif2pSJch3UI9YFnFtN28X4/L0bHdFhb55zY7oWbMj5X0bHdFzz07iM6BPf
3ZEP8Hs6Hx4P93ZUSyznruinD3xoRcx8Umc0lz0iQ+9pInP2e9sFzfxoQkTn/PNHS7HhCZ6Fn4XiU8aKKfD7vI
v4d9cFXC9rMTfAs6bj3dN+bx/DeHe4G7s/P/mjP+fZe3/cN9Kbidd/PSJ61xnP1bhZCr8PRf+VgT/2Qt/tcJfV
Dihwn/kwp+t80+9cDwX/nRXsfwnFY71wke90H8S/ngS/r8KH86GP5TAByr8vyL4/dE6+fe9cBQbj9bB+78rkN8
/Cb8rgPdU+K0K7xbBv6XC073wryr8JgV+vRbefhF+pcIvEfyXa+GtI/Pkt9bCkXnw5i/GyW+q8Itx8H9U+LkKP
1PhX1Q43AuHDmbKh1Q4mAlvFMHrKry63i6/mg4/GQUHVPixCj9s4RUVX1bhhyq8pMIPVNivwosq7LPD3g25814
V+154Ue5T4YXn18kvvAgvrJ0e/36u/Pwy1x143iV9Pxf2qPC9Xnh0hWdV2K3Cd1V4xgffscLTT+XKT/vgqV0p8
105sCsFnkSmnzwJT6jwuAo7VXgsBXao801HrfK3i+BRK3zLB9sRZhsvPKLCw99Mkh9W4ZtJ8NA30uSHfPCNB23
yN9LgQRs8Q0F+Fbb1JsvbV0hNhq14aGsv3HevVb5vEtxrhXt0wtfvf1H+ugp3b1km3/0i3L102vLPufKWzbDFJ
f1zLnxNhc1fzCz3q/DVfLgLxbyrGDbdazE3pcKdFtiGxt9sAE1tSEX1tvhn1S443a7fIcKt9vhNhXWqdCjguv
MrWvXyreqsHYt30KDNR6HvCYXb1ZhtQqrrNCdBF0U01WIInoTISQifhJt0Qkif0AodKrr1w0oVvthL5BV1EFChd
S0sx5sWFFwq+FRoVqFJBe9saDwJX06CZS1rcqMINKjQspXLDSVhKYcmoNH1JEdSrsBgpLy4BjwPqBjtcNwZqU+H
6BSP161WosUC1C1WLbHKVCotsUKnCQnnyUIUFFTz5wUioyEiWK2wwPxnmqVDeC+5eKFOhVJwi156EkeheCG4V
JirwnXXpsjXpcK114yQr02Ba+Yky9e4zoyA0ckwW4VZKsyckSrPPAkzptvkGakwfZpFnm6DaRa40h0mJkPRVRa
5SIWrLFBYYJELk6HAAv1TzHK+DaaYwVKEV0701a/0weS8FhlyLuS1wKSJufKkYpiYCxNyLfKEEZBrgStUGK9Cz
gjIRjmzU0DxQdZJyEQRMn2QkQzpMF0FcadhLE1kIY3aSqM8cFo1NRoFUbhoVFp4FAhVYWRKqQgQIokdptVXgK
2tTDCB1YVkpNGyckqJCF00i1iwqEBtYFbBhGAmFYypYPCBhA819AAH4C6oI0K90AUEGxAvHd7Bt/5rwpX/HX7I5
83AeX8y/gvbpQKHcmVuZHN0cmVhbQplbmRvYmoKcJiZ1DAGb2JqCjkwmjMKZw5kb2JqCgoyNCAwIG9iago8PC9
UeXB1L0ZvbnREZXNjcm1wdG9yL0ZvbnROYW1LL0NBQUFBQStEZWphVnVTYw5zCi9GbGFncyA0Ci9Gb250QkJve
FstMTAyMCAtNDYyIDE30T1gMT1zM10vSRhbG1jQW5nbGUGmAovQXnjZw501Dky0AovRGVzY2VudCAtMj1C19
DYXBIZWlnaHqgMT1zMgovU3R1bVYg0DAKL0ZvbnRGaWx1MiAyMiAwIFIKPj4KZw5kb2JqCgoyNSAwIG9iago8P
C9MZw5ndGggMzE2L0ZpbHR1ci9GbGF0ZUR1Y29kZT4+CnN0cmVhbQp4nF2STw6DMBCF95zCyz3QRYSCEVkJIKQk
Si/6otAcg9pBaKsYyZMht651JW6kL0Dea98aPMXhdH1tr1vjVT6qDRQzGag/zdPUKxBkuxkZJKrRRy62itxp7F
8XB263zAmNrh6kso/gt90bFr2Jz0NMZ7qL4xWvw17E5qPuQt1dnfuCEewiZFRVQsM05jz17rkfISbXttWhbZZ
1Gyx/gvfVgUipTjikmjTMr1fge3uBqjSyEmXTVBFY/a+XsracB/XZ+yBng1T3X0V0Cxe58gZc428Y26Qc+JUI
u+J8wK5IC5S5Hw75Af1LME+cBMMkfW05yamT1c0Q95TzzngNww45xE8r17ZM5fY0aE8+cZ8i3/CZnzFxkt5Pb
1uBq8u5+VC3X1PqybLpj2jBs2FsTvT+Amhz6vgFvv5oEcMvUZHN0cmVhbQp1bmRvYmoKcji2IDA6b2JqCjw8L
1R5cGUvRm9udC9TdWJ0eXB1L1RydWVUeXB1L0Jhc2VGb250L0NBQUFBQStEZWphVnVTYw5zCi9GaXJzdEnoYXI
gMAovTGFzdENoYX1gMjAKL1dpZHRoc1s2MDAgNzUxIDYxNSAyNzcgNjExIDMxNyA50DggNDExIDYzNCA2MzYgN
jM2IDYzNCA10TEgNzcvIDI3NyA2MTIKNjg0IDYzMyA2MzQgNjMzIDUyMCBdCi9Gb250RGVzY3JpcHRvc1AyNCA
wIFIKL1RvVW5pY29kZSAyNSAwIFIKPj4KZw5kb2JqCgoyNyAwIG9iago8PC9MZw5ndGggMjggMCBSL0ZpbHR1c

i9GbGF0ZUR1Y29kZS9MZW5ndGgxIDE1NTYwPj4Kc3RyZWftCnic5XsLeFNVvu/ae+2d97NN2qaldPdFKYS201K
ggJi0TUuktJ02PCoykDZJk/SRkKSUiKoRxEBGQTxMYwyDmdmDuPR4jgeFGTQqXp1ZLx3x0P16HzHGXFm9FS++
+D4zUXY3P9aeydNK6Lf0fd+537f7TbZa6/1X//H7/9Y/x01ERvyIx3aiTB9Ax4o46bllQjhN5CiMno2ZoQLv2
sJgTjj+BzKBDtHTj3r3d9iBBOAM2e3v6RgLjrybcR4uDRXBL0e30jz4wsQchaAh0LgjBRIw4q4fk2eC4JDs2/
UXNrrIfnnfDs6Y/0eJfP+20pPJ+HZ/uAd1v0NFMPzLLgEQmD3gF/7Wfvt8KzgFBhWTQST9yK7ryG00KVZD0a80e
bFIlyeA4ixH8P5hi4yJ80hgryzGK0VyhVao1WpzcYTeaMTAv6/+mP0aA4YtFJ9Bu43oFRCwqjLWgX0grjT9AYn
X+ZU5ELzt7nCe076CRTDnQsuZhKYM0iM8CnBtY+AfoAPB9F3XT9Mj5Hr8fwOXYYsbgdRu10x1F0Ei/j0Hx0+tB
dv0E3oRfRcTLmz6HDQNeB3o0rHrjf6j95m70THmXbQd7UNxElUon9Hw74IuYdTNv0uv/4Zuo5LJXJh/V2EBS
WGw8zRwPybNM+VMB96MAxBrY2DBcbwSZu9BYW4zXGX0aqT2STaw7BjI1+1F59jb2HKujDkOcoiMc8D/0LoJ9A2
ApivhwxL98afoK1Y4Qzb+ZXSL8haFj1EotyMPri7hGuYxRT7ajLzjD3Bohb19qJV5D6QgC3pewXOYZZBdMI2zp
W7fu00764X/1FW4wD7jUTAphXHUPq4fEU5eu9a+nsvju8b5We04VDX01Rb/4esW/7DAvqp9vTD+hqtR5ura3Ah
znethSJ5gGuZdjQuInwLiYS7AH4MKoES5Dh3301K8zqj4jSyHKif0T34Hmc5Pnp+syjQXmksLzYUBD12J47wrn
4iH1Ya//o+YohxybgtCit3gpzz0PUdF9k61UavEOKLnu21WM2Yi6jsykUVXos1V1KgyDCVZGr0JQyU5KjNb0Mt
0abL6QvV58mWi/3wHVS6/eDG7+ov18GW6W0VQC/nR/J35v8rnmi01tTXm4tpCazFoI91rrIuwVwj3LWf0sM1ns
IJ8n7n6wpkr18+EedTxr/UKy/X1fYnCYfaqu3eEqc3XPubWQgxYQONWx2xWZR1Fqkf5/bqc/egHGxt1m2ep1NZ
c1MfkGudRDQGCtwCJC1cmJ03iBdPFjLoqR4aQ78jfkY/fRm8zb7NvW9/04zcyG51CqyWrpnrR4hoDU1yEaheim
mqkrICxQsmtvflnJx8f+UHBIfvnL38qfsJk/fG/M+x5xYH7djxoYpHqvQ92bH9mnKkRrzALxfd0vPjSmRdoEqN
loHQHRLoaORwZi1HVFg6Nsowd1Smxna/SmC5dqL5yAYCbrL4yWfXsf0Deqwat3rWv4Q0184CAw1cjiXmxUu6S
iXAADzm2NNXPz1xggmFw2HeKcI1DLhoUTaa78hE+3WK/QSOHJxrbVQDGDKAxqVJCQnwzXOCLWJjmY2l1YtqF84
BE8H27KTRVgtig4cefVjQoYcf0fgX8U+M7dNPGZv456bPLz49NMLFz7rAUTFplr8rSiK56idGoRUNRBHFuawY
wP2aXVa1sewOrizMKs36BmTwaA3m1ijz4IsBp/eAPmkNem0HM+p1KzGxPPcao0WxSjTqMZ8xKxTdmNkMRkNep1
Wo5LSzpihsJr0v5VdRwLt0vL1k9nV5uy6Ksg0pYn/PLtaaVJ9Ln2kJ/7z9JmuomcLsiR4Mwm8VgqvFS4JXsc/6
VmN1jfksVsgTbXUMqWa0sMN7PLtbWGFm2TYT17m9zjGND6DHdoE4YsFaNgMVar1GqlVqlXGVRGG0RdFphr5Ww
qm9qizdT16G2GOWg0M4edg8u5Ev0cQy27EC/SLtQt1ruRm3Gx9biJc2mbdY36LYYNaA0zn12HN3DrvtV26DYZNx
k2mTeZNGZsyN1k2ZQWQn/GzfhzkevmAKqDu1fRqAzqf3m8IGs0msDloGWLjeCs/pBpSb9UMaYd1Cf2I8XbT7eY
Ry9qNaGNhsZopZmjmqZkatTRS2I6Jx8Xjx8SRQ2zh1xixvMkcytT9xTz9EHuzJf1fBi7wT34pd13G19fdTPt
0AP8Q7EWQ56yFGkUmfy4JZRM+Kz9urVo+Yj4Ct2ATLZM1R10ruhygbxDfXh/KRpAnx28WK1+Nprpoum18x1Zkj
DJSZkYkyAmIkz8SarKcuUbcqpzHUgB+NgHdjB0fhGqyPLke3IsYJolhALjMAqNk41iY160QaX5EXGXEwz1mp0Z
Yt1BVNTnYXfcf0tq+47807Vcz8Z4KLjo8u60HKxS/aWk+Pi7nhMPaExor0GZK59DHkrAUvobjHnCz0ZMXIdkh
fcEhpHs3cgkaVm/V72Qy71YTLIJVrZ+Vq5hWR9Lpwf1ymswXJkmSmS5CeIKdalRsKt5RPF4MZZAmMakvb03CD
FArm2i1UykI5YerGRcfWfTTsYn/9ddXx3666GkmcPLg3gc0HHhg78Fw+Ce+Tcziy5eZRzt8bHn4y9+EfvXmR7/
/8I9vUN9cBt9cg1NWg7Y5ZnE+XsFjn4JfjRUcwytkKia01Eozh1it6fzExJUJWnUuTVBfkBz1IE04SBOAVk0g1
SIHMNUi7bWzUoKoBSQwlepKDUfg11IaCr+W0FKawtpCM19bCmXdCtHGi8w/fczUj+6dd3T14eDU1cQ/0na2
g5yy02lGosyCjHiVYX66fD+mHzRa9LhtpmGyFyqAxMRnafN01CXKMAZh1dSTzL35xUbxoArUBWW2TpknbpGuyN
F15i1i1m3E1Z7XAgc7Qw6WWWcFI4DKKaN/7wU8Zg/jbHJ24i40MXbmJefXwzh/ev/8It4XR1C3/4ysfiG9nw9p
J8dgAM3H4y/sfPv4AiYd141ruJMS8DhWjNQ47g8z5WM8dytWPGotmjwqaQ5a9uZuFu4xHSgrsSJmbXVtos0ebU
Jm+qoScQHA+Xpo0QVYJidp8a10hcZz00rHS6EKM5LrJU3LkjGbYbWwxUVzyqRQZo4d3LvvwIF9ew+ehMpPAuS
vf50gAQJHAbsydPaNP374+4/e/GUfhIj45uXL4hubfGIeZC3JdQoKIPyN07oJTX4CYvVYnhCb7DCbYXRofQpt
TpWv0K1UmKWZ5QKBXoiTykhNsKqJBZVlsum7ciK8u62mpQIk6Vm8VjZjQHH8k06rBq1KzJ11JpdYqoUjrMjh
1RvYsCLEPJ8xSqqYY6feVbFOuZczQsD6S6Bsg1GbRUJsF11yvP+KQxmbFYZSNM/QjNyGdWidssuw3hpAAWXQE
LS0GEas96A9BouGUbNqrNYrjLyNz1Xk5TK5bC62aW26L3VaLPZcq15c1E5Y2ftuFrbqpurLzPMMc4xzTGX28p
zy/Lq2JvZ0nwzNKNuptnQZGz0u5XpYrtlwIE6tcZ1hjXZq21rc1dmxdmwWY9yqD2qAuq081+I2BvG3oduZ29
nZ8u2JEP2xIGB0mhDlh2WbdmjVkg8odyrsX7Vbu1e7V7dbvNuwy7rLca73Xdm/uvXkboGozmcmKTUJAK93k4s1
6/nk/8+g+8W0mZy9o3fxnv9wv9u9jSsWX7mY+2MHuZmr6+sRhcXE4zKwQX4HvQ8wZUsRZ0vPx1+CsVsHzd5PDp
1NGDN0aFFErGasRcYYMvboEmbWFWRc8pL8jeVcnFYvJaqhmVdlt2Tuzn8im1UxAZhMqLGakUis9MZdfh0RfLL4
pfvi0Ay9XQdzF/OM2C7uCDMtZCK4V14+TBo8cav4I/GomEDstWViHfMs/zEUktko5CiabcrWYXV2Vt7+TLVU1
HAK84cFOQuyEFZkLTkW6e26qgI4Wa6Qg+Uiya5rKx0mi//4mum35gzIqucrhc1CVMBTp0QbjZ0CuKTYeS6eHc9
h0cbMhYugBlstCkgzgrSzJspqa2YzJB3PNVRWNDirKhsWQBX+9c9X3r1ik3JNx4Z1/JaKxsaKyvp6cbTvyy3HT
hUWfrD51rE7k2fHHqgVGjQX3eGozCvU2pBJx4zaDqr2ZxY+ULxv9v45ezM3g02sjuEFC4r091wjKtPyuUVZ88q
h1kGpg4oB9SFQnYXxItf/MkERQ/cYK4j3WuJaV5k3ra5Z+dxo3PPMmfZs3iCm+AnFBPWiayJ7LM5Z21nc/Xki
KxNnSmZ5uSxM6cWTC6UCo2yuFY6Gis/333vfYn4vpPHxGM3He078Z8vvs+wB+8ZvMu0afMv1/7+z0zrH7a0Ru5

6gD199Xfh+C1Np4/+7fPukXt83e+V139Aa/q1SYipTyCmDOhBRx10a6wPqgrjQwwUFTi2GbVGrTewyKTRqFcZK
g30Tii47pVeo2aVB0GqY281gjdnVQzLtF68bX1Ano5Jc0gjdCCMCqWwxw0kT1MDpujnqWfZZjLztPPM+iVTBZ
TxmBAQs3ShJH7HybAvMCc/Bnz/IGr//LU1YsPsQIf/vJebrSv7/Jh0vHQm2ox+PEDqJk2NQooyjYrdHoLe486B
0o+0qLezCF7Nl+G7cxSg91UpqvMJa1Cjn9y+sNtEkp8Nc2bquRjmuTp01yUK5+mz+3IG89jUxGaSyM015BJfUx
hmsvMNDTpCWEuZt911j8tP1bzN+Jn/6E2bRpd0e9R7myKw3h0+HIG++xdX1XH3rl+6d/wXxM7XgK7PgEztpyp
sxRD11HSaEJevPZBWY+YILufLVWJxRyWcw050MHy7J2WH25gxmhss3zBK2uMF+VbbSyRqQqtypL8suRkZ1HA/S
C1NtkSHXBd0E57XbEi6n6bvgchvINXGwaD223QyvMr5rfPj86f+d8npid4SGN+IZV4xZa3udRf0ahecluIzFXU
64t15XrywvKhfLC8qI16jruNu1S3VJ9naFu9tKCpcLSwrqjCvujVt7i86tX1ngFtyF7qIuTRf0z136LqGrsKs
or07X9Gv7dX36voJ+ob+wvhUPKYZ047pxvQjhrgCMWGscKxopHjx0fnj81kouYvN2VL2L5xTyVRA91AitW1SW
2nJKmBmw3HNFRbNORPnT/3q2VNbn/1H59hDJeCf927xXzil4//7e1vvPCbvc/Ez8fu7aS8TDHf7xvV39i8TL
nK3935ncL5ouvPL7vzrG+0eWLb3rpifc/JD/aQd2AvprdTOuffbkd2VidMw05YoIyATUP611ZptFursqSXhfJq
z19YTz/FrwyZqe9MmbRKMqCS4qiafUt1RC/TspafWVlw788ffWzcWiGU/Xsy7nQCHf0yX3+FuIBZ6EtjjnKLJS
hGVWhWdx+2xZQyqI6Y1DrmQUIjvwytAh0s5qroFFbfqEagn+CtGqkdFVDv0+q8sVqUpWf3T87Tdn8qmk+XJKm/
7AZbWbYm2cz5NCTsM6HF/LCVBtfu0ttypZbvExnW6+HKRffm7z61Pj4+Au/aLi3nretat1yYHfflaepBfc8+3y
0DdHffNmuRxcE9m7bZfz+r6hARX9b+91/1T+U/J2NnDyqGgX5cVE19eMbQsoBMT/957gZP89ZuXMooL iMtnCnU
YDdj5Zx/wXu0Nmqq6pAGv4Nu4QtQgM9G1+GzB8bLuLuRTRFGW9ixa8u4V2Ht9WuTfBlazI+hp2D/LewXBHNUhEb
R7xgF08Q0sUvZp7AK34af4yz8LD7Av6loUjytzFD+SVWhepBqZIWEYZEUQSb0KFJcp6d1/idYQn6Cy2VWpPTel
LIBWjR4YuRdH0qTxxjmI/KYg/GYPOZhvEseKyA+D8pjFTKjx+WxFvz4jDzWqw+it+SxAS3UJGWZkE6T3GtGnOY
JkMhwang6o/mxPGZQjtYrj1mk0o7IYwzzd8pjDsaPymMexifksQJ1ad+QxypUpP1neaxFS3WMPNZnztEtk8cGF
CxIyjKhnILkXjNSFfyhIRIdiYV6gw1hbk+5UF1VVSN0jwj1oUQ8EfN7B+yCe7CnQnD29wsdhCoudPjj/thWv69
C0+gPe9cOCT1B72CvPy54Y34hNchEh7r7Qz2CLzLgDQ0maTr9sVcGphLpS390H6/1x+KhyKBQXVFTK82nkWQig
yA5AfoEE4no0spKH8xvHaqIR4ZiPf5AJNbrrxj0J5ooGdGD2JKyQZgb9/uFbn9/Zl18QvgWWldoNF0bQT0vIH
0YvZcMM/jebfjqowQ3IIVBQSMa/PP+CN9QmRwEwUgk27PzYQi1P4gDroj/lBvM/M05jw++xCIAbGwzYwGGCyC
4mI4B0cEaIA0GyIdCfA4NBgL0jpAaUJZSL01xH39vREBqJATggSqeAOIPkH4wBwEYWkqByY+QRvPB7pCXLBHID
YMzTgH0x4E0SfQKgfMJ5LONINQmcckBggzIvKqSYxfzQW8Q31+CkbXwgMC3UPJfxUh2kb70C1nv4h9Fk0JQIR
oYSoMxASBZE6GMS1MB2KA70xBy7MOCnV1P/xoP2NB12IrMyEhPifvADUiIdAVdn8GaKJcsA2SoB0yNBRQcPByMB
XNx3B1ZigyDQTzf6IKI8YhfjQ91hf0+CzEgY90NIEoN6Io0+ELEjv1sJ8cCStzuy1U8tKKIKpAKgsFIAtwQ1
2aJV6JTESCtCfGgF4zq9suogRoQ5N5pdkYGIS5iwkAk5r+u2Uj1j0oPeEFQhaTU9NUB7wjhPxDxhQIhEmje/gS
EHgyAqdfno5ZL0JH88sZAr6F+b4wK8vnjod5BqkZv/0g0GCebSIR6e4BJn0xI6h0fKUmKOJ8EmLc/jcEMjvK+p
C5THEHFwf4RITQt1MGkmJ/8+zxKSwZxAibxTTJF/BB3fsmA4UjMFxeKUr1YRGQnF4QikrpFFDbwToucM91+yCb
CdQj8QIzYGgm1FPNvS0DWCN5oFFLM293jwuS/cB5hm0C3oQQ9MaBo39w0i4gbirCfcLQoE9WuGh6XSmSLLyRZ
+ORfpLZ1HXEUv6hn1QQyJckYdTb0+ftBcMgFwcjqfrx7QNrmiwoWqCivz9A1FrpEpraWj1CZ1uTZ52zwyW404X
2jra17kZXo1Dk7ITnIruwzu1Z2bbGfIwBFh7PV0yW0NQn01i5h1bu10S641r3d3uDo7hbY0wb26vcXtgj13a0PLm
kZ3a7NQD/ta2zxCi3u12wNMPW10q8zK7eokzFa70hpWwq0z3t3i9nTzhSa3p5XwbAKmTqHd2eFxN6xpcXYI7Ws
62ts6XcCjEdi2ulub0kCKa7ULjABGDW3tXR3u5pUe02zywKRd8HQ4G12rnR2r7ETDNjC5Q6AkFaA18BBca8nmz
pX01hah3u3p9HS4nKsJLU6nubVtNcFoTWuj0+NuaxXqXWCKs77FJekGpjS00N2r7UKj7Wz2dU5JYSQyeZMwUE
2NLtaXR30FrVQ2e5qcJMB40jucDV4KCVgD0i0UHub21o7Xd9dAxNAlxQBDlnpoilAACf800A1o+a3grmEj6etw
5NSZZ2702UXnB3uTqJCU0cbqEv8CTuIjWzAT+K8V11f4iMy99XoACqyWzaw0eVsAYadRI2v0EJ0ubb1+KMJEtt
yckv1kZZSqX7aadRKRQBCuHkQE1eo00Iz8gsevJIFW4quciRbjfLLykfEN1wGkn117fVD1UwTkoJ5EeEFJPhU
Jxm0hyDAxH53It7+0EY7EpRQb309s02eErN6QmVPBcjRsBsGY6FE1BMB08QzMZCo/JRHJOPqpkWECKz9Y/541E
4qUJb/f0jFUAbI+cZ1SQ0C03WgGw6ha8nsTRZQxNCL2XuA80hKasQNDfs1yqHQ32hyhDUqG0V0WC0U16UM3tG1
ADtehSNoBgKoV4URAkkoLmoB5XDVRpVwVUD026gEFA90CRQHD4x5EdenIDsM0tGg0BfASMn6odLQb0pXnH65Ie
7H/ZshW8fUGpQI4zCwGEtGgKKHqD1ApdeSinAmPAXgMsfEeBphv4h0B0gP0Rk0ulazP5dFIPiRQATSNw9X3t+
tfNr6Wj0IwjVHY1aFuDaqfRfx2XAN0j2ZyQ8SEYJMCcpagSLp9MvxXoK4AuAvcYWOWne2PU/grg4Yc9TWnckng
k/fJVP5A1grGf+soPaEXQMNAsr/yfbwqC/jvk60mWMPCKF3nr8aVbI34d1xE+n9ErF4f7SmbQzKKA133Uh8PU
FT7YC4Cnv0mXYh17ZTfAOU2FX0S7yBd88t29VIpqzTCfJRPgK76U9IkD0vRZKd6RaigGg3R/VI5wSUIEuCZkD4d
oVEi29MhIJ3kmqBbTY9wLVD00QqjY9yQHQi3pLkWSH+bjcgQXpUVJEFuc2eu9zjVqwf2eGX7pBjsgagcoFwSd
CWJTwBG/XIcz03pOCWBZDpRPwG5IMU5kT1FCZmJwnEpAxRPae08VELEjTWumE1QVetMR5eg130pR7QbIhykTA
ZpjEQpDmfkJEZoHPpFiX5x6ZFpaTtEMXQnuYdMh6g/kz6eip/47Db/jV22FN2VtK6I1DOUj5IvEMyqt09f20rk
8hJ2kZTE2YEXVTFg1TPAa+1YRkNgRozRyULfSnSfTRbyLDTu8EiTBQ9FB+Ek16HPfLVTLpoR4q20c1DsmaLqX
Z6ZF3eYFjhFaGKR+k16IpBL5aCQaBPiFnQ3wabTJXphBLrwHp+wRqs1f2VHeqbidjTUJDquTeG/gzQs8YQfb9A

L1P1Y9v44sEWB6155Zxtqh16132kswGUnpP0CzL0Rz0VnRi04JuepJM5KmBFNfms/Toy55fhEpE15DwMVL9yU
t81FNib8G09DoBTpiTCei6XVUC+NHi12kzJm4hP/RpvSa5xvWoR5qY+up8GNNZkubyYu19PRLvu9n+4L3aCqx
+QK5Kf6DUzjm5yJpyIzmTczTxG/X0/80zwwTK3y0f1F1zkXi1J2z9xB6J0nb1FatEm50zLjn0mmeR9J03VIzoe
kJ7bCaug6iPnRNorzoJzRUbikU8xLk6s/tSPd/5L0N86YIK30Ar3HZR39NKK+P14k665Xw8nqEKWajvD1UBXSk
Ev34b81Z+00eibP7KmsS2YU6SD6Uz1ITN4xnW0URnQffPfKHpP0xUGK7cz+4/9Gxfp6q7r1HEnI52IghdRK5KJ
y21ArPBE5bfDkQeugn+yga26YE6Cf64CVtfDUCLON1C90ukLWi2g2roMx4diG1lBeEo80+Ca8u2CG8BboM31aB
fStwIvsdaH1VIYLuHVSy7KezXMsDdJd0RHQ0wswaeybgZkW5UktcKuzw0d8g+ouokqQfmp6R018pNJSY1Ww1
PHcB/pbzqBN5uyo/ob6dIKXFrsSs8mWVMnxYhwJjwbQKMW+kRm18C9Heg6KZ50ar0kbSu1oQnWJVtcVAPJE5JGD
XBvB9mEohn081AtiCSPTGmnFhJ7Gul+InUVnZU0a509TMZTXCpkLCU9CP5rU5I7qf0tcAnUfg/MeKhvnMA/yTc
Z082Uw+pUHK2h9jkpDm1UQj1dIygSPftS1B1pXmmgeBG/Ec0bqSQnRaTzupYkuU33zvWiIymhmdrnoki1U0p0w
NEF907UjBSPbmr94ytxFOKeykmWtLQbaA2Es9+F6S65JhyUuymWf1CNF/ygrJA075uyENsynvt8rebuj5uo1
G2VdRWUdz0UWpnNTXnSkUmmj+rpy1X5MWYUk/rpHjsy212XR8k3mUpPs2tUPi1ZQ93YONNj5aZA07U2h8M1+pd
rngX0uh7zuJVN2efnKnd49TXWl6/21Pq7XpnYBuHsZsp7cAMuq1ZqT5LZ9bU0096D3e9kyv51i19FPdb7L7kGq
39G6U3v36aJ8u9YLxVFc inR+RVGcyTFenznTpbXCAUqS/78WpXMmyIXnHTF5Sf+m13QKRFr80mjcb6oWa+IUbpe
S9JGabhNyZEPUgZFoyPzrjrTg2463qm3yQt0Wb8I9Rf0f1d6oQRZj0kxuY3xhKvp9NYUIQkH7dGpjh9anoI9y
Wopl9KMGgN01zn+xx6ZcyI1Pz7/h9rZL13QefSqqjj3Z+FbQLj8Lc9I7ym35nRFj+70DuQLf0/H+wyN9Jdqfjo
9dF/FomfnWihH/Vh191cBM1+Nd6/MrLZfwrPvxyGT67Ef9q0z6txae0+MUXLPyL1fgFCz5jf9BxM+L+Jci/oW
InxxifFm/sRLPN6MnxHx09vx34v4KQP++XEd/3MLPq7Df1eNf+bDP52Nj1XjHz/p438s4id9+EeHDfyPSvHRb
Rr+aC1+YhV+3IR/WIGP3DebPyLiHzxm4n+Qjx8z4UcfMfCPluJHg04RA37EwT0MGx+24Id3cocN+LCDe6gUH/q
bKv6Qia8+mMkfLMUPHtDzD2b1b08yDoea0/B9DX9Ajqw+cZJDDzX1fg79/1tsf2c7vP4UfuFPLP2DGdzi4fTDat
xTv3XOK3yviPfdv5Pecwnt2cvfvLuXv34jvd3C7Qa/dpfi+XWb+vtn4vpPXzjqucbvM+B4QfY8P/00VvjsL33U
Y36nF030+fqeId/Sb+B02fMd2A39HNd5uwLePGfnbM/GYEY8exiNmve2Dh7cK/PB1vHVoFr9VwE0zcaI2JWbju
IhjIt4S1fNbRBzV46iDi2zHgwMr+ME+PLAC9/fp+H4T7t/J9elwn4MLg8jwZRWKnuJDIg72buSDp3BwJ9cbKOV
7N+JeBxcoxX4g81/GPh/useJuEXTFvH1TBb9ZxJsq8PdEvFHEt63CG7bjW0Xc1YjXi3idiNeewmtE30nDHRb83
Wrc3mbk27fjNiNudTrqcYsW3+LD7i1V7z6MV1bjZmzimzNxUwZ2sRreZcONDz18Yx9uqDfxDZm43qn1603Y6VD
zTi12qLGD4NjJ3XwYr+AW8CtW45uWW/ibVuHlyzT8cgte7uCWafDSugx+6UZct8TM12XgJWa8WI8Xibh2oYwvF
fHCmkx+oQXXVGv4mkxc/R01X63B1ZJ/vqPGVZU5fFUjrqyw8pU5uPIsVzFbw1dYccV0boHaxy84j03zLbx9FZ4
PRsy34Pk0bh6oPs+Hy+dW8eVOPBcUm1uFy+BWJu15S3GpPocv3YhLijP4kk5cDNuKM3CxgytS4UIhhy/ciIUCM
y/kY0EsVwDCCsy4YCc3W4Nn07j8YjzLiPNKcK6tis/txDbgaqvCOSLOBqHZIs4yYavFwlv7sCUzk7dYsMXBZwB
iDKDLOIXNAK9ZxCa4meqxEfQ3HsYGD0IWA8M9D1Y7+B0ItbC9gaxpA9rgEazHat9Wku08yoLvpqXqg/mFdxd
/v4aswBM24BBqasBj0dGImY0cn47tnHzP9/9g/9Rytww798hP43K53LZwp1bmRzdHJ1YW0KZw5kb2JqCgoyOCA
wIG9iago4NTIwCmVuZG9iagoKMjkgMCBvYmoKPDwvVHlwZS9Gb250RGVzY3JpcHRvc19Gb250TmFtZS9FQUFBQ
UErRGVqYVZ1U2VyaWYKL0ZsYwdzIDQKL0ZvbnRCQm94WY03NjkgLTm0NiAyMTA0IDEExMD1dL010YWxpY0FuZ2x
1IDAKL0FzY2VudCA5MjgKL0R1c2N1bnQgLT1zNQovQ2FwSGVpZ2h0IDEExMDkKL1N0Zw1WIDgwCi9Gb250RmlsZ
TiGmjcgMCBSCj4+CmVuZG9iagoKMzAgMCBvYmoKPDwvTGVuZ3RoIDMxNS9GaWx0ZXiVrmxhdGVEZWNvZGU+Pgp
zdHJ1YW0KeJxdkk1ugzAqhfecwst0EWF1gERCSCKpEov+qKQH1PaQW1rGMs6C29czpq3UBegbzXvjx5i4bs+tV
i5+s5PowLFBaWlhnu5WALvCTekoS1Uwq0VvcXYmyj23m6ZHYtHqayj0J335udXdjmJKcrPETxq5Vglb6xzUf
d+bq7G/MFI2jHeFRVTMLg5zz35qUf1SbXtpW+rddy9ZY/wWUxwFKqxbFTBj0wuwvb5BVHJesbJpqgi0/NdLe
bBcB/HZWy9NvJTz/aHynBLnGfIucI28D9wgZ8QpR86JswK5IC5S5EPQ75GPxLsE+RSYNI9BT3PqwJThHPKQ9yn
MOSE3gXF0ws050XLIX2DmJ0TPdshr/iPymj+nhaXfjqvBu/tZORN3a/266YJpz7hhpYH9/gRmMmij5xtsNpn6C
mVuZHN0cmVhbQplbmRvYmoKcjMxDAGb2JqCjw8L1R5cGUvRm9udC9TdWJ0eXB1L1RydWVUeXB1L0Jhc2VGb25
0L0VBQUFBQStEZWphVnVTZJpZgovRmlyc3RDaGFyIDAKL0xhc3RDaGFyIDIwCi9XaWR0aHNbNjAwIDg3MiA10
TEgMzE5IDYwMiAzMTcgMTAyNyA0NzggNjQwIDYzNiA2MzYgNjQwIDU2MyA4MDEgMzEwIDU5Ngo3MjIgNjQ0IDY
4NSAzMTkgMzcwIF0KL0ZvbnREZXNjcm1wdG9yIDI5IDAgUgovVG9Vbmljb2R1IDMwIDAgUgo+Pgp1bmRvYmoKC
jMyIDA6b2JqCjw8L0x1bmd0aCAzMyAwIFIvRmlsd6Vyl0ZsYXR1RGVjb2R1L0x1bmd0aDEgMTI4MiAvTGVuZ3R
oMiA3NzMwOCAvTGVuZ3RoMyA1NDQ+PgpzdHJ1YW0KeJycvW0wLV2XNXhs+x7btm3btm3btm3rHts+/bxV3
1vV1dX9p2NH7MwcY66Rc441MzNWRuzYJATySjQCxnaGJqJ2ts40DLT0nPiYfjaGLk5KBrbSNi0mZi74/4CsUCQ
kyhb01ib/i4UiUTVxdLkws+X8d5yQo4mB8z+AsIHZP+HK5i74A1i5m+PiM+AzMnEwsnCzs+Iz09Gz/DrRz5MQ3d
bQzhCIRsrP3cLQwM3fmxP+vXXxyFUU1Cioq6v9GGDg40PANPf7N4AubOFmY2eKT/g+Bf1BXE2s7exsTW2cufKF
/Y6trCyn8M2sPe3MnfANjYxPjf2moGlibW0GLW1hb2NvbeKTC1H8T5V/EmWg+eeLBYoEX8zE1sTxn5L+Y+C/z
BK1czQz+Y9a6NkYmfDJzZ2d7Tnp6Ez/oUz/RdE6mdLamjjT/aNJIJrLGRn869snKCg/jVY2MLRx0if4j3o/p9

uW9naudl6/S/Y1MLW+F/S+MYu9nQqthYOLiYSwv8n+B8I6r8xMxNnfBZ6Rno0ekZ8Ewd8E3cjc7p/nVLZw97kP
0iGf8EGtsY+XvZ29vimBtZOJj4Wpib/bKC8nAxcTfCdHV1MfLz+78T/PIJiYMA3tjByxjc0Mb0whfpv9X9gE9P
/PJYxHa0cMfXoqf9x0N8+n99/mtP55+JN7aztbf473BZAxst/P9V9H/RgoJ27vheNAxszPg0j0ws+Az0jIz4H
CxM+D7/U+q/TPi3Af+ByhtY/J8E6f9bUsLW1A6f4z/r+MfA/1MLPp3rf7YzPvm/2pkC/3/o/50hnb0Fkck/nfJ
f/ahNz0L/Tyv+s2H4f+3T/8H/V7f+/2vQf2kJ/Uvpvzvf2Uo6mJt/R9ukv+nm/j/20mEL43/L00tDRz/d7yBj
YW1x//Lip8VqWbyn5fk/5eShLPBPzUI2JpZ/9tofDoLJ1ELdxNjeQtnI/P/bLX/Q6jYGps4WlvYmsjb0Vn863a
BT8PAwvD/JJXNLYysbE2cnP6Z0P/gTGyN/+dp6URsjeYMLWzN8JWc/+lpa0fj/wL+RRu50Dr+Y/F/zPI/Y/99b
GrxT44mJu4mR1D0fMH1j0bW5Zp5P5/yzV1k9JQaKtE+48cTgDjXyV//lRHUze1oj+UKUXZDJqUzeGv8DbZsjH
xg8guDRFQBRswHqRCBiwCWM+chX+n0sysXSMrhpKGFwN+jQi0hVmHISYjH/078xHqbV9XfIQt9tKxTHz4qD75
on2PHFl0BuoaFLNQ/igY8bIgo0spdn2+wHiPwNIqoTg8vhHWDpPPFAYEDobv76vA4RFCsnnTc+9AX+LZWH7x
pHWAVhYTl5ABgPf5mz+YW3NVX0cusKMrnzbzwV4QNsBH0nDck48SeBd4MP04nWJgNG/lpnf10p1wNnnwpF9rZ
TBZY0g8hKYnViWS6WmuhRv7h7zrB77anmrscxB+fM3LPgE2QfNDvJ/hW6CDyvqBT0oAeDs39/kxbI29kCjWw/J
Qctsyes05RA1JH2CkBkdceh6juCkm42AUGhZN1RPUN0XPjyDPwEVUjAjW06mD/Bp5Bak3itygvnmVhmgYfoXVi
x1XiGBXT6B718nfrkqrHB8ri0MCjr2qndzgxyJKtD0AojoWwmZ7mKEX0bU/nK6IhuJqG72kxgQn18Vfu0Nx60gF
/BfmXD9aWY+hz/c5azHidHQH1V5kwxufwRAHPp8bWfeo6GfeQFMaXqk6HAPcr6av2txuZrpA2dM6ZHXRPTWS
B+BamdgjcBREwDTY51nxC7n7G77aSb0GAxVWrA7eqdwzNtpjp2H2yujoHPcgouRS2+o0MPoF1usDUSUNh207i4
f2nU4BYAbtSJuzLkUWaqxLcDNinCCdBNdudcFP+p/sTX2gWnLRRKA8FvN6RAnsFHAstSCFjxJmg/4xQHYbJhWiM
nZMu3PM0X/TyJMu3688CUVcrk8FvUgC4o1kYjA8pQ01pVKh9ad3hxcM76QS6UL70YwxpJyVxNchiJgtuu1yu5T
0CKC74jlxIonpDLdNsGL2k1Q61k6iSt0V4Vdb30s7WhGdxK8uajqRp/9LQY26Qhy000EEmYViQWm/83CEVoo0
7qIHAQmbEPLCeFNurhrbV6mEc6tKrbgaeMS0PTMygLaF9dmYkMKSUTrV1IGmSb77euKEPKJ0ytRHHfWBI5EP
4EY1FhgG+0ItYQt19d11cIhaNbMmKx6Ei7gVhhZiCu/Y8XrDau110ked2nI871+SvDwUzP6mJuNWgflZeh04e
i9xBn4w9hj4R2XP5B1NjG09aXIs4Q8ztMT0StTezbm8KifGOE3I3Cx67AnXVcSzoCilt/cGcPwJRDyYVpBEf1T
ispP/mVyKGFrp0l+dis0A4gYkkAVNL42yLVUEop6zFjxv/iLTzgohkQLswjMrvJ1U9KtHHU/RNABgUQ8iRcNTb
B8+ea63o0+AbD3WSZXH4QjIo/dK2UWEaRa9fSPkmqNESyanPnq38U3YsvVYZkkktDce0XG8bd1/Fxdt2r66Th
0VnkMwPnNweZbXh3HH073mD5rIs0Mf8sa1rytgiMjUyDj65U7gIx143EuGn04J7H0uYs3ERPDy3MzW2DGJ5CR1
9vUb0c2fWtWigudIucrNQW3tFNHhsLk2fI3jr+UI1qXPX1kG/LLKkt5R+3X6ymNwfhtwLa1i/K08m95NFy9o+
Ao7r2CdbSB1esXFz3C0E5XWuR/AbwWnaAKAbIeAu2YszIsNy5F2LdA41ZQXNg8LREtSxDng1PerjyYCBpt/Dz
yQJzq2pb0rPn/j54PHzP40IY1ck6V5eqhrGbxRXIeuBQkm9DvQM7/V56CSPsXU1Y87HfijTeBU2C9I1kKq571ID
i2IdAWOITDYPQQ11E4H10t8FwXtXzt20r424DG4IOA3VIzxDUwbV9Kbw7zpdZb+hvU2T325sPf8ynH0yD5n2B7
Zakjc72fzf5oVLabd1qk0UfZRymLhLV8UgPQWLGVZHeDE+1vMmfV4eytYwYzifMKZNyBdiWbGbQKPxJJfpbDR9
sqMx/XJpz11ThQzrjW6UuAPq9bHAhyaqGQk2oViftB1009A7Qe6AW6uzbm04dD8AjaMpymJ4cyjVL9uZddP+W1
fp7S2Ss8KHDrm3jt7B1QXG/1VqkGp7JeXSMk8hI2070PujxZ1/IUJsvZ1qBohbuR/qoPd0mdlzvBGTGsqMMdy
CZ3MtSipoKgGGkrrsV2Jyzp1TcmXvu7uJGRDTqWOIPVbivYMsThuX+N1813U5r6cYP78vXa16qJa5vjbNI6cuZ
AxPpn5x21CP51haBK2ojamc9r2Q0pqzV6e7TLUoT09FGtocxf0Whj1NdT2dBULPdx9b+dxeEEkNrYyyCzH97Ed
CN6WJjljDrfbp79JKEVF8N9idcD1Qj+XX6z0U3y3BIPD0vML1Z13bT0Nkis2rHSicWjW4LIWQo/7jSwk40w9wy
+Hy0bKS2Aa7tA6+h/6oYPo3pUtXvVHYB9opaP371XHRvZXwpoKLq15nQ1vPb6MzcnY/fg2i0To/FXyz8FpfwoA
k6r703fd1Ug+WIVllyUceuChRyMN75mauo6EqB+x1dTjTbUPMQHTXCbpjbYwAGkiSSAQt2XunQ8c/uYh0f1h93
eMByoZAsEq8d4Hw8y6isI2gd9t9YFqeMHys4Ni57h00p0jU9xtosoxIjDqk3HSEYV3v7PPDQY6Jx8mShgeHshi
d5sTIqTiZ1taMgjuLcgE2Ddna93FFiYUg3AhUxAMsnUuXPE7dy701CHunWk/tWELSpmcxYoewA41gYfBdu0P9p
F43JWZpf1Fsvk+VZMr2U0RVfAdxVbB09TqenL91ydmLdYza3LR5qVgh6S1kaqhI7v4o/0f04gDP2+fhKVRB1Vj
6GaMdZq7RxCz1oD1b1SGnZfw7hDWYHVoEotF/b2KWyuuEakjIYX54r83UzvnK80E2T/2MnnncIgrPTX3qg8nWu2
ibK9uiBtGVVGT6UDIM5eD4CsEKJ1yJ9kod0pza1w36Cock53vD0rVaA9/hK7w4kYhTrvj0La/XVNXPIm7Qbx6
iynn94nt2eGAtE1aSqd08BdmfrU3HfGcMFTiJu0ERqvMkXCYLLRkPCF7pYDY7178T4LzfS2N/mWwoh76SPyWg
7xAA06cJH6qAHkvixuqK2HkLyOBtt0+mGYNKnXcX38WpEhMX3KmW1EZdA1Ygutq373nJicq2qfVmA6JmNXWP+1
dJgK0BLvZ8JUFUYiMEebadzc7N94mNmcaMqyE6YI1VrlmKX9RbUEEus2yisMNJ1kFw6n2Go4E8eatcyeKzIw
4505eY/w3yUHPKq4KqhiACNjjb0mqjUG+YBoPuAXxPh5qq9HL/pT35WyDC5+wtEacvB0m21ZMSaePYmSezbJ3y
is1fXZeGsweQIZwZqCFRSGq2UXNxs+9ypmJvH92br518Fd07FY0phc3SVUdy30TMDrLyqGqDp10JdozgGGHhm
qBa1/3D4iUyjfvwo+PKg+hnv7mm2i94EW/N07FyEQA0cFwTijcZVShaxVxmgzIM5aQrqpNSaPr+oI41GYLjvh
6QR0nIgn4S49EdmjDMHtyNfF1Eo4oPFz+Sxg15cH7XY2RizW0sjnq1Z2XYchA1Z16Kn160tqG0gAPe5mxRPFbI
PzxxXg0ZHIKftz9KY1xynICf5DSMs+JwXWM4HD2ekKsBTFXWsNFXglmp2ukzizH0BSUpC6HVUDLVofDiy9LMZ0
/HvEHc6t2QCHQmHNjA8AKVeeskaSW1kfoF1v21BRVvg11FcWZTGaAintthk1K/+pe5kpq4uRibNDEGAbXN2800

K+tjzVLUHUWipRmNvkRV1/fZV/WBtTXcauwhaM2iG0cdqz8Pehb33i2QEV0p2dq1itvZRjDDW4T2s+0DrVtoac0CUIsY31uuEJZfkPNvMvbvYJXXmTkLJa5tYuiCzLnGqFiId2V+Wa8ReuZXkjRPvY3jZU7i418/DpHb3oFfB1Lfuk1nIb702aAtv7Dv7gLubtRQSSwXL45Dkm5hpQ4+sF0q7jqmNbRHYxGQ+8Sf50YIXWicg3u6WUimG4ZI8c7MV3Vhv4tzLifSrRFoQDCS7iBGQ89RY6J1zqJ6wDV2L8AiTMAudKkLuVX1MH9f53Su4mxC0olfBcwttnSheZw69a1+HMsHnQ1r2HVqI8iqGkBd6sd5+CYMpR6xYDNyWGab+EJ/LNSRgt5vGEos/v6djJ3IoVWcpcjNob0o2Xw0Bwb5mMgX810n+B+8ywTUQG/h0s9MGwbeA5Rn5L3HNeG12Jm9YQWFE99AF6T+/3Vhx04EjcIV0ZvqUA1PKu8qCuW1V+iJ/xj5LYmyLY50j6DV33kTTq/R9cwfE5hzlyfBgyCvrm8BDCwKrPOPiNKTDiDakPdmhhGjIiPC8n/j6IhUuEreu16Xv5+DTqvkE1HkizXwhay0fmc8E4mGCEnSN1G6021gMIEfuAvLeb9LSfse/uk2xijFljdADciOFmFRGALGFWh0n3EnaLwlpFE+jjtN2FuDcw6bMXjBRxB0c1L79Kv1evs+8IKEez35N0Qs9N0NelsMMPEf1cVN2ye1THppJ6ley0mCpjTd6lsV2KEGNY4d8HCWrD0gE1fGHdAcKofEmwSW31G5VDzYmEVH0JiPdp1QuxoJTkRgXzI7sQoEZYZepvkqMKwA/ZRx/qutr+To2zpPRLZgYzdGj6g5zYwDnx5LSTZgzvaQxBCSZKtUZcyfg2tB0v3Mk+8IFnZdxSbpwyBsU0kC1qQ7b3+iu70vmNzy4e4zVevq37aqIfsDdNIMYZ+xawv2znrBYYbgQWB5h40ncsSzLQYIdxJHYcULmpT4g2Xh4fxsEGJ12ZV8P+qCZjsq0PBeY2WDIpcnkdlH115Ydi39Y+JQzMPFi4XHTAdhpj3poRLwCL54/YnZeKvpMH6T+XDyISBy0fbFLLWwRoYpDGKKtsItxWe01L3Ybsvdd0wfUgT0jHx/PICiE10+7Gy1IM4J+nEupktMID+nQTeHZXoFFC/6Uw9L03sUwJ4cwP+RaaXrq+5u1TR7NhTrXEOD1IgTyinVxPiib5WIoD1P7Scd778AZQeULR1nw4yP2uTDyBAgP/IeIxt4k0EpsRkP10jpirWEEQMquWF1Z1N4YSD1KaF20nVZD7EZr8SGIjNzIa1kLNVM4LRavTJkhg3sKFZwgBWoCncUZByswt/07+BA5+xz1/W3qat1BayR4s02yLqMr/QSooqN2FGqkThaAczStdNN6/1S0+YwhbZu6U1TuNRD0U1Tzgs0rHP37nqc5hUT/MUtbZiFz1aSd4418MOFi10PFFBmWVgdLbQwxr0Pqi7YijGb+/GmL/Lh00BjR Vf5hpaVEDejQKpYbjMLFIux10d7ndhwmqgse5sr0swm0mSSsy3M+a2xWTOLwcblxdFyaZMHfjuqZjEQ1Wkx+0MwF+RqQEGPdJYjeWXTRQZKJ1ezjXwBwQaWQcBYdZTKwyeEug51MhZRXPO1UEvZkbYppMPXX+1pXSpC32m5NPT7LP/7Y39AP25ySwAuR79IwFMqBjtahLdFzizS4Vz7tFD+4PEqd7Nvo24hmEIuMo9YCTUScKFY32sEkREPvDMjIqLOmkMWSdWCtmtcFh2ZTlcJoE9HBpm/yaIN2WZ5YBE6HRcw4vtMFr3hRM5gNArNmTbbV0apz/1pe/b0gE32Hhza dd3hVUe2xEVfyswAy0Mj7KxsXCV4W5SasnV3iMudr5y2ID4hx0v2Y0gcNNoaGEjC1tg+EGm02ncgWfkhnJNg7oIjL4/0Bt1X8Z2+mEIZf5xpK0j4RbdnTy9Vph61SCdzUkVvLFMcp38CnfKy6YEATwqeqQa/GkHw7kWdyn1dCIEmNGU1hmTy01oQ5H5g0fgeY8C7Q80MsDsruRv70+NoTjk39hLJzvBWiszElq8N8orwAUdzAw42qDEGPq+H5mThuhFdMNoehpZJmfGWbghj/2xYwJBrvdB5v60MKdP5BiXyAi4r2XLrsafYl32u81TjoZ+wSeANMwtwVjd0v/ecGj2+dUeP1dH9baVpVsTBcQ8P3QCtw1FDecoXiqjCkjbyS6maLk5MXeRaNfhsDARb+ccv1QumegJ18GaCEdI+71uSPDx9zNt+4NFr1o5dj8T1HzvqbD878q2X0LZ77yzog6i5J2k6Te9Ww8F5uxVGfz0V13++qYj0GEDJs4Ij5ziuvFFaeZsqgC/MXWcXh+OBmqHAcqH4B08wpRlwcsBfpgAsbZtldFEloa50TtRevgvxSk5Fca9toRErKLKb0f1KYQMFTMSQJ3ZqosnAgUbaSeUabAH9v98p7cSXzahoa/MDqCg9L4LxCKt1NHa1sS2kbp8cd51d1PXF04InRey/e5hs2L7D2fZ6yQqhtZ+2y/SgNTAwct4oNrL+UyIX51DQC5fomqoyxSpa6ZR3IboG1wuuDf9GpzJKDUE86e4NWPuxv9N+UQb12vs/5/KLWCN0osUq44aqSJH61xG1wnR2E8+15JkavDhE1UjbDzEBGpzmrFc2nWfEoF06Kxm4MxLgJeS4YIk4mXDrjtVdKF3aeVdyyzpmh6CYGoFPhv3HrrBUCt7mt2cp1uR1GAkok+NSXWeX4Fj2ITPv+uQL2JXHzL4tCPwmf1uUjXZeNqdyo7WjJGNUS/UV8T+Hxzsa/+UqsGvxSvoN00Co+W04+BvXuHa61rjRTfKPMm8G9knYIsHdUF1TquAd6TE5Mb2wFyzeDIbHfiJXTioN9aj5SXa9DYKqWQ+Ncd/Ic2hN7j+Gwtl/thb6X7ezmUUfhdxi2rzo1CZ6PxRb6XUcgmtESefRAMGz5Li+H+r0wdwB0in8FqmsbUZjEYgKigTeQovNbX/q9a9yY4RPGDB++o8g87jbzVQuw6RW5b5wuJv4oEy30neplrEIhNEIjxIdq4XxC0g6n8j1/hD0oQEGv5p9Q5K/HYmk3gXz7zeUahR8HeREA+j+1Xast8gKjTnzeTpksYeqJbBeIb2KDKWS+RC+pozW+Q00Ny6o1cPNmNdi4ksxq/KxVXoFJB3Dn/8EQBL10emMBQxMt70o4xFD7luesYzJw9QJZ+iiAf3KWe8AX6NI++bzXH+MezG30p4ycMhCwHOqWKPL1rfCcelP9/MSs1loPfHooDw7mqStIyjtcnNPAYVRx5qrkNmfRqimWZTBrq72uNyFXBc9E11h0mu1zpoXVkf0gD33QbrQ0w2RSoDw/KV1MSEqX6ARak1keIHpuXGy6KAawg05oFm1d1kNRyatDHMBbqedFg6ay7j0xVn1cucrN2ACQjeMnQE3ZeL21bRxMj5PSVTN39b0u6RDBLz8DHzqQ1A5A5iWFTCBnSuVsP/xgCN2hZip99xqo+0QULCY2U1v8MntC2t1PAB2RGHIRkAh6JArchA7b31AttLtnGZeecf1pjDdyj3ssxjs1PTunJGkJyB7l6Chipa+Z5AQfWHIPQx0ronQhrYpSG6C3p9t3/kDCU6m19/Pjn8GLEm4DE3GXINipTgE60q0yvs4zF9oL87TRG3T0m8loE9/nF2drmRIXk08gdEmGX/mwj1J3TaDIDdFw9RLA20o5EwR0Zjm95xw4dWte6JMVXepJatc161eTLRS4RkHf13ECUZ0BLXuuFu3aBdQr3sV+ctVvP5tvRH03Kn+7EzKoeKKnZQA0MW5mVIVF7788DL5aHvGznqEc1cfwx3dTeItVTwboHkA53SjXGQd72g8KpZSt1Nw3ghVT0a+zGk4hzH4FpRVgE2r8c36AA8DeqG3FhWXrjZIAO1iC5kxRcsHn0miMLquS0yJ+asxi0362pIZmWAD8LdMKVZSVpATRSAY/kLxRusVDrvpxwco095L1vN8i10iR0uZb+nW5nVtHP+wC8K37oMqwEbp0AeQKkWUZ/kaqw3xyYujfcSbnB8rsQAQdChUnrBKDA3GjWkw0suTp4wi9YR1YUydDjZqZHnB8YGcSQsuRsV4scXf8I0GWcd0wvMHnHMCKbcI81t6cu1c07ZY3vw7WmuWIFCstYaWQZhf2p5e/RzF/4qTrhHRS1TemX+vgpc0gs8ZwvcTA8Prw/nzFgmQtvsYqjZBS107YPRQG16ZdI+zxCOqsDjCg5iZ8QR6t31kcM3SS0jFEj7Tjm1K1ysItRg0oUdQIz2I5vpsuo7MaEfaKcBV

wpyd3uKUNuXjTgXE+kJwcwdFdjSYn6HD3r0rGTND30Yo2A33m0qC6IuUYT5gAvjZhKmbs0/ARTUQCriFyic47a
buHdPeVVG7gai5/32qMQAsf/rx5rmkhUNmeNrmScMJD0IajvB5SsGryNr7YFH2stHBVE/1T+nN8SntpTdluCbf
+1RbRNyhburqu54IEZRD0gPeUbPCb74oDGEIAp0df8QkYcmbu7n8ZHuvhDe03BtkgXab6n+FuJtVY8ndGip+6
uSrs8b1psfPJb2Gem8Ircds1V/NuBhUhD3wLZruSwwVqXPifrVrWaUmPHKrouYTdEMbLfSpisJmMtRz5cJh3R
4iVcoDcIsevnxncxIS40J1h0gGhdjNTF4bTem8QAQrgtkpSbevyTTs0e9Bq57Gy21X19EMPgLueIm2RraFKgKu
2Fa4iQ1g20yVQUJ17hNexJ0Mrd5X7tupgd+xHBqFvjY3sH9XjM4u/KWT5ug+8QzwnJIhhCeoQFgCvTim8t1n5g
4DvhnbyafCLnE3y0S01JSJf4y+GafDuDt3oJ2H9i0va58eNoPV0e04eV+1eXdJgwH1rxZvdDThkhCTKsdbES33
TpQScMyRbve094S9QK3U1N5m6UMsYPr1wfyim/FxIDkbsktgAwvE5qrk+2EAtJyGJ12H68IciAJXk2xaSm61j
nU/xqlbpLfGvyg1Kre6R8mGF9rX1r+qpTF/jSB/FFC42Zi/Rzd9VUfql8iyqh0XFteoFrRZ1/0Gd07kijhMB+
qrmvLco//y2kqnMbWI+cXYXNuf6ojfcYoXXEP+4qPLEz83ZFcYdCmcdevkNb24k607ymVgC/9M2LIB1q+oXKPwb
QTz0K1dcIYaqbUC44jmAbd8z7g24U0wTv1sfn2c4X7QL1iaBL/cmaZeC7rmzEmGGBeIzjmV4Bbr314ME4uAYuS
HRmz1s+jFDF8En0fCR8cF8eqr56XBtxHipMhQx0N20DuD/h7T2kL1zg2UDXvmKTiAsWKTFFIttuMqPa+eHijvfr
FJuhtTf1XpBx+A2W5fbKhQfgMbTQG+w4+UWHM2KcSLnvIPigiBNTh1w6Mt5Q0J1eYCvv74PrB9Qn8yK2aFc55S
JUmkqdEGSVsnt0pYpNn0AtwySS3m57L08eGs3KavQg+rjGy1E0SHmuKaIFqA7wS/jJLUSyi1/wnHKJEX/sf15K
N8Z4WBQF54Fc94nHH1WyKq1dr61kWRpFg91EfjnEEwm3dEW7SiPV9N/JadjWQ+FW3bs1S6EInZAejoetqaA8ZM
eaarc18Yy26hokz2gIhfVLi19eylvyInfCoA7tZgagAFYO/fYf0gXq3t3RVB+Ew8haL1zRfGF1Rj0XrWj+lamF
+3BsM97TkwY6I1FxzfkNQlyi0Jdh1jAo+0o4sDHE5VaXf3kd+e3P0F+CvqHrDVEFts4YwlPbZwneIyzvmPsqMy
obPQub8rwSrvNB6Js4SN7suXajjKabd+FxtTtQijj0f9GccnVXBssRjS1cpZKpqcitbXAHDxp9y4Srhc2uX062
dbgCfs4RVx+YXXn99HFoCLYCpc+NeOWgQ8Sot82ICn0BzXmYM1KmfR03D8SYw5V7uq107c/+QYE10aUIDWdcn
BhBt6im8VItUXIVf1IwHu0H94kSK91h2kZ15gGaSMs6J1qGIJGSLR6RGxr5BSW6JRHs+7w4y1UouV0179kYIOG
DkojY/VL0/8ojDrfm3Hjq3UXvLHmu4gKDekP0zZe06+CtQ/hcP++GcngFa5FQNXAEP0FjwqrAf2pPZKP8/ksa
HZUQ+psm1ltUuiGWQF98noyCwEC0+LemFddCz84hdUVZnu2NmTTA0nQ4vN4A2fp0kgSjVeMCLv03H50E8bMV1T
GOM2E1WbYeS+VUiVjLdHcS0MBF60K1j6rvhaB7C++86BrS8gJ31LGUQx12QKMne61crWv12IKni69fgdbP9mN7
X0Z9BwXYf7DkyPtRQLstL6MqhSutxjnXVpo7+jiyGKHWzbAe4RwDW2L1v7WIEA22BNDz6hhVt3LEtUR1dtYZ1z
w5SSWXIPRw4ghzJ83oeTeAm2UUZTtjMGx0U0yPTXT0BQGw0tLp/pmssPuF60XfHTqjqu7ykv5SHwz0HfB10TrF
Kso5fxCvKhNAuIN+PHCntrv6Yt+pt/U50pNW+k7wnMNB6FOIo/NwBUeBQIeszGUxDlamL6LYG9i6vWg4PMLTt0r
5ehtBcjwUB+Lg4U12xE74eSOXBBMndZZV+j0etyfDoo5T5uNmgzHRgVRF0z2PukoD483Y9LNzdh66rR9BZwVKX
x6sybs0XvvkLre73HT9foQOHxkM1F0+mQdKV7Nps0ddz0ww8mRGaEGVXAejhb0Mf03We4Yr+1DMFUwCEzntzM
Bi6p5C2AvtMb5+0oTk7BcbGS+JjM83mRWmDNRDFWpo6HNRwvVrEoF42CNw6JPLMhiM3puRwbobrXbXWg3pGiY3
CF15z8vJ3rhBMGpHWtAy8Rv0Ksneke0pam3WBKbzbeDTTSQnak7VeJKQM9aA9QF/i341D94kz5Magq1Fts1JaU
zmQJWNE+GtEtCG8Pk6bYKxTvmbpsjYTrxPsc6Zna5BEPmxZ5YJ06UXxuBkw871sQVtdLjTknoECN77sztXXw1
gKy1LeuFzG6UGdQDrcessiELglc/f9puz18tGtsDjeDditM0g/UcDe3TQ7JB7Ksr9N5Y+TH8I9F2LFaD4ii1JuA
BZq+0PJN30y64w5/jU/hZAn0p15kgj8b18Ct3AiS1tqmFmp8efBjS3ckm5Z0iQt/o4G+KqSFw/a0afvG8gDYrr
5DFkHE3wGJ+B1w9abkZ12kJfFv3vaQp2n/y3a+WuVGHuWOHCCYAvnfqmSOfqVKoATSonryAoa0D5xKYX3HPM18
2Y217nri93fSsP8KBY+eQbUm245TILExmBqgRReA8JN51wUvYykwLnfNSTaf+k8A61LPugFJRH7xcAIZuydV7U
Cfp9DSJbWh3FiaU8/Ca7XgLg3X0ZVgtYXy5dLXeF1beTwzxCZjeIxtQVtPjZt+8RftEWJihY/cQbA8Vnk3FQA
cxEsQE5iKQ1D010Fg0vzds9GBAUykn+oWkexvWVn55km78W61Lvv/dfUd6olGtiH4iuNAD8dcqyI0jTcwB2Am
JWTwhyR0uoMBV2Uc9XfJrpKfXrl9dtCSPN9YnMcRJ+JLvgH4d58hwdpUoo7paXURs6MXIxc8Xsash0mTjx4mgp
wPqxrWFG6TGjwZpbQLFSTjruU3fH75BfW4RkVPin5kqe/K8D7U2rR11Td9m4cj9Edi0lmlVzGuts70yqvXsAX
t6d8/GXpUU/w5A0zZzeU86U5YxZI0Wb1rBpxohRXW02Lc5L0Au9ihDz1twJLB/KhWq6jE21GDEC9/d2PMTZiPg
g107b0M4bN8HBCyfZMyXK2ciS9QjfBKfCNIdNpu7Ycov6uInGpe4YI0UQBnTak9Rqlv2Gc6/78o4GFeSp87RbQ
QMwSj01u9S06d0KJehYbtLbgiu4Zy3Hdu0K1idoUeZx60bj99/14LWSgyxFb01x/B8NtPxfdicajL0sdKsdGvji
B4Kr863egIyS+aEHTA/I0z0pseDvTHY6xzBpk7F1RopI04Rczvek9pXbnNeONQL0FGa9M1FP7KPIsABH8VnTD
vgLMMvMmmpo1dGtDla2mkjYNs0a6AdRT0yS+/Phaa7sNu1vC/Gzb0JTxp+1ddu2SPbFkJnS57p35Lo9V40eX/o
n7HNV+0v7oebHSGGYUibF76nEeqoQ5jLCK/phEDAtz3sEfeBVBHyhAnQoQVjPZhJkxxjSjkgf1M1kaxb/eF4a4i
xwqb+8bTP6k/F8er8IIIt7HD6ji+puYnGoMRo7m/gx6CXyUwsS2SbJa3asZ+yo9vs155w1d1Y/3zAHyaOZ67D2
H8U8IR4VKRSnAC3Q5c14qEmSBmDbKPHnSATQnxJF/vpr6I8yaQmD3XY6tCgIumyE+0z8Rzzod9AIn2PXW0i+9U
rSrwwUF6M9rfXBPHgFxK97z4it4H1+TLa+XAf7aIAeffTT9NspaH1kjtstD1KWC23J0sm/IsxY30E/k1KfNi/V
SqVT07UNGK6rHz0dg1nmtegGagtU32xvToic48XYH1Cnkp+upv3tSXgSJd211NgjqSE21U691Jmj0MxkLxiFD
MYPbsKtPXep+BKotfZ4uBDoSUBuZg2mA+Q3ZkdGmRVPwadvD0YFz+R3sGwXL/Q65emGcEvb6rUkBHdJeF1N0o1
VNClyHwpEnWUUtNXjrerd7PaG4AzneGvDTXZHSDD2936IiDfi0mq4VD2mkA5w0q0Sj/TbT0w1QYmjJi5IBZxvc

ZXWVvyRSmRCf7ojnq0beQguKAbC3c8IrQodOoM9i0/y0E1EymtSN6GnjJLsJKbWukIZBkdUGQDIDLytiVoHJ9Y
FG2sJTz5sE+qgRf3SJJYZQLR/zm7BnDYN0AjVN0bzS2/We/PgMtMkOUYXC49m4cikU6axnK5vQx7b3SuLoioD
HDL/SAxyv6uqWo0TdstZTJR2cRstN6mpxGnt6KZQJqT1iRbo+M70kMIN0ZsT44fGS0k7Nn60zRXoBXnYgIHH+D
8BgC1mUFt/HpEzbHsL/jQ6HTt7+TphQKxvImtNwTFSf1r9de3sNHAWbp1gJbVTKtRDMQbH0r+ADy9TjZY3ouCw
2U20PYTt8mBF13j6b6k1B2mj+ajH+XacsGeCTX41gAoCRo28GEoL2tomHZWMFsJsdSpw3a0vLt7CN/t2nnnkoE
gzIw7gWHGcz4U7AztxbGbyvLkGioeh4D1L86gaS2p8qFHQkCaQtADTkc1cX8bY0A4/iVh1or2NuoHgFN5C6IO
eBEUL0eHnwDITkBFoBEHxj15V4DKyxcUz/5sxN1p/crC93BcszIyPP1618EeQghqG2RnsJqDGfQA1RRgi20rc
H41+Jya2yGVfvSFJrIC3BbwuqMLJ02Pt6HL954WHJMI1rDhuXFdsPrj0DydzNiQfcBz9sf6W3ra6nPwgHa5hhm
eKLUvjuNU/dd4MYqVUN0oyia2Y/a5Qah4q54v0tQAAfYHt6xAtiE5IsUthAODN7NX7dQmLuS7SP/46b3v6tk6
yZRStwmokQc4JtizRIKUMFvoEQxzmCmujWIRCCdzhr4gPkzbtzu6R35s3/mHkIlxShV3wFs/ZHrKLJP1fWa3s
jW9UUy2EWAHBuCQ4uPFB7HWx4GmGAbqXfu6tgz+2t37jMMx3IW9Rz58CFqqi7q0G5TvRhAGrGU4pyUAKzUXnt
FdnizL9gBZE1raTUV3FbR6I90Bwpxe1B/T83A6g9CRIoiJJ/oJ5yqAR6d1gEhVb8jemIpTv4c7y4ES4fnhX3L
M1FqKwbPMTwhpFMokC1+1zcXuSELw6pm/BTIC/wYy5mCBjohNGD8xu1EYxGUTx0cDOLi03BsU5Pi4x8mkow6+u
nSj2oILuCqRT0Xv69vLiICg5Yg/MHAGp+7DB/RbD8TjAjeFC0Ds/aSpOKjzqF4DSnZaYtLDbu1YAJSKXClnI20
eKmNy+vVd1I6tpA51W7jev3kzjyRi7NqvPiw77N8LXZ26cG1C/CJELdFwFSjizKW4kYtSnTrZK570ntpJf09J2
3r5RgaSbV4CLfv770WzBdqjAMpo8i0kNPCVFmhdqgvBrBw/05Nj0q0fpDm++9a5XX7tI9vRYII/W7z7mEZs0
r6mCmKiJqkxcQSp4vxLE4UguKvDXZT1ZZWDKiouhH9xsQx/x4LP21XHKL+vbP0hRjwLNcvURqSVk0daAo2PGBT
wrVPtNqGf16VmHyP8/nmv0FekbMd6aAFefJryWe7V7taNteES+MJD1fdwpFMDzTwG+w2o/udyzwEirihPY6CNI
EFpXzzXq+S0DfogU9NFM/bd3xuSqAa1WX3u6Mt0nf72bJ87T0RdbQjJnnhCfb1GYGsabmAvhZwoVL1RXhvZBVP
fYc1UHYH0L1uGrw6tCg+EAJ02VZ11Pkzz49/dgp0mp0dZKmU8M3p20q8e+6J6y17NICpwaQ20HBq7doD62hDp2
7vh181s1/qYSYFBPa05GMP7/0x46wSAgJ9afkPgYuqEX71AjUdBRuOue72Jvmuh4wbKy+owjHs64rAnNzMqjsX
5JLTQFJU+qjGglszQxwAMKHGnIcgU24Wv4PYKjXweqOMCCSwI01bLF45sno+bQI7jP7Vv7E3i7MjI3XWn4038a
a1je9nzYoYiCL5o0QK3gmb0dQWncYoXIR1scE+EJUbFEDSH5UsnMmZtHXXYE5x9LFzU/PFis21PKnRNX0Ex4x
zKVS0rn+Bc4HTopUoNgB6LkaN05qTEYA1tpVrg0t+aSZeBmAL0G8pJ9r3Wsk9fkknIX1k4CySfx1mPet5Qt81
rDcHXcC+wc8vftgQ+V0K6npXL2tV3BeCVw6TwAmIQ9D8szQhDGuN7yP7xQZIbDw7PRw5yVbP/SpACCz+hMYvbr
A4VV20bTqQoS0iS1DUXez0JBGk4I6FPz37wEdbxK4Y7NaRJAiHpjPz2NgV06B5B8k3IL3hUGXLb07xQq/GQViQ
Ui1N7KeKL6qu9JWnv16KetoDpz1n2G8ZSPdbXTkfCAR/34LF+xX6RxVKhj3wxcgrS4j4gZoCUBS77YM+HMSH8e
UHOYGTBPedkrFmgPv03P0zFz0mW+YA+NA590MFyPQgrhSYyGLfSLBH/m97a9ZaTrZ7pJS+3P2EC67FAfDi3Uwo
FFeg3+1206rP83ydm/uv/y6es08YQrx49bL17EcrasgJBu8iFJU0cgz1z0ilzwGhjMcps81F6rRU8z0rwyuw2
Adyrg3WDrLZXpMcegYk5G78PHaCrgxHV4wqw5wE4nZ505NC2NqS9ubXxXDyEa8xoXg8foBbFWG281zSn1PeJ9wf
Q2GQKugjopDeh/o8XNsdgiFEs92q7i30qNXTlp9qMSSwdRNAEagzRqnUa0Ys6u0021wJpuuuR6mLSpe2vQSD4A
8vwMrpQwK4KhWJDoi4P3kjSzRrRfBG9H8uYhEqqfKwt0iK/3+IQYybV7PK5jUGMtI9Iuow19d5FFv23vzMzUbx
JrGaG4o36ZdyfiDGfJHsg9HQaLCJwxFVtzBQ8Es1lDDKV9yF8QkS7RR9iTRcKdgcS5F5RfFdApvM8tISwWfd
zWw1DFweaC157VWP+nhfErn6M2qn8CBZottDDgz5xQ0+ALECw6jp17Td5C7F3q/0B/DFVswADzje29QapKdsIO
uNpnTTsEEB83jedLPVvWg2bELz/Y1PapSkNYFdJjmp/XprBoGoULq79sdBcwI5msBderQThDmoyXV5+KsGDY3
0GeAknaXjJ871GRSQ1qjT/wMNwSX9BJrQ56gHIoP1I1NQ9znGY6oHDyov4EE7Z8E5007kZ+YU7X20addCIT23b
VZ3JzW15K1hHCZ7NqQm1y2pGgN8vYJ8JAuW3ywCSeSS3n1/ef66sHFKY3+vX9Fb03TrZXBPsqaQ1y76nZHiUK
U8qtC7Nn0ieG4BWryEIk/BssnEpnc3RiF3g0SIA66C1vql/b3N1jjo0ux8EWiAHdG7GTm1NeihIqcfm8p1cmF
cHQzD2kRkPqIEEjNeW816v1Ir512z2vkCcvM1h/g6ZN10Ksf eoPtpsmzW79wTCCB3upTWGEJ5Z/u5RhwnPepTm
hsA82xtcNQfCx0NEB+mzbwUdX/Gkqg0RevVaTU0XnFqyri2i70CL11U1zY1bwzZ33t3xRpK+uXnIg0yao/nz+L4
wJPVHjh8/I7ckLf7g2Bt5uQRsTbNUrJqf0oC+RfQKDqilwbF5At/kRxLWznbawNuZ1L9PtcrFZBr+pWmVIvyx
KqNuUir01Bsa0XVNyxGi865GM2YLK9/IkISK/IoL5BKxN82ZKM80YmyhPvKctqzIqwb56BdQC3ZkBCu/ftQLKW
/brsJpFd6TUh//j3WmB4pneR0hc/yZuHydGYN1JNdJhxBxNinJk5vvZ0dLct0JBsQx+/0wF/5wDcAV4YEDpJaI
Q885oa9szdu7GkZIpT7aAc8DsmYtrFN1YjEd0kcm5Tf1BOZIu0YUZKACTfS13g7Wd1R5+W0km6wthmMQiRMGy+
RSvvsv20arC5rjQADHFeCEm3Lmk3sYN3RQW0119HNewz4P8tHeRD52RFjKu2CFb0bMIR26swRZGcm1Ee2WP0Kw
2rY1DjHR3EQcZn5mY1fDv4M1u4UeNHZhjL14rV0ySN/ZQW8EACHD6WQ4cyLSTG8iH0sysNMejR0jd0Qfak8FmL
11YB9Tm2vXm0tevJ6aGQxFkPLYrvFBS+rYPshuI1yYqBU6a05mn3Xbwg1czBrmwognUxL20FSeqiJxTdrSN9TY
hzuAI8vebRtPinpTaYJp072S0ocSdUwbJTyvBcm5XfR1F2UAGOEdZHUEVdx+hFOZ1FeeQ/p8rALXnkM3UORwW+
GaIJf1XNYd1Sz/53IDFCTkc+eQZycyUdE/yEPLLAE+athQWNVuufVZQ50PBA1+Nx7fbpH0J89WEotm5qSkbt+8F
Xoh0NT+j8zsPGvnCnkXM62ejBhVVfJu0gio/wNyuuwh16TbJ9Dgw+vJcCgbK/Lr2cpyES2TteSmvRuzvIW/hwm
kkUJgy9IBL4W4lgndDX840rxm1sxu61vSA0g+u+1HDzBTf9dT34aIL4MGKuuSjhYCh8b6f7jwP4pQhwUFyFQVM

ajMNKWpzDWU0apYhOS/sgX8w214MoS77jo5ulck+sbMQMCr2RDxV/kwZdsPhKvlM2pMcJJxJtLQFGFxzNcNNU4
9wjxGjGor6ux4aAIrbDLuXN6+g4jC0qTw4UZZd2qgTNLm3A4U4dcceYku/wgYdcJIjtGNLmiU0PkX0kF5AfL1j
7w01g2d2Njef/XGi30DYiGZ14A5Mfj235e/WBbkMWJvdNa4U7CkLK4jx5A5xPoRIR2f0kbMyyvJdX0hGzB+cRQ
kHaZLFTTduNGLNy6x81ir/mmX03qbTEypisefjKgE9g5JhXmJAtWX/YztNw0ZeUwZxhCM1bfkkeqXQkIi20E
8165sMQX1HTjATI2LIvpweyJHCSyMseHqba2Qd8r+mS0vhznc9q63ZJAFj11km0nA9oYCBLhPC/1ow12euoa94
gNj983bHFx5w+x2Cd7nGyL9zhThtFEKghuvRtsr3SvfQeq41XqIYWcnc11fI1cf4SM5KG9epeuXY0Kwnm4vMG
FXD4PtyjZbJXLtnLH0g8MthJ9fZVV6ZcG7kv+1sBaxDf2gHt6JDFIu3/Vdk3WqA3NCWpimWLHQpjWA6xWL6Df
I571ZM8YV5TB9BFndIJMguCwn8agCuRDT2wsRLCjD8LH4fYANBsEmM09U2idy6tlwIqWz9Z36UQKkLC0NL510P
zQ11RzpLNLhQJBQAoiuGgv8payNsRsWkks7jHMomrt/ruu+uz7rp+vXcf7UKmUeDmzDwK4dRLSuIVy0s1gWjbI
ygr/ZxPK3vi4J6PKaYnq9/miotTW3Kqny9FpFT+txMp0/2mfXF1KxGpKaG/dXac/qkeHy62eNuKYzh1rDboYe
ofNbnYTudFHTGMU6ZR0c7Rz/JgUL8Igaij4zHQIPF0oxCREwRmeTs0zPkRC4RVo9XMbohX2Ssw4VrGEtAIfAJX
gEY0iYpphGS4/bjvp6yhvuhzXV1o1PUu3w/keU+yBomHTIdcj/3d02Ac/u5oFfLLF8YlLst6w24oqa/p6zT4Ap
Muq07y+Bh99tgW8mf1B0vL+BanBWVq8B1kF1kckbIJ2UVkDy2D0bLWcxs9AYs/uuoqOQ1i1qgCGDgJSEUbVbMG
q13nVjwSbCDzsH4ndETSLdNVPy915Jm2AwNQ456NH88BzELqDYeITEkcGhqnwfhXMUIh1tVCz2GU4uR2U22N2H
y94iYCDoQ11seGtKhapz7Uy1gJshZeDN+2RaA5TD6D91PmU1oL+dYwE/9Uy0euvVFHcWukfszuP4abBT//Tf
a4ALsZSVj0KTqhhNmmRrxD6DRF2154qJHr7pAn1c/2x3vxuGN9Sg5mUU3YUROv0ouyQYD1/AtxAXLUWR5wu9KM
qqc5w+8YX0edogBxRHEBnjY+Ez5EQJ0u7+1dkf9TaSj0aUoTALdUAWK8yqz2AXe1m/g/qJstJLu+mdGWbc8zvY
nqCuk0BmpYgtTeL+A0h1D6gZ1XP1HLwWIdGY14y21G7cz94NYxHj70AdyorL8QoQLSuQSgtPcn+FKuLM+H9S+
xQ071UoNCysksvJUihiH1XbAgBb4L9r/N90sW1+h80bLXn9yWz000SmVG36fBTMqgvDWTN0+WFRg6A1xz4YCM
hA0hj30ymvDBfHQ8h0D6zJS8yPHEjxWm3PxGx/Xy9ME/HoawNe4sv0LNuhf93aCZNCuHeVzFJtZUA3knyp0gaV
zQsXdf20RpxDGqRgTp11o3HcJW2HsbF5mMk6GE01SYXMxPh6tzmEP1mzRd7gXQ0W3917eX7YLu/8EW3HN6jCYE
6xZ5B30mg2ad1YhCQC313T1rKJZnaGj1hZEvigb9TwKS4j+5RTE5Gzx+9jDdUjsGyoDyrLxykI0H3TV1NNRrI
CrZ6p10akw3IYpNDR/TpUzAQ83d1zL6vHdAldIEN57iC3kkpC2vIuizjD9BT2E23eH/XCKNsrbX1aNa+T57Bm
IhdqGnYq4okG2SBm5sfGtYLeAQvG+TLXjEnZSwRIBzx00MNhe/oLjxDp+ognY3MqaXF31VEZkzCG+C3JB/aQe
4XaPjwEgSIKj8qjG1as7CudpeE0RjCWHBh0xGpr1knFfwzB24/eFXU+3MxTDBUdd/6rhK2SCWYWhNY/HyXnPD
Y7XbyNkucp7PA3dbuB7n1h145w93c2KsQavR/6Y0/8MvpakAIh0DJDp2w6nTIdfctwQPLG5PW0a7FvEK0etx1j
guJzPn7BLwU7PbkhrtMMoReWJiZ08NKMndexgTvY8YCKkPupYWrZGAuDvfUyid3iR/uGBOTRF/p2usdg08csaH
q3wXiBjorPPqP3VazxWpcYhh3dXNX9zImS2eXB0zIQYl1aJ4d1y5KYEB1fXbx3QcI3/gdmR4golflJ90NZ+
NXuV7JXbSoTC0cCb8PKEQSuy690rXPJD3TM9Nh24hMmYnKDon7AAZ6ZVti3pgNwtzh4JExloFjyVl3Smjnd3zJ
Hk1qdQ3Ht/JT5Nx9yPCoj2z2WL0io4WDQpF0hp+1qF1bUfKKvxz6FjvD6vP1fQ+3C6em9W4ov6hd7FPr/HwxB
PkGu5s/T5LTZYRCvp9HOXx4QobUsoQX8jupyZq11B1GUYf1DcC60kkPnR1uF8ge+ysT7o/93RzPDY/dpRURCKR
H3NLjbvsC0mS1t9Sjkm8fuXehJ709CAxS+27op+EcNjclCtI/4tdWdVE++T1znbPts9aluH4IqZ7L6WomSwk
joJvoJ12swssWzOPVaCIhpdtYCPn1xWQ9VGLV6a/7L0c5yBtv2UTgCQYW3eox91jFuCAVMLGBpX2wIx84a0en+
40xodXluT0yhEslRcIiuMp0+k3p4A+Fa5kDw1i+FABwrEQpbpoT1wdceUDFDwv4B3CoFJ+VksuVEx2/Z9kGewF
1L9FF1RPrngLAyT7PxNHudnvp7ZmP9rxr0R5x6geVvzG8JcfQimSf1R4Yi1CsJgH1+g0dTwvAtrkXZLHnIumgW
PyYYMxn7G7z0U4t030zpgtadTfqKGyHpuWq655vXYSDafSOZAW4qJisPZgnhVaRbptogVHUqRJmk6wuw9GdJt
1qYZkrEOUTBm4VXPP+MjqqpvJkxILjzS885NmL01U/17rDhYJrjFoorjmU8/a9TeaQoHqQTIZuRjbPN32hKEk
ONF4vuEwE5PPMaHs6Qnizf7HXJIVoe9fts4JrtwzenII/n+7Md0ZiCWFcDyhYteJCBfHg420G5Z/TKp29jxkz
Zh5jCQfUusDSwbU1loU3qT7qCrZ1eXBBo5vopvTjNQ/b3S99enDgQmDwB2JqKfindHooiPrm62nRS23oWCX0iq
fRv/TtXX7rRtQJguh8013Nu6UcfuU2ZJFqhc/ERNVSrAKJWpAlk1gT4kQ5X/+sCn1Pyah9m93iQMLzK/qSbh
Uu4YTRL9GbDqBKqw6Lhs6Uty7g5L1LwMrzgdZdcgUFgadcuLhwGiaNPKeLaIiHJuRsk7w0zS9E5MCT1SbKMN1
K/LxKGnzgqvYwfmn8s8901lnukvYqrEanT9/WMLV+zTqUb+9FkhUWX6wkq8ffRKp1fkBU3FCVLL4ZRgf1uW0eA
Nf1GnRL1BG0Qg4Vs+YuRq3eR4UqRtI8J5G5BbjcZMgr3knU4hth+7WzSROkZgKvQExnBPNxWBvynG/ujkhSe
dF7PqMgHe8hAuYN+Lj21BVUriUN3fI110RmR2jK0zxNvexTlw/9KK0Com04Z+d8vzF9kn7r5Wx7VBPUBwJ1DCm
1RodV9X1g8sY9YzjxhymBgTUCXx/ErXVKemffZ8Co/A4UHiRnJvQAZFj009w6Rjmg/LJCH+UX/RWibih5iAMSG
ewsNjUz7hq2qpHA6CSx29DIocnU9B9daWi0Gk27y25T6XYxuJAZ6GZP4yUGq9ehimZFDjZp/jXSmQMytDyVPi5
YHDWnTr6DWDT7bWtA7iMk1Pzd6q7UM0Y94bYPU70GxW7GEou/uqshZp27nhhPKkHktNf2QivLrltErhgGtmPex
EEQuV1hD/UY1E7uZi2BqjPFSWcY68WGFL9X0c1j8pdYwr4UcNqZ5RY28LCa++ct+8mpchQkISKwPnIrmEw1iSn
VNcB0IN8bBzSC0yEKia/24rW0HoxMsfdm39pWAnPH1+VyyLiuytP0zsCG7m0xRG3eLueMt+p1NLwG/68ryymX
V7vxRtKnQIZ3/u/85M/cyq87PsxnWZZ+m9RcR0FsEnokWz1bj9VL8tEVfYhGK10UyUoo3dHEwOABWfnReSeWx0
ZvuqXzyr9HPo+wtQXJ2694EW0GW7UUf0mtkhd7CTkM6iVXW5NwnXFe7o8iujX45pI13CkwhkTAG/k5ThmHxsqy

VbTNy0kEGjiS6DYnHY0RWktKtacucuci05Z1Zciy817g9tMMfhD1b6BaT1CmGg5H4yJ0VrCXfMNBGQAe7csyf
VNauAvGMp59uoTcvp85Ev/7dCEuxCa/ItwrI8UkZf3ia5p7zwDYZ1fgqXjwYfZPGJQJs3B7ACYsj2/7qcosz7
LDMxZQEnBn5kgxtD1f3gF4I4Hzzj8SDqP/Sq7T4JYBNUu5Ir88aNZEwN8XfySwm5isk06VTDZreuz8s5fKMgM
3KL9Zr/1VHMQvPijxPkGmU0kfgguUzeGu0xqf6uC+JbWjZbxnaRcNIslC2FmwfPEMzLIfv4TP/9SmcahCoEcG
MLH1FK4EVbFtomAs9Rod7bd09eM3b1DCxIF32ce4RILnx+R/0NZ3u14Xtq0J1v5kdbwxerZ4JvZ915cdLrXN
WKLsxJ18PHc6m6k7uog8EdJMyfc81ttb2luUe6YOMdu1W99Yrr6YquUJ14ICJfQWzfbtNDzfb7D4XJYxA4j9e
zwHL0SbjFLMB/WXn2aEUx9XdURubbffhwrss7tweNxLoSfqFwcaiwik1g+FaUucKv1IJZZWYHvmP1H4MrmgEO
LP1hky60fRNv/4UJGfGoM3caoieILjTSwYMHSwswihvobeN4N8sP1mkjpmksq/J2USgUK/Vrq1pi8cS6S6rdW
0HYcMhmch7wotyVfIc8NTer79r327zIJjn4EH320V89B8RSEGyyJZHn5JGc4uPMZ+AdyOnr3JPzuZcSwog/f0Z
QvEEzrUnqzsT9vMFtv8Jq2K3DT3huyJdD1LeZIZA60LQdd8seLUHulvdkzyKuWv4J2UXP6vXJ80UoypLi7TyTf
VGcWnRWAphsGxwKn4ZEr947QRmxrhT4ln5S67kMqip/i4DNBnluXIie6MZQxkp9eVnIFGdRU0fVUdu5IK/t1L9
SfyH7asWXFRWQSLACcvgvt0MIN96ZBfqp/LNC5QSz0XRUQWJ/Z9NdGqavT4a0DqyeL7ybqg68rhGx+A2aj0wR
Qnk7cJ60cgLfRXFmDyEhSq0AIoiXq2PdhHxn+Ita0PxR5uppvSixnk3dslwKu3G3I1QEFZQ9GcRhsaIc6Rt97GA
8FpvHs8oPcv8wPutnLL0D7f7zGjzLJKhPpW81FY8gs2mVztr/T8ZSsC4mjIzyn3lhx2SwwQ7iN2AaVm9ytvTE
Mh/UuYUL256pmbQiB+7Vx2c1nz40xxZyGG1K6hedxMc0LP4LqhiDjT8+5Y1Iu0bMsmSND2irrbo7VL5Dof3cT
i6esykpjxvZ9sylBUcwyadgk6kqr+JiGHcTfUyMwLwQfHF10ZwCfMbYowuZ90d4EUopWUDOTQ1zqMb471kgML4
0i30k0ngISnDBU61IytZOE2zop0fvwjCo0L9aC6UAt/oDPLFF86pQQFdj6q6j2EJKbLoD1Zhm00yhg0dqKvGh
/0h4gUnaUuLjW1vNDnHnei7T0LQ0NroHUsJ0bEUwe80mwtabjDgFuQWpmOE3wN0RYHAyFkhRKzGNVJjvju0Sji
SSCuFJkYnu6xDNR0lxMy+1U06JNcU2hU2sN7HzfPH3p+gMW1Czw6I/HVCcaYC9XSKBRqxRz88V41uTNsmADfMN
KMRT82id+BgGyRmBAtLo91h25tk5sXx/qRFwxz2X/vLH6HQjwSn7vJBj3nXPfasTMu7NVcdI7u5jtFqYTh4Rp
1m6T3h1mJdfRt7n6Yd0WY15F7ikRSokmB6q0UC/19XS+CvOKW0z7laooIbS0guozGFTtVB/1mIR7d4IZrChtl
V9U+YZ3bHcgeqUZnjTif56yBK9mhI1DqxiwEBXrnTopF1N4pLLC05awYrk7XtCoMz02YukRN0bXrusLwUpX/in
fz4l+9NuqPNGZf4+ZxBFZxP+XSrh1EcPoJ4s0DJ2zoXxbFizDCxZ9Xv7lwE5sH21u58ZobUvR1MUnwo3B1oSIT
KwcC6MyGps8fZuuhbTqPbW2PDn0RHhzkZewg77sD9/cvL0600kqoWdQn19RvsgIA+p+01ZMnxB910w7wIMJI2z
rLcq015xx9CdTau0aKB3+2zyCECPdVykW1lZYcjxe7C6LUsi6Sba3Bpo7HLW8Q42khez7dbMLruisMBzTJACL
jAe63nKj8hgzbCQk4Vm8vG4jYe4Hpo2BxRQdkLV+RMj/ZWTpQ1gIqWUucNzxU3ts1VUqbUI0g6apetmVWprhDd
jXx3+8pFB12d1kZ6FD0tVJVsY+kCc50CF0Fg8p0vE6Nu9IUilg59sGMB0qmjFO/EqBjI+iVDpw4Lq8qot/mr0a
PjExYiRI4FFG0x410Kb3pcBIDF3fh/SPyqtYrU0csgEPFgK0U8jHrGqZsd3kIqm0Ruyz1obnRs9KaGyuEka0V
eJ6ZTynyHjM/kpxEd9bBwWRR611vqMYvTE+3fTj0fR6soqZMV1zFoY0xLYi/M+SRFYrZewm6noGzaNPgwc1zX
Sfg2qI+PsCsdYQef8bUeXY/cxAQXQ8JSI0SXZP1yxvorEXG1CFU5A3GNeI7jVwfrWTpdTyLrain76ffHMOmkb8
cdFedbXIjeBV0k1kGtT835F66PCXe7v91F/qWFpEu19yA13naVrjkgU2q1QTa9Hadk8PX1IF0oXeYhzc2nI2E
oyVEhy1x1Z2R/geqSMe5kXaUKjeNh7XRmxRhyfwbaUZ1jekGfJJSEatEYVLaBusXzdoakZPJ8dpZn5Q58HSQFb
/VQ1fLzeZWzujRFB816xm28H3Nhc9X08Lt4uMbxP2zCgDokF1EJazrSR3eCgHsmkI++2K9JJZK/8/hh8TgrbcZ
5pW6SQ1P9IYEmhKpc0yQzMgTAXh5WsdCwv1jNEDyQQtrPpEb5RJmDrK8QEH+m19ke19fFAMJ+T150//Jx+U4P
FEF8LLvjjfI1CpsDCPwmkoLi2UQsLSXQ8mvZ1j0RGBDWah/uSefdRSaCDkF/1A47tA5oYtE3iMsk4pNjGZPIx1
HdEu9094xpISOGIwjBQqt8Zh8zwQUV2N7QR3LhIuh9544W8uQvsw6cfZis/Co5TzacJSbQG9pLYzjFD33SYA5N
c/00xEXHZVGEqFB41NQkSB6Tj+z1QE+pXmxJuekjNwYRbhviEh3UGw7Pchct4RB8Njv1Ri6d+Fa94vpx9901
Pi/t4H4v7z/KEA490d2v9Iniu27mH0AcSic/qqCgBSZAL3cYVN/BC6041EBNVSr/PXdQTe6VXe95zZvrcDIzuD
7NWNPV97Hwn5IqJH8BVudjq0rY/03eZERgZWYUJP8xIs9R00Z/8dUYBwh1ZiwpCgDf23Mh3S5r57VaWR6SBtCQ
1Z+w/yU3h8zugH+5f6Nv+Krd4ZJw87sJ6ZpEdnI11ifuST2dTNPWDNbvh0LS0Fe6E2Td4d2t4jHtE0RqzEIN
B2Bn+52Z/bJGUTfpTZYooAoQ1fyPOuC78G1VEV/w130NTMU01PkKGq9I/g2CJbYR2ePGE+y21XOAzu9wKPgkU
2P90fo/zpeL/sKNEVSX8DOVSGagdFzdRK8WSLQj0CqP78YdXyEgSNsBRZS57NqkpKJT9gC/XS6Nyg0Dr6SyLH7
1oYsVtQv5W0xbEBnmSBVgMN48tirkN5LDvKKdTNdJalnZrMeBRp3cCPCCq0k/dv5S1zEM1Zok89GLi65jGTu4
IZXk6417JAwa2JASDB6azFw5q3Rq3e/NxVvzFum21Qn7h7BkTX1X0QBLUSvJuzcjhMcbJ54uD5ApNx+n/OPM
78FPjNnYHxDAYNugnnng0MNI34aI1GjB1SrmhphsYHaqmrPxQ/dAVJxnywMPsGKQxiPqoPnuV5I7vkY70kT
LEfgBDM6U0g/D5IrdG8HvBCY+0nnJ1d7vcRFCuj04dM4CaLiHEfdG1DrQSgFz7+I3WNydtfz19k6cZaeza12w0
o5/wZkKdzT+qGeCqAi460gX0n7zN4rVJKZbMAQV9DQCDaUtz1o1d6k1YdRHWS/bWQriNE2X0yWyokvobZnLLf1
CXAzfeRERXTfQfapPeYdh75160IoD/4hKBk5QcQwrbW7XVceQBoPilRXnshU16y5d1Jc77VCuC+QB+R94S1wZa
woAKtFprRK6B28zTgt190qsDg8q8u2s0wPqbnHaa0r7hJk4pGBFXqhrpagR7zi5Db0TxpjnfY0rKVS5a17lUMI
iD1oRhZ03AoqZtv7U9C6uc6amN3PZKn40txNPgbjagU39Wo+B5Texp5FEmsiugin0X4Q/cZ2BbjtYQhWo0qce
wj3R8q3UcWItcKUj/UAjtKzqaFoNm0yUczxcJe/OM01YqesN7KLJTp0PgeQ+7SMkD5cL2YD34S0ciYiBD6qagn

5x51gi7pv+wR3WlQzw+zfVFM5nVKQnW9mhWi7r/oIPj4y7uY+zG0Mt/fXx7V3/FL3dQ19cR+fbRjR4KZP9xMSp
r4MqspA2hwWy/7dt+dRRut5LnpjcpOPI6FiXr6GQS4xBoTmbTKh3ANTun+nzsT51N6whkTUqHXWSGXQr3vV7E8
GS+XV8mQZRB+A2FZ50w7hE9K2KmM4S1FMr0fGzuPTsbn2bjGxZoW2gx4DDV84Qy1ezonX/twtk3RLwUhGt3NhC
o9aj6R9W85ygnxAQ+/w8U5wPypTssRCxwWuxT0sQMLAtUEFPkozJinnoi1yMRakyhHLL6SuABpCiftOA2PwJX
U4xDiT2AR3wgXL2S1ITZ5ktDf6rmTEt1d103mAM3dLnqda237IgZ4+GaA01yB4MqQHsa7LbIJj3VbIwST2Fr
ATGB0V3YcgpZFZJGte4RIIzi9f5J2guTp4e/3ZXhKM+95uHIp7dcnbDJUnThs3BvNQbuIfpBcXMpWRnSx0Pzmz
Zbju2tRc/mqvcj6YJ62jIp+NAbrRAEP4txx4Nhn1pb9yAIbeYlkSbwPZ3e/p7os578qB6ByWpqZuwIx03EY4
fcynKeB0szrAehhfH0qRY5pv0j9gc7sHb0/fg/NPpPsTfRebyW+oQ/av34MKznlSJQexthBMnpQbLBhgiKiLrz
o62lbGaylGGv01WwHvTk6A0AEyRkPH1DEKnFjLqdTEhgOdWWmxvZDaHSnTs0bVauoG1M9F7JNtDjS37Zp2r5Dm
lnP1m5N1wLssD+E9Ax+VZfh0g2vBbheh9ERWg3pSc17S5wJW0kQBAZzrC1Z8qzJvHm56iEZ3YfVWSivZqusMGe
sL8UvtejdBws51zhZ9HBIBn+pCUOEkZ8nZD0Asq8Rcd5q0Jc6UjPFZKJeOznTgc6fJHTn6xifsfT1zRRg58E18
t5T13h53gd6dgWJTDRLExvbxN/NfDudvrukRp2IxRNpmT6RT5FWREbIQeVXMi3RBI4Mu4+CXRW6/sMgN1RNQom
xW37C7LmmAND5uDY9Bxr0rZUDh0VWP3ME1Ess9xecBfkkUhp5Ja0pfZ3Ao1r0IokNj5cL9He0i+ZVwm4pdNz
mGiuvF2Cm0G/iX1vYqHeKXVWN4FkRCVDFUpVWRhAVqYyZSXQbQCKt8SvKaxAdMPAg1vAb4gjd6lHseUp3Jgje6
FFJ40v3cBh0sRQB2dbRjUmuW1hPf0G8Eg9vbPZdt9C4N3wCom3xFM ZyfeiV33CBm5Kcu+fhtVtvgCWdfbxGwv
D0yIZI5UNHv2BQFvXNyo200M4+MLXogMwqqm7GMTukq14yzh9jGvWhY7mMMUF1F0AXUPBNCjif3Wh/+F4W/us5
2DWknxVtZbeeyuxJ3CWi0u8awokpwKmBpbs0KZzVmeZn6rDTHIXgiQqqqnXpIXaNCxDa0AnxonOsbahMhZWGz
yxS8GX7DLt6F3IEyG5pY1XifhmlwtawVYfqUyxu06po23c77I8RFsSCIfe70iccEK4wk6Wtd6GWijJGGnuoP/T
rJzheazeEaSwH5q51Hudzj1rW3v4xgE7dHYbQ+r6td60D30I07dKZ2JWc3/IZeWwlJoiyJun5SONp7q3m/Moel
TWuxw6d9JGs8Am0fjT8aK4K13oS1ckrrbSD50+c0CTJbi4Tt61LJR0+ikq96pwMkd1fM/bhyavWPLTD6bLP0B
QydvMnrYJcF58o9+6HunNywBU7dUA7Zem8dmIuye1rT80H35a0vMBbDTvdk28GPK9vAW1uwqpDygL1TnSH0KFK
yq8FNT+a1Qq7SF7R201MzpQFs5oeBo3RbvbPXgff1+CH/UnMvfzWXp1wFstyp9G/6V633kfg6a7Vr1KMPzdvb5
4xArnmdExrqKmickFAMDz7BZbE7JN+G6VHgdzwFRwNU3sCmQjdpB/5GRUD5154nS9eglurn8WTUuGrENvsN
BbvWBIR2tLWCQT0DX1mbjuVvRKndjHbVTFF5bPRj1YsHInxWo2B8p0eswhXwhAAy+0A/RDaCi1V/EL1M5Dqhv
sMN9P+NT9ZfbIcZnXt821XGivpR+oXAzzq/vM3IVORDdYhGwSpVPIbwjKyc+17kc1SJv3Wo/rdLPj4BZ7Xbt
krYYzPH1dBxAVHXX3IOHHmoCKjKFhgJdeUBmBNIsy7dBpt50rs8d2BZA0wpjdXUiWf7FvthI+DACxkPF3NX7K5
9g0oCg01m1+nELLyT9m5FCvxd3x8y0MNT1V5qm0826L4vgVRZJLWkZgvt9tbHcT3s4wp7Riz/kgqgx7ENgx
F5VjHiS8RQVe1B/HzP0aDX+9rV1RuggPGgomYk8g5a6w1CSw1f9X0pkUNrUZw0k5V05kWYwSJ0aGSC4026YtQ0
+TSU4wGFp430CIC+/9AsenHN2zsbh0Ma1JKXmtw0qfoG216AiAWlyNILZv4SJ1zFXV8Hi0Wbf2HVpKK/Sd20ki
7ftP6xcI6dua8AL9Db49drC3xWFhtXshhNd7xuDADIhbvPSJ0x6LJBCc1DTxu5+jFByDaP08G5114LZzx1kLD1
FQ8HtmRrzCiszj5eSIECC3fr+BpGpK418n53xdg7Jho9CNbqzUj/4EtCTtx/xyeRtJXGHRpsyETI065B5VtWH8
8Vm5zeSe+Jxi6fEgdfwKNmvuMUGdBIWaQ5Gh02Ha6SX2+C+BQWuhZsdyFa/HcsuLpDhUVg6BxEuxTjahMDODS
4PrFZtSXmJ6qYjheIaPxhYYZczRLGqfIA9a9ap1R+mB8aSbzNFK4RCMbWJ6hKFxuf1JgAgJMzgK91p2s0aV1Jb
iPdh39v5ZoaWH9sMBAqdh09rEZmPQhB6r+9H6y4pQbmvS01zzJdfNk9XzHq8k1/AC1KDyupUIPiYa8Jt0+0YnS
rJnxGXmeBHLke0IkpEjhq2rc3WmecujjkcthRs4PdHuoIUC0tjAua3iN+DgwtPfxnj04p49Npuyidpbka0zyUr
fXX4BWp0IL9PpWSOVTBmU1+Eh60ntrYjvz09mBEP20vwqZeT3oZDRfq0gM5tPfeIUiVSSoLuH5es1pk3bHSr49
V1015ahmrFQx0J129mk903b1XoQXW+/6iIiUV917zrpuCE/vkL7ox4KnnTPvXUOVqQdXhs8rbw19tpGL8+w6
B2gAh0912dEtBsBhvURONpNwcAd3/6KzV1vu8t01UdhPQE8tRSdkZb30Sn0jghtedUmIeEVeq5ozSvvqy+d4/B
fQy4x+sg1aJDB8g+Ka/z185P+fDupb39YEC8bQ1A6TKDBLAPYhvznMpJPujA1/ja9/TkQELoJSq7eSx8wwAZfw
/aHRaS2S/Lf/mn/5uu17F/wWEeCz00ak1cFL+Yt8FcE018HtU9ThZ4UqHqtSPi9SanjVdS2gFeAtIwNPMpGs0s
1T+J42Fzqny119zGffwa6HyuCeZTBsk/GHAMT60MvC9h0ho2kryQZ7XANAA1yla4pZS1lkmp+y150HscqSdq0
7fgA0t+NRqCi049i1s90650ofPiHQGfsJAEOy5oZWAF2wokhTdrzSTJ1Dn5KiE70zUzfiip0vwi8WCgr057e+
30Xx0Ef/kJJ8StgR5clxXAU+z69EDH0kBImr2sUm6T2I9PzYtKhwZRF5WFx+svlQ44qcUwOSEasvKBmce/oUaG
SrQpXMSznW2cwJywBreuvAQ1U6wwLutg0JZQHicYpeQhUycI0KIS2Yw1GMdnQReXyAWW9JgnRX6wV7t3rnD/Ga
kn3KzP1Yo6ZdIrByFLEDGppj5MLHIC+MbWUp505crB7eSg3Kr7IUFuCTvyKKVb2wLgzq1i1hPYDgLwpauJBGs
+2kCjTX587ADPIWK6Y6nue0G4L/wHWW6qVvh15MqNNAunmY8qHWRDyBiAqr15IGtqK92J11JprzYgd0gz90r4
5Yny/hbTKvSw5kHckAXQ0u1mtYPQJZ9R1uW8aXf1fGMk3twKZubmKqUNCIYsfbVexTz0BvaIF1NwuIKhQiwVx
5t1GvP6Jt0zdLoeS9juJzqYb1HQ0nRR7CRrciUDKvq9nWCeCxqpwuYB0JDLG7sC62HBDXMZ66J0uSmUsvtx/
wZozrayCSmEYBDGhr8ixjRPKCHAela9TnEB2F+0HbKx21yKk1TnobR1Cq1VTuopherz6qUsabFLdyZ9w4iEF1
2qRH8uGjr4DI5MtU5Z3fy0mgz1Y9xvds06fcpaVbTKwsHWKrqJ7aThWuKpAuL10w/1sBFmcJrUr0rdROIAuL+
YA1A9YVCK3oR/yPfXfpzGgKpBFH4GEWVYvvZdA/TrcxcgKMTe0uX6W4Bpj8ygzaFQNB60JPnF7PKsu+AAIbL1C

Iaw1ZgL3ZNot0NEEkzRq/giGkci0vkdBGqrZSiyr/PZ/1A52c1CAqqEKbL+U94ds567t4pA9CZRCbmJ+44TdAmxzTfV+rzDtyaAygJDRoDa1CAPWxeAE6zvE9BTDjgZ+1yQXg4UMhRzPfXMZ+RKxo09iVM7RRivfmpT0Fr09fbGJvn2+IMc7zzES3iF/8BLt50agkqU69ww97CKSYnb69egbAg+/Q/ntXEHUiFeF6J1a03riq3+0c6Yi31Kp/CQgR1Byp0KU13LFFtJ10/CJ6MCubKM2G9vAGe5zervFlf9oCvxDGudGx6Wk/H4szgPDFq3N2otAYxpeAX4m2q6vydAS0sbI1TIL5UHf/AlbiQ+AIn+mRwTgIwg2e21uybbZ+7jg3L1gtVaffR1zinTt/h1419BoyjDfCGEj5kvD3glzEn4o8S+MvpzXTdxhDmRuw41Ty2RAEYNbAoH5tGLupoZJw/JyJ0DxnQ7mMIow+pLdqEa4fIxFzEy6+WkJ+j061H9mt1Dwe1Mi20jmrnZJiw7vdpxrxki6Q94nGLNQKD29S2rHR0b1vIaXevIFB6yj1SR3dvh5c7VcFiNHnQhDNMvJIWnCk9HZxZN5JDOA1RYCF8zcG8YhKMACem55UomEPqhfEySdXuUGydep8xCh/qvIPuU3BmxJB79RrA3q98KsAnu3qsoyV9RjgnqzNkskre/7djaR72hG309VREBEVM3rnJ7gsmH4r3vLwL+hvuhAdL504TNgv0vjpruDPxIPKRzXD1M5g6UZeeWCam3upgf3LVGMAh2tynWFnQGKqD5nR6AW8TulwxmK85VA5YmgzN36j+GsjNwN0tmzgtsFXDn02wJc0uv/kUu7YH3jF20H6V+7w13C1LEm7WzjRL1PBasBkpfUGqT/tQUIh74VNWj+ZKDU0Ub/LpoBpSXgEh9F1pp7fFXdU/8XJ+r8gmVAucsSU10UTWdJ4J+anJBjCHYv7ZJBINZGryaWtYaj4Iyd74yp62jiiHe8UWAh2458WP10kewJTUhX+/yiQBPomMJdAXn3brQEYv3ikYx1SZUWIMdCofzGDavDOEhCL029EWTHefrVwiNe0q1R17mpfb3oCRBPLHPebfof9SjmW/6jFGByyQUsTbkGSXoQutBV1/KrSyo1P8c1LJt986uZajVrUcyeif9TT+8rhFTDBD1xwi/RPO+4xGQZNCb1I4Q/C45j0JpvpVYGVntykDpaLsb7vRbiWii5SNUAD74LYxFEj6o+aIMKwbZPU5UsuUzTNWgI80Ds+eCV+BEgxcpXYcxyYb5wdAZ7Lrxj11IERkBtDcBxOsuiFnR52uh+1TBy6qzfmMnFuoqR2Sm9dBnq9c684PMhdCFZA+1StgKmIf1z5/KJNFF4rYcuTarKcF5y4Z1JFb10GSM+nIi67mxehdxqfORM3f2hRx0iMgXFdt66Nda bQAG3ZuWwkXYBG/0kit1glCvGcoFDr0v3IePzhsx1pTDLVupeCjBwcZrcfxKh41P8eCiKQmqUVvamcfLH/xgyFju7B8hJS1xRsOpRupANDeD9UcESAQgr8bmNic9VYssu0Yo9igVery85ux2/o3NwBF5NRL4iojbagpDvf0sX0yRBcYfTD0wIcNsZip0Ge4vuNaY0iKZz+1f/kPLDgrC80UzQkfguMnvPkWqjwqnnkUmeNoSvxk1/hVsSt1p01HomGQZDIQpnMV8b3MNJGMHD3AxNzZaKe/OqfAHh5krBjmlLucNPeqHOIH7gPKBHEYeGOHnBZbal/3r7rGS8h7+nHBjkpyfbCcB35E1FLMzJqB0G7mfERevDnGRErgUj6qH1WUAIF44+BJEshR4iC1VhCK+SazG8WKpcTemmxv2AweLj58v6sUE+C6aZ4takhWdjAkB2qBqWjo/0Ha087cz82mRn/8vA4CqpXuLqCgPvRotnb96sNS+IR2z8iz7bZkzjYpnE5YXP0QsoBsk/J9D1o22YNh9iAqpeieKjsyr9bzZdhDUNFrQgo9JnYhTAqxFK+SG9Yxot6LeMK6ySaGXabeKA39SE50E9scgxZzirMjwk0Nqauf0n90EdC/LpMEw5zbDpmU0+MtjGjFk5BTh2WNia0wjyVVUcbsyMXWdQY2u4fc1hh+g61xTrMA8qN7dWGSZmbr8D1Z0GRc02fqAhsc7Qu9mUy67NPsjXwmVRxPKoTFh5aFtDX6V7k/efZxKx0q1x/y7/fz4aEniuy/l1HSoVT+GX2HA4r1kcZ5jk310m2iuros49qLd3xx28314AiaqZYx5DFaw6sfKVjE1j3hmo pwNnzBcEYM+k6vnc+vNunezsveNxzuQhWgKxF7irapQL9nvzltvk5Cf5uTt0/jamkzCvhM7gxFo0bnozJmgPjezQpwrT1Up1yr74pW/tAXouNQhjrH+ONVBj+akaYjeeuL5A7Xc0/M9VuY88bmipASz1DgvaDH6goa2WHX2i6rMAmZ6kDnxjTxv69dEJEd7m7BEZgPWXM6CvyyWbcSiRsQbxINGA37C+ecfwu3Zn0K5sCBgtx8eumtmk5hV844LpJRpDmQmX83zUrmEaSpAPwTZk3DfsZV048Xdh8xWikyLbqCRD/TupfpaCBiVq2EtGc6h9WUMZvA07z/QNksxTbs/JUJrV8j1mAsyggUMThP2kzsInRPqg9Fulq+5o628k8hUFR+UVmBZA7zg9WVytXikmG1BJQlxehVLeT8HjmhxL7z0rruW8kQXXd9+2FFFa4uo7hUWJR/m+Cgezik1XkStJoxiS0q1ppv4udCba3phFH76XabyF3Yf2CbgkbCAZpZkSG7knIIxTkNXKv7fQUSScTEzVJ+IhnGXT2jAreX8ma6FpZop2uxM4/KHx0/dU0nWD1283Bsacb60D5QTX29200PXShKa44Wg1ZJYkaQeeR8YchTsXQNVAjLMZ0aZVzbY93F0Z8f0WUTQPCq5nf5U6V/p0tdFplJ7RMJw8nB/kElGBaj7I+cwy+DLoGmZKABUL3RKPDP2/zxIGxtE2fgK2GwUGkaCq9VMXctWnzEnELmjPEOpEkkC3F1po6Hzq65L7L0sM/CiJ+/16XP9M+Wv8juAjPIkFec0M0xGpwQbFKere3A4kYjDn1lwhszuRZMrjfMeXWTkKTnb/0Ge6giRC1Y7p3H+V1v04ZtbL0+5m7G7lxUgETXzCXI6rMVMWD5hgPaZhUHlwjrmB3VAIK/ut0061+HK1yMX5I01MhbrwBumJA1kZcqMLdAM/ZYPeq/f5infbcrg8WfxAwqvgZ2ZW6ZRV72kQLCp+AGrywN7egQTpBM1Qk0q1Am5D2XiPWcdZBM26S3becz2gf6nQ0isSeHEZZFYvJQqP027CP07AKZcE/ir10uzACePxpjirRAXAFKuHnvSf9mayfuKcmRMewR7J5S7VvSII1DX+0ZX/9axuBSRCTbT/4xtI8U62NOQFWIP7R1oYyyzeNVPkL0dwDvEZ3THTn8QrCcGBHn+MvcpRtEUxGmN3mUCBG5t6rJImMLG+c9au4jef0rB1XIhopV/mHwVLF1KW0gzJh9n5otvdNKh88kAftfbF93Z09nrk9Nj4pURCtUyIMHzsCh4ozcmSKi8EtMyVZPAvMzGh6uEGJ1bdr8fxxesITrTT1mNpD2LvdrrV5GRx9A3d1CIXKjR92KiChDqcobc4IwBUApthrVeAMNDCikvbu5aDg5goZReEnfAjvoMKXk71GVoSaQezFjIunzUPMPd7uU60mNAUI59LhX6WehMBDZS3uztgvb0F1J5JHo7RhW0tuzgHryfLWBOYbSrI106kKgFdmZH2Fw7CzHjkLEzYEqp7dmjViYRpmIMAy4v3fn7t1VdNdRVy1aXXYPHDfVPU/jAKgt/5RKe4po17Umy0jpGKu460CJarhAQxiFgj1CF0ygKjFnu1JXqLoacahU0pkYACZvfe3901Xp4pZ+f4ySzHkzLpNjfkTa1McW0ulfGx0rvIuqTuDaIpYgMSrQCHCzGRnFboXvGqmWMhag1Dd3x2KbIoR7b1z7WkiJ00uSwPi4AHC6Caev5y1Io9ddUda14LppaaEiXLM29LC1ZqyJMOtge+5NaCMNUVwQdXUIXMBdDnU4IdwEIr+Rq6xMU6Xhh+5MB0vYi4r3n54eKu4vTFsMdcm01CUG4Feg6HqkMeG1atn6xMgf0ur4yW5th7+5CNssesvpzULv2VdPrSoWyx9YiSYERPnjYu8wtIIamPK9j1N0EvA2M1I1hZVQVNWFnMLMN18Qr0/K9nULvZMM1AZm2LPJjNzGICGwVm5GT2ms1vGgop5G6e51snV7/4bNb0A3Z/Pf0RLNdCGRpKwmRrJsjQ2xYZS/YS1

tYuxBcS9/2Ym60QIVn30GAX1pMNgdlepDWkbbAka0Lz/D6qdRRQ4PfdtdnKNqSQ6S7c075dvoE2YtdZZ2ASy+T
SbSM9TvYBSjsQzng06td+cS/Pv+PniUVGZEBxCd01s3WRBiCfnXoK/xS82MJCte68aG0d7UvSBv3fVIfQP56m3
5yjj7KArYdJMHMapgVDG4fAbzRx0Bjatz06x++/Wc5pLj4r8rujbUKeTg7R1CFd89YNzT2XD/0Un3mBk5jv9
/1m1FaxrHT7PM8AJ4IXy80RZJjhfUzw6Z0ZXuS+u0X+8r8m4JNdYzSD5P/8zL05wmIRzPSKBAUSPkJEwGYXz9
sB/vIUsxxHJMvfBndOBxNPSF3uu5dg6xYzdd8VoLOQmv2HeNkACBXj1HzbBrwKdPeqpmZPM6Jap0xpeHZwPrxQ
fsSAK2bFQ161JEfOBzW1Jq/BdFLa/QxsLxzCE2Wur01fBt1AuV5A0WxDmokVRpJTo4LZ4Z7q76VCi60fbEe+g
31rhUvTJ+Crx56q01TaAQvRQ7dxT3Q/Q/h2wGwKtJsTIMk/Xox0G3PWEsNy2YeZ7auffFMA8IV3pE2JiPeez88c
g6kqS9VSSW6nyg0PsIYh9pV8vAZ1KY0THORwb0FNLL+mrhM03mXjMVAW00aMuT3Z/qcsFMJ04xNwztnrZTLUv/
w1KqKYT7zMccXRGLEL2dyTygogeXUGstwsM4IJUBmW0iu2i8/LfgKRKOB1v3JoW/QUiMQX+Ci05yevTvc1A6mn
PPNvhPH7oC+K95h0LHzmS8YSs56R0bmX10R2K2IdYmWzWc/0B8tDFf5iVAzbY0tptRRtFJ2/rLMVgsfaDT3C9W
ifMuGVGQ4A6HPqhBB1bo2tKDOX19s6Yz92fVGFeaN6+7Fqep/CxZY25H50Il+bGP/w4qaH/tGv6Ip/+rr0BpyF
CHL30ZpENzZnbnrrWtTppU9hEid6S+T091hpHX10oNX4ykSE07zRD+4o4DHHaSj1oZVOWPwEBKPCdEb/ar0sjRf
e21/9ca0Q7ktCu6gfsjGrgKhhgyjVARDhffVa8IEQTN7yfcmCBcKH+90ZNx/ffb1C0oUhpBIzTLjvEmpq5FoP
wpEUJ1+uDYqKF/+MP9N0Me8zfGvxXYCdIYX1C16Q/N47IEKSQm7VnZLjnS1c6rpQRGm/aiBTC1MU0gQ0iTZ2RD
FTFQFuVCPSHtTQC293a0vJ23t4/cMp39/VYq1w29tnjIEBogZKris6a1Es1FHPMcc90AzPghrSEmprbbSF8a61
c5e4xms0iVRuu0w3P3oSEWsVRkL82JD8mjbjaaH0Mw47Duv17PXEsrD2YC6wkPnMLDZXzTAtg/QaAA1b1cB2DL
on+7Y0fmlf0YjfBcVJqsRx1xNLfUL5UUTnLsShTU+myxBcVixgQIYZzzcvN0uL+8XMqE0398hAeci84xp051cZ
JvqtFgI4V5x+CM2cPrcP49bu3495e1uxuWYd2atPV0n9NUNV5Lr9ErsX66oteXSF0voccgEjgZQDtnLZ3fyq/E
YtYsg0VfD03tJSKaR97vwe/f2+bQWidQS1s23h8F+/XopWZHIYGJ/3aXD9Nss8ss1WZRbmFevU6mpuE+hgX34N
0HtF1+Ch690ST8/1KFzueF1vqU7j1k0cMTi+eRCAoE2GB0s4jMvj0he+Taa6No0dVLWsdthZcKojoPMuNvrR0
9kcadBHRPFZagP9nBy6tmBdRosP391N8/wB4irFnv8SIeg0bqGhPCuLNvf6Pyrsch1lrg0PYpi6kyXf9gsTpf
jpI+LBgV3YX10C5QpAVXWEMhS1rtLbW5igo3VqvTxnvuM1twkoTwn/7K020JMgNmZe8x/VXnRZF83tzm5K5mX+
mfBF1xTuMCKI/cj+VrZ1fZY5xXuvM5Nv+X/uBzY4VR1D6AspVo+5mU5Hg/gNtloh+Mr6UmuFzeVn7095cfXZG
FnDSn4ahN2dxUA7pD5AX7M7YLMMRoMACsNQP5GSG13gd3bJyYw3inm2IbU6y0WRKbgDtnloCQ/vqLgwxGLKL4S
zSKiFDDoJiemYup9r6LOJF15fp2QGw2DrPSqo/M52s2Ri0VFwIKk10G7hEwHdHgpWbo3njka97o8TzxC5DGwcm
ECRx+s3DDFGbsn+P1VqUesfD18t5dcxsxRWc6Y+4m88wZ8Rx4oN9eeSVgHgzk46e89VVFIPFusTkrE+vV43UNgt
It/mQ9+6apnb4+Ea0fnxxAuTgncTVI/uF3fZq3wha9mR66mKL1xTxuy2NdqFN60/5H11WwSkD5+xF/PtHhgAxt
iWr0zw15hzcV6m8MSQmIBVYU9f7EAzGgOsNATe0UNz5U2n1yaRum0RoKg100J25KpH3V6brkg5+85DReNmtM8
b0TsNg9YuQrEo2E1lkArugSV6w0AaEVd9BAuGtTMj0yTdGZTVq1V+Bx8DCuYTt4K6vX84NkDOC+/rWXYHFWrPO
IBYecLgsPHbKg8ijs48ycHoXeVV/T907nYJXxfHU/km7Sg+t95JUPkF5kz6A+DrJhoQbuQ8sTakS0jZd+s3P/0
wvG9K8Joa2s01z4Qj3mrVfIpU0m58fZ+jyeFmcvkG/+zYwaJ0rzxGoljxbP10GE5oaXLdBpeyvo1dk9U4IcPQ9
Wp6XsLDQ0EstdZEaotFc8STcgLW0nn0UmsvV5aekd1uYVs7XPm1cqq0V/uONK1QsmwzKvHj5ZQ5FptRUHoWRn
2TgyI6UhJNCuDyUDZALFcy1QbYyzK+kYgj0wsfC+cnpS64M14TWHIVEx8b44s2LXJK/KhRxEH+0Db/tHE91T
m2sXmpv11gQh9juSUKLqWDHcjdKQ/Mi7GFJatrCrw19k7QtUbv77FtAtI6ozBm38sEhDeLVz6pIbp90qG1+GAH
YQeW2zRQFYhWG1ZLCdgSCP4wbFZ7gdUUiEm7Dzppule0M1qZABQ6De4yvx5Gab5fHCEvVmnuxaW1HqJDPDvyA9
yqoq2UFbvE29P0FlguXAR21JJFd4n2mM7k0X6QYTztaeV/mMo4DcdRpjFNxyIXHL0ECUueY36pI+i9n1lVMyg
U2BjX5hnqr+K9V0Ur0rpmB9Kp6kA31cC2ze236nAS9I4Tu0rakI3CEae7iTeVX/Us37lax0poMsvLYCu5QmHN
cNe4CfvAC4mYy8IT/nN109ByN9aNRwFi6jya3iyaa+saGgDXcFILTR20+Nfss7q9/44inmp5xvEgf04DeN/J7z
Hr06jkBpz38WHRkYzvqSTNOB2P3VkDaFacjw78jP5fQ6MLYCvK4KesgKqI8R2nEntGX1mrXuV+yHfdyaAtAaL
1+GZGh3msp5Et21MLcPm6VmtXQ4kxRcQ0Ch0XhsC8KUqsus4PBSC4ijwiIiNiVLubd65MDZvhn92p6Szecojhd
cwrvZYCahwWzIacHbv0fyEBNS58qSvQHFQ/UAPXJiybqmI1Q0+0i1N4cfTMbljWrkxqfNqcDeQQ0MaraB+07Qm
fh+Lc8amcMWlwh+U6BpAFhfxdpzuoAXYJIAFaz61UpP5AeYFgjQmjZpfjN+3901qbb/dLaUePYraxf0Bk+6BqT
ULwBK3rkT5QCQA0o39rTTAC7tvBqTc6Hm/EmYI+Dz1ne8vJUc4N4SmtaAAJd7oThc1JPK840e7chiQ2UUj0ykL
FYQ+eqkFTRpCGZSW0mc6CersZhuGYRMlaTzb14oTSuVQQjjq0m6ecto8FOTIhA89p5Hib4S7Eh6T3y/5Clvmh1
h4Xq10kU2DbPtHpxKgbb7MkAWBxhydSa0j3Lg4A6gdy9kS9I7r/0bUpQ/ET/VpSdopvUBE+RIyzk/M9A3YuE
a07vyjw9Shs20g56QnMWvsTEEIP7grDFry81EsNvMe50wTusIk28j19cBKmHHUHRPCLA8v50KgMITdgmL8MFNE
oxtXnAkAV15khGtNhrjG1rgIJdY91m0DAP51kSXacue0qzGTJDSpYkxTHReFjUYcXThy0izJ0/XTTLRodV9+4
U6/MmdWgtlBUcsY90gtYIT0oxLKhJCbWkzfoIigDS0ZKQjV3pYo90zsLEL8vTwHmc1Dzqjn+zeF+TSYIAduY0
oH8fj4ByJcdwkDjSb2LKqkJn0VnRj8Ry+0jsTUyUA0SPWDZahDQcHSiHUYdpBsLry0rYyzGU6y15moQ0srPHcW
ZK6LfdJ45gvAtqvwnd33yKrWnH54xZfVin9+1DC3K7Ujh1xtb14u85bRHZBA1mH6cVRrmsj6N0px3FzJNz5ZQi
b0a96ztW0jf5XboD6u5nBY9rwANSzUeUYALBsj9Zo4IHBid81nTmo/ZLz9SoX5j2zTpLfkGMXanGwMrE1lVxSa

++nOV/k0rBmJWEAsPSLeSxrz2jBERYiocwqFGi3kUsFeBFQEWzYlHTAv4kSBXi h+m3oKCliw8sof+LngKvFpj uGUvRPUqEpAS+Zm0RcWCd+0xFMraIc1GXXm2YwWQDimW9tRt1WNx6JDSoRH1jYc1V3HINVb1qjthRfoD516k+s 4XZ9bt+CeR9E7HU9fmVTlQttPSLsfqUcLqLCMfi4YqIv2W2TzPHhgLjwnS07nm7VnFhg8ulxmpAKsI0kGOXn+R 38pphjsinASsH06Uf00w41Dj2HqQPP/+PoF4y5AjWB0b4psqFjaWqK73Zszhws0t4KywcqB5HbmP6nqBZFN5+Z +ft8MPM3gg91orUGMFLSIufrLb9tKzLzQhrgJnuzzXsBpnUeBjTyTiNxEzHadjjX68EkMkEgAfVxqLKKs+aKhI pFp9yEmw9DdMZVQnWAf13+R/YxFpEW0bjdeQIHC5ci2Ci51cFo0RhVULJG6xc8zsSXPPab+W6U4fhtkrIjt1em MyQAu0rivW31Qo0RAzMfrsmq1TxqdZAjX/tfv016Wp71DdJrNEXzdV4NPQJkmKwzhNpteuu3XUUQhWAmDaJ9d t1q5fxAVhCdkKtMpNc7w7jATbwXog1Q4dPuCy41I2MLpsA3DmUIhCvt9aIixGiW025w0CDFShKwf0i/YoWpa+P /fu6CH1kNggRMqSsfPGwDbiI4jIEBIhWj1oU5XM+5AIbwmtudNi4vKc/2GZ5Ep0X1EmAJGfggpQk5VbyYoP4V aMZRq9KnhWVj9y+YmzGgv7Kf5PDv9/Zj08mCRjaMuh730TcX9AaEesTYYqvTWkdPTThxhDdSFVniKKGQgYohwv /A3ni5SAZm+pXGcLXb1jtPjG4v9C23UoD1Y3CM4K/1oop6WT8XxJcNWhsEvSm9d5/P0Wn1h0vZhIyYLtgfv1aQ efWZ1HvBsfv3E1h7VjmlzYHx0/Fb404ktRgTtz1R1CAXLuD7M0/sgb3s09kFk8CYVzv5x7cw8GjavnqN1fJ2i fyk4I0aFmRCPerdSgsip4IdmjZSQYcZB8d8C6g1c76qyUm5UQjf1NcZhmz1LuJQ0qPTuJBDpAfPyzscEjYYsHk VuMlv6lsBMJRWZI0fGaaqS+ASfsSRC7gRejICvMht6yyBcMkI2R321yGX1I6QztX5Elvy/cDjOPxqAugodocLN yod9mSW3rpun0dp01+Q2mePFMiU/pCcHzwXJHJugvXG08dqkWR93DU3YQm+3Ygy4S6tZX+9CUMRkQh+THiA/5C 33E5xVE4IcqHIBadALYgeZz41x+kNq578Dc16EG22Y62xDKw1sVXGn2znEQHSZMXQ1BkTUxxGCbWIFRjUs27EM 9JcjdSBzIvulz0UI9p9ycEipDDNqPPd2a60QMdVpcqUvTKAeNxVD72Y+qUpqT1E4Qxq0wx/JAIik0L5iZdEvcQ AAFb8/LvnD0ZI3fuI+gix32m8TzRejUMFV/zMTu0y+qzsnSxZSkubpSHpprQZHNN7x7Vh8QZmshqU+rK9tHzx +Swyc6PFs7qbzooZgbHD0oEbIu1p1WCJIKWsy076E1Ud+rsZ7qaSsONIhFw+3LoIcX+StilvtccUFynKVNHME hGweRQZ3dRjFHFBNzjmK8M/KCw1ecQwdUvapHX0U5Q1Jr10pZ0KIttDQy91Uorpa1BIs8auY26Xx+GsIbcyhzB 6ZFcMzrdllgUurGkn3ebFDHdW/wl+qdx6a6YBgc1omsBZS2s2DiKOX0bp6nyrV6Xh0pQore+gteSxbZGbFiWbS n180It8hpg4dR7jZfZelwKhs33/4TzRINv305wIFKMMcG8fG1Z8zU7y0Khed4Czpsg0A9HoWzJHOTJ0p7BQnBm S/dMV87eDMK+viPYUJJPe4p/LMug00BEmxax7jdr08fIDA7YWQsyRLcDf1PwMzVbXAX5mPwN2Zf3fqNrs1arL jkEa0ynyrHomQ501A7U/2pFYna1DcEAGKmUFYfrjTUKJ80GNcmxbVVAojI7cMPWWK+hkLkguc0omXQEeRwdz KxNpuwKPF0kvIxWoJ1f1bk/oFeU+mmrnSUxyZ2dsWA0q8FtYSJIiDjdqnh/1yQS787fvfNETU4tQyME5ihYq+e VWzfeNAL1UlzHi2JS1g2UbTvVddjVREn8wRSkRvNRW2Jjf44Ke61UzAecwwbXhI43dKcM31TporCekey/l+XTBh b88gVaB/NVPM0ysEdh8rNbgXcXxV1q8NWUy0a5oEYb0wR0fYgsBLf1Twdn8rc0qirkvm8TjqhVYY1P0mnvYGLP upMCwuoUxYGV3udB89126kgkDcC/PG3dWwRZJ1r4E87SGBX0BkuPrer7XLmLX4WldQ+gqEVpvq3/EKoHRVrh10 JNvRU6TJtKzQc7+3svDu05WRXhCtyaa0n68b5T3ek5t5AAUruleo5jqCxqbKMaS7tBC12QBDhB8WjtqoB6pwH5M M/Ab0LOuD88W/W1Q6ejCLV9NTqLEogMnFQcet4G4n14aso4hK5Mu/coquv8br1MCzFjCFRBfpckRpCcKAruie FZcwyuAqLhTqyX/+08gDoSiWxtY971EC05Ae8YgV7YVRmVT2CU5fbbvHCRWAj0MhymNPnvjXIjkW5+/j6X61L Kb8mSIT01cCSXyc0c3vPvPOTnaHRUsjj/VbNUyCyxpnBsBUJZ9EaAfSfxQq0Z5f4Na/r8NMUBpZfUtY6u+6ZmE d7/Qe2E8f1ZjxL2BPw4e34Nfia+3UIQVYALKfwBIa+G/25A9EtTQLYxyWxy2DNve5kY58Z4RQvDu/t5PPFDd6B Itk2qHYakvoQjhM2Jso04Jhv+4zdMHz7RhTDUe/EYw39kSIdPKSWkFACjkNDatTUo1EYoZpTSaCMczKuhn0cDa Ylom0UN92Aq940w5oSNyqDuVxran3uSVoYFeiDqDxam9jsF+0hhvLE+BmZBePxyWM3bY8WzfGxtGtZ1T1J97L0 3q0JM1zg9WGbvDCeMgLWZOEJcaNMN4rP0TQ7cJXon9+TBJNzWi3p9kjXzcuCaWJN2J7HTL0D6G7+HFjELI7m+1 X1cCahFB+1WB2v7aE3wFc24Vxv1527RX3CIk2zA6XrRVSmWzh9qkkZUqukxUV5V7Ce898CKmTE49exTGA5/Uw8 G8fojv0t2dZinLYuLv0vieu2x1cCi93FKQzik14904MXmoT9Sb53uC1Aja2CsXJVC36rQ9ymIPy35NQP8phgSr rs8gvcK9WMTut0/VEkjicgK6VWTUgrrZ/It8AZ/PEyB/w1Vqt2gHA3yU17M7VIZ0cWCib2JmnJwra1jAA10xGa ecYAJwAb9GYab+0YbknLc3E00rDqieU+ZL1fgCithioKy+G9XJoEtUCA6/Tt6BLPZmvYBI0rJsT0E41Xe11g7 TE56Xz/Bt2pSTFD0dvK6pT5x+HBAP1GSArpwQEWbjMYSaZwJspJpe91NDjpzJj078w7sUqPjwTum6eFQayJNVM IiPpCvoD6o2p/gP41KB2saXy1k1fSKWgL3eezYhju80erX8SGqR2ycEf520x75jsW9qzxpJoIDeCg0DdX/TzU U5MMngtTcMRUgFVpydwBsXRxICIFawxaI0cMxRm7WFoXqg+oR1/UWQg5nE1hrhBbTQf6rT+gZ1KEtCZmWmSGsx 8UsF2p0iF1Fzgrrwlbp/skjmByy63TqCraeHkHqInXmJUs4m7wJFjvZPfZhxBwzGukM2FVCx2vVwNrvzIT4nQP kYUCn/iXQ8N0dP6QGCZVmCVb4ay7xKoTnkY8rc+KSRZgE8kbyKRk+DTQAggeeW83Yd8ijJDYE2HEUAWZkgqA1z 2RnGZS1JmWBCXaLh9yu9moh/Cj1WwAfnYQbnQ1LK7UFu1q7voiPnuS2hMTMBLs4QjvZYeS/6ktD9y0C+NzTRVE /ZUPHD10fCJS2jnJ7gHkk7BnhA0KrrB3LJBpZJ0GJVgdNQkLLaZP5fR87a0yX63vTjX7mCK5nkSJ1n0yMVK4W pTKwv6DnXD43fqMkWnkoK5m0xy54maTRP1NHJdhndwM1m54uD/xcAhoDhf8hG0tnd8h6QgmB6fd5yVEyv6vBR DsCtH9B/xK/FhvR/dTWiTlzbU4DEN91i8eW2xVcCws4W+dcp0oWPF5bE0p3c3fPzDCAFjoJ7yzr9dEWCPnwmcu /Ssf+iY0s9Xba9XisBJL4TImzS9VPMmgV8V6TgoabRyKJ+S1EgNbHmvJa1fEq0VvBn1pK8edE8kdd5vTBsftG WURDIeyWe1rYtJVRm2bAUlvaikwHTU5Yv0DVIEdGdb8errwE1JpnJgNKURUMy1a2GRR947nQmj13qH5bbN/48F

t/CSRlViFu1wkATmLB1YJCpoZTYID0MeaDq1iJl36L/dBck7y5n86J8guXpBkocm7781hSkHuLQsSfSsdu5ErxBWWyt6PDLVM4LPd1lx7D005UTqXGCWNHwxNkdZY5JyPhNc8RFBYNoFhg1PS/Mr0Evvlzpqa9h/pKwf/2LFuy/r1hF9iJRKnYL2cXxxKZ07UDPvbrb0u2tym8Rf/Y3scC8xkXpKxMueaanXjF7hF6habzitk8SyAccEXa4QRkI+S h5NVxh9YobYjLn7joyTaiJ1DfkFqodiE1rwIFy1TLw3s1MOLf5keTUx9jDvdh0L5NPN6YeXkCDUibW3ZRIvmb px1mtfJZeax8RWs1wpDwbfbJ/wG+g0e0hMVGf19aBvDMfZ0r1jdy0e8taPszig7kmvsuv1XjMpra4N94jXmPkq 1dFIOJwpF2VA4XhP0xw2SRytWH0vuXhSdxuhmtL5TnLAQDWv/VpFTqj2BwAU83ZIq60Wvt66d0dfeG3K6NWk7a Rkk5aLiGe62fan4KLiy0BBigYErUXGp0/1nrX5k2JP+N04XxRmeNxcBph9VmVNS87wBRTsAij+etKLRs7+s7 /6w3+2j9aWFDYN444cBERroUBB0tE6HoHMaS0bajKbwm0buebuCmppv+8EhS6uCr0BrLGqEM5WBzR1Gezkn7A BqT908poGvme9m0B7VV5n7mEgVigjrg0fngMEpc5EP7Lw51XPyzdASY40gt5jX3MIJiyosECgHncUVgmnS6hcD tLQMzEIq0+i/thh/Ye+xfcMhmY0e2k4xZE10T5rBLn56oQWCqQFtbRJkZ7bPC6kNtaG1YprFFMNTVENx+maxpk 1vFKPND6Y8/7KoIkvcck6V0ecVcDYPHwu1Q9HMfaRZx9pBoPfQJGB41pVvArVW3ddGFvK9uhLPvcDZ/Iq8WNVBY 1meepnWglW1HPBs5d8zg/Fbli+1kTEZR0kP3JY6GFjMKkbW71CWAXJVRZ1w3XwvCM+DNCWYMeUV5sEsKDYy+c/x82h2P1JkzQXX8xKQVk5aif0+sc+fUEz7MwCZzzs0DWc29yQBKCxQKDzd0cZ03Qg/2gqzeGAI4Ps in+hMkRPim KM6TzWNC1QU9bvHBcT80rnwYhpHX65Q3cFQPR5JgS8aQyvZzzjY2XuPxBkGHBaEgvJDH9n1A6K1RzkzkqijBj Qhw8i+/XGRKL/STfs0qAr9SRW81ctn1KRYirLbk+8cYkoSxjLh00Ik4MgvJKiLDEuEuArFFLUDnFqjwA3lwujQ Ww7uSmkPCiKz27wi1H0614XdsW1SGGaFfMm7bj1zypjwKrCNDgNPaNwomTWewkYHy6aboNq5z4lsijgtoAxKC sAqfONi91xc3Bh514ds1qh3ICaBMyZMIB19tm6Xg84pNw/ogN7oYLD9A6371LqGSKzrZk3m3pTnu2AvzvZdYJF EPRIEZOGXT8xPXLLibd865ctnDtYEAHJ3ldiX9wZ7H30cWczchN9j0DAXtj/m4h0AwjBDjb/BWe2eqN7tc6nN r7+OR1ooRB4DrdiQ4mwy3DAH4h6FbgeI18eNNxj0elrzjs15eJ5E808Y5IiKjBhs47+XcnIjvrijPwf12i7k7b PE/N11I4EAG6Gb0+xB8UmK03Xohk8RjvzK9EhtjMaZX9dvh3/aUEWTsJk7LZZZ29V61W+206Z7MioCRsPqr0s GH1tzXG5Ih/n3oj7hk0oyMTuSsZ909/ywWSLFG5HNFPW9BWQB3LTtDV9eJttg2u/3bk81xPB8wpr+t/vKBumA9 9ZDBr+PZ5CI8UwczK/Rzf5IdFmpKhrJfKazpp1X/rjcBimsHiev1N2eSzqeL7gaXR31DVR/DgAXcrVy03n0coZ qEn+TrMMKcuyCq/hY7SsWorC9ExMzn6aBw3oSubHYb8cK+m701zPnntBtIi9KSD1EaLeEcA5v9amzf1Smz22FN XzMKTfxdatxynHiGXhWcvXNmSpr094iPsbIviGTswUAJUPWaA32ABo9DtahWWZ+8LerNIAdWN2ppE49BwTcj1y F9FQXysSN5d6B1q3nGedHR4KDRSrg6psghh5M7h1lJFz4AHQJA0j1Keyne4FJ+Hx0WmkM6pbelrqv0n1oocEy xCnh6XcLy5+0weVplrT9osaI3pa1UCHdo2nR6dVc+Yws3oKdw5w2oUsJCVQxS+ZkpNIhgcVot1IV4EbPacegyF 38i5rvyxLH1cr0ieIHAYre7EmieaKsHJoaAkqCsvdDS/gABapI9TXym7vNR11YEB14QKq9ACup03mR018Gp5of r3TuhAr5ev1kN0axdCrDy18jF8vV51EPq8cvb3xCpmVHeRDwcqsNPUUqSJSdWfChpx1Jar6jApz5rn1vrPrUP y96PNRSUeAr2U59Hhr94CMFq+wQp7Fi1HuPg4ZQ1dcVXUrUMG0WBD1CbKsCL/oaWeEIm+/DAefepMk+0sy04vg C81IKY+LR96uCOHrSSvpwdBPcx1RV82Tqoakq7DVyqJxZ1vWBmSa4mYR1hRkW3PhzT4jU0gNBe+lcnZK3MjkeU Wfd3spCdYDW2wdSBr7IZswn9cvhqapzJGQwuUo3nSRteVKEXYyo5+NT77Id6tKpFkOKjK/ETwzvFI0jxGgvG04 2VcM7i/XBycnZkWBx4Ewfbxj9mMIAofG4h/kgn21AfEsmi02NvdMnd4n9phLIQi31Eix2ML+hfYqZ+TBK6GX9y Q9TUEB2u0TsJsE//mczfGVLI0xdXosv++LjiyTSnMstKs6g4Sj2VNPBZUmBw7vIPZsPaArTUPw8TQuYfQxF2XB eVhptFf+E8D04SWr1TL+cabmuHFuPwkWK/hLRBdzsf0Bd0a2ARK/DE6vf9j1qxB3qsyavQ9zIcqMt0wlqtIO N/mZXcgp40PtCp9q01SRo/ZAa2KYeZd145bvDXojd4zshVWGd6d8AI80rvhmHuGzn7T0xSxkHIm1Z7T9+iCqX HJ1zow4G7b1AdwTkpLxpTgHgn15sfK0+NPXWG1Lia229J9kEtk9cEVV2ysiCqFnDHyFT9i0Gq78pDxFsXJ9dNo vU0m1+7AYaTx5fSSDw40J1Qta1/sg3YEF28W38jAfeRgiwwURd6/SjW1HZhalufJV0c6B7wK1smN0zbRbcYLxF Is3yN78r2kwxMRdLSfdsFg2VPiU7LbU7opbjLaJoSuX3iet30r0z7abh/T2sE0p9hvYC1vKrThT1zsd34g1F31 RqDN1oZiiJsLWHRHp9Ua2jbInupE9Nu5wqSud0Ebmr1ISw1ZROUGuelPozW8+YPLLQ+PjxuUaM0z1dkELN0cUC u0UX+MmRWhvqE/UP1wM9068soqWpgmjxUdFTbzup6mz+ApIsYW+2A2DI2zsMeGkrb+q2zDu6305zpIEvjp85ca hmLQZWGYdj1LbZNp/CwDNW9djmMs0a+25PPhozEkUb1K9/b37Tin1TLB2EZUwBm0mR61/gIVYQSq4WH439ecPG Y3UKaQozIacmq04MKdVAfvGzNtu+2r294/50AnFts1mV8Ky0eivugLWf125s9B8m5AbsZyeKG52C70HFuLhh6R rmVIOXVYR7cvQnm9pH8hz1PUSU4dFnH+qc8nRbx0hpafWnwnbVbJ9Mn1nG0uoNYIttK4hM0413SAG4wyL3nF6 ViIhg2XE44Wv62UEUoUOLsBULZKec0uBZqTc3ZB25pDcJI60ps537adg8N8FzQ3y80pAs3CvL06FoVYe2hKQrr j1UdFXjc/H0vJEk6KjJ1s7PAa9B/MdaQ0RVZK0ty1ZedTaJpgBQk+mt9AzMuM5nJ8xExTEW06gaLtv5SZe/3v Vp+1/u+aJIUtsyVzXPNos5QnQMosan1fMcLvIErfAUMkvba5y9gh/H3BhYty3e4NMAjjHFg1tSkt4BBHX1I/Bm /Cya12DjPA4R2o01xGHIHEw7hEIGM/eVzpqD+gbZC2VdgZiGj84Lr8EX3gcLuJ/oKmqmlhg+G5WxbzrzuuK D1fUeVWVJUEsv6adwLRbn3cfocsi49Wu1jqsXbpvfm0g8VCNJhHp/Agl1KI1AUkyLjWsl/9aEUWtGmPkzUx3Wk afcigiJ0zzxUOC/uX1BEb68mjuem6r2vZ3KY7Bco+iwhYzEXZ1r1TI52XbhAcq7Kv+W/whAXm0qwTU1ss040p3 c7FEjdhpP+njCSjaykegWMHq5RjjTt0TVnt6quJtwRqrCwdmxgHhM45oUaXr3f8D10iYalPAZln9UDSH3XGq6 90ItSTYtiuwbtmWh607r8hvz6WSGZfv4HUHD9TZ0qvww5TW0Ev8P7ME2yuxtjtpC2P18RrPG98ZY0RGK/15gy1

0pTa8eanSWEenj6A7NpfATZcEJKYUpqppN9oQWa1rSb0meosHnGrWK8s82g2ZI6wk9TLcGjYijQo1zfmNkxi31stRASTgjTMFzDPWdIH+Mq6hP9b/Hik/7dUcDZYGFxNs8aRqa80vH9ZP0mYCaykEo13RSSipFMH7H0n8AnmLt8n7eb2MYzp4e9nZLB9wcbJ3Haf7pQzTXgaXShQ3hm+PuJpA0Se3pvBTyI8fMzzxQuApyiNaBRX1eExnq3ziNjbLyLjdvEPbdbz92vjoWCauD1ITHo4YZ6u8CA13t8w/gP08AexNQ9ePkTcewRCJ1cqS90MA59XayiJffgU0f1ZsLwdrYytLQWxpPfKB7rFdxnZfJv7Scc4VA1EVyKy0/EjD66NEo0PWXDTzU5Gq80n68ydtu125na1DeC+oQEMC+u0zVgYwizquoq0tL0vFYva5Y0dmbjmip7j1AmetnV26KdYtuuYWxN6Rw++vLQJpL7inzD7P1EHijVYg+WWXbCKXwDBAdPLT5jc30VRZUnS/Phka6839bn3A4ckdWsHnURI8vF1F1bw32AyNRYcCwtwoRFcUXHbo6vyVIXnJNEnWLXab9/jBmP/+BuBYVZHbpZk1cNRSFiyKSTEaeqo+kxGQriaCz13Mb8Yoiuc4dkeUN20dVj1pWCc/c1KxGUeMS1KTelpIrdcKy61r2pvQcUXpMc2Bwzp3vY02tD1I9WizGF1D53736NNTQPH0/d9Qe6rwnkU/H6fTgAVD1UkvNJDKUF7w7mmdj2qu2y13qbwmBp6E7n4Tzi0T10chD0K4630+AyWhkGND2U0ZieTbPYHdwJBW14/z/HSyYmk2E10g0EG7RIdbSNxHohXAozp07FGR/aKxqh6H8PeVz+ueTcdVVzc3XBeckPpX+k1ehMtga7+S6wBZ2H0W92p7eNisBtMCrrvqC1gypkcr5Nb66fmL7hJskPe/9iSe3cGXhFXHwDjC1RiIVA+1XivGpfoc4tvv4WnwbFPQ1vDF1zcULjK/Z5i3VGI//SMqVCfjMGG/qtzlpMhDE0XiMF33h0Ygf1ZtMYHvtHQEGashAIcKCFbLs2J0G9K9EudNj4ISYuVPoTY8MGh+mZSfkVcVeh586bHih970fvtmGMVq35v5Rhw9sknRgo9a100a3f1I60XSVwKy+DSirz1v1yCor8fuSQt8AkPExmAJGJpPDkAJEt4/DsiEZ10xTs+bTJCr7I0ry7n14NgzDi7wAhW5dDF9T7Fz1SRzj3jUIrg/SofGqCC/sWhxv/+u005zhGxmEokS6T3E71gVXodAkJWcV7y5+n0g+mquG8UHeW07fndVWGoSdqwdgbzBEeJ40kcFT16XFkcVqBh/1kUqtvR4wttxPYHHrJJB4BWdfqbTKCF8/Rx/K/hY+1kw0ucbk4r0d0brPAcESNmCQHyLsoHwfNSS7q1yZffknujolAxkwLLX2DDUBxEYPLZ0VAisRrz9iQijFdIBYZAMOPBCLfo+HuMg70C4mX3haXr074bEb7Ecn4VRqPQrE0g5BbqfkK1MlmrfyYC8ULRT+9IYEX0KescrMEMmBr48iYTdEAOWWEIVgi0iMW45krWiUY+91jS8CskVaLAua+KkByIBjvIz0ieTkqMIkgTYBdo7Efx5K1xTWVTMzK5Y40QZ5nBubTAIabR2c3TV878tJPmkDba3SpTdpdIyNUEDXUoxVZfyIynr98IXqgD7UFiNagR4KHBPUubgdc18NBox1mQnkdnyNtaxIyZASrJedwptpUGC7ALb/a1xtxLB0S6b2eCwIqNPhwfxP6VIy9EUuVERYLj1s9mpyTX9mQp64bMNkANKiZXLgHm7Vtu0uQlql5CFHRaCCJYT4xkYZzx+xAz5rFCrE4hklbwFdSWpXcqsUQK8cankHw0umNecc0Go0wlnncB+CfTmm6Lp0tM0q9Gw1Yu196EK2nFZw2B6CiwhqRFjksydUJ5Dz4dn0cJPunFE5b1EEpplwG+E7J0rc8BjfhV75BsMMdZchez+gvrQ440P08hSsXT/kSiX6XrfZHIGLRelfr1BFJn51WMkIgLYueoDPTSCeJFvfd8xam9E8K80bPI6yzSrnf41zrjU9U17a7Gn+04F+Yn1HyZMlgk1HBSzzToNhCLp+4N2R9MAUX5dSSmSWYDjIgM8ckGLgPdN14V1Db/lnfISPGEJU96WsjUVgoAP/DW6+8zKW/PenQmsQbTfckHWthP6Wgkx1LuM5K9kmd24cLha3dgnmUTWA7+bj6Hy4Qo+ki+L3isTDBxEM2sDIHw1G5f9P+ixCNvuVBCh+7YAi01DUJDS1Y8ts0yA/uHmqfjwQRG7T47oBSwWnH94E6GNsGM6gS4MmFa0BD7RguE7duUj2XnbB4i7pJlwdMcZ1DSox21822rF5zr5cH6mRzaxB/Pi8jhaPsLT1Yx5SKKTxB3UPK04LorJch2PPnqkV1F7ff9oQ0z9IvgVNF/D14jHKnGvi1KhHFbhI83cf11V0Z5VDn2HAbNHQjv926+WvL056M4op0pJoa1Wfu5IR9d6DcF0Mt0oC1v+vFALZ1d6GxQzEVT3LcBuWCntGvMOG0Ak9kes6WuqqR/aKMgv2zLBIinG4Iy+Kfw/6APiMzUQrooI616dH94pICGoNQ5JGbtmj26TDWJ3agwVmUqppSTtXG8omoDEMqfayz1Vhr8e1gEWd9umn8oXJzcymmLqLLR6zaYCtLWVEBz/+CK7tLxf+jw9ZmZaNS0tTWhg+mE1mvmpemBcaDjSvzcu+FPN1HQ1THpA6Ry1hpuK7M/DJBKh18T0sT3Kc9Sck15CZJ6pR+4o9AZDUjuAXDkOZ/mpw7R8C+z5pnp7qfVgN9PiPDChvwGuxrgueUUAQbY/hiEscWkE3ugIHZ8gCJeD3W0H3RPbAhyUAtHD0SbhgZ5jY+vejUgu0Jc4gZ/2DAAA9NzaOKVq56fzNE1wpC8Sg456+2KTgCwrx0s3vkRya/iz7LCt2YIKi66Z+L0ER2xqmFRiFFKevGE4uEMLSuN0ifJSugGEhv2zKFuCnEoJ1wJXpVoXf5BKBbwMmp7q6up2uhS1lylsMSkj6v4V6VgR5G/01++kNDwy4t6bTeNk96euWf8A5RQt09t31LwLwmFccqfw3kJf5xBVUXaby75DPqM9ZQj7/7j+/1ifU0VS4j31Z0G1Qmv+44hnV1AL0JjQ/EocnHcWk7s/dp6HDp62KWH2XnZM1s56vzFEYVqb2U5p7tsyAz6PIGZfnU1EBpeTyMSyMc+sdnTcJPDA3ieHBqfUFeEaeqrrsPxx++sA3xxcd1CbMKRg4EhSDZFu9KBZYnek9BFiPjobkEaqR/3WFJkv/8D9AxCM6TRshjN0tFFo4uYm/QGyL8088GPxjje7pRVYigBmxvzAXcNRjZP0QcZaBB1Roz8Y7uFyuAitk4Z7qLjpktdD70BQ9afyHDnUG21QNAWrAy6zZvD1+Rv6nkyZKzA1bKTcjp/+YJPh10Z0SPhxSD26+Cxstj7F/UoQ8+mvnVWFGoSp9F22Gsc1RfGtMTwZCFC2b1eQKjtVz1n+5xTjJ16V+DydmRaQ/q3tVW3jSNnZsFapqzNFJ6YshPjz1fPZrfrTuk8eY3X0AsqqeT1A+AR0q6K1PKdv9RdnS84PzSaTmdMVCazp1qlb2mk7A6+ZnpHmuCuWitILz6KW0sHF5J5WJT6VP1B6Gk6F0xQGmG/ZkLMZLg5XKh2S9bTo7ZRe7JZJcdhHsfjVQf1Mvx0zb1j87BwHSqB7qE/x2nk1RUzyTshi3vPa921SEvFXnZ3E5Ib4tTddW8reYuqAZvCgfW2k51QmBHGGqbY51x/FV30L86MPssiJmIj68fJURvNSz18nMYU73i4mF5GCUc32SzhjFv98bTbwvWkRggfMgJevdZDjsa0CxakEEb7Ajjt4eIX2c6hKmetly0/l1sgkWxtq/JmSnjoK22Qm9j0Dpzfz/NwmYpA89a4msf1fM3qDBfsib4A8cRGHWSRDr3lns1dXcAnRJYwM9d6YJD+QX1dH3SD1IY3TMTWscDcedYR10IcnrL1rk+XQXe6LWGHW+q1xqfWwuQGI+ZSoRiZhpoUSiRbrcRFkN6XgNnjQXbYKZ2JUbp1yZ71rDdAZ96ZBXM7fSnt1cEAmA6+BDDVkfJyh90mCT/ddc5SHfWzbUmXjXt59X5x7XUgnqGcrfrGJntcrgyyLptK26GYP1Fep/zXuG0D6T2zJZpemA6Sry/i1rjrxq59aTbU1L0WvH3s3d+GX5cPVdnRK/r6BR9WZV3Eb1zzndgprqRnB8qB23cGNyxHk3W7p8IFpP6TdtbtHxsuy

ptH0g6UTVa0bqZ5ZkRAEpjYTy7q2cs1gvgKsuFmXtyrQZUle17HG+4XmywrgV8eE+q+U3CqZa51GK8L+tL21czL
kmaZSeJ19KKielzYfsAp0gg13TCCfQKUITHg9196x/pMFnMP6qsE3FbTGSmlGtLpGB1q6U6kuhHr019VmWD25fG
BoDpL/rUxjX+hjTfV6eymo2NytrFV5+eo2J711LfN3TsWDnh5nHT6h/uDKkZZ2ryF4r7x0Byx8VmQXt+v9ZY5+
BMTNP3CDGEoyoK7sgoIHu9CkXeuRAQ035ubIF8TMQ/D8B1pW/6zNGQXnmbfaMPeWY6gd+anpr1I89usFBGB3oj
PIuP+kQIdhH8f7gfdaktUYPwTjm0zqfxkyao0Jlrb40meiuRDY4NVA0di1WxWhk6UCyr0k/BiyJ11JnUaXSwb
Bp1LmcvxHonY5+J0HyYTA2SKeAJoxQxORGYmV04c66u7esKYjnBl+1nW/LMzxk1P9L1EnY9Qx5mer2qyb2nU
GGj810z0cLN6D7RFsdM11r3sW05vxhaFPgpIL16EpE07vIk/Kv5I6ezPdeqsPuX6fa6nDM087ZqM/1HvCNW2X
8d5Y1ZjIMjXJQBCt9bvAP4zBK1QQSdmB7ZJGJMeYjysEDaEQDHjQYjyVaCFNdFJbbLvr0TrmvFXWVUWbbZ6pvF
AZI2BH/A8VMAV+305P34Bb0X4Zxq26qSDkzq/FYoKSIswxa9C68Z5SHCBUuWv81aiDFh9IHkRoYBFFhva8vle0
NGd0u0rWHWN6g8Hd84zUnHOLzuN3hgTEgWWYRXNKA91jHf/Po2qRiQMQtD58zxE1cGSEp7RSrIqdcUujxHkfU
sHqYNhduEETU11lIz4au0s0AJFmcehkIdD7x011WixJ4RK2s+N2GC/KXa5EKUDUadLg/bD7pWumIK5i6qW+z/w
aA6j0mdlWNBPIfCibiLxYJj050Qha2uzi2+9TfGFckWrgssijWKjGFZh6ugYM1DhVvycuf74/l33Fmi8v+IrRTq
PueqjB0dkXnwG6rk5d9gFLy5e5jY8rcGiZCrujX7ISVUEY3RGJ2ivR0bJEwjM5Wh6yjv3xrMxjyp4Eg+2DXz
apIP1RkUApAfb3hk+5td1vCSZqxVUdasDisFocBjHvedpJxBJUsJFq5NwgQULcFGD+oKkQVf91D1dLWHR39gJT
1T+dLh/3sVpI9p5LIZ0tqgt3dUP/5MSgA4NU+5mEyPeMTBcf/0Rc6PW/rKKf6kVIyPuQ7yD0o/6D5m6Httzc
0iIZNLb6wQpyDP0Ry0ratr2WtxXLloJz6mN+HiiyVuw4TPUJid0R5v80RWIK4rRGRYSyW/P/6BH1XgqPsmp29q
wsX4LIVGv0RzLpLpp+8Z07oEkiRYvPSXYTm4Tsw51Ibmy3BopVswVzj1KnAG80F3GQmp/0W4E00VtG3NVBbDEW
m0dFjTB2wPJfpBW2AZzU6u9E6SnUqoYFucZq/PIXYjHBSY5xrw/eVw+DV9YJ8AN7bwBTzRB33LaVWjSPUYz
0iYYcXrzC9w1HmVqbsqQfVYsWnpg7a0AqjBWeBdFCqAhGxpCBjEc07Giaz03+YUgMY41oVGJFysCKtL8ANVsRg
NUqGy0ZKQULEuqXNHwRpaFwVQRedYPNCAScGJeIpNnk6P0HYTw5nCHCbkIYYYJAv1C1JeVHNR1rk8d7DN7syX
Z4TXXqi4rZhNY0/w9w8Mo7BvZgdsHpehquLQpdvEthbqnT91ds80rJweiwzQjDN6VY3czwrjY71C+4edvxs6j4
jeyMpcAhdRLW3vzis2rZnLSLhVQZK4A2RV28DbHkeV1aJyxCxP8T+V9yDF+v3aNcTWaD1Rtq19/Zsnlwmmru
S2VjC0CHB3K82xxSRNCI/oZo2P5hXdbf76ZiruYe3EdXUg0kSdwUy4jLaL9/IuK+2kNvt3aYaJ0XzUCPSuXX7
e57P0gxfktFDHakuGhD7b0t+35y0NxSC4qMMFBEaEwcWbqH+rBS71Y2103M+BYGZ+24j3GI0IFr3Q9mBQFlrA3
Cjbq2cGn52zNFtFEWJDzmf0F9yH1K6qPxBa2vGyR0GVPx8hWeFmMWSXS0fjLdGr1dMV/XZ62XDWHi20QLnuyc
5oHUBStcy3GNQ+/70RJtnveISB2tSEUBwh5ZBAUeh74KBx8phbfJ3v0M0zPun53yAoTkp04VUw+B/ft3hCNkV
L080drmBUB0+9f6e9d0dLqjItrnPKoZWvf1k3KNElHBtgt2LvvG+inko01fKyeL9Kvi1RhcNC+tzMkq9Vt0Fza
AheInZQdEaJ6AYDY8bEwwzGSf2WAio0zPRGrG1Cj1mWhxFA6SHSVttKPhZPaiuxeSuXZR/sFE1CB/yxKo8M0Fw
jbJTn4qGdt2N1+lWtRsjiRyL9uE4aEh0i183yy2FQzVslme0V374r2Pwygds95mn0syj1NAMA9BIHDFnuXemCG
j/KBMJanjC2mQsXzS/E/yHdOUXrWC4DBGLC7onxt+yfNTd/zmCIhBv3UYjNZNc3JX/ce3Nk7CNOMMCgtjGgw
Ehh+FXqWrhJU0br5pWKZ5wVeaRVsRwS4v4q2ASCEkUm65gtxKoM8IctK/FS7ugDb24zxS9dWEGty96QVgo0ym
/z1VMV1eV16qp0Y8Ldu18xf9QTIBoJeXgEB1dv8ne/MjNZU8WAQ0KRLjpDMobyMjB4MuvqzK801hZFrTc0bX0N
9ED+TzvUMS6wX+HP2K9eJmf2kH0US7an98uWh6vozqKxdFpKj3NgxZxK2KiIT/KwRj2tvoe5ts3kUfUCx0C0U
C1Iw2Lu6B1P2+3MUN7//mDR6HjGT96qXjtHFWFQKA0nhfw/tbzDQnB+zAW6dE+kRjCKo6tnFgnVYNF1htuQxB
Z3qh23hNiQB8pRww7oEK/R9aI0aDyj3Gn9iVa54Y3HgTowakNvasHks/wr41TRTy4bD951jsPikC0QrsrDd1Ln
zBPYVbLL8nsvRjFI8TE15JaZyBgIxXLLe1YdPMnSJe1oN97CwcsuQjR49nF9czch9Fh7eqyXcyA61dCJBor7RP
7PryDYrzn+J1nF07j2/MsRpHgml7QVE15NuEB8sTKAv52UC20yw0Di7bpkJcx8vyu1RJL2nkjKPhGaPx4yln4a
8L1cVe37Gf3ipEfka0XAr12YWi40GS0Ep75YHYTEs8t0UMISKX19pGfDzw68+nRmYq6TTASJ148i6atjNshKDv
/1sTmDSpi9pvU0THgKnuELFotYC1dLSBBHTRjYa2hU4RC24k0p/+k10L+t08GE/IZCbMMnWH9hTqRxZteIRH5E
/18Jp7vxGYwnE1Tgn00FaqTS3ICrLWK5yVRa2ME4fxzD1cJrb40i1cD1GskcTS0cfjPKhFnx6ysCeDYqmF9v0Z
wfrG4YtpksbVqBDkS0jMMyd7SCU9QtmXXd1NMfxNS0iX0zypKyq7VLXQ1RPnG74mn19DyItG+uxc5V688qqm0Q
eybZsCT1IchnAxQ15SMDvh+UuPj5c19t4r0IdyMCsPXjgbXYSGVC6VA53DaL+QsrXJ6fnmGH07fc5h1H830YNc
Aqo/peJaMgtmXIo0hygeQ6AGccZfRj46UJrQGyB0o077h9KjmGMYTq/S4XAT+g3fachXhQ2gjttGGHxv09p1jM
PST1ZCtSkCGdX4iqXDe+L9GdD1Ys2pD11svAJ1E8wHDzgXbpHOIbbafBTzskTIQ7R9uLyCmEUU1Ku0/D5731G
vR9TaKfW31znDy7ciYZLOADNm/Ec3IHRbbkCyIPD8D2t8N84ySiCmHxMcA96IDakMT4tyHswqUez0A3t3bRbtg
dhwTev7fcZ/ZQLGmm5i+q0cmRjkceA3fCvaKfiW10+WIr8iKx1LyGh+yr60d0/HJy7u49J836hYwSisod/VQVD
gs+E5+mYtjNJSDBitp/RgKQmPNFaLTiCD1gU+XM1/OdocrW7b01vPTDhY8AqgtR6np9kzIp7KgSJvJxhVNEaRy
3YbewVfBGFXW8uajYHJWjucKHKgDBTkbTu7a8YFBxGRJttOK9hbZ2Wa9dsPDFF2T1BrU2tSv+JWxcr12K8/H
GC8Bs6qBvoGQs+Egl6VjQFj8DSBLYrHRg7Bfn010eCSkJxTH3C+ideijq1rqx6ExQys8yi8ZJvDytX8GfHKqU1
Q3R541PNph/typhfwK0v/yGw2kYwsCHBPJPJRVfNs1YsBmhXmfvH+fWvZi5cNj4mlzaIT6MScNaj8CvrED0o60
6k0r081mrJKK5gQZkv7deXVJLLUGQU8u8mkkaYfp5a3kwYe9wZzLttkwquJZ2iNr11G0vnFWC3HdKfYzezFbks

2LBf/wcPognd01xEJRTu9VFxh7KdqysqGWnr8X2FbUuzIwZRuU6UYcRKcamKtW2j1Jbr77zC1RdtVmJzdb38sJx+g26i0Lt0D0//XgbUqia01B5y5vXJ/3zzmAyEppCI4cbbLNfSX70bGzDVuWixa1m9VfluI5NI0AH+uB1d0ukyaGxNwLCYff8rQQW6zyZCI3Qj06XsJHHXj jcWqpiuks+sQVjMIBdNt7BQl4ESNKWWZ/NxCmMTjHI7LITYprwk8rN1MzQlmYop1+2FfwXtZoTMAvFeskIrmZTk3nJsMKAFz973m104H0gYERrb0sEujqbQYUpSg6SASC5++Z2KqBs1YzWalGsCveR+BDTTo2XML3093Yiv2m+ASFUMM/cxR8BzAJZNTiZAV1Vo3GP04WQ/wRQNz5DWG6xp/bLKRFrJinBkoyyS4uHMI8SkoYhEIOMNOSzkeiyu6G5ThxPxjCpffB3Jm10h7ELuMwxeBzcl0JWpAP4p6N+KqDWSuWEMTzJnL1gj9JXGXbIfy2IiU9mZE7MQGSJWdvZIt6ekpReu61z1dBeX0sgFiuEKhBBob6fKx1akcwI9y+iE070W27ij8b3XbkwCP0oZbwFpswHr1XPR1oHIBjjo79HCR+EakgBm970e+N5tHzcBLuFhap4cyhWJEBgnNYzocz/v6dX8Ye9TDN4b+vvQJEEdSA6Ue4f7gLEihuHwZxn/t0f7SZ9emXPruzUzRVxh2G3KaMxPsTAEJYPwSd8L4c/2T/f0djBvCG0btgeKX5+/nGZVm7tuUxJM6yb4patp2knNU7hruizozjuzMj+QPjRXLJnigPmjnmNaeth6nXXecX3v2Gvm2Se9bYo9ivpDlzVpSjHwKvwEpBo71eoD79nrV6jUSU6kXpTYd60+SZfc1by8mdni0hANxBUHIHuUBmE0t3JRy0/hxejR1zpJycnpaJ3CYhu/D2mBZrhk7FTMwkzoPUvLz6KBzvHyq497FCwte0Sc6jmNS1a5FvYE6+008HBwBuBnAXTWDKDCRNPaL1NSxkTK1pgQbXux041sNpjhhHUcmhwh01AvBDMDMz4gPFL6YdtJ1gPmWg+HJyHHVeZQP6X4IwdreiS3ns/jMaf13RM81kgrQPEtWLrsYHD+CC8j+mrAT42JhyaMwMxKhT7+BInRacDQQF1sWXhwZvu75Bsg707bS7iXir5IAmqtUzfehtMKo/Kk7N9fiKLe98jM6y2s1RpKq/z1I2/kdmRfnZZSjDs50AAy6/pXTjM1sv8BIS01I9BRaUONEmz1kCnbTm/HYyFKChjd8wWqCQ+vg45m+DV18xc1b7XLuvQRUE2Ebqf1odPDADyphMhIwrmwnMGM95tcgb7rpI9nLXCprZgZQ6qUxCVXUY9Q36wmmbiNr+0SzS3kYNWrAdVoWTVMAFGr04M82vwh552Mz5/aJpAR01t4PZY09xsNNLHBi8Y7k4pkvQ95/ebVTuMkb0Nf+y75PtJ4Sg1StmXpbkQ1gpPoGViJyWh51fkYRjBfyhJ3B SV/I8e81Ih1QnqV3uBiVqvHTjMSTiwFs74rX/41BPLsz85B+nI4QvyTS0KghZyKzWXJYdyE8TrMAvzcvjMz0WP AOBxIQTbhzbvN8nI+M22re9y9cktxRSrqucyySj6ndDQDjQb8bphuphev0+1KZJct1A5Fswgp3e71EiQ+UbEOVcJJ4XkPTbwYnvNADkaw4Tn7R7YGeizweEr5EJ4D+LS0lp29w0bWXZmiP0v+ABecQdCdtF/HkGRvCUzkLusdpp0ISzK2Lig6jon1qMCMQOQyTHhJ3eHDLgWn63K1ZuzdYAc1qnn0076KN6E0ZzpwRdgv+pAxu1CdYEZF5XzrFS3HJ3N1wqY6TaecL/H303D+rebk7JA/JdiHsdZ9xwupZtNLS+alkvSZJy7g/YAMM3LOAYamTqy0Q4LXZntX7AQVHNYVW1l14Qu3boQEpIEPX4QSpbSu9KDTDp0X0wN/MNNxPpXjQ0HjF/c1dUet2fWHvZXTnBGAVdn7tCgdQf4dKXfP/MCL5A15rYbxw0qbsZPs15poP4gfm/GKGsX1eLJ20vnRNnptLQd2ap4c1C+p8odTdmokfETnvqjUkpASqJHrh2KPhzxWmZaDESNXLTAkPhWL+RA17NYH7k9ZQzXuFxXCB55rzR1/W91SFxTEctSy8W26D1WT67+tb1YBvhbctLTrBEf2P1nGx0zAAFwoRyxDe9ROAkMEF7sMhgWdeagTkGdwohWOWwFtaFH0mhxtBp3zR1Davm/f5EqJRC0N7JFua1Uwv/UiSsAA4w80aSJcvRoDVJNzs9+1Ajp0izby3EEZCjxCADMOVeV64sJ3SikMLgfE11HV0fRIuQ4U+rb80GNqVuy73++fZLq45n8vQ0yBBLGB80vt1Z6ymBivJ00aKcD4WcPWN4kjvsPQwPHGL1xTw6iJD3pAiCAR7VH37Y+IZJ+kBGhdXs1E9Q06drx0vIgtuJbU/qXnTOTqURodICre+pkhRG58iYiktvHa0zWx3GiDv2zVNLNFcc0121z0QCvWBtIm0Trj39F1GY9x7yE1/Vy6g1PNY5+847C8ajXjuZROSgMWmSGFXShJaDrTzU1qHaFU7m7JaDRor04Rs0Mk2AW6v+R/vnsF08w+HF50gnGaNE6NwMy6L27A+DptJb4Wshju+egaMQjDPyJp2LsEHHTh7RaKXhhWpqjkdYK94SwB7x36XaKj46KAc01HbP9F68RB33L5ErRrdJY50wL09JV2pKRwFnv+App+PJKWRNYhQ4gTXtAjReueAyURC570YL2qtkGk/LOTbbjH+sRWry3IHTGamgEWdtVfb14leq9d0aBMuLo9zKbTdxGWao56h42XdV2jEaRSFuc0PZYffNLY013CcZGoSgcPwoM3Bu8iqScaXL0xj60dqBL6m4J0bBc3Q0sDDSnftw3VcrtReEwg iWJY+RUKFRovd3ML9vYtpf+sz9t8gXvI2Nb1Lz9KdVQBLJxvYwfb0DZT7qxUVpMzufB89//8XhWqTY6QNDm8qtVQg/OP5TCWp1WD39CVA6AvMvQJQ/pvUMHrV23dCCd61PVEGQuek+UocSex3R/QFHh1a6d1J67zjPca136w4ah196mTPQvb/Idk15yyDYj4v85j6dZ71vUxhj/JYZGLkCn9UiX5feKEdRKQHFF7vsRRUEPEb8sIrpDh5NzNt0zEodsWVv509VMLpqDZ1SNJ9Ma80qbb+zThfrewGVH4eeexXGSz101Xh+/PSsxTFNbPXurjt28gHdi7SIQsR+SSuIFIrdEBZqc2EONoYEBjB5aun+ClyMcsGGH7Rum6jsrJqhaMz5Re/UwSzwdRwCKjUORh1qzDX+JZ0nHigSNtr9Kq6z/92efaS2dhGz3mL7Zkb0jgwjxKLxbqI1W7NX+Ia3L+gZ4y8KoB0uupL1vTxXeeA54V1IUC77z6sB2Q0im0zjd9/yfHDVzBKD/GwKpIXhZ9QMGh0qrmGcMP3rnrP1DnDEDI50ankSLQsd+/i4pIhd1Wgvh2m4GS10mLquHMxrwBy07kXST3ggqWCMW0jX/WQJtv2xmc+uZgRgsib3bxzxHTI2D92U9QzLbVwhL18tHSHrsLGcUk8fv89U4L5yaE3XcSmMF1bp8U7xmLRWv/9ixGMeNmKZ8a/9NSuYsVuzbVt1zm1RQ9hQmhA0drkdrJsgqeq3bDh1U4jT3gdHRYoQsg90gzUbC+Tz0GKIY7wGxLsRo4oPzr/Vc6C1uBEiF9MuTI6z3zI5vxmG49tPSNaDj1xHqGciSrZqEV3010vK8t84N+aB11oXk+3+Nptt0QWhM3xxqwTTWMBj0+617nEqYdL01w51qf3aNL2nJ/cRe6BeLhQTE0AvnrbKMFf1NdZ/ZBRBhmSIQNjfChAOFQ/N+i2A9f9kN1jVxziA11NSDrwpHcZoNJKPptSfhD5pK7+EKDNEFAR3xKEYzjbqgHUFxsYefoaidRs5St9Wc1ek+0LSJAo0bwo9n4fubVwTJBGj75oFhC/zF2CJ8V1NHyErZg0u5IZXxGtu6cvwPCMA1pEJi0uQeD8/wqyQx1KZKZzDLpUT9nj3J0WZfVCbyhMiyZkvrc9NU+w74zK6R3KLJewx7dv+OCFKXhK7ZRHnxo/gfZfNb7u9fgkp/XqmdAtVurJ5/JDCawwb7vR5qV7sHGF8vTDZV+/V/Eq1E/Ea0HxxNhyz13Mb6DqmYnFucU4S0sVjpfzYwNk8K7be rQ3uJaunGWYsivqv0+UNMN5I0DVKHvuEQNSxZRCnw7fUMq8XLm3T8KUg6ESwFO0tjZEUNFR86kb5qhv1H0kIVj

f7oYtGlsSZyFEXOTQ9JmmT0aVf5n+dpHQNK6spRNMoiiBN0CEg9kXmEkaQSy+4UaBmfpxuLNJLwSjJPIRB35U
6ff64uKA/Tofa430arHRufpFFbHziNid0aegrZJHgI16V9fRtSQU+i0JbswrbtmT/cwX312j+tK//InF+zgya
E49L16JMDIQ88vzR4yWIJPeH0+/oskMeqqMn19nM7cpVsYJzJyENMrIx0LmjuxhtY1gS1H0S6YAb0WJi57L3
FLvfsfP/TMZkJ44U1yqp45t618qY54d48S9yyx9TC1Ia2ZdDYjR51Z60PbowE4tzUAmnzNH55HvQH3BbFU7q7Q
JAut5DcnY6eqs03WY/RSm6d+0Z5U+xtp71bgyj/aXe09U9HPYg4wRp/SScaR1CggdHiuz9u69NIN2AsTaFhj4a
XpaWEkw166Zet2KRI8s6qhmUigfDeiweW3eWrNPWWhJMP0UCfrM4VNN1XgdWKu/GUdhy8mG2db2LvHU0pp3p4+
Yah7TPR1FFkyuuZB5feMPYIEG/tc9jvoTi2ZhVVUELbyt/D93ISZw+dqt250cHVbCW0+4ajU8UJ1PVRKfKeaj
Zj5LGEVN4ho70PEC9+peTPggwg+WvACGYmTW5MD2TXALzs7c3S9pCIxwFLSrqQk0K3wW/PtRLRiJyRj7TETeqe0
cPQCGxw7XgGKCFddYA6wnX/TDZQxp0LUz0nDtvESj90BtBL+KAXIw3/KLH2u8cJKDkDg3TVvCNNGK0gXT+bhVU
nrC0SW2/2/Jvgv7EJxL6pgd+0zJUDY0Cx3dbFbjQtHBIxnVUdK7MTKgWSv56Ttfru2UDaSabQ2+mkEqZKcHKSe
SJVi0JYVGJ2Q+Vx70PLUPE9/NOvTo9ResVqYz+HGxwf00utovGeQMW16IL/N1/Dhj4IT1N1F3I/8oAd8tX8IF
8Kvz7NNbi1n/i2IaE8cdmGT95MYHxyd725wzwg+4133V6LWx3nYU8N011PG3/mgljm3jzdbw87vp2XmPuEKzYR
OARKJXnH9w0GJ/dA5PDGt2KBNUICsp6GsBZLc in IJEEMKLT7bid05o3KfmBpv1EcTiK7yXS61FMtwu4Fh/sR6o
YJ71goMU nIIVDb+IrGf2Af4iqIZ3wjzQwcxEjBZy0c1PdnL1WUZb1vyrfvTfCeyMe5YjjHuXxxweGatush+iF
pnUZezeWZd56iqwNjXixiAvpqNHG4DMRAMoZVBHWbZsEoz06GGH2zNK1fU1aRMzTdfYcKbq0SptTUPRqDwAodW
xL7tw/wZFL6rqaDVzzauEgE9Tv3BvDQCmqAL++2zSB51d+NahpJ78TuAuzqhW+egWJsryD0Q4MeB8dfgUPyctr
kPt1ptIBQ1BHQC379KLVNB SyHI5vDZ6p4p8PkTu7LIJf6itUNzH/kh46p3ktPTqY8vUN26LJ4uJvU1iWSSpigb
Y6vpb04i7jX33QKZx7xizrAcpd+1QbZ5jXw9DxkN12VbyjIB67uFutAaP8VvAvjuYadY8JBXiGdKdGbRqzobK
2p24e6s7Nhe0eFSi0gDaNVW8a69pjDje+1qzWBixA9KUDqm0/LG9JcyZucYZ7DMdeM2mecye1ehSASN5SbGrZ
LGiEBpeDzcrY/LC5a0eqeLZ5IsYgb85zdnpsIXpiddF2LUTshrbC5dQY0ZmXA1Y767cNh5tIjpitVkfUDJB21J
0wsfxsYvWvHf00krbbBHyexQ1kXV1jLRF+o8L2fedIkv9L35HtQPdB7F iVoOT+RewJQrsca5DY35AHXMLPRud
7wGxUD+/5BkJ+tr7x71BzjejMrymNvVRYej94ep6ENxec22f+PcS0+100h7ZxFrBIEbstLsXoVFqKhRe0hW23i
RmB1cRP371AmPc8ZTsJgJZY037qQwdNZX8Zh5ouTW/1LW9oHufPpnLZLbphSWzBJ/FCONWUoC+T036+FkWYNDk
1p8rNx0TcCnbQm0N3tQkGfWF5bCOX57ccN+z132nGdxnQ4pJ1qtnA4GcTMfyap9+0XB0/P+LgP/WMy5ynD2W2i
D0Me0ZIL0YVmbX7Uw6h1fLmqV01K10Kb9HtSm3JFp4m98DX0971YcVPER8qiao05h/KBK1UfCVBF9HB75dgU0
StnyThJug4Is69AN9X3KAoRjqlvotZGsTx3fAnBr0jjDiHfv0J6izX2YpNaTbNHMgRCpVZiYibUSVCGRpRN1
+DOE2CTFcWgCFCj1Awa6AnQ2r0H4XzrhB0o5zXf2s+h+xRBQMvtytwde0ZVRCcr6Yb9tHnWmvi6BMRGMUgMy0V9
bNkA15DoJbwGRUWpZV4fESWDuiosLnijpVV1hCx4nmQGvDQpuBi2FzU/eBa/cZCgpFk+pnPQo33DBzu3v0PEhq
Dox00zInTf5IEm1BA1xVR54Yn78Qeqfvf1KRIZCeeTxEMnWdnYPQvKJLsTdcpoEgQFZGVKQ/EcVN+sRpqTnqZz
bTLY3HDjvJsbWJFsZ3BgfpxpAZ6Ws1aZg9rej2chsfkYADHNCbIg51agHany0opQamRcxL8XcvFRfLkMzr/DVj
HDJ5NFdqwTPx1sfEH1IyJr9yM0RiX20epSW4z2W01PwmuV/1LpaSdHwnv4dd4oZCML7nXlmtrd0TjppKVAVZVI
ObjdCsRX02rw+dxv/4X1n63QBjV1GHKXfxVaZZ2MIS3nNfhRSPQnf56BBFpugwCNCg7nwhgNwDHE51Gx/BGx7
RQQCg2GzHkVWAZ7EHetuzXgDR6Ape6BnhZCDG1QLdEV8sBU8S/9K9mbaZj0tH5FwqDhxd98d+tL108DYfCam2
IhRSUEiDR8VMU1q0rNUcBJWiY3x4LSDXMgS9yNLdE02Ja/1ByL+isqEjXJKYg4HPrgeioCa3I6dAIPY67/icIM
w03CTS8z/qwgkd1cRj6PZ0eDnFD+kivvDbecJbHs2DCC9xUTFr5u0cPpj+1Wn0fWYvr1DxCbt1LCcVT6qy+jMu
S03io430P+cc1XuyLFFU1zsM09Et+NemzZGfg5MK3QifqA9RcTo9+T4+oIi84TX+DgjpWeqrL1tJf7CIH23zQe
8VX1CZdgs4E9XJ+HkobDH144Vehf2yZtkS9zebc8H2EEhvfEbQz/+CarzsraJwIILfb32/2IPYULFKZBHjEMqF
RbsIVk5k18crwU70WrXQ+zvKUiyyuvxj+5ykYg71ZprwRZAGw1V0YYjq108Pf+xrRsn+xtLkfBGEX997Pq/nyK
6PYXGk0jIQR76KLKHRcQrCOrX9fwSWqaQSTRQLJcZLs1rwWMXk0ht4H1zFvJOAnmadYUA+YIWz3SuIN055HyX1
tLZZTf2i8fIy5q3o0fqDAjgWugiZ01z0Ms0KPCb1bmmP+M/SojLshUq2yhs755AE+sSr1N01qTdREQed5cHA
q0ox05Q5Qel+UdQdBVmBbdgg6Ig/CdNNu5Jf6YZtory/4AjQAJK0rtzBhn8ni83/ds0F0hF+iRmYNWHeKmoU+S
KDDRVkPqYEMSKDh26kKGyhvASbDtjvV3mUwirQq8sFmdKdK5bCwMpa/Yew1CQ1xkFT6K6xQJ1Zkz1JU1QMDcP
ShCJarpkn3Htn3Q9La+iwWFngfiBVXXcopztTf3SRrsZuStHmtjZ8Jui5n2Xxbdvs6RDfodRhxTMWAXs0Fq/aB
28ZTfN5yaqJa2NdA6yk1V80KhfqRDWDvN0shqcNcVIaUPEgasRWSxv1VjkowehM9YjNhQ4TwluEhgSQw10Ev1f
/8e65aQuJA8jHGrAF2TL43W9N0YkqoepXs+8Ect6F7bbn16pwANhVUynq20fF1uyWLEHNDrnwoXugPYFBXcsf
kxxDeoCz7RoG8UxaATjxCj6wWkYINTIITeZCGX/k0Cx393BpoSTVWaFbQKCTY60GnEInvvsyq/f4mnzhIb4p
qkjuzWF6yyqz/5MFHqrtX+Zeh+rjouPr/japA7IZwczq3b6q1fu7nYevCes2dGT8wVY4Mc91sLf6bh/b+nQ4aQ1
PItHR8mUGyBa8Zt1x5v8cM9KBih0QttnE/tPVnbrIq1n3zvNUT4WHQdcSJAHzusXQT+7iFiCaLi61krF6o1Hu
SL5CM5iP/VOQ0hZN+5+m/SCo9c4vhHBw8pN1btguSCXEAsiRZb/fUZB2f/C6iVQ8w4pihSm/f9BfWSWM1FVW/7
URPdMkbbfmgvY2p+UmgDP+DaFxTGQXA2uaeNFT+c1aVkZ003oo1NKKuRXIq7tc/AxqgMscNM0ZEtibzILWvZmR
HZtcLf5/YG2mkntx3sooGuW57kAiUsRsv7XoJjGChLBvTLBnUcan7RSX4bvwwRKiT9cHOSFSU8BnHYMF38F/D

PG8Rsujp914j8d3/SuEi1V+Yc1Zbifw3qq9M7sKfm2qLJmA1J0MTNPf+IHWaR5VNn3qRopQRXqErr1/ExuCp6g
2IXcDjTFBteGvigi7+fF77U4GJPQvUrxtEZpQL+10DYDBvmbMRRtooQ9d2+1LRBzst7ITHdjm8jHICEcAGNHT
JS86ur5yAtVHNCvHZKVFy1P198APc31oXMctxV2Sdc0Ugv7HRqtkjJfhbxM7205JL2bbkCMMYX7WEZ+3zKDLK
53KRQ1GEI8urwcrYzsav8GpASE9TjpLWxjY2bQrnUVXJUmvVG5sUfDi1Tz1vdgU/CsnPdNY2E25AEx0ALia6F
URf11ST91BJsm1D6uYGnys2BoY08BRLg3/pjLyaID/KMxnYoR0UYLjufkptDnwe30S4chLHewtp3Mz/AZHMrQo
VpXb7RwMY68zJTnnusUonkBggCn816QGJc1N0+knGiX4mkqv6fDLPS0cI9cibaeF7kjy0xK9GUPreb7aM/DMF
MWWKpe1NziyjYrdiWjvIq0LCy0z9SeND2gx+5CsNnDZrfDmaD3p6iDRvEt0ymWfNWKoxGnLQFv4W3N1/MNpKVm
PzpjvrghtYTbgBZJK/OwyJpakcodRWt5AERC+Q2S6I7rfTYzW0tv/9h11Qosf0qxK0rBbH1Lwn89jEptSQmM
8QWpBvDOUAAgoCvF6SP9IFE9pXcA2yUs5rF1v98/Uw0Hmsi9/TCVHMC90kQbUQkfRmgDI7L113H+1aCiMIgEgk
y75m1IMitFQHibIH2A0IF3Sc5qc1z0k8Ym3j9QUYRgTDxi7nQSksJPv4N0J1YSSxQ7S0A57RIHg0q8zitam5h
zavH8UawU5PI1Lpm3H878LK11Scx0GU9IB3ApJS1ZxZ0j1fS0dnfZDJ+ssdZEtTTH8YyENDyRqfcldk32WAPb
W+1+86KduzdG7aJvL98/oqMiUDowKWgcbVDjv23M56ajfZgRPmRFpk3DfdvfHz41DcMmfeQ8kZNkCL3npsBxp0
DqafdB8+KpxerPbN+K0BZoi8j1RsJuU/G276FF6YMrFCvsc02oqXerMcxXJrb3Uo1v0BA+IUAKIDcEj0wmsY
xAZp3tnz9bnnJWlMAoikDS9HyEf210+nzQZfasLbnzQhf+Uck1KfoIEbm4EHKwG1rPOKKQ1Ic0dIYwF2XNeRN6
v72/b8W0z9a8dfmfN6fSWeYBMQyneUvXX5GJZ/+DpHA76z/+C37UqrExw1TyfJmkni91H4MQmHHi1HU5UQPcWg
+RGDjxFmu6S0QUBQ4uJP+QdLkc/vDsVKHw2KksA0tDndZM2F1/0fUj+GCM4testCEU0BVf8J1Pi9z7Y9tdEGw5
RfCDSMhDC7IMjcSe37xHNS6hP4aJN5aBmPlqgHP1voYIyz+X00DixnNsxtWCqXmxLkWffMbDovzUztAJ201Fip
wgD4qY+ppEza8rBsi3LqgEAb0A7rTB8d8XzzE22A6ohyZaqCgPEaxBpGwOsa/KDDJCB EgAyEIQe1oAQMeXMT1d
Dd3FAZjZ1Eaq8mWyIkZd4NeWauju5gAzVBM18djiJ1Fzpc2V13FyQcStN8I90Dp6GpspsAWLay37ZnikxjRwp
Co3cpkDpoynrv145b9I5U0zBjTcw1v7dMpBt+eyhJnGyD0ROXs3UX9rVXRUEkeymUQZp1fmuHGf4jzK1ImN5S
eLNEwHEdmiQBGVtg4C0Qv5Rr6LsrzK0wXsPWg1DUAEKOuv+9nNRSqRVah8rfZepkh4kkGfKMNEHF092FrDaBU/
z9dPmC1g1eTaekLwdMjbHQqfo1AR6dwz/JM8h+Dhnca7k1BkXfe1s6r7JcvAok5bYlu2Nm0yfiktVe1GyIr+9q
oZ5MNknyocVj4gzhsY+XSK63vvehB7n4C1ROXtE9Z8djJtnxWb/HECCOHC8BDkZ3kzf+ZNRew1e1mSnRefkFw
/1CTR2fhD00q6S99eA9e6jTLvEx1v8p1vsRnRhF68IzXre4W2/Bnb6ppTNAunCigdmUJrnwn1t1U6WdvFhMb
xoDhRiBqNrqafjEvhUH5zCnbQV0ntxp/A99B+2WpEHmBbg8oyVKodV2CQmjE4e2aii2oybU4wx2SsdSY5zwly
LXsNrigrizYYlp0UUr68Sa1GRw5bKSzzKESAbNQ011224eDThcwE3jicJcrJ7EYLtt0DphVQffZUvCR7drepG
daLnLp3txMbGUZT15sU4+RX01nyI8jeQDt6j0e4Tsjjnh0mySp/0tUc0Fj2ME4UsVsrgfWPnCLucy6X0pyzW
XsEedsnz+k3znYtzPTzTaHXqPaJVFx2ewt9L38sznS7mXkfWgWV9K9dUKCCkD6TfrJuhp0pjesYg+F1ZhrPEp
BD2tConchM9XCyhXWQJAL0JP1ASLVSRHkYZnwjlhnxI4PnKZ/ZyVpyCsC19JL6P3ekZLOLva1PzaVSkBNGXT3M
jb0ia9F0D/cConJEsWX3WHscD4pRJSr1Ynppzm1WPdn9kzxD10Kz6XX6T80qg0UAQERqSAMbGKpBPGlxddgT1S
AGH+/6iQ69wkLsz55sRG3evvJNu0iKWoJNAKTVwT6g+JwA/N/swcQkRmRutzh3H2r1Ys17rhgSpRo0F6Zw0/9
JcrIJ+DGRBnpXb8+3L1kQpcM8PiwXfq2IpDoGRuw0XKwAqgyV479b3e/8TzU6xQCQ0YryzKT1Z5Gg6hpPHkei3
0Y1DQLp8uZce/3KUXUsaNSVwAk85y3Sje58G/QKfm1fIj0ryKpF9tNaIyWmASdM9SAkceADS4JXaHTAf1mT4j
Err31TXCIvIN/ucyQJVy02pUQQQh2SsGIYAYtG0wulx9qMHURd+nBWez+n1QnU4nx8Q8MfqzqZjp/pEgmUo8IWY
KJk5CaZkYr/HLIChe1WYLMdk5b6rV0bCauRz/XgubW9/qDL3bKsC960yPDqIxw9CrGbxjeUGZck1zQqCMMz/YH
bHCKK9rTPSZmAbss1aR2d/MH7LVEuMySSEajr6r10KqsmkhZxSm90JBIsoawiNhJmqWgBpVYVwYX85g4IQU5X2
aBB7oijefEvJbuZIj67c6LBZZand2edfGjs2mhzvignDDjLgajrRJzigs1P9chqrz5xbnHbCRz+kn+M8eCfJ9m
SehnYmvQ0E3Xy69ifAAivYNSU31RdWChitB3JAUkIPyGY9yNfwJuj3JmoR75n/+9T2NjyV+BVKSkNndUNCiGc1
wL/rsDDBQrSpIzZfVUvet6xYAcTyEMQGewuDNZauqXVj5igwo5NdGrTVns3RVG5VLiFCVgAKqETW80B6J1IMIn
x+FGRvdIwH7FktFRMQR7wzqKEmwXWM7Zk+ty8BsRbw/pNr9Eg43q9rcSy/qsbpReYEkvzcgaExwWg70DK1Yw0x
DavfaXMwP0Cb0Z2qjk5zs4Gijck5h+IAq+2nfj0qMgqWt5+0SPjVLZ7ECa0cUzul/Q0Bhkdc4fCmua5ekc1I9
fkjH961j6AicNAK9K6JhfJ1mVjC4Q/oFcV2xx3vafIT//1R14jqh/hMcCUke106C2D0m8ss0pRIS3o05i0eM0q
VZen+xamVX0jgC63EA75wn7UutB4bRRZ9ATZ8kP7KVMCCNxUdyPFrgs0MxEJchQ5oPjn1557DYSA+W+otMvMBy
t4BnoTEt4T09NLEBmnui97isofkz/1Qip7aJxzTdszSiVAMGcw5YoNqxRPF1m8Y3PUEvSPogysmMSIag//NK0o
LwvhmacKbfH6uhvL720F8rn08ed15pW1DRkfNjoCSwC//pYg/6C0b1Znrjs/ihtMzXN0ggbGP1HQwahzauuWs
LvhwmJ668H0yUnJalyaMfY/mInaoDGIFzjYn3CF1C+qgE+YM1/r1B0GxpWnrfQfJ0pW0uvK411t/fdrNu5wYcqQ
adSZNNP1VmUDyuDcF+x25L9DouBBrZbE0kaGh0sY6A/6pRF+kuAbFIPmSV3ZcJjKZu+enFdGiz/kolz+1xU84b
pyhupz/MiQiMeyHeVmjcVrS0VvBdRMv9gvEAUNclxRQ1WM4kwus2U0KqX9L4yFvD0kpy0p28iWVxxip4offlqr
UQWgPE/B1gNX7pD0zwTh/upAo/jUbUA6fPQygeo908m+LeddoCfcEt04jCTIP+nq/ZvuVd3zf93BYVar2J5qr1
uNyTPfa6dX56YHRT5I7Ts03jJG8+riLRxFvVoJIfkI5LtgpzEy7e9Z6h66vigzcb4ebqJppht+39fj6R13VwzC
3wFS/Hmha6x8J91onuWV9LIWXDu8YX+cY8s+ZwRDrivYGMvW7D154AkeXCD2EgNyacZ2xyLaaumUbCOI2JBMaqb

Amcoj/n7U26A4azt6DMax2BLzv09MZCCRnoNtSEhevGXzo7PLCyWR+w90+EN/B6tIYMcVxd3ATJqnYuOafqMYw
pTygjsEt7eGvuTPmNln5r9itEU2aUbLVwD6LHbmqJpsZDhoNV5nACK704Ij4VR3Ej06Z2hGpo+R8EKF0uXuyy
Zf9jWdItftBdGFq1EXaWHsKWjEBhN8fm+Yl6it+l/sVTS+YckdvZfyHnbPrpwY2Rt2bME+oQIzo9zuJ2LCwEj
BFyBxpxLLMjDGLf0yNbtT6LeuCUfMQfyC01+9HxdQ2TVkr+2jAfLIEWHpef/6F+TwYDXHc5hSLQZJ4Zk7yZCc
ARzCOnsvqVokU7zq/X3VagpUITSz/10AVpZP3BaRkjCjrwEBJhuYTrYco0wLyJ3qFAitwfZDyRG2Ma0fCcZ+0
1SRv8KY78j1zdIgMGgKSUudh70S19ICuPeWe7cLTf30IHw391q4sjbrxNoZkCmqPee9be1D7LyT/ZP1QxeJJDP
Qxfqvcdmm+vduNkryj9Nqzt2Wv7GbVxQ7NbJKfbPB086JMcI64DMD1EUQr+CBKopZNVYTTcNQxfBqUIwdgoI
j6g5F9xV2JzVaWNRIsavxSnnMfz2C65Ln2W5HRppkh6sqq1xL4LguyQxqSTI1TRzI57FuZde43EvJ4Zrl1hwuq
XGWzZ1vee/+84QGINPU5LFWBDQyPyQkB+UdkBh+5kHRLUertHrNDYRV6zCmsU1M30379BcRPA11okLuzLMicV1
7EGkjDw6+aUCzVkJQASK0GWoQEM1u1i8NIi5Nak8z7FwRMQRsTQsopcDFARHwC3uwP70PvQZ803rP6JX6HVE2Nf
Cc7oge9P/G4d1G0EbGEctA+zcDf7T3Tjoxmbwuh+u25rGkCH+F2qZTXLKxYYbv9/atYv8NiGHw+JHs7pAG3wQZ
+pHbulkSGAiVcA7UgSvvi31ftnLQzvCZt45Y4SWHJH1EgSVVX1sijahw8+05Zu0f19XTTRRH+7AJItwYgZ1i01
/2SaSRTqh8cdC6a77Wy+he7+7i3h5NkD3CXHrV+awPqRw7tkMbA/p4YHL1Rsa09pUAHdRsVacvAleZSox9q06i
z09PH95XT1bQdRBw3RhrsP/sVWQA7s/iXG13WXHQPiEFk4+4As3YMyPnqM4fNWCHngPwKCfGg/w8Tx5KtXuv6s
j3RAeVbhZR657iK1JbG4qgAJETi9NgqXDQ7auyaBP4Q1XUWzyod9D5Mz+xMq9o5Tgp4zpSiYWG13G/mC/qwVLM
sp3v0ERh44tRhjkR9oJXa6NHn4GITQ9P3MF6gpEFXomILXDLMNrb44qg5UGJM26tq44peBWELCxGAD1j5m1XdI
B4sb15x5RPiHHbrRkwyy6VfSybQR47CJgduZiMUqIfWUKOAiiT536FnjDwvgU2rzEoDiFnrLuX88trV5D4Na90
S7cN75baGwSgmvIYd252q+R2BFfBewz74NnVMBPaadwB7zh6/qw7+qmN+Mzb/oayEsxuz/ZGExL0+whavX/v47
kj7qFypcCd09TGZBCwP7d0KqrtHwp601y6pY/XAo0y7xUrxcRbm3EuVmASu0/mEJadtK9W2g5McbA+f3mjsRHu
XbkgWKnApCcxXhvu/30ZoNneM3+e9aXqqzxV3AIzZJf8EHAKsepCVHFc1gHLgvVTf00aTOXHik9S9FkyaBt6x0
yIsoHCHdyrh+KpvdT6/0pSpV8PGh7GUu/L503T8e//msyMVKFYvpz4phEmS8CRUDQCZrifyktNLhP1A/octiR1
6RoJH9UKMC71aCLu8BwR3uov6IaXKzJrlhd1K11WE1/FDZiSed0tTw8V6CLnFeKqiv7cGwd5ctkvkwV5K47K1w
3JtwGVYjRuplqpzWgtWH8oWHvnuj0LeVsblgh6vCJU39f9j1D74K1LIBCaYI2SWYamKHmi1eAVLRJssTq2Yhd+
0yB321WJ1ZN52nKSn2HfZw0+HAgXs0FnGouo9FRvRqkrKcGreUrrKKBjDB1aER0bggXJE4EF3uL5QTN6fjF7yL
enadKq5Lew+z0ubVraWY0yhjkbHR6sUjusHCb1DoWIWR3jITw1umr/1AG4ltf7euPy1shbQhDdtHkdRmAsSsD6
o00RMMh4I9QhSd6PH/T0m6XqpRMzbw5UXm6ltEmbG8hrZPv6HP+/NKORv77tBAELGP33f4H3dvmLYM0mDMPqGS
rvYa5EuKV1Z/JGQ6InmPHo9D6p4xRX2xykxiy6PwriMsg0u1pm4iULIC+Bt9HkvJE4o4nkAcnim1RpDBTGSGr
1mOVLtp+8Cz9PyAVK3A1E9t44AQGneLFpjzBF0hzxkImIqYRgpU7aeB4RWp/cxybTwsc1tGE3G2HzPhCcDmEa
rrQmoyC3wElFyKd3KYy3bxYKEAehvJU2VQ/TnHvJG9pNh8uNm4Y68vcSbR2Ga+n0NjFGVRC44wMzPtzyJ09+dR
9aDRm4H7XMoTSqBGFcm1pvZ1c+TuVYKNBkhHKnzFSsLzkoFzseJE/ajqiUoewp7Wo07vR1kL1n6YVvMwW5Ig
Ntukr1Itx33VIc+x8g8krQ1KXLfi8ng6WRp9w666Ad1o7W09h64pqIIcxzWQsjAxIyME8ECV3bmtdRPxjpdYkE
cvJpa03xW319sVM3M4zRafrI5u6oFnVw5kZ951SKy+ut1t42+fJ79LLjvCARwccUArcom04JCwTvG7d3R8zNr8
PJhYIH9Z1a1jFnGYhb4YSc9be371yfYGrYVfzJ5vd2QU1U152iD/deXwdCIIfuJX37KUS43WWpd4Q+IuYN1MjZ
u11S/iAQrv3KUTU0tbKYuDt6P1nK3u5DL+U3ACdQ/c+s+9WA5oKDfyK2vRVIVb60afrXCPWDV9MwCiTo80T2cT
DUS9R+mWnHXF016zC9rqYJGFdEfH/FpH0gYARhDaVqEA0gDWDB0996JDN96Moh8ujNqkaiwmpQ5znrP1DnBID2
T4og1kigFJcRXb2vDa9fmRwYyoPH1y9uqdpDZv/w9KoMb2QvezoRJbksBnWV+JKnQVjEJxZ1JV6tDe8fUG+G9M
FnB0EIMy8VpgIJTpIzQpLOEJV5bV/OSR4wYLQ04ULh1zba8Q+oAlmCBs0r/HUiD8r0a0kGX40C+3Mmjbd1e+T6
J1z2p5mAQdcykrMzsrd0zaeuX0hV9Z1tbdJfEkVpLWYamtTPBt6RID/+Jtn1PRfwWR5ab8Qht2PqzNoHmykP
1X9Bb0x8LrqJ95j2wDXA+3j/IyeSrrGDi+dFGtpHhGKZXfcLvvuVaPxi9tg0oRtByYNNNhJpoeXkzrxTcNNcrx
mcK+0sQfqMg35IuuWf0SkjyUf0AJI0qdK2mAC+iPCjVd0xKQRikXUpVGdGZzRo/qd/PMWrfKaQEdNA+h8000In
GqRo5889+87sW0R3aRTxKR21rIpL79ILEqrTbMoP0vv/NzVkr0EVKqqETjVTwgXnlfqUb3spzye3YqlCBH2Vn
nQTiH7uKW3428A2DSp0pCYFYCp4/3fCYB4BMOoVNzDQYcJcygjRjzoo4hTDNidzaZrnHDLknPhnSK8vuLPOUh
wnEFhx3vwJQCIa2rzA880SIAZHWwuBChicR7Y/tE4nq0za50LQqmji0p7EUE/+oAEIeGv2YIR2bf0t8scV15u2
Q6I/Me0xcbCIH0oxgEg86Hchohx88HiBTfsj5M2rK+AYSTVUQe5PYPZqBsQUKUQzx1i3/gjMGkjqv4k5JmHqQe
Z0P7iD7/KNQV1tP9tWs/mxXZefqqzKpbkfvQ+k172ygBX51eJnQH+yRWho4PNi1eTxJGqvoqzePZoj+vhad
t080WfGEG5g4ai97pRyoqbujQNAss/mkRWGrZ0Pf9Nsbs1/TG7DQfCfNytF6vxk/1LzaakYNY0s+fwJphYkhfg
xdq2eSWLIXPoa1VHAtwn7fg8EJ8y70vjKkGy3TixIJz77sQqARXK9BdrrhZfJ7iSBm6yh5o+sLd+4FB1EqRhm4
x11T5e5qMbtU+0WRetpRTDIIhUoTP9I1sUSAmbHxfVuX7cnu2E2HHRetJja4Jqz1A9Z5G88ZMpVy00F0B81+QzBf
fGGKbYkYPV2xXcE0m3vDgjIaea7210bXx0KxRZea0GM50xJKuHGH4brWZ/kE1zTJL1t0BUTb8MIkpCHxNu3V4P
BkXZ+xyw8C0QWNn+npFX2t3VifEmxNWN5pS3j1SP1P0tba1NSsXVHyQqhKAI2QxCwQcJ0WzsIGjPy2dXtj6koU
hrMXBKfT3ffzVCfxXW+INIQImevuVZoq55n2X6sPH/z8UIb6UiJEWnnwayaKFslrpsGdWyaG8lajyQ602L+E7c

oM1r4MyPfmAjHMPK13i3aw74Y1EV+m9cQINV8dRSpKBNqsTn6VYH/HbHkG3iIBnihT320w1HrurWHfeLj1e6H
j9RMKvSd7QULzstAhWTSeKxc/4zjXgFeSm08UcfSDucwP3vNIeYjELkrOR17A7k0k5md+e5zPX1kZMuwrWx0jk
Rqfw6CqSi5i7ftaDHHd1aX1PdVtS4F5ntP4Pz8SqhFeczTIZXnBLsgyBP8MX0xnomb0B5HWFZeEAI6PiMq9uCd
giYXPx1HsUWgIFZY8jZ2Ic8j5n3eJCT1gSDzu7VNs+9I4aTw3gBbgML+j6s30767JWowxPBfJMeV5YzzvAlrPo
hgu+Asyx1Atyn18tKuzeTU0/ACTXAT0us09fy6jEcfc9m0L0SXvAiCxTT76CaUhk0nkGxx8Xw3IfwCmsXn9Fv
psS6S1BGrfXA550ory+x57+wNqYi/ASTwsfwYh21yVsVWUgP2ZyufzUtTiPCjG3oU70pKmoVP3c6MQgtEK3nJm
9TSTP9pyjQ8IE0m95jtg5yZf1LbzNhuJN221rC9k2pnI+MzMquRlwV6J0T+6gtX+WtT/KWGM8Z2Wv+80Fqd0
XKs6aGG7GnevGsau36CQTmo9rDqYDGBIaeOLcTol+V+mjxx1JEQBHDf9V9aIIloeJ74qobSVkbZGBvhFS0tnF0
txGXckCcwKnxWkzddDbQUUR58095erHaQpeF1MbUmwxjg8t/7o8JVSG47+vXD8Cp7+0yG0uZ0300zd+A0euWeX
+SF6k/2M8ru/yyaecAMj0xDUTEcvyiM4RQerL9C30f1cZB/3Dpj0oVA1F9NG4LJR71K75s4fqmG8PwyTz7PZB5
VQ5Exeka0vPNCsovjLvB6b0KeG5FURoE3ssI+1m+8RCxfG/1LZX53WSjjRYXLJkcrswbXNkbnBC0tW9IquqRh1
I7BsVhrslwdwgBHVxvAQ/o1gzhWSgku7BTFxK1G2gi8Tlru1iA5HyG1B8naBmkYkkBwkKTWx3G9AGDtZjK8jPi1
Y+uEoIF9DP1gbF9yJ9W3TkdtptG/ghPoy/4DexlrkHfGuaYx/hyoJBOP2FTK1Zc4Daf1Tmd4B3MjkZJKIeXAV
KAbfE3+Stf05ftKCZfnmItW1Z8gXJ4bP3B0H0icBolB2Cgj7+4J2yeIkTK4kHsj4jNCftKRmUDbfioHcfaEuXJ
GpRz0tOya+YKXWq5W2jnpkyvS+7XpTxyRATbfD1b/0/y0gGRwBcYDQqnR3TAK4ycCLEJ9zijuxc0W+9itC9Wai
JZoUt4WFcp0f/qkwyrMoAFRtmCx8haJiQHuxLQOZIP+t/oJpD1UTxxd3UfYxj1n3HWWXGKwEwqkeZEXNepMKL/
y4/kLWYBWwHFFJ0wR6+4qgYpw+IsKNwpwC9wlrtfc6s1MVYpDssU0hw0hLQGSWXjwu2WwDdx6h0m/KvKRIPt3
18reEgZWzG850WYLTdF6RFCAec+UxI9078TzYlxEmm0Mx1KTOPHRXWsYfQ1n88PA2i4A13ECMbniQpN9fUFQ5
On//ewj/Y/Bumnv02wpieFZHDP0Pd+w8TtBjXvXYcXXooVlg+aimqxDdz/bJHdeDF9Y/SM88Lj0MC+eTLaHor
Mh3CsE8G61Fm55NANf1fc6SXyB1x1mTfp55dm74nSYAZDs/FZ13RQkiuWRHn20gjQJwDS1EMA1QGWidI20PkjZ
btEN2TmGX+4NNTJnxZ43pQshpxatHvRib1Mn/9ecso0D8YTgz16mVwu7WBx400RtK0XuIJIMLrGsm4YdkGQjtv
4Sw5lw0hC1aSVrIvdT6915GMzb7dzSuhtyEU9Si1/W9Gw9Kf8g5NGzF1N8vidJXc8fnnw1oJrpMVvBRx0pDbE3P
aMo6VDuHKKXi1JuwVAqlokCqk1PAy1QITdsNh3B++agXc6DFDAT6BthXnN9IjI5y5pmDXmWpPA1PV3mnG1FKn
BXY7maWYyr+cxRrvqXSIFifZK86SpVwP7KaKU0iD44g8B2tISIjvoZPoKpKFON/81ECWqwJOKFnrsI9sGQ7s0G
Atqe5dUsD+ja8vEa8EaX1Punu1WAHg+1IFUtcIZYIeXd4aDYbQmk1YyCBDQrZDLwonZC4DA7vYHsat0q5tHSEf
8YpPA6s7PcX4GQNrXy4ZHvhpYbSTAzo8aG1HTHxCwDm5QYTH8KxQZmJkNVZCpXcrkpcmbIDpv9x73C1h4LQrM2
HB86Z6WhrE59Qxa21A74CM9EV41oeDzUjaVhF12u5+im0jkdT7L5KGhsrBTH30dhE8njt+8Bw32/Infq0ITKsF
wFgP0Rh/OSqrea+GGDioqqQhLsd5mCb0U7pIB1bIVU7Pufs7I0QpWP5VYf+k0cX42rGgSD0oPW1k+1DU3mdh19
VYajtcW0oCPU0LoxytPOHL6K/m1X5NZBqXW7iK46+XebiEYp5IeRXxrdT+dPVEcsNjt+44KW+YYyifQHIQ/zE
hDi3Z+2ty9Yd16BEkYw3cDCmMLMKMXkygLG62mkbdSVh5bFrgz7QKRUQRAjtaayJcSfld164V1KAxteNPD/v
z3cIK1LMmLrGBHha+e0eX5GdBeqjp1EFOEpZ6XPVWjwDPaUWZ8tjG3m10ci5s6xb0KnX4w21MjtGLypbKwvMM
av90xAi11pbVaWKGf0WnCfpn0NrGW3JMG1YpmYkuw/5cDsijFoRa21Z2ksz2jrk1aBcyLZFwZjsjEU5sIorbL
aI2HuJQ1Kng1DrDG6HPDVkekJHdYn0RRGunu0usYGrh1q8o14D0rpXm3ee+goUeJxRv0TEwohircp4SVA3txZl
M8RiYYk1Mt6YddcZhL3VNipguc1VJWz51KFyyYw1bms5jnXfccZ7LZn0E5+4hJpLHPk+Uq4td0fX8j6duuNn4s
BFzZP1dQUoN5k1Knu4nozeCmcrg3qkv2RjbSnSPPSBpn8SJWJ11RCacaP9Zzs3WFtZ7963DP6Hwjkoq3pkvj
f/zv8fU7edoSD8sNXZnq9XFho+w7gu4asinukTzMLaEuuJo4kR6o6cwlyVvZR9mJj0CVV5S6g001n0UY9osANh
vHdAqEnfFtc/I/Y1ZwZ4P2wEmRE12G3nVxf6GaWuGibmhtnsixCubWTP+n4kL9pQoMmxP5RoBuiZ//HwJsbZx
A3BXc0IzIpzhcsD2BvNVLx3Z0mAXV/8iS2t0ISjmJXWQogU3KNrj8rcmyFmf+oqT7SezcuPSeSfgYprsIe8rv
DQ3M2kEodSSb0LQvcPbaufSu6QzGIHRB1cQeo2ZwVZXYM1IwnUZ5G8AzB/Gv07+WEJUzQrVKs0QilL6ykn13f
+5EWs81qEQcWXE0rTmnyQ/uHMn0S+4N55XwrBaRir0cvaCPQk1G9o9phkrPn2k0Eg9U5009sKAJAeWSC9y9am
h4Wb40J/22BQDkVA5nDxF/Sgoi0Q8e8SzC+Q+5zeM5Vkm6qfqF0KbKZOUfDqdA32Hqr2x8gJoufh4n4o1462NE
BQ5itLj0i6+5gIBhHiz1dNY76c3Et609EP8i9Euu2i66ELbw3oUlniAuRx6Dhr0JLijWUQMw7+nomArQje8/X
nTp1D9+k3Njwz02EYnoY6jqwskb3RgN/3obeu0finPHS9/StBpcC3EV3vNpfr306TQ3AniEooIGes/dtrkLfAu
7N1WLEz2UmfnxZGkZ7TCrx2Mx7X0wMx3uRB4EbjQKZhpcJx9nCz91tFjQkEJ56fWD8/1g4JWZHBw1AjROT6VKi
woLH2sRk76iSRI4Rr0m0KX0yDSwCEPvmszRNgeaTwPqMGTlhf1kY36z4dJF/AhCo7B6tgadeot0p/8CcbVe+
JH9/C1wiJDFGmrF/xUhYFFfG4hyGjDIXlngol9aABiqCgcPS5FStqpT2kr1qrVJouPXEmXE9I9gkmqqEcjbkRo
hZLnmQZCZPanu1YDubjeBqModqNcFx8egbhS70yyntqAw31bnHE1FIItBr/sAS7QzNf2yhd7PV0pd8r3nIToDC1
+fmczjxZtGc+ATQ5bq7njhG9p6PL9gIMJu1w+Ewgv0UQzxIgtxCLjJodEYteQMEHXbCYZy7bgVxW+6ZmDt/4mm
DxsDwWD+KZi0yNaaW4+1ro0AQuq3o35cY436QifJpVu3SN4miyHVftdGrXFmpxmDiAW6uLG+0Pi0JaIYLCKL+b
Uas5ARCGB3rg8ftTG2ddcJWE7mfstml/AZerz5KY1lqXKB5v5ARINIKiXAMiw/G9Cxf3rie1FwJQjHuLpFr991
zNwIpP4NCPKVW4jCZ5TcRr6A00AMA51xrLboL1q6W/xpo01q8bGEPOzNt5bkDZDy1ft4CGkr+TpWEb1x6M2JDJ

EjvSLbaIaduLZDeRw+/1T0s9lHRC Dwh2a0a/7G/wqL+7JYiTKF822JtdUVVI9kz/2ya4cPMDoCat1A6eM1zo3TwF2wfMIMr5F3RsIj+CtEZPyzAmq9EsiE3uAMk1gXTMa0LidhIF/Sb6qYagpKudYjge0hX+8Qhp47d1g5unUxOB/un2F+nUKRwpIY4Ciz/hZD+IYule8SzvY/OwNRSqHQxOKYVXQpFs/wX0bR33jfSMa90+XDIeef3WPejWqL+pPgE4dzbS9FmHE230DUBzHTU2+iJSEyqNUxrDGBMf17UYoIY4ZcM+NqK7a9Q/kMoRUf1fQ8Ej79+7XJdhhJixkUf4mExbSprGJ/OuYzdnDs5aMShSc/rvKJXP2FxRJzSVePTw/XKbLS51947ZoSqSyY6Fg+EbPQpHGKc1sUX0dDcFSR9fnb0mF1v8ozeeIqxhqSa3zHUVQER3t8JCI3H1QHY5Ae09bT/AL0yhBcVaVAh603uKRpyR7TEiD9FtrKYSsNc6spqwI5t9KmlmGVL+185vG0bB50fYQNEz5T0zt6A3vfcAzPtmn5ZkTaeLHI2sdINm5y1arec2a5R8vfQPSFSEitjrzrEjyDLKacepCh8ukaPuhEI9+Rtmilin4LGkb0PYwDMore1Ac1zWSAakdg2j0uHNoihD04EXUXIAgGUdI3DxW0vNGfWN1Z+pFc4b4GDrwJ+1HkgIJodDqquaJFH8MkyY/K4WTmUztc+vkMWybn4tHiAnW457iD/CX5LU1DVQI/nn4MKlhMeHFiBgR+Ifw0qD1b0t77s7t0AX/AOJy3qCpiWK1y9+D4GvW41JMPaH38ha7ZrKrWtz0Ym5TA1s690mzkD/L1XvUgGKtdhFA1BRqxLKs3Uz3XR6KfcuQjtRSBdf4/xUVH2+q21z2aAjbd0pgfl+rsvDaT71DR/Ez90ikYkPYq0seC0Rr4cZtDxyes1CvjY/M3g7dMOTd9utdIIF4b8yWBpy6hG46w5JaZhYNl+yJ7cXiKYb55JdPI6IFAJeR+h14jnEtEB1YAkJHQ2iUymA11TsQzjafWbB09Mb4TzANE1KkMIcCN7D2/I2Y140G5YxeBrPV5z3h0we4/wz19JoURPAf2XEQdpay3xiVKz3mOsMemHwnWkp0xC1svLXvbmua2FYOF1Gt8YnCQip8cjnpRER/Z1ML0RczR2/ZtXub25cGLrnkg4YzueJNOKGyW5md8YVL8fYRruN/TctxScdr7kbT6MA1Sg4ZCAEN0ERv8iilQwfHPE70kRUwteT7HNNXsxhmuin+Y41Lp7h57mVpIPabRQufMUusoifZG00IhNUQbYzjWsowApOnJ1hAvji2ent2+6Z91wGYrZPG1MpuK06KNibSa+q92nwaJaqtib0gBcTyDevbYhIkZM9Jh/aCtsaxjGctbYJsjKw1qd9dhpjcxdcGZ1ZLQpkASvuZCcXpNsdp2vWaZpLV1Ix69hiyH99mFW4BQAH56tKLGizhrP2objyUWEg7LzuBhU78pEHufFerpC06WePc7yWJmgF/BWmMNrTsaqXBAdqgHdA5/uq2ZC86yy0S0gfPX0sf3q13vSS5AyUrAxuAtQeZLQMs9od80XIXWXa6HC5LEBTYry45Nuc1A8MZve6A0RBVK1JA1RASf0GPYiTXs6BXiGIEHMyKf1b1pThGDKyDCQ3Bw5rqWejwA0SqhJeUzqSK89uUI9FSwp1ShDAE02Wfrunk50NPuWVERXzW6bmu/iHUVcJs6KMm11/EXfy3e1e6+jRzEAoHDPcJaenFcZEDFb4x5Dntesi1UI4twUuLfeYOIeUWlpoMT4nAfPVGXeWd5GxgMXGGtqH0tGNWG83obHRA6ju+yS1kxg8YNdN6GTBDXsDWwRxf0VVY+N1E8+bxFj1UUjEXXeaFudQ0jNnt+7F0tEMTSmJSsVks/IBamieaNUfZMyuxZdnyk307MtQLEns5m9VixXnLhbcXRrsaKuJYRH03R5nbF/gFnMTx11HD+ywh5jaoBGoJsXeGGOU2+0citosig8kVtCr6UkfeDb8zuk4/lirb5s8he635J1oKSMqM2qI1QIdib27i1qeMybCsh+5UuJG8tE51KOJU2IGM0NAZ7UPYeenrN1BYOnvnTPgbxWculCNxAhaSFFJ/BTSuGcovkL1ow15J4qtfTG3rkNF/TPFhYHxN3m2qJcBB10yMV9e1150tH5Cz4S2w3837bPjUfPw7YpYsmhUIHqDDVCtyT1zvJyrgzoYvvK2F10W5JxZitqDCweANHIm6k1wrWBLXg0SYde+SqcMN41Q4er3Jd8FXH+5kYqYTvx2wRUNeUGmaLb0LoNW/rRwvLrDbKP2aFB1LLRSRSLHcQX7Ccmvm5zM1JPgZiDqbtCylhwmDEH6MZ3CEag6C+aXiDdgo40TYFRQ4R5h2pBBTjxYz0qY/JPYAkfDiJhf9UNNbExbPZkhDVli6xrfdTaN4dpox9tnirlW92vds9PiB4c1PfUQMH8GMr0LvvWzr20qYQdtWQS057iTMfguTQYx59i1L8EIDYed1TVxHvXs190vCQuKmP30kEMbdLShjoTB7ivk645K3GbNSKRGKhZ4S1c7JNjM/hTfsBe29VXyZQchQnms9J3+UG1TN6HROEr1/Q82forLgePuhJXL9orimkeqC4J/wBojxx8sYXA3YFkwqylndpv9GcfwR/L8vWqV8geMrJ4f+DtfP3DNnSkPcmq9r4eYEMftNb2/q9P6gbpdLPUYqAUqUk1PgayCHF4ZMQfD5wLfMhBp/hMxTmPpWWY/nQ88oIniIZMjrs8bTmshg2h7a3RIHTwLccIKVV/1k5Ejd1r2bCzE31d02QpcZt40W3dAWDJRagcZ6C2G0yoADpZ/zYdUFEQQ5WpDSNvE2Z0SdHifWV4Hp2Y/Osy2G9457kHVxuK4k38zbZN4kdDeD/3QeGIMAh/5xfXNNFMYh4e1j+QtDiJw9buo/Z/LInhXVaSWT47T1/VQrjDsWjImHcsSfhgRqSt06GG71tc001BeM4+Ti8Addhn6LxQ0S2u63BShRhgDwQ58J51mPVY3gXUHHS9irSMmenbIPvV9jeAfHfzYEeCF+zv5mMd0XEHyAtNporMk7FkDcg9i1iFEf96cvsdLkMcK+Nmv9B9XXH2YXPeho2/h94R4RDVsHmaocs0C1/GG/kGMuUsysiYqK1vFwa1Zq9Gd16AtgW9iHb7N7LLwgK26fm24Mvt4Xsw3D0X2CrzV2cU5240iNtKwgIGW+7tpUIc/V7G2ihfPGvpQRY216Dy3z6gF+cMnmloppvU5KDyn2d4a3Hg5bTj7BBfGQU7nixR7EkTzeI406Lpa1Vt/zMSKjZGgbnyiuCSnAAu0zGQ/0smAIflkm1J6buL7N0sdqiq7mfUCbGQ101z2oghTF+Suf/KZjXTs+7T1Qfq1gnVzJTY4KhyYDzASTFqE/SfaRo6WsWCuDUVypDwf/u4VQ0VYwq4c2eM8s9KaZL+tdYE3H4x7rQ4SgvcVEtugp168a6nxZNonSKycz+rx0rhILt8NjvYn1v5cZwML4b+66f4hNNDgFpEn4dUkcZ+PL7wTdWJhVV3iuzXeirPI5KhysGdeRkYGFcb8gmXhd30p09fosaeoYJhhMujm34oCpLsoPLk2WhtbRr7HBClzscaYzWFxbMibSPDVI06V8deM4xnUVIL1GjinIn0ZJ377hTfb5tFEHFqr6HIubxazAVfqI0qZuoyNAAFTNuyw23fo03+DfAWhD1jnB1DERU0AvPSQX8ERc1LztHDw+rqFx8EbJ284G+veCbGPSkyIMutSvbchTZP0fLbrGpjMeX+i8gn8LtrbHQiVr9nQKrjWk8LQDvVTicd1gDNpuK1Rj/WNfDXy2a7bfa7/Ht6m1YZN8cOSI4nWE8iVdWExNYKn5MVDUt1NggoM1udmsm0JICSuU0dALGDn10tm9VYUSJJ2QFcnUD1/bI4PCWRuFpPD1X1TOIBeIW+Bvgscw4DBa96GBfLEd1L9p2+LszZPcamP2I/Q3exc9NcPMndbRcphtTLeZ1Iz1NAwXeC1WuMWvaFrKbf6rgoJ9w7mC11Q/N9Kt45Khh0CUzCJDXYq25jks9FQE1tAH0DT8nX4bHh0FAW1xgDXc8Fd9PwZKjGT1/twW1s7SixrV502noL7AL2/jWbb0WRqZqI3wiDcnF990PVI0as7nvh/wd+xUqr1FWRbpR38XRuqc5rZutXxFtzJQenQ2qfK0Rs585AY1linpE4w505djnYTVxkHLmwPOFFRKMMBN/WCY4br2vPxJhtYj7iyk12A+CC8gH2CzzFM+x1As6gymgjswhS1zPb1lCJWKFTYo

3It85YpH6nppMVGEr5gAxQ+14bAJTHQFu9TLrx fVSd+1YFGAs+1D7KH3qQxf7dfeqZUSFv6u0930/ZT0f40Bog
2ms3LizGi82MqJWW9TRz1qKARzW54T+9zeyj84vb5o66LTt6yACfUe1tZKHSSX/fnQLQGSFEgpFgPIz0wU35C
a1UCSVZ6CRPm8iSSzBNqlt/p6DXrHdSkt3JgSUN4u098UIjEgwjlwNUY9MgPJZIen0f1zmvpaHuVGwDGEKriPs
H8EV0fAj84RigbGrqtkqT0QZejl4B2Cwjsb1A2t4a1ljkjyZBsKH/qvpQEx0qMhdZbmia6xHMLNI8/B55cZY9
6jwtfLsTyDSrTgg3qC+cAkcL7KfBdFTrJUiajohC0u1zkjdntzqf0YngCNBmMCQTSXG7aIaN5rbqnjgsSC03
TGUjgHT+nq2s2AmNzxD7JtNM510Bvhrlw/ALmB7I2W2yKxr+z6WotbarDy20aSWLrTn93nungs4Ladj0rk+s9a
XMd0ojgkZ94g5480B2YrSAIkewXNEhG3mV/gRktT1T4BQLjsicwL/z1EKEU7eT0gBCC0NN9CgDDEpeZpxusgUX
mh3YNR4vppgMJ9NmPz0t1IWkPpi7Y0qMLOGEQPXzHUPWrShafE6JL5gfk5d+updFYg14qxpY5v6v78anLf6g
tRQCLCvqtCsPvQ6m+6Nyubx0RxSLPHxjgi5+qH91tfWqu09S+zJI5FItlizMSG9tCcKrq0kjUDDiTf0vEv1o2v
EC+7C0v5ScKlmMuAMBUrDI72izoxmS8msrWEYZrmWWtxzSCeWI4gnaY/TtsjfcVj6ecMRiK6iDYKhbhCwfsmvu
+M1QilKVLja4uvJkYZ8BMFljaHCukeQTGFMRIBEm20US026VJ2Br0Dwvsxlp2RTDgJboummRa1Xj8w/c709K
MhUvV5K1rWlqsGRpl/QBLqXbdy+CmQS5z+wPPWEFvudUJn7uZRQ0+upGk3/gXBzcBZufmM0J15qazaKfpZAl
mddRd9zj7B5CpS+VXWHUI+MWVa0YujI4D51Fa9wbLzHlol+6cmaGzo3U23guLn51DiiJr4m2B6jWdtCr19aQd
4Ud8DRgWYZQLgvTgtUZVggR7gqjzUeii3FpjAh1dUJ3HTsz9Eli0bSgxFZopGZJImXhio3uTGnHhCvrprERke
Gz/QecNLx5SvNppXQ61JhwQ1u0hbqDk+l40Dki6Sb6wBvXluPx18nKjH6NFru9c2Bt1lQKfNpAhoC/ONNfyUL
0DJfJNYndliXGeRNHJtPTPrfHE6R6uWtVETACURbtkpE9fVYpUTOcvk7+ybe2fgkI0yWwA/dBDtm2+EBczLm6
DLGLJYBcB51o0oKENhgLK0gKZ3/oBmaaUSb2jWiC6ov60iUH1gQbSe4+GKDHWtSk7luHJLPEpAyvrqYfZe4gG
IXRqv1ao32+cc4NMXmd91kZ919CsMCgstjK6E7QnnoHEugMJbR04M+zKCUu042TtFyL2CzJR9wne7sXj8/nNj/
2PmeeFQ8QLBACdfgCQq/ItJPJuLh4YIGf6ASxLwBwZGhF4fNd/GhjQE0htuNvJ1D8HETIfUsuXZHBZ0zSn74Bb
kPjVQapugJ78eL3dTBxKx0/7Poqhn1zxB9ymv2fC+Cvnk6hhnR4EVkPN1+t5HPSaQmv0D4+JXK5XLHZfGDQM1z
/h7dqBy/AvHhPcxFmJ93fVP96LZRWVnQYDnHKTLR+uHz9DCsdbvAP4mm/g91+derjho+HGUseR3w1PRh0lyjzh
346FNs+YHBNkP8GIckpu+8XKZrReXCh0ZkaknG1dA3n9GpdPvL+gT9SuHppyo4mwCd05aDIPL3I73EfM4p119p
uaC7JTk1a3EGIVSPxboBpoH1Mng8fWlx3y0rCunvVarYqGTMBMZtNXKI4xLWILGdxKWAo9kW0aPX/c7gQNzA
sZ8NYTqJ+Dp/rGHqVTKqKAKjSK1HCVsM117y27stjwqFWQqFn0IALHWrGMj6qSpFRbtvVu+TKbPlumCf40at
fJPWGUXU4K6UqJHpLkHe2ooN6K8T1ndWLxNCjfmC+ApI1rGbxPwBoZG8R7rJ0rcgcxCEqNAMhs4BHva13WDoJ0p
qUrNCu76WwAs1rMhebXBvmKEZ/9QCaVy5lwRFXv9z2QCMmV6Wf2951DUKJ0SwB4ItvpHfaJrHvqeVbXaGzr07Q
P5S2mIZ1sXAubUvTTygL5v3rmViyHzssIgpMFRzyYsU6dnNM/g3axWTb2X6BU0qFMHMCQ1xmxzISLubBsAV0gQ
eUKz0S9vgufTqdeNXJyvvUziH1SsvHYBiN+Dz9r/7WGBKTesnX8zLdKsI7yxC6p3nzFBZQUtZGTSq68po1EyV
GL1wDLN9yMN4Ye6n5zT8h4wQQCXWzHPvsRnDHLLDGfkGQ9Ki8VnqZKSMCMp0FCr0ivxGU96nNeCs+0TH1GffdB
1MJre/yH31di+uMrI6nH0D8EKejSj1xpoyRzDEaIt0AtVZfMv6sRz0GAU80yv1BtzfRCSwdBqbBav6miQISaFTs
msJhXJwbD1NVP1pPq/4NgyuY85FKbwAW8k0H35CCn93R10kNV6muVxDWATXn/WwjUqxHHJ2C20sP++5ZxEdvAu
wyv0BR2vGQC8VEqmR/MdAEvKNcdU3U5fUzaLpRJqBpm/gH1hio0Gp1MTWE4GwM9NEJz1hyS1NintqaLBuVC+YD
2hhutzHc1oZ3fWvP+4V2d+V2xpWdmsY04N9o/N6KvGbHh9+1RfdR8mu7LxAmvaLL1PSS+xCJUWE8420RMcKry0
jD1zdosPg2RkZzG5RfUwC0+rs53XndphnLsXjTR81z0gfpAprHL8WOYMUkkfih18VMp0p3TdkrMNgFUJPTkgy
zxbxbv3UD2qWrdjbyfwBVW5BxnLuV55pYP1t2zdy1IUj48xQ6E1Hb+z5R8aPVaElKwuGBu0Jh0KcqpFn5mWzAH
Qa4eW3inmkzKq8xeYgAVzwBxXwHUJXBHTMTy7qgcyTgo1Uf471j2BbUL+LK7L1vAAu5M1sYyWm/eTHcL/XzKta
HsvNqaZfWCsLso03DgxSeroiRtepMWfVfYRmtxyahd/XCIz/81ckbiz/UFMd+mLIopJ3ZB3lcs+yX4prjpsrj1
/cUt9BD1cEjTnURiBu/QUXR5QUu0YkLLJrvN2wVJM6y10QhZnqf/bLqp/TmbpXtc9a9Xf8ALFhg41mow198cZE
kAmvYUyNUy31zBcNSGtNuHdEmrDAcUXLhIRO9f0RjN5f2fL6KU/bCivJ4+PLQRmArgkUP7RKWYt4s1TrnYXwA
XcgCGxwTclg19ii1QDvY+t9M1CVURc9hwj1VBhvDmdmoevcWDr6FbH+GZ3/UPPnX55dDGeGndKE1ZBSD5w7hNZ
coqRgxHtux3TzNNuBcT+SowPCVqMf02YjDwKugYxk2hQM1RkZ16PMn8171PTgEgPEx47vysbSiSwrVmU8kqAPF
0vM1060Naq4aCYH0tkNz4odduLQn4DYDvVuIKZrVtD1v/Vgb/DMGE0313rJLHUEXPbxdpVzH8XBpKy6VAx2QZQ
SwF1uT17R4qgC93a2aCc9gQMT35fSfuMQHhXj15mqI91JbtIRCCi2qdGWTAGRqvwcBfMvFvTzDT0u30tpuLhI9y
R0Qugmu093D49UynjRW0chUJ5mY2c3DFF7fU9nix6C2adqDjy6cNuX44Lichgx5y5PpMjNQm5L8wTMTkoEz4TJ
r7fZNcumJat0MlaGxw1us428Mi4nBHsC6nojh7EMEYIjv1h8v2XBtxCS04cLzzPbGT2CNM3jtRrn9Pgb1QXu
HgFPEh2J59fPbbaY41eGfijmVcZ6bxSIPUxncv/Z8CfxQGeoDnKRyj+EGvqmzF/cT1eLo7t0N1vnj4yKI489Bv
JwXQI5vqNjmxnZb7ZzTdiUMgKxj7Njs2Nix7aRj27Zt27bdsZ00bdu23dH867zCuZs19QB1WVXfTSm+SrjFsD3
s7W9bHbp+4DvTdJh+0htvo7uy7V77dqydcWRICc0LLz1pbvquYEcya2TjbNawpzsh1TmQjLIFphjp15dSg3j+M
aadIUabUkLWYitp1AU9q+CCFaEo5tW0VnTjik5JN8nje2DMnzFMk5EM28yR+/wmidVLS+r5VZWWtJURZXkMa/l
uhavnQgbmmDTbBZt52zbaW+yN8TDOvQNYBH7MkA7ZPw8IHpSRMZhZohLLt57K47JmcGd7e1Q0ZK7XK0SUP4t
LdKF/shXc/gK3HsKEdZQ21gH6J/9EmnkfOUztsAZG0rVL1hn34/UuEyVs0T67xoXreNWRU+B/1IYSYbw0IfLS

1s0p3YAcyx1znyACUCn26470wvjuvQV/AnXnVvD/JukjbExhvHtjwMIgoAlCoZUv1I5SWwUtKnYlnk5nrFOZmBYXtwgRxRqhQTv80i44cjxiu0tJNkQreumFVgtCA2WU61blamQ+dLf9m4EahKLY+Lw9yIB0zuKBG4UsktAjDxjnF6tcrbmFpziqL0tfCgAqPQSh8e6KqTb78SC2+MbjeC61wBWxtaUsFl+Hxv6Ip/G8XZG89QkcLRt51bGeFuJV+1R6/vpkQvDg3JH0akc0+MwOecUwSz1WEEDeb6UeNGxrPbyhRqnPua/zY0716emwynGocgoopzska04YhjbWTMuderP7oav0EKbtQbytBWuXLGacBmLaust63cgvrnRidkXGDZiWke4++AwMU8EGw88F32YIhs0MW/vqb64arKLFPBvJUVWoL4W2Y50ZK4FNLvVLG2sqbqIONK2d0BF6qsQD9Hc4bNjoPZP8aanrb3VR/Dxf3cEg45VNA3TRC5s7EZwpKRyTgnAR+/7CvRURz99PvaYfe+1Gx7jy0htjdgIwOP8Rb17UfGy7i5JAsPX8Xn4F6HHP7X0/b9z1Qcy+JGgznjpo9JiJh0HzK3CX73zEJZz0sZWTzn8ZADRahylM7gCMXL80enRJtcQbgUho1ywAbcnkzu8yTehOfBHKCKgD9WqLVk4BqMUxf9ZbtGzYAZdRkWwFjNQ1J9UP8Azbtx5Lz6B0MPlpo86FRNRZIH2m803y/4rq/ZcMA9f/LcIeYc8v3oLNWS1U0AtdbyqbkzuWCZNMrATrZqEZ1pbxI7nshUnotMwzJDV+qi6UTAcnz1scduIzzBNo8ISVPHqfhShiQm0zdpv1qPQqcweHRbLZMm8MBazeVPSAgA1FWp0QVSojJK3/jlwrrgNAidwk/TQuF9B962mjTapFY2XBjk9os5T3Vtk0gEr+xbgN6AcikhkFQE7bgMF00BC0DcM5V3wYGc0Glx45x+VJnYZW+657aNi72lgU80fkyp05ujf4t80VTGgR7Gyleb60iAzEwa6oDeh1TH7ICTZnmjcr+GFAbqSKU/WawgfrT7I4wiM38vt4POZXLr1B/3dCrXR8g9WgzlSvebUGUW6dcSQc9q/H62WkCPjjmppQASGubyFIIVfKhv1Ts16cmTDKISSvaY3Ea3QZ58Na9ZyhG8tfm3WkYOGU1/KhXg75D1aebbv9Rfp17SmLxcZN0v+tsplJYwpfM0zif+Krd1TzxiUuftGcuMDiBLsjM8mR15Yedh70ChhXq1pJ4K3ynaJ7wEb1rgovUM3L0ySCd8p42S/ls5HMuW/Q8qwWhtzIWkNFajtk1HGi+iEyu2BeeMyrH0ErHUQr1KSIcMcYv24Fa9mRCrt7Guji3fkWPBX/PIE0b1Y8uhylp+D0AZV9mnQYbKPzDiTuait0JQnPiXHiqP0fbu2w6JxoJETCNPGHvI26kLQxK6GBT/+EpU2CarkCieMTsyswjNbeTw89ffdKeTiKc8voWGKv06UfZer3MFnnlfakcSorhWjfP5VGHYSxu42/du6CI7vv9pG5roscn7gE3XG0vnn0M4u+m99e0SNPlXSkYhpNISxoTJvriU1ZFTPQtJYX8aF17fEaWug1kXBy+K1R79Y0yAtWoHrtxdvsPDHfga14fz8jb8Aa+RP6u/cA51cf4uQQ20D+bAymN0Zk7RroYA75eyoDkUE01A2Ninhr2pwCrLJ38k7+1qZSZGulaJtP00Drm+D1dPSFuW97qeEsoikbwPiYfoGhN24vrrNI2yvH01KzEoNaVm+GVzaMSH/czzvMWCeSQ71aPrZ34f0jCbAZxBm4uRLsebE5ny8w65XTSPTGfogw39GA7M1W9+prtHkRNvTbQhjpDrQUIEMI/UMdHh+8nK5oE8lshlap91ouy/XKmPbxWbeBw5Ht4H1hoU4cfyeT82BM0N1xwQ+0ymqkJD0FYyu3xTj0A/AxQ9nac3KKtdhAj7HtIHq94xBmhb2k4V9SP5kf0gZRzkcou6Zf1cmg1LYx1R1bEoX5sk4prSrUX0qnYfMLk1fkczjtkJCoSNdn+IeKvBkr4wgNz+F2DsRkCtmEvIjknAF72GkPdFE+ziynUiQD0vq20mW9c94zHeQMjojaI1CTa+NDmwLTZn0M4WtVGoLVWGIeMB74XKEvELGrg0LC16F4LtF5nxW2K0W5T3oJjNrzSoSANG2SksCjbRinh1W6c6X10nFLy4EPTf4FQqQAXNASEXoC8FnAmwH4jdUx4k1IwXxXk9p/3svY3NETp0ABVaVh2GZroBwZFBX8iJdaSBw7r7nExxf4WbT16wTuk61T9y0Hz9VNV7RruIOPc1jCOYZQ6Wn5vnTvQhsjemDcRe/36we6QkUX/vjZ9FWtUFOFLCZ4cK1UKH9V9LEDrfKaDt9obpsLmHBbctcea0KnXi3PCsKGR02Ue9kAbhRmIyB4huaDzHNN6RGMxhTs61Hngmcwnwn183oUv3b4eQZRLb85tsiZPsxuqmqJVsZ2zqGvFF81VEx0IfMiParg7JGLs8EuHIk2B5k2uycR8CvDXChwNOKnY+XNYpo5eMVsfrftee/rjIPX3k8ss8n0jHXx3j+AVxoMw415+J3vvDpE4mAA59B5ELlw8B6LRS7Cke3uavfqLerXfuKchv6DgHYXG1n8k8bPtR8x7YoP8eFYpKB5trg25fw3CFYT9jeMx0zPkT1LhCmYz xuQ1Gltm4yB4d1EzeVai0pHkdzed5o5T/hjeLGM7yte6Z3K1ZC9QGwCy+Iu093Fzg+8arayIrnHwiMC4y5NODOm323M8jH7juj1LwvgV8cPFBwwK+bkzwMd96p8HSXQfbPTUU29rFfaJnIIPQnu7wbX93pKDkcrj9guJnvxZSVnmF6dkTv3z5iSrgaha20CuDk+gSB/n7923/1ImTWw+2XBhVG/zE4ed7estYFMDs/BbRdUE9yK0MDmszJmTUL2G0/YgF+0idro+1UCbc9x2W7c3eAqDz17g0KhYj91cYdg0uH8zbGzufzunu0jcEdnfZVgDH8m4bC5+l/Wo+HzKeae2sWg/Y7oRy5ftFgQVoiV/Iu8fTV2gI13rZN7G6L+F2hc1t1IPMVdckMb096YNe8yKUAs14eGs2bYtUhtPPUrTIeCnypGSFrWxYaldeoivL/kRp4TnaUgewJW5KJ416mWZGr1NPIsLj1nw1b2ZqpEj1DKros4AVUvKw41Fz1qWr0SHcLgM91M5ZkTVBUEfEgulaEWWhIkDPsVTHUviVQXp4Jxsilxub1+Lip72Led6czBvP1qUMDMY5QbaSSrKjfwgIF9jVBXySop1D4mU6MbPqcC9phk1mJv6jeuhwFIRiMtzhVubXjKzNYCp5o4Mj2LU5ma0mt4R6dp/tEZkHHq3phpF943wubIepD6FN/rDJ9PZ8QsuUPI0BxKyv6r0zAG1PEimTkBu2bDzHdeDwW3hUA0GyRMEibzyOxKS6ijjkFbB7uQB5a9X/OPB4JCKa+Kj8NLbwhEcys1zVtgshj69wyho7hPJ6Pj0W5uKLfztkZJ1T1lchhMt84+SznaLZYYwMddVW2kxzFG4SKwdtnEzyJab7+dQDr7CSsNyRDPqwHX9MbrH3nuUkUnjc13YQqYNTgVal5+WWR/FdFc0cK9ku3XP6h0h5/qFBgXJj0F2N1u2R0M54sd/o3R8paFjbARuNiZCjFsh9186qu3TauaHdsbapy8Yi071HhXP7a0h0qYqCJ1oEXG5StxfC1zJBVe8B34wyH1b3Rjph/NT0E9GVnDeq/sAn/BMwYrr707R1UKMpLzdxNwDwiupzT0rX09+mj1rH1jQ58v9R683eEJwtJUJp+/3s0761eQ+TrA3Vqym0hwggd5yZDpTSMg787WPcvJHIMTEWb7QusG0DZ5vkm1sQXjq2Dh1bSsP1uisaru99aaWwIMeq8oFRJWRV7X1XWtukxjsbkAysnJH81c0ZMKP9FbtI+P+NCM0sdZ1qsSjUEQ+3NTe1YPBWLnt0q/SqylC6FTG6LasbmoCCgkp6U0mr6dPTGBT2zrAr1mQ0xtWXqsUctx09uoF0ve6PiV5d0ww5ueaxBQsvqc0ron7I0dWSg9WsgEBHqk+gZPo+WjQ3ma/Pxmu2vnRmjkk/yZjdtnlSijwy6XZkUkinScAnctXXDdfLZ+uXB8Zu3CBRGhbYzcTdpQVI6RrqN0JeMNsHIQmRfBB07nMByI4eUcf1jwz0X21j28UnkMFPs/f/IP8E3T974E4BmYT1ApG+crVXV/ZwyED00MmZxAnN/ BTkoNxLc4E4aNrZ+IraqYJBnSSNIMwWNQCiza0FRCY5FW7gDFimyZvmi

NJkUMEkYv4j0mQX/PUR80jkwlYyEw44W08euMOPMNJBmWJoVeFxEmwdW4xIjHr1hNSbn6FC8i81AT5+0geORN
YEK3ABwovX4LegneUY4FgMEH4wZX26p/kgDm0LeMkJzfisoglxU36Fy1uxRKzN+f2pYyyqNpV4sEhC1/oZoZww
2vWgUhcoZtzWqdDQGQW00u36ZmeS3d7m7JGFL+K/D+uuhAtd6gye9qR57Smx1GtvfWEq0F3htZCxFZhdQ/n7
1Wtj/B8RMONOWK/vXEtmTQwYOG2z4sdGzzbLW7GrBq65Ycquo3ZnSrTJ0g10740qA5fL9bFnc5LKqJz77yBFk6
xyP8ceKiPv0CLfy6NpW4Hz+JA134hIx10/m5a+zFNiQVA0dMRDyMk8jMIYx+E+QiCpb9e3p1cYMcfxdL41ZMKH
0gJAtNHTCLnj9YQixXKF1WxrVSm04LJuksKFS3N/6v3ciJJIf7yB7jEGtjJFSV8sXAEs2DBI+5z15QdXPrT3z
bm6kFCmxVx9sMT6q7e84baGtXzumtUU7d8KbBdPRLjwHD64NjirM6nIoCtDAzozJmV4V+EzAcxafOUfa9q+ai
pdfTxs7G5h1zQTA3SSrCGQhP+auRhxi/q0iUQ0eCnPwCLI+Kdf3PUvfB7Magfcj864IbvlyIdwhJyF59ihLCEJ
zn7qxWjiPkQ6RNR0HHNeU5SMVgBK6Jkc8Vra2N2dnSSIfmFd1R0wTE10laQtUIiat7uKsuu5PkSVYjoqv9g3Ak
ikz9rT2/KQovrQpan83LhDN0oG5C5KodpDycaApvzzQRDbPx9b7R03ZV2Cm0LLsF/sPjultJmtsCL7xaX2V0oSf
w06etu0RrThFU8Avne91o1tHS26zHsSCFQaIqRZSfswxTC2U1GdXK3Wu4VU/G/J97cq1Ti4+Vw8U5JYPha8Zi6
b+k4wGH2k0uBT9v1xCWWxSUQUsmeSMH1P7rDG19zAcP2+FzQJxjGhpC8CaYYoRf5D8/XHFN1Rmj9BCpqff9d9B9
ucDTozCYnz0qfcxMTB5MMb9BpWACInuxek7x1WztAjmBCKjJz9aGIY6ZZvN3QDCzscQuwkjrfv9y1qarYtmd
/Xqv1EMrYslru3bCl+icKv3klsnn2Mk3Z7uBkxSVfp6+uvkgj9RfZD83Ki3S5haHiibMrUvBD6iIycLHn9FfS
S1Kb0GGW/9gzVrskJxqc618zF6NPI2B91QtzuCz5o+1wBPyM6d5y82DLI50d3zDweR0ig7oey6bJolwLIR1DT
XZ0sm0/gJODH+VZicsz6b0Wx0d/EW1n9evL7UNYUgTJHFwaE27zy/CoLIUxWAsLsasWtIosAzfCsUy5vLw19h8
3R/KrTH1d/dBg7aZKVDxoopfGLDpdgJs/27Fs0myCbWRq6spUxJf+mbLTAs2F/hG+dq1/Q885DVupB128ppXJR
531PSNxji45v0rwwRp9K1t5kQntL0H9Yyat6B1Z/pQqYulu/6o20Rc7zRrU3Auuq5Fq0k/y0qQf3dJzVI0nUv
DSS7Y5L/RBOIWVQEUsDwLLQZHOAR8a1EzzCThfDDjCCQBuYzpHDBESJfymY0hwtVYdaQ5F1L2c0DhKyde+fgIm
jZ7hKX1DsLyC8ry01C5aPHzrHxzYy0ZVsLPdUttRae4vTKcQ2z2J5TALIm2ZMKjap06U9JwfN1i6a7KNPd/Wwe
JULs7eZby+EHCXV1G1MC1vA4LH0ZCtRZC5IMD2J/C1MoKgE4u0XY4mT3MrIRGQGvkzwPwpUcY/HMOUawX98cLI
4km8cNBU7EQI5CsczpNE7AkD9TSujoje/kowtFicTX93JbN9chjWJnm56TJu27NMFShb4RiIZxm60n4prfbr0V
NI/9wULyp5wLFYIiupVMXpjtv60xM/Ro1I1Mxcd1VzFV49L2UhxW1+bulk9cDw+Tu5Cwys7kNfxWHeptTjoJ0+
/rx12rXJ6/ds2lNKT5jyFjqQJGSZKoU13JrzeuyWB/zW9wJTPvdfl+sdJh0zKHjXN1XbBXSC0Wzq/Dfnyt5ef
6jkj7tLVZQ6ExDGYbhCUBFEPygP8v21gKocbTBrqHz9dFOQYA76qsH+pWwbLkKCQcfWFA3J28UvbL3aVu1BCUH
dSRmTHFm8eRRn+F5ozp1i3ztEiHj3p0dpUq59wd8J5msgF9bEf116ivoTwmMVOQI5geZtXuzPspB1HDd9XYJDY
v6HqkJyz80JtV7Dqn6Z6xuXa6YxS+QWtrdibeocoxX74hdA1H73Kn8GIP2ipkZCxkaNjegNWI4XijST0/fbgt
LWXmMJy5Nf4jbTYT7U8Ii1GftDjhj4mAWbuR1KasoD1E8fV5UHEfp0+CREqL/qjhB1PMk13NoLdhsg7MIxBE
8f+74g/9hA06zG5Nnlx651Dr06c51wFQDHx/QVgUt56vt9DEjtS1U3/HZtc5gK4mG1HWtw1E2fc08sJctLFU9u
X8ZIYcvpaxp2DXJPOq05ow5erw+4fE9ifaEV43r/6A58M3x+E7JbJVHW0e4xI3Hk9mPQwLASgVRQLvneQifGwt
PyhKm50CF0a0t40nA/WJX/N90X0tZ7SkqgCYTY196o9QvnpjgXA95Pq0S501gJEB0J2WAy9pIPkaqI7uzRJEaU
/iKuJ7IMc1NNJEFfg4iReD11PxcM2RCMsIDmCDT9jm1g/UEkeWPBhtk2Nyxo6kaVT1Y0ifZSkQUm1F1eTkD2pm
szaEYZXeVXatcu90PEkT6HVbxuVX5iqQ1fK8ErKZa7FsKNES7pP52NGb3K0ZsKdDnbflv2jCM58Ye6P0M+YJGe
DJ4VGI1MxzFt2RUY65YvEFNjVhD6aWWJtSScFKM3IrxuzwcI961tde4QU0DXmi2hc2u2k+ea0iB5zxpDt08voa
WVtLoUadG7FebRHma+vjGCQakPw/RN71bs05swcF4WUEhxj5WxV4G2tovt5RTwyh0Y0vwmB6+1YokAsXA/wxe
IAyB+BXsIJtTLEt23cVyiBTwAGGmSW2GVMAiq0sBGHK7ru9pfkea1TC/r1h0K5od4udS2+hshiE6CvFGH7D+o
XJShBW02VbB67MdvX0mmsAS3FezWnYDd1YzwmKc6ai+ln4L9b3LQ3UV0q3EDuuyXdfIezF63ZR/xaEYhptrPDB
tmysCF1LTbPQaP9s089pc1RxaecmvS9d9KGu2UbQ11Yn1xPjmnY03H1S0mNNI77i9oHk1eeQR1A7m41LWIQo9R
+uBj9RFGft30VX4022POPZ8ree1/c3bcW7CSsz8QIVtUQSLWx00hQbGnPx+WnRKruvwxG5YiHY88H3n1wi83Gs
qep861VBNhPYvMC0mo9DcPgcZI/k6ehIsTG3msKczlsqk+jM9lpmcDJNNhC+XQ/VIFEqAN2tt404Y8/2+IxC
Le0Xg9YHG3zoam9oTyrmyTU7Uxybp1cdfkCvGY6CmdIpM8M12t34sRCDXwUhM2q7dEaimPHoB99WXjb/V4y3J
CnB11Uv2JIY8jytd/WFs0KDLLE6QEL1184gp2ae1Bsh5f4hDXnK1vIVE+ordukeyfrm8N4h1omKmTmCVI81Zvu
FEF/1mpKyZ+XLF1Upm47iFIBLq0aup01i6ZEGd3qGsb2JSqHP7hp6ohLS9wSR5DLk16Uhp0MHRfXc70h9g4HeU
NcB0N1iNQZSGs1KKC+mQE1JGZnYdv0o/0QVYyA3lufk6triP3k3WhroTkn04w+gyPSiZuiFigR+iP4zUJtkoHF
6aYYDZNe+47vL42kqwJ7cdLS84rjqSpg49NNbyfdsPxAV6E1brZ2BWNV+vcnh0n4DGOKVeZ+XHl+ifAZJPKFmz
GJ/EzvWJyd05eVTrnfUfBEEGU5p89XqsR0bov/CvVXV6r92Kv9gqIN/U7hJR+9iXUvr7jGIucHMDkkUhmDZKX
eytxbT/ZGAdvZJpfkKBxWuIn58zxbBGMVHAjnkqhmxG87NpQjExdIGsRrvddfbEwvEzTahk6PyN+HLC0Hn3Te
HP4mYtomeawq/n7WX3c5AEf5edNxt15C1D3Uqn0qAcxE0S2Bx0I+qbAhvo62AOYlqWAcI7ayuyZBY72QPZLE
16ano9+nBP401YyiE1dPTdhUDTJo5MKWB/XELH+nYBoNTRCasowYPFCeY1oRRj4pNhLKUSGkIfuQi9H0QYHqhn
FAuiIx26QjKeDz/TxmsVQLq4unTtmQL7+vWSgkBj6Z4DC+k03RN/CjTNu2viSeossJPEwpRraXk08k/20WNuQ
cZIwKMi2vqc3T5fMgN/exfG2Pa1/T1CcJy9ZA5k0BAatdc4xPyewGsIFJ0Z9cjqqsbd7P8gqMD0rtyFnxXm8Pf

vjszKU9DcY4Q5QazCwaZXZRwgPPtTCXuF2+/hzV0dFVLfACoMQkzcgIer/TxN502K2uQY4IcCb0M6KzHf5etmW
oYVG51/K7zbVC9MKE+a9c9WrtZh23yq00UGcTZAmyUvrQ4sEW1ptmfDPnYDGGvU8Tzw7ssUdLjSighHu656
1cfnh2RAv151m7L7na7gbyxEitJSVm9MR6h+j0Anc38UnWXuiyvCjv2EAM5eAuW+DYgbmXZUAeHnFbSExQfFkW
aUIOT+dL/em47sLCm7M65s5dxkbpczIf2eXcfyJ/gDGibT7Sxotp4PcQQxvJKchrjctRoL4UWfetLUioCyzkF
zgsMIwUUXTs9s3ysu160Tg31Fcy2DPZ8cuydpPrfVQdBOLW08u1ExLarVVNjhous/E9CWA5Fe2LvgEIzRC9sE2
LghcCzxF9xjq/dSgsnr8xnx5T8kUT11/WH4gw/prtrFSc1aos/HHuwNQt8wL7SvbdBON5w/pwC7oyN9gTAm1Sv7
bJQ84o/ZqL7vV/pGRoo1V0wuQv09tR0F0GCV9WT4g33mwI+j9nT2UyarM+/AxSNaIz6KzSv6j6rKD0BpQV/V0W
p1po8fbMDW+toBWH0iw16kYE+5jKzn++GCXa1Ah4CTtHXdPCLLeGdiNpJm03NPqHD+PhG4pW7TiayHJ8Y8Lt8A
fVqldNqVwHqKnBa4d++WdBEWf2FG+r4gAbmJpf8RhaEDd2n+PFp3mf0xL8VdHq+XFUyZq1daX9Amzev1rrdX3
vvn3MDcYMMnoyesLWC4Vv2csD8Kc09qGKzQVGAYL5H29V12/pc1EERYbJemq0R4euBYGzwPDNgEQX4VXAlty
4hc7f0aIPr1JbK0Nn2WdUtADM6cqx6+1g71dJxdZ+e0SKG3JGK+3W0q23Ct/IAvW/lbcPXRYD7XbnApPPRZMGK
50nqjTc6+8KZBQs1/HN1xCe/cc7n1SuXYWhJM+/uDWcBpubXKWQ500BG1eORBWoGVxH8kjfsJSA1QlwQzG1kE
TOGKo/+1T0zJcaVYDSCkccX7173eYGiKaMKNW/Z1QGjpT/38SJvA3FMGQ8U051E9IKj2Sgxa0PRRQjKG+NnNdI
ymH9g0G9UGwsDymtqjjXWeq1Eq2p4Kiza07v0mm1NK2nhwzouoc1cs/K7zkuKfp7j2c2U5iAkxQxN59anAWH8Z
bJdyLVzweuQpyfa1zpPFG1RkbJyGjYdQDDv9wmXdcVA1A01t5VRVvLXF9qJv+wK7ogruXPDGjSC7mXT7ThCt3/
9ZP04Ru8qUvQsY9rMJW3b38Ch/F702EED1K/HIbSYu4RPNn8YQwwVcv0eCSSb9CjV0utQWKjPMEzxAzBFQ77v7
E0PosZwDRuxu81ZDg42aW0Jw3BfUmTib2J3itvbaEoeCfFrUTWaZ7D8mVaz2M0rExq0kYp3t8T+4uFqkn2mLL
WGk5/N3Vn1o1NGgwBdoBIQAIqLeAtSX4cG9FRHHwpeBkzQRZgQ1wVPAZ4ILNDsP3TcVMr8er2dfNVB0pc1xw
hEJd7wn9bUD0CLpAkGJmAIphXsNoboGyjxFc6Ndn2Cn/0piNnt/+W5sYolgWsWTyxIhfE/wjrt+QhQiDtRBkW+
zifLR7Ca0XnRrKApoSmMjy932WRmACm4Ct4ocGfSzrWyBdaDIaNPnKTLiw1AmQwGg9xE+8aVFBE2Rpr71IRuE
rbLyP381vCfScv0b01aP/7INX+449nCL0IEvelrtJ8M+RftDGuMis95yL3Kj1+tk3WwiaxJZMiQpMk6DCq3Hu6
JCA5+nGo8YJv6cKPNUkgEc/CJV8ZU9nNzSuLqexLDPMe2y5roufDskz3hmpRTKrhZUAzAfhvTfgk1jHz4ksp
yrN8ExjyFCFS2KoCeSLfYi7z02Sh8Mq0udVzF/UoM6xbjSukHU5RUVWvh708tM1kQf6cMmdyF1NCc1hN3WX5v9
S6bhapjL3dG5gsLqpRiDXY+OF/GXsENxfkPO/GPfcWsnp1hH5P0ZXDmBGZH/4ZMwrxPA57dDdZbmYquVLK8N5h
vItQCk4cXkh1mmnRMqY8nmfS0cCcYQEt3csvpvPuP28yXqSzUQ1R4ekJtzq5M2+7I5NjxGSorw6QgK01tBa8uR
zda+RzliQT60dPHIuVXm32s1oWW0glnymkikHnd3KnMI/K5jSoLQX5BLGsgVqNgcnmnIxwxgflJuFvYPbiXKFA
YKWozbDSSqmx8e81GsvjweicR8nRFpbv1SKNqda/ZiJCI3tZFa/nLU4+ZHvDwDuxgjhFPB2G7kKHVmZvJ4f
nLxkyk0I1LXKq402kZ1Y8vtNYFqUp+fVWYsXq0SY0ofAdWIPqgEYhAVTxop16UQoaF06va64tgzkc9PxPipLpU
kPtZGXctdZbQFWTapaiN78x9pVEa4H8vldTq2nvJqsLD/6PT4yFoVrHYUJjqxM6VSTDu81LvnneIcxrf1jHR6sR
m4DtqCa7Mi544bUUwi3yrgB0fVR50Na2I920rvGC4S6uFmb9S5/QF+gM0fpoT7Qs580c2zUVS7YtH2HpjMT9NiC
F+6bLfLc4Ub5QUmrZFBtkmu0yrufTcKS5qwYDjZ8t98106rDLnLRclmj1o0fxe23o4I20jp0fS+GpTorqRSeOG
k634xpVqwzoQty10RCpWfCY6UBDt10yeIvt0p400Y0b2UpCbJNoW/0A1dCm7zrzmqakM8Xxwow1R2rxZqa90S
MB+1/TL7qqBZp7zBjMGA9p2wGM/zN0FDnC4KG0z/NrL1k1tjCSTz2J3ZrpkRL9Yw0mlkaCmvfDDNp0zpXzDGrw
1/ciNY7+t44v2Z0c6VdjCpkAkTQxhiRtx0y8S6Iz/Q7owc10UxGaHK4YnAjssCKnYbU7So3Z8DPG1KXK0rMOGK
hg0Ixvbsq4M2nUghxXiDa9CmQqqv5fvtYvbM4NRwA/AeXWQdEFKd4215nYmaOnyUM4CqGP1GUzM0rB4kyrKIJM
PwhTJ61+x/wBESAujoFNQ1WLwYF0nGGxsf1E13M5bHrjiAvQZAq2C1yoXcxvMIYwd56IM+j8Wx0I9IfgwUKVH6
DU0y+VvKb43FPMxZxVz0x0B1Jr54TJQqL8KSQEIMG30vzEneFabnVdggNrNEqkDvGpuNBxChWww98c5Y0rk3DX
1AjdytZ9GNhHOEV5KmAWpTRoyE6yRZ/jqy+eH8k785r14slw31UmfaQfMUGUFGgtNPPcueA/rWK3QXXnbutydv
Yis0mWi+7wNT3HWYFoyhMZYK0pkwb4Rqo8G9fj14Lq19EgcspH2t7IDL1kJ9j7H91PksxIUzSX8x+dzjEBGCyL
tn30bKekA7r1fXy6bFS+5kYecZNCg+heJzjd0ZzwKvch01oGMb2Sj819z9KknP7yxZIxtqsfyoq0AEx0+/PfN
bHN87ZScN7r6JZDgVQZGtYFQNHTIQShphK1bqoy2i/kM+P1+DjC37kwCtWZI1d3JMMjwUG+V1/eB+Zo7g3XI+
rYx2g4o3/BWF+r1Fj/jkCzVkjxLwte6xW/9M/LWu1ahq6TTvndtDeUQ5uiNECpkbfRECMFnzBgdWQljAEudkgT
1n2T6Z1tVmCPxUmVkJ2zdfB+WebRs+fNL1UJ3mAf6fXv9BwvHfCM7cLuX6nH3T0fPVZ5Wnnrp60Kpn+p7fLcc
vrkVsuAgJ1k8x8zFc2Ixp685qWkTsou0ydhWt5IiXo+UX8ASHKPI5X8eCQyWV+9GSuh5gJeqZlcXbevmCQ68SM
TZibWKbSGpdT0wzzgIkh0ZJTAE6++OrWw1I5c+CBC8jsacomMa1auRWN7HBrtn/n2v7+p13ifItCRZ6Sb2X1R
xwttDxBjn3sCMdiezc4y4EkaVeeUsPh2rfZtmR5o36XM60Q8uQWgsZtK+XrjGqXWH3KshNKKr16S9PdKr7u3Fg
jrz16NMh8WFw5omEy0tgl+afIZA1DbsB/mUEYxCA98p40ogosQJjgJTyPmyZNRV10hFkNQfV4Wt+J+1PwWmk
Qsu3XzqFxvVmH2P8UhzNW+o9H2/6By4UKSWyIEtfGcexcmgGb0YMFpq31TPOyH1rTXYLCCP1YpSBT9DUhBavCe
APtpPCrT55dcS8QUm8suxozvppusA+5+Rq2bT49Kb+5Pzx5/Pz7o0cXN9RrSAQWtbIMCFJffxJV56ajbX2Nb5x
+iaM2jhjcjkBXYL3IeR3g9UJMwaILITsld2EBFvBtoLtGPfAP4xAVQ6Gjla10tyqI4SkK6Tq6asqtAoT50UnzC1
54ZA5ZZnCNiikrfvWrokAdcr7QitMoQLpSids7S11DaRFK19EePuQqpn4FcL1bzdxivfUckzRyuyPoyTHQtcOs

MDVuAwtschWYtksdPiHWwP2jhqb1irHHW++xaKaFApyuavp3kp6GRT163kbr7ZaXndKFTQ0ZZIiDtWMR38v8ua1bT+btQRKpz8JzCe5+PcTwq3qoetSsBen5CK/zz9TePTt154qVDegVo0DoT0Ez1L75Ga4VVx0L0TuQLwvkn/FE0feVrTpV40/uUVnbJuaVLWkA6kPDG8xJ7C4zLgLrJl+cJgB0iT/nq8LwlvIPqnrVbURUtoisyKdmxaueYjMA1AqP1E60zZjU/YFxkq0L1y28S3TU40/BmxffNBy9RG9z15NffCw5CWz/bF2iPFCIauQWsRoB2em+PhG04X4jTzX0/epaP7YCiRedP9VDXwKurfaamqGpcnY618UOEYbf9ue3GYhHHdRPEr59Km6craz0n5nY3z1cUrJxUKbFuL3H00QEFLycLkHEoWyyd5qPkdgIV04FK80WFpkcFZGxQL7NPDm1MtqJ51Qw1zwwGU4SUihEgQLSDksWsAEf1E/YgSKD9Tx85L0jkJLjEsBY8oL3cgziQNYzGsZLnt2viNptMqtmqMNFKCQd7hImKt3bZThyqhK6D6SKdLzdXH7ik1HSAEj7jHiPDwCR2JQ0YKfSIZhfvgDYyia4Ba6g/0SKEv3H7m0dzRxRFx0er17wxk/7+F1qm77qRmMkXaU50E0R0osjMJJ0Y+nyd7jCmMmBDz/Ge4m7IPhL397a5fWcuk8aE6a5JtnX0GTtnIsHuILZsmAVK729Nc9YMNNUnEgX+0fuUEHYN7Y/FjVv+y1I5/3FKoyVs1Vd/ehwW6fLJYnRCc+fxtNPyKcVURcg9z0LJntnDw32D4RynfIweH8aItPG00Pg1ah91BYd20Bq+Gtm+L0f4nKV+mo7wQeDLLPjjIyyfso+5ViD4Pj7RFzdiazJ6PPMsz1PC81ImVY0/4absBGn/GH5qqgXvq96zzpVzyF8v0vDGrc9a/baWl32HecvWY9teums88AUw8PS1ddmeUJoICV8ApTqDsEQVKzRfdItijhE5GBNWILTE5XrzFVwbIW9ArsXRwSP58XIb05FK2doCD8B1p5be6+WlcjQYXmE3yx3W4CKfRuILD8vqFN88XrxJsq4/qZl603S6F+vj7Mho9RB1SPE1H7rb618AeeH/b03R8G/R0r/2leChokfR6NqhGpgcZPMKH8WmTomhwqTsXw8f84vKWL3srTEnM971Vt8D5t+p/geo07X0WtTbgyKtSG3sC4p7i30qQ0hUcTmsbL0/9ux0w+rk1noXW+TCQfx90JNCRWoZ9oX7YYWGVb9MJ3DzQn0gLxg850XBrbn0Tf1v3d7GhVjAh2kC3P1Mk10qYDzot4c9m/ZNvVJF8ApaK1V4T0Z49idBPAQXrlnyhEkCiOKAJeRqAKIPDuHeTduFTGjRdE8vaX+U70yBsYKGF/27SbFMsCiaumVS7KsLdI1rR1rAkjPWlqrTFG01BeRVazU8rkA5adqpo8xdF9zPmcfHSPuLKK+456CVNb68sLMj0hsa8zzGBrd6tbLcx4gnYnrUkUunnCuNK/50+7y6nfZwCa50CFanAV0QKUorr8v66dSfN7Ur1UWohyxZ05NBnkp8Rt4aip8wKnDWFjbAzSjhgXJBgMLv68tHENCRV/e73H5PbcHqsjKFFJU7kd5hgNFj4RaE0zXAARYSNOkzdD0c5cSDcBIWvFqfXl0rVgQeIDA0yqBtETtbVows3Sg3WW+951b9qw4WAGvKntWW++XKKK/uQ4AudwvYjNWYgx1qYqsXcNVLYLAss9IaIevYzLZXXXvAHVgY3Kcj9WXdj0pQQ93oduNKcVd9EvzFXEptDfP/VkzzYNWHC8/VFFKSK3n5xVz9p+KiA5Ywv//0BruptRnXT0q0ubgAPvUFswQKsx12V/cjMhR7FBulTWIw5rnk9mvzKvf4mT+Zp4C8ia9JSXyZN4SglwwJ61UQjqINEXUbjr7g/NzDUn8SEiVQcSZPZQ8YEvOMBosvRBXkiKMdoM0hMkuPsw6psqCJyYp1/0XfRvPU9jPVPGNtXEzwBsZVgTiYQbXbycaY7Sp+0nkKaUEchIn8xARSyv/Vzt0fI5hn8Cx9K4k56305+KzKyUbsac3RVXSxsCHJb0nFQiK57Zcv1+CY7CFMQAgsW+rEaiy2vDT9Y1atb0yqqtG1Z2+W/qWZ6+ehGfoVohCauEPqPAoP4GCv8ur//35unyduRCi8ZjVg2+xUWS0g4ncEFnHgoIhJDWGJgCjWZRL97EawzTxENvoQontx76rF4wHw08qKddLw4i/0852cdVQtdFOBHI9jdIL9wR30p5hMzC7pnkYmg5iNirvRMqw4AhVPpzG3XLPQw5zir7zy/aJN1ojoRTf4jdmyRlwW7Iro9bfr99nYNHDo/nVkDCBFP8QX+34ojvF226hgaSK69HBFUY0s0RYkxL1T4x9RMRH3V0qK19Q3rVQ3KJrLaA22KCp3GMKVZNNwtJfn12oy2y/ioHC+y9b889Yq75ykYapr9B+gkxCaW8b2LOVQCGEDoxojGurYVXsdNpTjC1xkmx+xY2ux1Eom+d7Jka04HQpqEK4BhUztZYoHu5Mavpk0c4KqVoDcZuYznK6abCYBK/kclCBo+hiommWuJe8/KYZ1+n8EEDsTe41cJA8ro3+zT7hZv4/MK+ItY3BqHK+Rupmj1m17hJjg0lg7Eqj4uRi+vCCxeg3kzpANoMk8cHNMyPx4V/XRAkaw9iEpkh6JUpqSF0nzFbrmuepkJH7qgRYNXBcwazHGvq60RiUeNaIiowJWKUxWe50IykXoEn0kh5B6k+y6T3yAOHotFFMuz3qRSzNE1Nv0V0UGP0j4dyiLzpD4iQwAxv9P/Z/B/w4DYxtTQycXe1tDj2uAt50ps4u9k6mvprngfwa0vbZhCmVuZHN0cmVhbQp1bmRvYmoKcjMzIDAgb2JqCjc4MjEzCmVuZG9iagoKMzQgMCBvYmoKPDwvVHlwZS9Gb250RGVzY3JpcHRvc19G6b250TmFtZS90aW1idXNTYW5MLVJ1Z3UKL0ZsYwdzIDQKL0ZvbnRCQm94WY0xNzQgLTI4NSAxMDIxIDk1M10vSXRhbG1jQW5nbGUgMAovQXNjzW50IDk1MwovRGVzY2VudCATMjg1C19DYXBIZWlnaHQgOTUzC19TdGVtViA4MAovRm9udEzpbGUgMzIgMCBScj4+CmVuZG9iagoKMzUgMCBvYmoKPDwvTGVuZ3RoIDg3Mi9GaWx0ZXIvRmxhdGVEZWNvZGU+PgpzdHJ1YW0KeJxd1s1u2zoQBeC9n0LLd1FYEmc4MWAEIGUJy0LeFk37AI7NpAYa2VCCrd6+OnPU30XiY1oafhzJ1Nfd3e5uPF3Xn6bz4b5cq8fTeJzKy/110pTqoTydx1XTVsft4Bq88/+H5/11tZ7PvX97uZbnu/HxvN2u1p/nz16u01v1Lh3PD+X9av1x0pbpND5V77529/P7+9fL5xt5Lu01qle3t9WxPM51/ttf/t8/17Wf9eHu0H98ur59mE/5fcCXt0upWn/fkHI4H8vLZx8o0358KqttXd9W22G4XZxx+M9nTV3znIfHw7f9tNq20Lau55c5N8wNcsvcIgfmgCzMgqzMiHyZI7LNua2bDfINx2+QN8w+npgTcmb0yB1zh7xj3iH3zD3ywDyvcBvoD/AH+gP8gf4Af6A/wB/oD/AH+gP8gf4Af6BmQ6Y/wB/oD/AH+gP8gf4Af6A/wB/oD/AH+gP8gf4Av9Av8Av9Ar/QL/AL/QK/0C/wC/0Cv9Av8Av9Ar/QL/AL/QK/0C/wC/0Cv9Av8Av9Ar/QL/AL/QK/0q/wK/0Kv9Kv8Cv9Cr/Sr/Ar/Qq/0q/wK/0Kv9Kv8Cv9Cr/Sr/Ar/Qq/0q/wK/0Kv9Kv8Cv9Cn+sea+ifqQ/wh/pj/BH+iP8kf4If6Q/wh/pj/BH+iP8kf7o9emP8Ef6I/yR/gh/pD/CH+mP8Ef6I/yR/gi/sf+G/hv9Br/Rb/Ab/Qa/0W/wG/0Gv9Fv8Bv9Br/Rb/Ab/Qa/0W/wG/0Gv9Fv8Bv9Br/Rb/An0h0cic4EZ6IzwZnEr4vf54n0+QU72rJzyebvjSz5ApqNF+ICEhaQuIDkhZYNyMcTLzygiQtIWEDqfNy/5GnHjEumLGDItaMHZsyVG9YB0reefTPNgRnHZZY9Y97MRmc00n0j9E0237A05sobZjQrL040Md0Z4cx0+oabF6fPy5sj4+bolhsa4x0bndGHzhvddj13c1u7Qx+6pdF+buQ4Lnxnz0hJt/TTxxPHYe4Wm4/vOA5z1zPD3A2cF33YLV88eHo63dMvNwfq9j5X4xtiz2vnnt7nbXxz6Zd5cfzA0j3qDL6

Wxh8Gw1IHxw+sM6D0sNTBtRtYZzf8eZPhAYon/M8Hc3V4nab5oew/A/xpj0fwaSzVr58K1/MFp/nfD51o2C1KZ
W5kc3RyZWFtCmVuZG9iagoKMzYgMCBvYmoKPDwvVH1wZS9Gb250L1N1YnR5cGUvVH1wZTEvQmFzZUZvbnQvTm1
tYnVzU2FuTC1S2Wd1C19Ub1VuaWnVzGUGmzUgMCBSC19GaXJzdENOYX1gMCavTGFzdENOYX1gMjU1C19XaWR0a
HNbMCawIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgn
wCjI30CAyNzggMzU1IDU1Ni1NTy0Dg5IDY2NyAyMjEgMzMzIDMzMyAzODkgNTg0IDI30CAzMzMjC4IDI30
Ao1NTyGntu2IDU1Ni1NTyGntu2IDU1Ni1NTyGntu2IDU1Ni1NTyGmjc4IDI30CA10DQgNTg0IDU4NCA1NTY
KMTAxNSA2NjcgNjY3IDcyMiA3MjIgNjY3IDYxMSA3NzggNzIyIDI30CA1MDAgNjY3IDU1Ni1A4MzMgNzIyIDc30
Ao2NjcgNzC4IDcyMiA2NjcgNjExIDcyMiA2NjcgOTQ0IDY2NyA2NjcgNjExIDI30CAyNzggMjC4IDQ20SA1NTY
KMjIyIDU1Ni1A1NTyGntu2IDU1Ni1A1NTyGmjc4IDU1Ni1A1NTyGmjiyIDiyMia1MDAgMjIyIDgzmA1NTyGntu2C
jU1Ni1A1NTyGmzMzIDUwMCAyNzggntu2IDUwMCA3MjIgntu2IDUwMCA1MDAgMzM0IDI2MCAzMzQgNTg0IDAKMCA
wIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgn
zMzIDU1Ni1A1NTyGmty3IDU1Ni1A1NTyGntu2IDU1Ni1Ax0TEgMzMzIDU1Ni1AzMzMgMzMzIDUwMCA1MDAKMCA1NTY
gntu2IDU1Ni1AyNzggMCA1MzcgMzUwIDIyMia1MzMgMzMzIDU1Ni1AxMDAwIDEwMDAgMCA2MTEKMCazMzMgMzMzI
DMzMMyAzMzMgMzMzIDMzMMyAzMzMgMzMzIDAgMzMzIDMzMMyAwIDMzMMyAzMzMgMzMzCjEwMDAgMCAwIDAgnIDAgn
gMCawIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgnIDAgn
jAg0Dg5IDAgMCawIDI30CAwIDAgnjIyIDYxMSA5NDQgNjExIDAgnIDAgnXQovRm9udER1c2NyaXB0b3IgMzQ
gMCBSPj4KZW5kb2JqCgozNyAwIG9iago8PC9GMSAzniaWIFIvRjIgMjYgMCBSL0YzIDIXIDAgnUi19GNCAzMSAwI
FIvRjUgMTEgMCBSL0Y2IDE2IDAgnUgo+Pgp1bmRvYmoKCjM4IDAgnB2JqCjw8L0ZvbnQgMzcgMCBSCi9Qcm9ju2V
0Wy9QREyvVGv4dF0KpJ4KZW5kb2JqCgoxIDAgnB2JqCjw8L1R5cGUvUGFnZS9QYXJ1bnQgNiAwIFIvUmVzb3VY
2VzIDM4IDAgnUi9NZWRpYUJveFswIDAgnTk1IDg0M10vR3JvdXA8PC9TL1RyYW5zcGFyZW5jeS9DUy9EZXZpY2V
SR01vSSB0cnVlpj4vQ29udGVudHMgMiAwIFI+Pgp1bmRvYmoKCjYgMCBvYmoKPDwvVH1wZS9QYwD1cwovUmVzb
3VyY2VzIDM4IDAgnUgovTWVkaWFCb3hbIDAgnMCA10Tug0DQyIF0KL0tpZHNbIDEgMCBSIF0KL0NvdW50IDE+Pgp
1bmRvYmoKCjM5IDAgnB2JqCjw8L1R5cGUvQ2F0YwvZy9QYwD1cyA2IDAgnUgovUGFnZU1vZGUvVXN1T3V0bGluZ
XMKL09wZW5BY3Rp25bMSAwIFIg1L1hZWiBudWxsIG51bGwgMF0KL0xhbmcoaHuTSFUpCj4+CmVuZG9iagoKND
gMCBvYmoKPDwvQXV0aG9yPEZFRkYwMDQ2MDA2MTAwNzQwMEU5MDA3MjAwMjAwMDVBMDA3MzAwNkYwMDZDMA3N
D4KL0NyZWF0b3I8RkVGRjAwNTcwMDcyMDA20TAwNzQwMDY1MDA3Mj4KL1Byb2R1Y2VyPEZFRkYwMDRDMDA20TA
wNjIwMDcyMDA2NTAwNEYwMDY2MDA2NjAwNjkwMDYzMDA2NTAwMjAwMDM1MDAyRTAwMzI+C19DcmVhdG1vbkRhd
GUoRDoymDE3MDUyNTE0NTcxMyswMicwMCcpj4KZW5kb2JqCgp4cmVmCjAgNDEKMDAwMDAwMCA2NTUzNSB
mAowMDAwMTM1MjE1IDAgnMDAwIG4gCjAwMDAwMDAwMTkgMDAwMDAgbiAKMDAwMDAwMDU0OSAwMDAwMCBuIAowM
DAwMDAwNTY5IDAgnMDAwIG4gCjAwMDAwMTazMTEgMDAwMDAgbiAKMDAwMDEzNTM1OCawMDAwMCBuIAowMDAwM
wMzMyIDAgnMDAwIG4gCjAwMDAwMTgx0DggMDAwMDAgbiAKMDAwMDAx0DIwOSAwMDAwMCBuIAowMDAwMDE4NDA0I
DAwMDAwIG4gCjAwMDAwMTg30DkgMDAwMDAgbiAKMDAwMDAx0TAzMSAwMDAwMCBuIAowMDAwMDI4MDQyIDAwMDA
wIG4gCjAwMDAwMjgwNjQgMDAwMDAgbiAKMDAwMDAy0DI1NSAwMDAwMCBuIAowMDAwMDI4NjIxIDAwMDAwIG4gC
jAwMDAwMjg4NDIgMDAwMDAgbiAKMDAwMDAzNDQyMCawMDAwMCBuIAowMDAwMDM0NDQyIDAwMDAwIG4gCjAwMDA
wMzQ2MzUgMDAwMDAgbiAKMDAwMDAzNTAwNyAwMDAwMCBuIAowMDAwMDM1MjMxIDAwMDAwIG4gCjAwMDAwNDQzN
DEgMDAwMDAgbiAKMDAwMDA0NDM2MyAwMDAwMCBuIAowMDAwMDQ0NTU4IDAgnMDAwIG4gCjAwMDAwNDQ5NDQgMDA
wMDAgbiAKMDAwMDA0NTE4MjAwMDAwMCBuIAowMDAwMDUzNzg5IDAgnMDAwIG4gCjAwMDAwNTM4MTEgMDAwMDAg
biAKMDAwMDA1NDAwNjAwMDAwMCBuIAowMDAwMDU0MzkxIDAgnMDAwIG4gCjAwMDAwNTQ2MzEgMDAwMDAgbiAKMDA
wMDEzMj1k10CAwMDAwMCBuIAowMDAwMTMy0TgxIDAgnMDAwIG4gCjAwMDAxMzMxNzAgMDAwMDAgbiAKMDAwMDEzN
DExMiAwMDAwMCBuIAowMDAwMTM1MDc3IDAgnMDAwIG4gCjAwMDAxMzUxNjAgMDAwMDAgbiAKMDAwMDEzNTQ1NyA
wMDAwMCBuIAowMDAwMTM1NTc2IDAgnMDAwIG4gCnRyYwlsZxIKPDwvU216ZSA0MS9Sb290IDM51DAgnUgovSW5mb
yA0MCawIFIKL01EIFsgPEM1NDcyMzRBQ0IyMTJCN0NENjVCQTK2MUY5RT1BQ0FFPgo8QzU0NzIzNEFDQjIxMKI
3Q0Q2NUJBOTYxRj1FOUFDQ00+IF0KL0RvY0NoZWNrc3VtIC80NkY1M0U2RTZFNTBEREJFNDg5QjBGQkJBn0FGQ
TQ3NAovQWRkaXRpb25hbFN0cmVhbXmgW9hcBsaWNhdG1vb1MyRnZuZCMYRW9hc21zIzJFb3B1bmRvY3VtZW5
0IzJFdGV4dCA0IDAgnUgpdcj4+CnN0YXJ0eHJ1ZgoxMzU4MDkKJSVFT0YKDQo0SAwIG9iago8PAovVH1wZSAvQ
2F0YwvZwovUGFnZXMgNiAwIFIKL1BhZ2Vnb2R1IC9Vc2VpdxRsaW51cwovT3B1bkFjdG1vb1BbMSAwIFIg1h
ZWiBudWxsIG51bGwgMF0KL0xhbmcoGKh1LuVKQovQWNYb0Zvcm0gPDwKL0ZpZWxkcyBbNDEgMCBSXQovU21nR
mxhZ3MgMwo+Pgo+Pgp1bmRvYmoKNDEgMCBvYmoKPDwKL0ZUIC9TaWcKL1R5cGUgL0Fubm90Cj9TdWJ0eXB1C9
XaWRnZXQKL0YgMTMyCj9UIChTaWduYXR1cmUxKQovVia0MiAwIFIKL1AgMSAwIFIKL1J1Y3QgWzAuMCawLjAgM
C4wIDAuMF0KL0FQIDQzIDAgnUgo+Pgp1bmRvYmoKNDEgMCBvYmoKPDwKL1R5cGUgL0RvY1RpbWVtdGftcAovRm1
sdGVyIC9BZG9iZS5QUEtMaXR1C19TdWJGaWx0ZXigL0VU0kuUkZDMzE2MqovQ29udGVudHMgPDMwODIw0ThBM

DYwOTJBODY00Dg2RjcwRDAxMDcwMkEwODIwOTdCMzA4MjA5NzcwMjAxMDMzMTBEmzAwQjA2MDk2MDg2NDgwMTY1MDMwNDAyMDEzMDcwMDYwQjJBODY00Dg2RjcwRDAxMDkxMDAxMDRBMxDQ1RjMwNUQwMjAxMDEwNjAzMkEwMzA0MzAyRjMwMEIwNjA5NjA4NjQ4MDE2NTAzMDQwMjAxMDQyMDQ1QTBDRFFMTkxNzQyREM1RTE5RjYyNDZFM0QzMjZGRERGMDFERDI0REE4RDc4N0U00EU2N0U0MzJGRDZDODAwMjExMDA5NzNEMzc2NjEyOUU4Qjk1NzRBM0RBrjgxNDNDQTQ3QTE4MEYzMjMwMzIzMTMwMzcMDM5MzAzODMwMzIzMz5NUFBMDgyMDU4NTMwODIwNTgxMzA4MjAzNj1BMDA2MDIwMTAyMDIxNDE5MzcMkJENzdCMjQxOUQ2RkE5Q0VGQjQ0N0Y3RURDQjU4M0VFOEEzMDBEMDYw0TJBODY00Dg2RjcwRDAxMDEwQjA1MDA2MDuMzEwQjMwMDkwNjAzNTUwNDA2MTMwMjQyNDUzMTE1MzAxMzA2MDM1NTA0MEEwQzBDNDQ1MzUzMjA1NDQ1NTM1NDIwNTA0QjQ5MzExMTMwMEYwNjAzNTUwNDBCMEwODU0NDU1MzU0MjA1MDRCNDkzMTE4MzAxNjA2MDM1NTA0MDMwQzBGNzM2NTZDNjYyRDczNjk2NzZFNjU2NDJENzQ3MzYxMzAxRTE3MEQzMjMwMzEzMjMxMzYzMDM3MzAzMzMzY1QTE3MEQzMwMzEzMjMxMzQzMDM3MzAzMzMzY1QTMwNTEzMTBCMzAwOTA2MDM1NTA0MDYxMzAyNDI0NTMxMTUzMDEzMDYwMzU1MDQwQTBDMEM0NDUzNTMyMDU0NDU1MzU0MjA1MDRCNDkzMTExMzAwRjA2MDM1NTA0MEIwQzA4NTQ0NTUzNTQyMDUwNEI00TMxMTgzMDE2MDYwMzU1MDQwMzBDMEY3MzY1NkM2NjJENzM2OTY3NkU2NTY0MkQ3NDczNjEzMDgyMDIyMjMwMEQwNjA5MkE4NjQ40DZGnzbEMDEwMTAxMDUwMDAzODIwMjBGMDAzMDgyMDIwQTAyODIwMjAxMDBDNkQzNTQzMjNCMzA5NzZGQkJCRkYzODI2QUFEMDI00EUxMUE2QTNFQjU1MjM0NTLGNDQxQjJBRjE1RkI5RjE2MzIyMTdBQ0FBMDZENUFFNDYwNTUyNjQxOTM2MTIwNDJBRjVENkZDMTNENDc0QzJDQz1NzgxMEIwNDk1NUQ2NzQyQjMwNjgzQ0YzQkU0NjE0MTBGQzFDmkJBMkY5QjAwMDIyOUGNjFFN0RGREQxNDk4M0Q50UVQDQTE1NUY3RURDQzEyOTJFQTRFRIT3MDY4QUMxMDJEQzgxQzhCNDdENDhFMkNEQkNFNDNBND1DRjIzRkExMkVCQkYxRDE5MUUyRjNCjcwNjBENjdGRTg3NkI5NzEzNkJDRDc2RTk0MjgwMKVDQVEOTU20Dc5NDI30DE2NENDRDUyQjg2MUJENjQxMzNDNjRBQjk2Qjk4NURBNDBFNkNDQzE0QzI2Nzg4QTY4QkFDMDcwNzdDNUE2MUIwRTUyOTU4QUE4NkM0RjQ3NkUyQ0JDMjBEQkJBMjDEN0M2MDdC0DdC0D1ERDEzOUGNENBNUZBQTY4QTM4MzM10jUzNjY5RjM4QjMwMEZENDayNENFQzZDMTQ3REEyREMwNjREM0JFNzIzMj1BmzY0RjhBNDg2RjhBNjA1MjAyOEIxNzkxNEVBMjYzRkRDRkUzRTZFRUMwMjQwQTgwMERBOTU3RkYyRDU0NDFC0JCRkNGRTUzQkVFOTg5MUUxMEExM0ZCNENDNg3RTY4NTI20DU1QkNCNzQzOTA1QkQwRDK1Rjk00DJBOTU3QkIyNDdGN0QxNjQ5RUZGQjY4QTE1MUQ1RDBCnjYxNDcxRT1EQQREMTUyNzQzNkNCRj1DNTI5M0MyNkQ0RDE2NDAxNzFGMkYwQkI3QzVCM0ZBQUQwRdg4MTUyQkNENkVdOTI1MDMxOUZFMtk4NDQ4Q0JBM0Y3MjY3RUIyRdk4QjI1NEJEQ0EzNDgwRjAzNTQ4N0UyQzdbOTEzNkNEMkRDNzI1MDEyMkI3MUNDMTFDRjg5QTBFQjIxQkQ5QTZF0DdBNjI4MTRGNTI3NDMxMDJGRTdBM0ZEMTc4MUIy0DZD0thCMUJBQUIxOUUyQkFEMzFFOU1QzQ4NTJFRjVDNUZBRDhEqzYyMUZC0ThFNEYyRTM0NTI5QzY0RjU30TdFNUY3RdhGMERFOTA3QjEzNUY0MjEzNDg10DQwQTQx0DY3RjJGMTJEqTQxMjE3MUZCOUyN0RDNkE3NkNDNzQ2RUFERkNGRUVDOEQwOTyZMkI0Qjc1REVDMj1GmzgRTQ1QjRGMzRERDRBMEE2MjhCmjg3MEJBmjA3MUQ0RUUxMjAxNDMxMDIwMzAxMDAwMUEzNTEzMDRGMzAwOTA2MDM1NTFEMTm0NDAYMzAwMDwMEIwNjAzNTUxRDBGMDQwNDAzMDIwNzgwMzAxNjA2MDM1NTFEMjUwMTAxRkYwNDBDMzAwQTA2MDgyQjA2MDEwNTA1MDcwMzA4MzAxRDA2MDM1NTFEMUwNDE2MDQxNEM3RjA1Q0VCMUY40ERGMjFD0T1DQTMxM0JBN0NGQU4MkQ1Mzk20DgzbEMDyW0TJBODY00Dg2RjcwRDAxMDEwQjA1MDAwMzgyMDIwMTAwNzJFRDUyMEEwMTc0NUQxNTM2NjgxOUUyOUU4NUMyNkM0Qjg0MkMx0TY3RjQ0QTY30TVBQkMyNzIxMDMyREZDMDk4RkFGODU3Qtk2RjdFRE5NzQ5RjQ1NjBCRjUxRjVBMUFDT1EMD1FQjI1QtkyRTZFTZFOUQ2Q0I20TcyREFDMEY2RkUzNUFBRDY40TBEQjE1NjBGNT1GmjVdQkZFMzYyOTZEQ0VEMzNERDMxMzUyRTg3RTRC0DYyNkFERDk1NDRCMkVFNzcxMzgwQTY5MUMwMDU1QUJGQzM3NDk2QkQwMDEwQjQ4QkE0MTc3Nzk0MDM0MTkzNDQ5QzNDMUUwRUJGMzYxREQyMEVFQjUwQUY5M0IyM0YwMTM5NzYxMURGQzgwQjAy0DhFQjc4Mtc5MTQ0NTIyMUI5NzkyMDA4QkZBmkZDRTM50TU2NUQxRDM0QzkyQTgxOEI1QjNGNzZDRjJEODUyNkU1MTkyNEY4QjVGRjg5NjExMzBBRTc1NzcmTJDz1EqzM30Dg1RERBQ0RCMjA2NjMxNDJEMzFCODk10DY4MzE3MDJGRjU1NjM4MUFDQjRBODZC0I2QTIzMEE2NUYxMTE0QTA2MzMxNEYzM0Q1NUQ3NDUxQUY50E1NkY0MzMxRUVCM3MjY2QkJDOTgwMjQ3NzY1Mku2NDZGNTVDQjkyNTE2QkI2QzM4QjMyNzJFMTg3QkFBQUQ2N0MwRTM20TUwMjU0NDNCNDY2MkE4Mjk3NzMzNkE4QzBGMUQ30DhENjgyRDY0NEU3MTQ5NDQ4NjhB0EY4N0Q2QzhEMDhGRDhBNjA5NjJFRTg1QzQ2NzRGMDEyOTyWNDg3Mzg50Dg2Qzg2QkJDMkRFRkU5MDE5RThDMzK3NzQ2M0M1QzU50TFCRjQ1MkEzNEY0RUU0MzMyMzVBODA0NTU0QjQ50TJDNThEMjg0REI30ENGME15NjI4N0QyMEI1MDExRTVCNzYxQUYxNjC2MDA30TYyNUFFMEQ1QzFCM0M0QjI5NkM00TyzN0JQzczOUU4OEEyMDkyNTRCQUIzNjUzNUQwRjA2N0IwRjU0NUVDNUNCmjQ4NUU3NjA2MEY0RTg0RTk5Mjg5RUFGRDhCMjdbQj1FMDUzM0Q4RTM3MTU5MThFrjNBNURGQzAwMDg3RddCRDjBOEU30TJDMEI1N0Q30TzBNERBMzNFNzdEODUzMUNCMTkwNjU1RDJGM0QwNDA5NjY3QzhDMkVdNTQyNzQ4QUM20TY30Dg3MDQ4Mjk0QzM5RkZCOUy40EYxRDBBMUewM0EwNjAzRjkxQTAxNjFFMUZEM0IwRkQ0MzQzMDhERTI1Mzc1MEFFRDVENEI0MDUyMDMxODIwMzY2MzA4MjAzNjIwMjAxMDEzMDY5MzA1MTMxMEIzMDA5MDYwMzU1MDQwNjEzMDI0MjQ1MzExNTMwMTMwNjAzNTUwNDBBMEMwQzQ0NTM1MzIwNTQ0NTUzNTQyMDUwNEI00TMxMTEzMDBGMDYwMzU1MDQwQjBDMDg1NDQ1NTM1NDIwNTA0QjQ5MzExODMwMTYwNjAzNTU

Get data to be counter signed

This method returns the data to be signed in order to create a counter signature. The user should provide a document containing a signature to be counter signed, id of the signature, and other parameters similarly to the method 'getDataToSign()'.



The parameters in `getDataToBeCounterSigned` and `counterSignSignature` MUST be the same (especially the signing date).

Request

```
POST /services/rest/signature/one-document/getDataToBeCounterSigned HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 9972

{
  "parameters" : {
    "signingCertificate" : {
      "encodedCertificate" :
"MIIC6jCCAdKgAwIBAgIGLtyU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMELJvb3RTZWxmU21nbmVkJmFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQDDApTaWduZXJGYWtLMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZH9ax8F1fE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdxULqpwz4JEXW9vz64eTbde4vQJ6pjHgarJf1gQNEc2XzhmI/prXLysWNqC71Zg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QD45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi40kDN2b1
dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUix00IoUvNbF0xAW7PBZIKDLnm6LsckRxs1U32sC
9d1L0He3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIfIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6ky0I0Bvw14PWtsWG10Qd0SRvIBBrP4adCnGT
9jgjk9LTc08B8FKrr+81H6uc0bp4LIUToiUkGILXsiEeEg9WAqm+Xq0"
    },
    "certificateChain" : [ ],
    "detachedContents" : null,
    "asicContainerType" : null,
    "signatureLevel" : "XAdES_BASELINE_B",
    "signaturePackaging" : null,
    "jwsSerializationType" : null,
    "sigDMechanism" : null,
    "signatureAlgorithm" : "RSA_SHA256",
    "digestAlgorithm" : "SHA256",
    "encryptionAlgorithm" : "RSA",
    "referenceDigestAlgorithm" : null,
    "maskGenerationFunction" : null,
    "contentTimestamps" : null,
    "contentTimestampParameters" : {
      "digestAlgorithm" : "SHA256",
      "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
      "timestampContainerForm" : null
    }
  }
}
```

```

    },
    "signatureTimestampParameters" : {
        "digestAlgorithm" : "SHA256",
        "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
        "timestampContainerForm" : null
    },
    "archiveTimestampParameters" : {
        "digestAlgorithm" : "SHA256",
        "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
        "timestampContainerForm" : null
    },
    "signWithExpiredCertificate" : false,
    "generateTBSWithoutCertificate" : false,
    "imageParameters" : null,
    "signatureIdToCounterSign" : "id-afde782436468dd74eeb181f7ce110e1",
    "blevelParams" : {
        "trustAnchorBPPolicy" : true,
        "signingDate" : 1625817757122,
        "claimedSignerRoles" : null,
        "policyId" : null,
        "policyQualifier" : null,
        "policyDescription" : null,
        "policyDigestAlgorithm" : null,
        "policyDigestValue" : null,
        "policySpuri" : null,
        "commitmentTypeIndications" : null,
        "signerLocationPostalAddress" : [ ],
        "signerLocationPostalCode" : null,
        "signerLocationLocality" : null,
        "signerLocationStateOrProvince" : null,
        "signerLocationCountry" : null,
        "signerLocationStreet" : null
    }
},
"signatureDocument" : {
    "bytes" :
"PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGluZz0iVVRGLTgiPz48ZHM6U21nbmF0dXJlIHhtbG5z0mRzPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIiBjZD0iaWQtYWZkZTc4MjQzNjQ20GRkNzR1ZWIx0DFmN2N1MTEwZTEiPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpeMf0aW9uTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXjhjLWMxNG4jIi8+PGRz01NpZ25hdHVyZU1ldGhvZCBBbGdvcm10aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiLz48ZHM6UmvVmZXJlbmN1IElkPSJyLWlkLTEiIFR5cGU9IiIgVVJJPSJzYW1wbGUueG1sIj48ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hhMjU2Ii8+PGRz0kRpZ2VzdFZhbHV1PmtjREhPWmp3WmhWZnVEaHV0q2VDRVJSbVlwVEg0Sm00Um1mV1ZpMzFROWc9PC9kczpEaWd1c3RWYWx1ZT48L2Rz01J1ZmVyZW5jZT48ZHM6UmVmZXJlbmN1IFR5cGU9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDE5MDMjU21nbmVkJUHJvcGVydGllcyIgVVJJPSIjeGFkZXMaWQtYWZkZTc4MjQzNjQ20GRkNzR1ZWIx0DFmN2N1MTEwZTEiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzEwL3htbC1leGMtYzE0biMiLz48L2Rz01RyYW5zZm9ybXM+PGRz0kRpZ2VzdE1ldGhvZCBBbGdvcm10aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPjxkczpEaWd1c3RWYWx1ZT5EenR3T1RtUm9PQW02L2xNSThSeW01eFpQek12TF1Eem4vZWJZWWtQc3I0PTwvZHM6RGlnZXN0VmFsdWU+PC9kczpSZWZ1cmVuY2U+PC9kczpTaWduZWRJbmZvPjxkczpTaWduYXR1cmVWYWx1ZSBJZD0idmFsdWUtaWQtYWZkZTc4MjQzNjQ20GRkN

```

zR1ZWIxODFmN2N1MTEwZTEiP11BN3NFTnQzTjh1ZkxGTW5LcjM2cjBQcXpNaVkzUTBzKytJR1RFVUMwc3BheFV
2MGRIWk0wZC95bjNrcExKTG9Va0k0TTNmbGo1V0duODNrZjA1QnFNMMWtoc1g2MUDKemFGVFBHcG03YwtSUUtod
m9IMjV5eXFUWVhFU2xCY20wNGl6aUtoTE16WmpVZng0L0IxWk15c3Y1cE1CZ0oycjJvaTzqTG9wOXd3M2d1NGM
0WUpvYUsrU1hrNmh5VE5PY044UGpHZTYzV1LPVE5WUFFGdmph0EJsd2crYTBiQnV3RCs4TjZmd2lnQ2RXNWEvN
ERKVWUvSjhNYjcwWkk4U69Pem5HrgZpK1RQYm1JZVztQ2JsnW1Vb1VnM1EveF1sdUpmTGgzbUDRQVhLQnZGNDV
vRE1IU1Z1Zm50L0QvV310QUNsVVZEb1FTeXd1bW5qUHBxRjh1Zz09PC9kczpTaWduYXR1cmVWYwX1ZT48ZHM6S
2V5SW5mbz48ZHM6WDUwOURhdGE+PGRz0lg1MD1DZXJ0aWZpY2F0ZT5NSU1EMURDQ0FyeWdBd01CQWdJQkNqQU5
CZ2txaGtpRz13MEJBUXNGQURCTk1SQxdEZ11EV1FRRERBZG5iMj1rTFd0aE1Sa3dGd11EV1FRS0RCQk9iM2RwY
m1FZ1UyOXNkWFJwYjI1ek1SRXdEd11EV1FRTERBaFFTMGt0VkvWVFZERUxNQwtHQTFRUJotTUNUR1V3SGhjTk1
UWxhNREkyTURjMU5ETXhXaGNOTVRnd09ESTJNRGMxTkRNefdqQ1BNUK13RUFZRFZRUUREQWxuYjI5a0xYVnpaW
E14R1RBWEJnT1ZCQW9NRUU1dmQybhVZU0JUYjJ4MWRhbHZibk14RVRBUEJnT1ZCQXNNQ0ZCTFNTMvvsvk5VTVF
zd0NRWURWUVFHRXdKTVZUQ0NBU013RFFZSkvtWk1odmNOQVFFQkJQRQRnZ0VQQU0FRb0NnZ0VCQUxS1VJU
VpidzNuU2RMcCtC0WN6RUNncFpra1E1eFY0Zz1NLzd3bGc5N29DQ2Y3VUVo0UJBMWQrellqc3p2K0JMKWJKW1B
nYW4yMTQ0QXZnc29HSmZiN1VJeVZNGdrbFvnSWwxYXJVdm9uK1drS25zZUZ1UU9mSnlqU0ZVRE13bnV2cDBoe
mNKWEhYUm1MZG15aCtuKzZOTUgbw201dFZvU2ZRCnRCVm1DTGVTTVZ6dUQ1RVBqMG1JUmN40TFwTDM4ZTNGT1R
XN05hR1pMZVW6dUZ1Ui9xN3o5M2xMa3ZaNFZBTU5HR0x2SVhPWWVSQLpNeVBocEJaNEwzQThJM0VFbEtXSC8xT
HdpaVhUVFNHMXNNNld2TVRWYmYydmJkNDduW1JRTJtU3BOR2pRb3VPQUVyzmVWV1Vxek1DZ2hRQ0hSR090dVN
MRy91ZnFGSGI0aldnMENBd0VBQWFPQnZEQ0J1VEFPQmd0VkhROEJBZhFQkFNQ0JrQXdnWWNHQ0NzR0FRVUZCd
0VCQkhzd2VUQTCZ2dyQmdFRkJRY3dBWV10YUhSMGNEb3ZMM1J6Y3k1dWIzZHBibUV1YkhVdmNHdHBMV1poWTN
SdmNua3ZiMk56Y0M5bmIy0WtMV05oTUR3R0NDc0dBUVVGQnpBQ2hQm9kSFJ3T2k4dlpITnpMbTV2ZDjsdV1TN
XNkUz13YTJrdFptRmpkRz15ZVM5amNuUXZaMj12WkMxa11TNwpjb1f3SFFZRFZSME9CQ11FRk4ycEhELzdQZWZ
tQ1Q4b1gy0VpoV3kvT0hKMU1BMEdDU3FHU01iM0RRRUJdd1VBQTRJQkFRQkszVk9MaERJVLdLb0ZycmhoV3phZ
GR0azZYUXRjd1JvT1BWU3NpL2dPcnpzE03MEEzMXhJVHc3WWZMaHBvVkExeG83b3ZIBhGRwTGxocXk5bzV3aDI
4Mn1DcHFCVUF0Z3JTa0RHb2crSzdDTDZnVXBBybF1pWhVHWnJ0ZjYM2ZIUzJvc3g0WkozdE1qNndWZWNERVVxS
VNGzkZUMkVzbTBRWFVuZ01LRk1s0TVY210dzJ3eFhiT3pVZURkNERJUHJ2K21XNXBvQVdyNk10c1YrSDJWUST
aTC9rQm53V0hqU1RPYUdGaXNxWFkvYUgvMVB0Q1hBKzE1K11JV2tSkJtdjNrRGFGek9YQUV0Uj1aSThsWU9KY
XJuWTdBeS9hTjZi0XVHzmZyYm8vaFZBY0w0V0RkaGt1Qk4zbTh3K2c3NkxvQVh0ZUV1dTA0Qs8weExae1VCPc9
kczpYNTA5Q2VydG1maWNhdGU+PGRz0lg1MD1DZXJ0aWZpY2F0ZT5NSU1ENmpDQ0F0S2dBd01CQWdJQkJEQU5CZ
2txaGtpRz13MEJBUXNGQURCTk1SQxdEZ11EV1FRRERBZH1iMjkwTFd0aE1Sa3dGd11EV1FRS0RCQk9iM2RwYm1
FZ1UyOXNkWFJwYjI1ek1SRXdEd11EV1FRTERBaFFTMGt0VkvWVFZERUxNQwtHQTFRUJotTUNUR1V3SGhjTk1UW
XhNREkyTURjMU5ETXdXaGNOTVRnd09ESTJNRGMxTkRNd1dqQk5NUkF3RGdZRFZRUUREQWRuYjI5a0xXTmhNUmt
3RndZRFZRUUtEQkJPYjNkcGjtRwdVmjlzZfhScGIyNxpNUkV3RhdZRFZRUUxEQWhRUzBrdFZV1RWREVMUTfR
0ExVUVCaE1DVEZVd2dnRWLNQTBHQ1NxR1NjyjNEUUVQVVFQVE0SUJEd0F3Z2dFS0FvSUJBUNiYmw1c0tCQ2p
TQjhUTWRhY31teC9XZk9qTVcxZ21Ja1ZKU1ky0EpiT1drQ1ZtdHpnW10Z2hmc1BRUGx1ZXUwRFRhbGEa3JTU
31oQ3Z6elBTR1B3Q0ZPYWhGL243aFFhMUYzVWFIU3hUs3JGQzVuT3dkTHp4S1JPM1dqVnRJR1JTWDJrdjFGZ1V
wUXk1RX15K3JzZ1N6SjU5ZFU1WlpkV3BkYUR1RHhWVn1EZxIzRU15Q2JHny81SD1NDRDZdXp0cGVUR1dtTTZjV
VNUMDc5N1hEbGJFeFNUVEdRWEZKQT1rQ0NzeT1EWG5KYThuejBGRThmbWN2UUh1VTzr0VFiChpHak1kM0RXbEU
2bm83VWRDwUQxSDA0K3VzQnA1aGhDckFCNjcwTmRvVHj0VG1HTkFGdDRKVDB2aXrQs0hx0utFSWQ2TghkY20yV
Gc5M2REY1dGdEFnTUJBQUDqZ2RRd2dkRXdEZ11EV1IwUEFRSC9CQVFEQWd1Q1FRUdBMVvkShdRnk1EZ3d0cUE
wb0RLR01HaDbkSEE2THk5a2MzTXVibTkzYvc1aExteDFM0JyYVmxv1LxtjBim0o1TDj0eWJDOX1iMjkwTFd0a
ExtTn1iREJNQmdnckJnRUZCUWNCQVFSQU1ENhdQQV1LS3dZQkJRVuhNQutHTUdoMGRiQTZMeT1rYzNNdWJtOTN
hVzV0TG14MUwzQnJhUzFtWVd0MG1zSjVMMk55ZEM5eWIy0TBMV050tG10eWREQWRCZ05WSFE0RUZnUVUrMnRGc
XBOZTNhmjNZUjh5cUJaSWLWV1Mzd1V3RhdZRFZSMFRBUUgvQkFVd0F3RUIvekFOQmdrcWhraUc5dzBCQVFzRkF
BT0NBUUVBRSt0dWQwNVHHT002RkVaSFdUYzgrYm16LzZCMFhRWE41NjRLV0JCaGNo0Wk1R2FkanFwU3N1dmTuK
3R1THE1bTZDTG8zZTrsWDJkSjdoc1BBdn1hTHFPSXB6ZzQ5VEdkaW1xbk9CMk83Nct5QWhUOHY5R1p0SDFFQ0h
YeFlzdX1TR01LdmQrTDVJakpUaXMzbGw0d1U4Rkh6eVJstT1JUW53W1I1MDZqRmNKZUdsT2d5WmgrVUxXb1JOR
UV3cU44RFRGMkQwWg9nWUJzckN4Q0JqMFBwYUgPcnV2RvFxcFV1dVlnMTRSMURKRmFoTHdxV11TT0Q1Z1BobUE
wSFI0ejNHRjNqSFN6MGk5a1hTVE9zVWNka3ZVSnkwdE1PbnVqc1VFa2czSDZXzNsejhUdzNjYzdWMU51YitNQ
zVLNFp2WCs1U115dTArZxi3YkZzY0lyWVp3PT08L2Rz01g1MD1DZXJ0aWZpY2F0ZT48L2Rz01g1MD1EYXRhPjw
vZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWR1czpRdWFsaWZ5aW5nUHJvc6VydG11cyB4bWxuczp4YWR1cz0ia
HR0cDovL3VyaS51dHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZQ9IiNpZC1hZmR1NzgyNDM2NDY4ZGQ3NGV

```

    "digestAlgorithm" : null,
    "name" : "xades-detached.xml"
  }
}

```

Response

```
HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:37 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 1488

{
  "bytes" :
"PGRz01NpZ251ZE1uZm8geG1sbnM6ZHM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyMiPjxkc
zpDYW5vbmljYWxpeM0aW9uTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWw
tZXhjLWMxNG4jIj48L2Rz0kNhbm9uaWNhbG16YXRpb25NZXR0b2Q+PGRz01NpZ25hdHVyZU1ldGhvZCBBbGdvc
m10aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiPjwvZHM6U21
nbnF0dXJ1TWV0aG9kPjxkczpSZWZ1cmVuY2UgVH1wZT0iaHR0cDovL3VyaS51dHNpLm9yZy8wMTkwMyNDb3Vud
GVyc21nbmVku21nbmF0dXJ1IiBVUkk9IiN2YWx1ZS1pZC1hZmR1NzgyNDM2NDY4ZGQ3NGV1YjE4MWY3Y2UxMTB
1MSI+PGRz01RyYW5zZm9ybXM+PGRz01RyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwM
DEvMTAvE1sLWV4Yy1jMTRuIyI+PC9kczpUcmFuc2Zvcm0+PC9kczpUcmFuc2Zvcm1zPjxkczpEaWd1c3RNZXR
ob2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAxLzA0L3htb6VuYyNzaGEyNTYiPjwvZHM6RGlnZ
XN0TWV0aG9kPjxkczpEaWd1c3RWYWx1ZT51bTBWdTF1MEgyS0lpWkVjM1I2czEzQ3I1ZDjZTzTVFZxb1BHnd
GaTdjPTwvZHM6RGlnZXN0VmFsdWU+PC9kczpSZWZ1cmVuY2U+PGRz01J1ZmVyZW5jZSBuExB1PSJodHRw0i8vd
XJpLmV0c2kub3JnLzAxOTAzI1NpZ251ZFByb3B1cnRpZXMiIFVSS0iI3hhZGVzLW1kLTg2NmI2IwNzcx0TY
5NGU4ZTZkNmE20TYyYjQ4NGFjIj48ZHM6VHJhbnNmb3Jtcz48ZHM6VHJhbnNmb3JtIEFsZ29yaXRobT0iaHR0c
DovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIj48L2Rz01RyYW5zZm9ybT48L2Rz01RyYW5zZm9
ybXM+PGRz0kRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI
3NoYTI1NiI+PC9kczpEaWd1c3RNZXR0b2Q+PGRz0kRpZ2VzdFZhbnNmb3JtIEFsZ29yaXRobT0iaHR0c
SMEtRdGY1MU51UVhEV3FTQWtsWG89PC9kczpEaWd1c3RWYWx1ZT48L2Rz01J1ZmVyZW5jZT48L2Rz01NpZ251Z
EluZm8+"
}
```

Counter Sign Signature

This method incorporates a created counter signature to unsigned properties of the master signature with this specified id.



The parameters in `getDataToBeCounterSigned` and `counterSignSignature` MUST be the same (especially the signing date).

Request

```
POST /services/rest/signature/one-document/counterSignSignature HTTP/1.1
```

Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 10397

```
{  
  "parameters" : {  
    "signingCertificate" : {  
      "encodedCertificate" :  
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMME1Jvb3RTZWxmU21nbmVkr  
mFrZTERMA8GA1UECgwIRFNTLXR1c3QwHhcNMTCwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ  
DDApTaWduZXJGYWt1MREwDwYDVQQKDAhEU1MtdGVzdDCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA  
MI3kZhtnipn+iiZH9ax8F1fE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH  
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt  
sStkYXdULqpzw4JEXW9vz64eTbde4vQJ6pjHgarJf1gQNEc2XzhmI/prXLYsWNqC71Zg7PUZUTrdegABTUzYCR  
J1kWBRPm4qo0LN405c94QD45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA  
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUA4IBAQCK6LGA01TR+rmlU8p6yhAi40kDN2b1  
dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC  
9d1L0He3WKBNB6GZALT1ewjh7hSbjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz  
D7UT93Nuw3xcV8ViftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6ky0I0Bvw14PWtsWG10Qd0SRvIBBrP4adCnGT  
gjgjk9LTc08B8FKrr+81HGuc0bp4lIUToiUkGILXsjeEg9WAqm+Xq0"  
  },  
  "certificateChain" : [ ],  
  "detachedContents" : null,  
  "asicContainerType" : null,  
  "signatureLevel" : "XAdES_BASELINE_B",  
  "signaturePackaging" : null,  
  "jwsSerializationType" : null,  
  "sigDMechanism" : null,  
  "signatureAlgorithm" : "RSA_SHA256",  
  "digestAlgorithm" : "SHA256",  
  "encryptionAlgorithm" : "RSA",  
  "referenceDigestAlgorithm" : null,  
  "maskGenerationFunction" : null,  
  "contentTimestamps" : null,  
  "contentTimestampParameters" : {  
    "digestAlgorithm" : "SHA256",  
    "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",  
    "timestampContainerForm" : null  
  },  
  "signatureTimestampParameters" : {  
    "digestAlgorithm" : "SHA256",  
    "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",  
    "timestampContainerForm" : null  
  },  
  "archiveTimestampParameters" : {  
    "digestAlgorithm" : "SHA256",  
    "canonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",  
    "timestampContainerForm" : null  
  },  
  "signWithExpiredCertificate" : false,  
  "signWithRevokedCertificate" : false  
}
```

```

"generateTBSWithoutCertificate" : false,
"imageParameters" : null,
"signatureIdToCounterSign" : "id-afde782436468dd74eeb181f7ce110e1",
"blevelParams" : {
    "trustAnchorBPPolicy" : true,
    "signingDate" : 1625817757122,
    "claimedSignerRoles" : null,
    "policyId" : null,
    "policyQualifier" : null,
    "policyDescription" : null,
    "policyDigestAlgorithm" : null,
    "policyDigestValue" : null,
    "policySpuri" : null,
    "commitmentTypeIndications" : null,
    "signerLocationPostalAddress" : [ ],
    "signerLocationPostalCode" : null,
    "signerLocationLocality" : null,
    "signerLocationStateOrProvince" : null,
    "signerLocationCountry" : null,
    "signerLocationStreet" : null
},
},
"signatureValue" : {
    "algorithm" : "RSA_SHA256",
    "value" :
"t7yd+/ltunh51z7wxK8wlQIpBK1zn9T1/xHoLFFcVSnWVhoKa0rh5x7Ddtlr7WwTRBeRCt/YI/Xq6HEJS03v7
Kj+Z1TDnviV7PQQqGPW64I2AAbcKYoyrzoZgHcCnyQ92avwYAUm0hbwNWRHuRZ3ALahEvwz6jVtip3B0cgQ5kh
bk7rZnGWuhxWe1ASceDrv4qsLFi/sHS7b1VeP8Cs0WgE4bT5+sZ+Ga5hsKay108QJg86Sn+pGpwxHXByaa5Vdz
WArZ0U/yK+YXT1LoJr8x1KKeGKIAKdd7PjIRzY6JF06G2WjTYE+x7aD0FFM01eb0Fsxu+9n08Af46k+0RKcpQ=
="
},
"signatureDocument" : {
    "bytes" :
"PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGl0Zz0iVVRGLTgiPz48ZHM6U21nbmF0dXJ1IHhtbG5z0mRzPSJod
HRw0i8vd3d3LnczLm9yYz8yMDAwLzA5L3htbGRzaWcjIiBJZD0iaWQtYWZkZTc4MjQzNjQ20GRkNzR1ZWIx0DF
mN2N1MTEwZTEiPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpeM0aW9uTWV0aG9kIEFsZ29yaXRobT0ia
HR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PGRz01NpZ25hdHVyZU1ldGhvZCBBbGd
vcm10aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiLz48ZHM6U
mVmZXJ1bmN1IE1kPSJyLW1kLTEiIFR5cGU9IiIgVVJJPSJzYW1wbGUueG1sIj48ZHM6RGlnZXN0TWV0aG9kIEF
sZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hMjU2Ii8+PGRz0kRpZ2VzdFZhb
HV1PmtjREhPWmp3WmhWZnVEaHVoQ2VDRVJSbVlwVEg0Sm0U0m1v1ZpMzFROWc9PC9kczpEaWd1c3RWYWx1ZT4
8L2Rz01J1ZmVzW5jZT48ZHM6UmVmZXJ1bmN1IFR5cGU9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDE5MDMjU21nb
mVkJUHJvcGVydG1lcIgVVJJPSIjeGFkZXMaWQtYWZkZTc4MjQzNjQ20GRkNzR1ZWIx0DFmN2N1MTEwZTEiPjx
kczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yYz8yMDAxLzEwL
3htbC1leGMtYzE0biMiLz48L2Rz01RyYW5zZm9ybXM+PGRz0kRpZ2VzdE1ldGhvZCBBbGdvcm10aG09Imh0dHA
6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPjxkczpEaWd1c3RWYWx1ZT5EenR3T1RtUm9PQ
W02L2xNSThSeW01eFpQek12TF1Eem4vZWJZWWtQc3I0PTwvZHM6RGlnZXN0VmFsdWU+PC9kczpSZWZlcmVuY2U
+PC9kczpTaWduZWRJbmZvPjxkczpTaWduYXR1cmVWYWx1ZSBJZD0idmFsdWUtaWQtYWZkZTc4MjQzNjQ20GRkN
zR1ZWIx0DFmN2N1MTEwZTEiP11BN3NFTnQzTjh1ZkxGTW5LcjM2cjBQcXpNaVkzUTBzKytJR1RFVUMwc3BheFV
2M6RIWk0wZC95bjNrcExKTG9Va0k0TTNmbGo1V0duODNrZjA1QnFNMWtoc1g2MUDKemFGVFBHcG03YWtSUUtod
m9IMjV5eXFUWVhFU2xCY20wNG16aUtoTE16WmpVZng0L0IxWk15c3Y1cE1CZ0oycjJvaTZqTG9w0Xd3M2d1NGM
"
}

```

0WUpvYUsrU1hrNmh5VE5PY044UGpHZTYzV11PVE5WUUFFGdmp0EJud2crYTBIQnV3RCs4TjZmd21nQ2RXNWEvN
ERKVWUvSjhNYjcwWkk4UG9Pem5HRGZpK1RQYm1JZVztQ2J5NW1Vb1VnM1EveF1sdUpmTGgzdUdRQVhLQnZGNDV
vRE1IU1Z1Zm50L0QvV310QUNsVVZEb1FTeXd1bW5qUHBxRjh1Zz09PC9kczpTaWduYXR1cmVWYwX1ZT48ZHM6S
2V5SW5mbz48ZHM6WDUwOURhdGE+PGRz01g1MD1DZXJ0aWZpY2F0ZT5NSU1EMURDQ0FyeWdBd01CQWdJQkNqQ5
CZ2txaGtpRz13MEJBUXNGQURCTk1SQxdEZ11EV1FRRERBZG51Mj1rTFd0aE1Sa3dGd11EV1FRS0RCQk9iM2RwY
m1FZ1UyOXNkWFJwYjI1ek1SRXdEd11EV1FRTERBaFFTMGt0VkvWVFZERUxNQWtHQTFRUJ0TUNUR1V3SGhjTk1
UWXhNREkyTURjMU5ETXhXaGNOTVRnd09ESTJNRGMxTkRNNeFdqQ1BNUK13RUFZRFZRUUREQWxuYjI5a0xYVnpaW
E14R1RBWEJnT1ZCQW9NRUU1dmQybhVZU0JUyjJ4MWRhbHZibk14RVRBUEJnT1ZCQXNNQ0ZCTFNTMVSVk5VTVF
zd0NRWURWUVFHRXdKTVZUQ0NBU013RFFZSktrWk1odmNOQVFFQkJRQRnZ0VQQURDQ0FRb0NnZ0VCQUxSQ1VJU
VpidzNuU2RMcCtC0WN6RUNncFpra1E1eFy0Zz1NLz3bGc5N29DQ2Y3VUVo0UJBMWQrellqc3p2K0JkmwjkwlB
nYW4yMTQ0QXZnc29HSmZiN1VJeVZNGdrbFvnSWwxYXJvdm9uK1drS25zZUZ1UU9mSn1qU0ZVRE13bnV2cDBoe
mNKWEhYUm1MZG15aCtuKzZOTUgb201dFZvU2ZRcnRCVm1DTGVTTVZ6dUQ1RVBqMG1JUmN40TFwTDM4ZTNGT1R
XN05hR1pMZWV6dUZ1Ui9xN3o5M2xMa3ZaNFZBTU5HR0x2SVhPWVWSQ1pNeVBocEJaNEWzQThJM0VFbEtXSC8xT
HdpaVhUVFNHMXNNN1d2TVRWYmYydmJkNDduWLJRTJtU3BOR2pRb3VPQUVyZmVWV1Vxek1DZ2hRQ0hSR090dVN
MRy9IZnFGSGI0aldnMENBd0VBQWFPQnZEQ0J1VEFPQmd0VkhROEJBZjhFQkFNQ0JrQXdnWWNHQ0Nz0FRVUZCd
0VCQkhzd2VUQTCZ2dyQmdFRkJRY3dBWV10YUhSMGNEb3ZMM1J6Y3k1dWIzZHBibUV1YkhVdmNHdHBMV1poWTN
SdmNua3ZiMk56Y0M5bmIy0WtMV05oTUR3R0NDc0dUVVGQnpBQ2hQm9kSFJ3T2k4d1pITnpMbTV2ZDjsdV1TN
XNkUz13YTJrdFptRmpkRz15ZVM5amNuUXZaMj12WkMxallTNWpb1F3SFFZRFZSME9CQ11FRk4ycEhElzdQZWZ
tQ1Q4b1gy0VpoV3kvT0hKMU1BMEdDU3FHU01iM0RRRUJdd1VBQTRJQkFRQkszV9MaERJv1dLb0ZycmhoV3phZ
GR0azZYUXRjd1JvT1BWU3NpL2dPcnppZE03MEEzMXhJVHc3WWZMaHBvVkExeG83b3Z1bGRwTGxocXk5bzV3aDI
4Mn1DcHFCVUF0Z3JTa0RHb2crSzdtDZnVXBypbF1pWnVHWnJ0ZzJYm2ZIuJVC3g0Wk0zdElqNndWZWNERVVxS
VNGzkZUMkVzbTBRWFVuZ01LRk1s0TVYz210dzJ3eFh1T3pVZURkNERJUHJ2K21XNXBvQVdyNk10c1YrSDJWUSt
aTC9rQm53V0hqu1RPyUdGaXNxWFkvYUgvMVB0Q1hBKzE1K11Jv2vtskjtDnjRGGek9YQUV0Uj1aSThsWU9KY
XJuWTdBeS9hTjZi0XVHZmZyYm8vaFZBY0w0V0RkaGtiQk4zbTh3K2c3NkxvQvh0ZUVldTA0QS8weExae1VCPc9
kczpYNTA5Q2VydG1maWNhdGU+PGRz01g1MD1DZXJ0aWZpY2F0ZT5NSU1ENmpDQ0F0S2dBd01CQWdJQkJEQU5CZ
2txaGtpRz13MEJBUXNGQURCTk1SQxdEZ11EV1FRRERBZ1iMjkWTFd0aE1Sa3dGd11EV1FRS0RCQk9iM2RwYm1
FZ1UyOXNkWFJwYjI1ek1SRXdEd11EV1FRTERBaFFTMGt0VkvWVFZERUxNQWtHQTFRUJ0TUNUR1V3SGhjTk1UW
XhNREkyTURjMU5ETXdXaGNOTVRnd09ESTJNRGMxTkRNd1dQk5NUkF3RGdZRFZRUUREQWvUyjI5a0xXTmhNUmt
3RndZRFZRUUtEQkJPYjNkcGJtRWdVmjlzZfhScGIyNXpNUkV3RhdZRFZRUUxEQWhRUzBrdFZFV1RWREVMTUfrR
0ExVUVCaE1DVEZVd2dnRW1NQTBHQ1NxR1NjYjNEUUVCFVQVE0SUJEd0F3Z2dFS0FvSUJBUUNiYmw1c0tCQ2p
TQjhUTWRhY31teC9XZk9qTvcxZ21Ja1ZKU1ky0EpiT1drQ1ZtdHpnW10Z2hmc1BRUGx1ZXUwRFRhbGEa3JTU
31oQ3Z6elBTR1B3Q0ZPYWhGL243aFFhMUYzVWFU3hUS3JGQzVut3dkTHp4S1JPM1dqVnRJR1JTWDJrdjFGZ1V
wUXk1RX15K3JzZ1N6sjU5ZFU1W1pkV3BkYUR1RhhWvn1EZx1zR15Q2JHny81SD1NDRDzDxP0cGVUR1dtTTZjV
VNUMDc5N1hEbGJFeFNUVEdRWEZKQT1rQ0NzeT1EWG5KYThuejBGRThmbWN2UUh1VTZr0VFiChpHak1kM0RXbEU
2bm83VWRDwUQxSDA0K3VzQnA1aGhDckFCNjcwTmRvVHJ0VG1HtkFGdDRKVDB2aXRqS0hx0utFSWQ2TghkY20yV
Gc5M2REY1dGdEFnTUJBQUDqZ2RRd2dkRXdEZ11EV1IwUEFRSC9CQVFEQWd1QU1FRUdBMVvKShdRNk1EZ3d0cUE
wb0RLR01HaDBkSEE2THk5a2MzTXVibTkzYvc1aExteDFMM0JyYVMxbV1XTjb1M0o1TDJ0eWJDOX1iMjkWTFd0a
ExtTn1iREJNQmdnckJnRUZCUWCNCQVFSQ1ENhdQQV1JS3dZQkJRVUhNQutHTUdoMGR1QTZMeT1rYzNNdWj1tOTN
hVzVoTG14MUwzQnJhUzFtWVd0MGIzSjVMMk55ZEM5eWIy0TBMV05oTG10eWREQRCZ05WSFE0RUZnUVUrMnRGc
XBOZTNHmjNZUjh5cUJaSW1WV1Mzd1V3RhDZRFZSMFRBUUgvQkFvd0F3RUIvekFOQmdrcWhraUc5dzBCQVfRkF
BT0NBUUVBRSt0dWQwNVhht002RkVaSFdUYzgrYm16LzZCMFhRWE41njrlv0jCaGNo0Wk1R2FkanFwU3N1dmTuK
3R1THE1bTZDTG8zZTRsWDJkSjdoc1BBdn1hTHFPSXB6ZzQ5VEDkaW1xbk9CMk83Nct5QWhUOHY5R1p0SDFFQ0h
YeF1zdX1TR01LdmQrTDVJapkUaXMzbGw0d1U4Rkh6eVjsTT1JUW53W1I1MDZqRmNKZUdsT2d5WmgrVUxXb1JOR
UV3cU44RFRGMkQwWg9nWUJzckN4Q0JqMFbwyUpGcnV2RVFxcFV1dV1nMTRSMURKrmFoTHdxV1LT0Q1Z1BobUE
wSFI0eJNHRjNqSFN6MGk5a1hTVE9zVWNka3ZVSnkwE1PbnVqc1VFa2czSDZXzNsejhUdzNjYzdWMU51YitNQ
zVLNFp2WCs1U115dTArZXi3YkZzY01yWVp3PT08L2Rz01g1MD1DZXJ0aWZpY2F0ZT48L2Rz01g1MD1EYXRhPjw
vZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWR1czpRdWFsaWZ5aW5nUHJvcGvYdG11cyB4bWxuczp4YWR1cz0ia
HR0cDovL3VyaS51dHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZQ9IiNpZC1hZmRlNzgyNDM2NDY4ZGQ3NGV
1YjE4MWY3Y2UxMTB1MSI+PHhhZGVz01NpZ251ZFByb3B1cnRpZXMgSWQ9InhhZGVzLW1kLWFmZGU30DI0MzY0N
jhkZDc0ZWV1MTgxZjdzTExMGUxIj48eGFkZXM6U21nbmVku21nbmF0dXJ1UHJvcGvYdG11cz48eGFkZXM6U21
nbmluZ1RpBwU+MjAxNy0w0S0y0FQxMTow0TowNFo8L3hhZGVz01NpZ25pbmdUaW1Pjx4YWR1czpTaWduaW5nQ

```

2VydGlmaWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlnZXN0TW0aG9kIEFsZ29
yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI3NoYTEiLz48ZHM6RGlnZXN0VmFsdWU+Y
ytWb2hnMGpJY1o0VVFTV2VnbENnMG9HTldzPTwvZHM6RGlnZXN0VmFsdWU+PC94YWRlczpDZXJ0RGlnZXN0Pjx
4YWRlczpJc3N1ZXJZXJpYWxWMj5NR113VWFsUE1FMHHFREFPQmd0VkJBTU1CMmR2YjJRdFkyRXhHVEFYQmdOV
kJBb01FRTV2ZDjsdV1TQ1RiMngxZEdsdmJuTXhFVEFQmd0VkJBc01DRkJMU1MxVVJWt1VNUXN3Q1FZRFZRUUd
Fd0pNV1FJQkNnPT08L3hhZGVz0k1zc3Vlc1NlcmlhbFYyPjwveGFkZXM6Q2VydD48L3hhZGVz01NpZ25pbmdDZ
XJ0aWZpY2F0ZVYyPjwveGFkZXM6U21nbmVku21nbmF0dXJ1UHJvcGVydG11cz48eGFkZXM6U21nbmVkRGF0YU9
iamVjdFByb3B1cnRpZXM+PHhhZGVz0kRhdfGFPYmp1Y3RGb3JtYXQgT2JqZWN0UmVmZXJ1bmN1PSIjci1pZC0xI
j48eGFkZXM6TW1tZVR5cGU+dGV4dC94bWw8L3hhZGVz0k1pbWVUeXB1PjwveGFkZXM6RGF0YU9iamVjdEZvcm1
hdD48L3hhZGVz01NpZ251ZERhdGFPYmp1Y3RQcm9wZXJ0aWVzPjwveGFkZXM6U21nbmVkUHJvcGVydG11cz48L
3hhZGVz01F1YWxpZnlpbmdQcm9wZXJ0aWVzPjwvZHM6T2JqZWN0PjwvZHM6U21nbmF0dXJ1Pg==",
    "digestAlgorithm" : null,
    "name" : "xades-detached.xml"
  }
}

```

Response

```

HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:37 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 11731

{
  "bytes" :
"PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGlubZ0iVVRGLTgiIHN0YW5kYWxbmU9Im5vIj8+PGRz01NpZ25hd
HVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyIgSWQ9Im1kLWFmZGU3ODI
0MzY0NjhkZDc0ZWV1MTgxZjdjZTExMGUxIj48ZHM6U21nbmVkSW5mbz48ZHM6Q2Fub25pY2FsaXphdG1vbk1ld
GhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxkczpTaWd
uYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwO18vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNc
2Etc2hhMjU2Ii8+PGRz01J1ZmVyZw5jZSBjZD0ici1pZC0xIiBuExB1PSIiIFVSS0ic2FtcGx1LnhbtCI+PGR
z0kRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1N
iIvPjxkczpEawd1c3RWYWy1ZT5rY0RIT1pqd1poVmZ1RGh1aEN1Q0VSUm1ZcFRINEpqNFJtZ1ZWaTMxUT1nPtw
vZHM6RGlnZXN0VmFsdWU+PC9kczpSZW1cmVuY2U+PGRz01J1ZmVyZw5jZSBuExB1PSJodHRwO18vdXJpLmV0c
2kub3JnLzAxOTAzI1NpZ251ZFByb3B1cnRpZXMiIFVSS0iI3hhZGVzLW1kLWFmZGU3ODI0MzY0NjhkZDc0ZWV
iMTgxZjdjZTExMGUxIj48ZHM6VHJhbnNmb3Jtcz48ZHM6VHJhbnNmb3JtIEFsZ29yaXRobT0iaHR0cDovL3d3d
y53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PC9kczpUcmFuc2Zvcm1zPjxkczpEaWd1c3RNZXRob2Q
gQWxnb3JpdGhtPSJodHRwO18vd3d3LnczLm9yZy8yMDAxLzA0L3htbGVuYyNzaGEyNTYiLz48ZHM6RGlnZXN0V
mFsdWU+RHp0d05UbVJvT0FtNi9stuk4UnltNXhaUHpjdkxZRHpuL2ViWV1rUHNyND08L2Rz0kRpZ2VzdFZhbHV
1PjwvZHM6UmVmZXJ1bmN1PjwvZHM6U21nbmVkSW5mbz48ZHM6U21nbmF0dXJ1VmFsdWUgSWQ9InZhbHV1LW1L
WFmZGU3ODI0MzY0NjhkZDc0ZWV1MTgxZjdjZTExMGUxIj5ZQTdzRU50M044dWZMRk1uS3IzNnIwUHF6TW1ZM1E

```

wcysrSUdURVVDMHNwYXhVdjBkSFpNMGQveW4za3BMSkxvVWtJNE0zzmxqNVdHbjgza2YwNUJxTTFrhNNyNjFHS
nphR1RQR3BtN2FrU1FLaHZvSDI1eX1xVF1YRVNsQmNtMDRpm1LaExNelpqVWZ4NC9CMVpJeXN2NXBJQmdKMnI
yb2k2akxvcD13dzNnZTRjNF1Kb2FLK1NYazZoeVR0T2N00FBqR2U2M1dZT1R0V1BRRnZqYThCbnndnK2EwYkJ1d
0QrOE42ZndpZ0NkVzVhLzRES1V1l0o4TWI3MFpJ0FBvT3puR0RmaStUUJpSWVwbUNibDVtVW9VZzJRL3hZbHV
KZkxoM3VHUUFS0J2RjQ1b0RJsfJWZwZuTi9EL1d5dEFdbFWRG9RU313Z1ua1BwcUY4ZwC9PTwvZHM6U21nb
mF0dXJ1VmFsdWU+PGRz0ktleUluZm8+PGRz0lg1MD1EYXRhPjxkczpYNTA5Q2VydG1maWNhdGU+TULJRDfEQ0N
BcnlnQXdJQkFnSUJDAkFOQmdrcWhraUc5dzBCQVFzRkFEQk5NUkF3RGdZRFZRUUREQWRuYjI5a0xXTmhNUmt3R
ndZRFZRUUteEqkJPYjNkcGJtRWdVMj1zZfHScGIyNXpNUkV3RHdZRFZRUUxEQWhRUzBrdFZfV1RWREVMTUfrR0E
xVUVCaE1DVEZVd0hoY05NVf14TURJmk1EYzFORE14V2hjTk1UZ3dPREkyTURjmu5ETXhXakJQTVJJd0VBWURWU
VFEREfbsmIy0WtMWFZ6W1hJeEdUQvhCZ05WQkFvTUVFNXZkMmx1WVNCVG1yeDFkR2x2Ym5NeEVUQVBCZ05WQkF
zTUNGQkxTUzFVU1ZOVU1Rc3dDUV1EV1FRR0V3Sk1WVENDQVNj0dRRWUpLb1pJaHZjTkFRRUJCUUFEZ2dFUEFEQ
0NBW9DZ2dFQkFMUkNVSFaYnczb1nkTHArQj1jekVDZ3Baa2tRNXhWNGc5TS83d2xn0TdvQ0NmN1VFaD1CQTF
kK3pZanN6ditCSjF1SlpQZ2FuMjE0NEF2Z3NvR0pmYjZVSX1WVzRna2xVZ01sMWfYVXZvbitXa0tuc2VGdVFPZ
kp5a1NGVURJd251dnAwaHpjSlhIWFJtTGRteWgrbis2Tk1IMG9tNXRwb1NmUXJ0Q1ZpQ0x1U01WenVENUVQajB
tSVJjeDkxcEwz0GUzRk5UVzd0YUdaTGvlenVGdV1vcTd60TNsTGt2WjRWQU10R0dMdk1YT111UkJaTX1QaHBCW
jRMM0E4STNFRWxL0gvMuX3aW1YVFTRzFzTTZxdk1UvmJmMnZiZDQ3b1pSUUEybVNWtKdqUW91T0FFcmZ1V1Z
VcXpJQ2doUUNIukdPTnVTTEcvSGZxRkh1NgpXZzBDQXfFQUFhT0J2RENCdVRBT0JnT1ZIUThCQWY4RUJBTUNCa
0F3Z1l1jR0NDc0dBUVVGQndFQkJ1c3d1VEE1QmdnckJnRUZCUWN3QV1ZdGFIUjBjRG92TDJSemN5NXViM2RwYm1
FdWJIVXZjR3RwTFdaaFkzUnZjbmt2YjJ0emND0W5iMj1rTFd0aE1Ed0dDQ3NHQVFVRkJ6QUNoakJvZEhSd09p0
HzSE56TG01dmQybHVZUzVzZFM5d2Eya3RabUZqZEc5eWVTOwPj1fWjI5d1pDMWpZUzVqY25Rd0hRWURWUjB
PQkJZRUZOMnBIRC83UGVmbUJU0G9YMj1aaFd5L091sJfNQTBHQ1NxR1NjYjNEUVCQ3dVQUE0SUJBuuJLM1ZPT
GhESVZXS29GcnJoaFd6YWRkdGs2WFF0Y3dSb05QV1NzaS9nT3J6c2RNNzBBMzF4SVR3N1lmtGhwb1ZBMXhvN29
2SGxkcExsaHF50W81d2gy0Dj5Q3BxQ1VbDgdyU2tER29nK0s3Q0w2Z1VwcmxZaVp1R1pydGcyWDNmSFMyVXN4N
FpKM3RJajZ3VmVjREVvC1lTrmZGVDJFc20wUVhVbmdJS0ZnbDk1WgtdHcyd3hYYk96VWVEZDRESVBydittVzV
wb0FXcjZjdhNWK0gyV1ErWkwva0Jjd1dIa1NUT2FHRmlzcVhZL2F1LzFQdEJYQSSxNstZsvd1bUpCU3Yza0Rhr
npPWEFFdFI5Wkk4bF1PSmFyb1k3QXkvYU42Yj11R2ZmcmJvL2hWQWNMNFdEZGhrYkJ0M204dytnNzZMb0FYtMv
FZxUwNEEvMhhMWnpVQjwvZHM6WDUwOUN1cnRpZm1jYXR1PjxkczpYNTA5Q2VydG1maWNhdGU+TULJRDzqQ0NBd
EtQXdJQkFnSUJCREFOQmdrcWhraUc5dzBCQVFzRkFEQk5NUkF3RGdZRFZRUUREQWR5YjI5MExXTmhNUmt3Rnd
ZRFZRUUteEqkJPYjNkcGJtRWdVMj1zZfHScGIyNXpNUkV3RHdZRFZRUUxEQWhRUzBrdFZfV1RWREVMTUfrR0ExV
UVCaE1DVEZVd0hoY05NVf14TURJmk1EYzFORE13V2hjTk1UZ3dPREkyTURjmu5ETx1XakJ0TVJBd0RnWURWUf
EREFkbmIy0WtMV05oTVJrd0Z3WURWUVFLREJCT2IzZHB1bUVnVTI5c2RYUnBiMjV6TVJFd0R3WURWUVFMREFoU
VMwa3RWRVZUVkRfTE1Ba0dBMVVFQmhNQ1RGVXdnZ0VpTUEwR0NTcUdTSWizRFFFQkFRVUFBNELCRHdBd2dnRut
Bb01CQVFDYmJsnXNLQkNQ0I4VE1kYWN5bXgv2ZPak1XMWdpSWpWS1JZMjhKYk5Xa0NwbXR6Z21pdGdoZnJQU
VBsdWV1MERUYWx1RGtyU1N5aEN2enpQU0dQd0NGT2FoRi9uN2hRYTFGM1VhSFN4VEtyRkM1bk93ZEx6eEtSTzN
Xa1Z0SUdSU1gya3YxRmZVcFF5NUV5eStyc2ZTeko10WRVNvPaZfdwZGFEduR4V1Z5RGVYm0VjeUNiRzcvNUg5T
UQ0WXV6TnB1VEZxbU02Y1VTvda30TZYRGx1RXhTVFRHUVhGSkEyK0NDc3k5RFhuSmE4bnowRkU4Zm1jd1F1ZVU
2az1RYnB6R2pNZDNEV2xFN5vN1Vq11EMUgwNCt1c0jwNWhoQ3JBQjY3ME5kb1RyT1RtR05BRnQ0S1Qwdm10a
kt1cT1LRU1kNkx0ZGntM1Rn0TNkRGNXRNZB01CQUFHamdkUXdnZEV3RGdZRFZSMFBUUJgvQkFRREFnZUFNRUV
HQTFVZEh3UTZNRGd3TnFBMG9ES0dNR2gwZEhBNkx50WtjM011Ym05M2FXNWhMbXgxTDNCcmFTMW1ZV04wYjNKN
UwyTn1iQz15YjI5MExXTmhMbU55YkRCTUJnZ3JCZ0VGQ1FjQkFRukFNRDR3UEFZSut3WUJCUVVTUFLR01HaDB
KSEE2THk5a2MzTXVibTkzYVc1aExteDFMM0jyYVMxbV1XTjB1m01TDj0eWRDOX1iMjkwTFd0aExtTn1kREFkQ
md0VkhRNEVGZ1FVkj0RnFwTmUzRz1zWV14eXFCWk1pV1dTM3ZVd0R3WURWUjBUQVFI0jBVXdbd0VCL3pBTkJ
na3Foa21HOXcwQkFRc0ZBQ9DQVFFQUUrTnVkMDVYR09NNkZFWkhXVGM4K2jtei82QjBYUVhONTY0S1dCQmhja
D1pNUdhZGpxcFNzZXZrb1t0ZUxxNW02Q0xvM2U0bFgyZEo3aHNQXZ5YUxxT0lwemc00VRHZG1iMW5PQjJPNzQ
reUFoVDh20UZadEgxRUNIWHhZc3V5U0dJS3Zk0w1SwpKVG1zM2xsNHzVOEZ1en1sB0E05SVFud1pSNTA2akZjS
mVHbE9neVpoK1VMV25StkVFd3F00ERURjJEMFhvZ1lCc3JDeENCajBQcGFkRnJ1dkVRcXBvdXVzzE0UjFESkZ
haEx3cVZZU09ENwdQaG1BMEhSNHozR0YzakhTejBp0WpYU1RPc1VjZGt2VUp5MHRJT251anNVRWtnM0g2V2czb
Ho4VHczSWM3VjFOSGIrTUM1SzRadlgrNvnZeXUwK2VvY2JGc2Njclladz09PC9kczpYNTA5Q2VydG1maWNhdGU
+PC9kczpYNTA5RGF0YT48L2Rz0ktleUluZm8+PGRz0k9iamVjd48eGFkZXM6UXVhbGlmeWluZ1Byb3B1cnRpZ
XMge1sbnM6eGFkZXM9Imh0dHA6Ly91cmkuZXrzaS5vcmcvMDE5MDMvdjEuMy4yIyIgVGfY2ZV0PSIjaWQtYwZ
kZTc4MjQzNjQ20GRkNzR1ZWIx0DFmN2N1MTEwZTEiPjx4YWR1czpTaWduZWRQcm9wZXJ0aWVzIE1kPSJ4YWR1c

y1pZC1hZmR1NzgyNDM2NDY4ZGQ3NGV1YjE4MWY3Y2UxMTB1MSI+PHhhZGVz01NpZ251ZFNpZ25hdHVyZVByb3B1cnRpZXM+PHhhZGVz01NpZ25pbmdUaW1lPjIwMTctMDktMjhUMTE6MDk6MDRaPC94YWRlczpTaWduaW5nVG1tZT48eGfkZXM6U21nbmluZ0N1cnRpZmljYXR1VjI+PHhhZGVz0kN1cnQ+PHhhZGVz0kN1cnREaWd1c3Q+PGRz0kRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMu3JnLzIwMDAvMDkveG1sZHNpZyNzaGEExIi8+PGRz0kRpZ2VzdFZhV1PmMrVm9oZzBqSWNaNFVRU1d1Z2xDzZBvR05Xcz08L2Rz0kRpZ2VzdFZhV1PjwveGFkZXM6Q2VydERpZ2VzdD48eGfkZXM6SXNzdWVvU2VyaWFsVjI+TUZZd1VhU1BNRTB4RURBT0JnT1ZCQU1NQjJkdmtIyUXRZMKV4R1RBWEJnT1ZCQW9NRUU1dmQybHVZU0JUYjJ4MWRhbHZibk14RVRBUEJnT1ZCQXNNQ0ZCTFNTMVVSVk5TVFzd0NRWURWUVFHRxDkTVZRSUJDZz09PC94YWRlczpJc3N1ZXJZXJpYWxWMj48L3hhZGVz0kN1cnQ+PC94YWRlczpTaWduaW5nQ2VydG1maWNhdGVWMj48L3hhZGVz01NpZ251ZFNpZ25hdHVyZVByb3B1cnRpZXM+PHhhZGVz01NpZ251ZERhdGFPYmp1Y3RQcm9wZXJ0aWVzPjx4YWRlczpEYXRhT2JqZWN0Rm9ybWF0IE9iamVjdFJ1ZmVyZW5jZT0iI3ItaWQtMSI+PHhhZGVz0k1pbWVUeXB1PnR1eHQveG1sPC94YWRlczpNaW1lVHlwZT48L3hhZGVz0kRhdGFPYmp1Y3RQb3JtYXQ+PC94YWRlczpTaWduZWRREYXRhT2JqZWN0UHJvcGVydg1lc48L3hhZGVz01NpZ251ZFBByb3B1cnRpZXM+PHhhZGVz01Vuc21nbmVkJc48eGfkZXM6WV5zaWduZWRtaWduYXR1cmVQcm9wZXJ0aWVzPjx4YWRlczpDb3VudGVyU21nbmF0dXJ1IE1kPSJDUy1pZC04NjZiNmNiMDc3MTk20TR10GU2ZDZhNjk2MmI00DRhYiI+PGRz01NpZ25hdHVyZSBJZD0iaWQtODY2YjZjYjA3NzE5Njk0ZTh1NmQ2YTY5NjJiNDg0YWIiPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpeM0aW9uTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PGRz01NpZ25hdHVyZU1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMu3JnLzIwMDEvMDQveG1sZHNpZy1tb3J1I3jzYS1zaGEyNTYiLz48ZHM6UmVmZXJ1bmN1IFR5cGU9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDE5MDMjQ291bnR1cnNpZ251ZFNpZ25hdHVyZSIgVVJJPSIjdmFsdWUtaWQtYWZkZTc4MjQzNjQ20GRkNzR1ZWIx0DFmN2N1MTEwZTEiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAxLzEwL3htbC1leGMtYzE0bimilz48L2Rz01RyYW5zZm9ybXM+PGRz0kRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMu3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPjxkczpEaWd1c3RWYw1ZT51bTBWdTF1MEgyS01pWkVjM1I2czEzQ3I1ZDjZTZTVFZXb1BHdndGaTdjPTwZHM6RG1nZXN0VmFsdWU+PC9kczpSZWZ1cmVuY2U+PGRz01J1ZmVyZW5jZSBUeXB1PSJodHRw0i8vdXjPmV0c2kub3JnLzAx0TAzI1NpZ251ZFBByb3B1cnRpZXMiIFVSST0iI3hhZGVzLW1kLTg2NmI2Y2IwNzcx0TY5NGU4ZTzKNmE20TYyYjQ4NGFjIj48ZHM6VHJhbNmb3Jtcz48ZHM6VHJhbNmb3JtIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PC9kczpUcmFuc2Zvcm1zPjxkczpEaWd1c3RNZXRob2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGVuYyNzaGEyNTYiLz48ZHM6RG1nZXN0VmFsdWU+Ym03RFRDZGtvZGRXNGNKQ212a2JQR1IwS1F0ZjUxTnVRWERXcVNBa2xYbz08L2Rz0kRpZ2VzdFZhV1PjwvZHM6UmVmZXJ1bmN1PjwvZHM6U21nbmVkJb5mbz48ZHM6U21nbmF0dXJ1VmFsdWUgSWQ9InZhbHV1LW1kLTg2NmI2Y2IwNzcx0TY5NGU4ZTzKNmE20TYyYjQ4NGFjIj50N31kKy9sdHVuaDUxejd3eEs4d2xRSXBCS2x6bj1UMS94SG9MRkZjV1NuV1Zob0thMHj0Nxg3RGR0bHI3V3dUukj1ukN0L11JL1hxNkhFS1NvM3Y3S2orWmxURG52aVY3UFFRcUdQVzY0STJBQWjS11veXJ6b1pnSGNDbn1ROTJhdndZQVvT2hid05XUk1U1ozQUxhaEV2d3o2a1Z0aXAzQk9jZ1E1a2hiazdyWm5HV3VoeFd1MUFTY2VEcnY0cXNMRmkvc0hTN2IxVmVQ0ENzMFdnRTiVDUrc1orR2E1aHNLYXkxTzhRSmc4N1NuK3BHcHd4SFhCeWFhNVZke1dBc1owVS95SytZWRsTG9Kcj4MUtLZudLSUFLZGQ3UGpJUnpZNkpGTzZHM1dqVF1FK3g3YUQwRkZNMDf1Yk9Gc3h1KzluTzhBZjQ2ayswUktjcfE9PTwvZHM6U21nbmF0dXJ1VmFsdWU+PGRz0k1eUluZm8+PGRz01g1MD1EYXRhPjxkczpYNTA5Q2VydG1maWnhGU+TUlJQzZqQ0NBZETnQxdJQkFnSuDmfd1VMTd0WE1BMEdu3FHu01iM0RRRUJd1VBTURBeEd6QVpCZ05WQkFNTUVsSnZiM1JUW1d4bVUybg5ibVZrUm1GclpURVJNQThHQTfVRUNnd01SRk5UTfhsbGMzUXdIaGNOTVRjd05qQTRNVEV5TmpBeFdoY050RGN3TnpBME1EYzF0ekkwV2pBb01STxdFUV1EV1FRRERBcFRhV2R1W1hKR11xdGxNUK3RhdZRFZRUUteQWhFVTfndGRHVnpkRENDQVNj0RRWUpLb1pJaHzjTkFRRUJCUUFEZ2dFUEFEQ0NBWU9DZ2dFQkFNSTNrWmh0bm1wb1tpaVpIWj1heDhGbgZFNu93L2NGd0JUZkFFYjNSMVpRVXA2L0JRBkJ0N09vMEpXQnRj0XFrdjdKVURkY0JkWFBWNVFXUzVBeU1QSHBxUTc1SG10anNxL0Z6dTh1SHRrS3BGAxPjeEdhOUJaZGtRamt0c1NydE8xS2pzMFjkNURRdFdT2z1VkNDTjA5a04wWhNaMEVOWStJcDhReFNteXp0c1N0a11YFVMcXB3ejRKRvhXOXZ6NjR1VGjkZTR2Uuo2cGpIR2FySmYxZ1FORWMyWhpobUkvcHJYTH1zV05xQzdsWmc3UFVaVVRyZGVnQUJUVXpZQ1JKMwtXQ1JQbTRxbzBMTjQwNWM5NFRZDQ1YTVrVGdvd0h6RwdMbkrFSTI4eDBNM0E10VRLQytPZU5jn1ZGMVBzVExwVXc3UE5JM1ZzdFg1akF1YXNDQxdFQUfhTVNNQkF3RGdZRFZSMFBuuUgvQkFRREFnRUDNQTBHQ1NxR1NjYjNEUUVCQ3dVQUE0SUJBUNLNkxHQTAXF1r1cm1VOHA2eWhBaTRPa0ROMmIxZGJJTDhs0G1DTV1vcExDeHg4eHFxM3ViWkNPeHFoMVgyajZwZ1d6YXj1MGIvTVVpeDAwSW9Vdk5iRk94QVc3UEJaSUtETG5tNkxzY2tSeHMxVTMyC0M5ZDFMT0h1M1dLQk5CNkdaQuXuMWV3amg3aFn1V2pmdGxtY292cSs2ZVZHQTVjdmYyds8yK1RrS2t5SFYvT1Iz0TRuWHJkc2Rwdn1nd31wRXRYamV0ekQ3VVQ5M051dzN4Y1Y4Vk1mdE12SGY5TGPvn2grVWpHbUtYRz1

```

jMTVlWXIzU3pVbXY2a3TPSTBCdnCxNFBXdHNXR2wwJWRPU1J2SUJCc1A0YWRDbkdUZ2pnams5TFRjTzhCOEZLc
nIr0GxIR3VjMGJwNGxJVRvaVVrR01MWHNpRWVFZz1XQXFtK1hxTzvZHM6WDUwOUN1cnRpZm1jYXR1PjwvZHM
6WDUwOURhdGE+PC9kczpLZX1JbmZvPjxkczpPYmp1Y3Q+PHhhZGVz01F1YWxpZnlpbmdQcm9wZXJ0aWVzIFRhc
mldD0iI21kLTg2NmI2Y2IwNzcx0TY5NGU4ZTzknmE20TYyYjQ4NGF1Ij48eGFkZXM6U21nbmVkuHJvcGVydG1
1cyBJZD0ieGFkZXMtaWQt0DY2YjZjYjA3NzE5Njk0ZTh1NmQ2YTY5NjJiNDg0YWIiPjx4YWR1czpTaWduZWRTa
WduYXR1cmVQcm9wZXJ0aWVzPjx4YWR1czpTaWduaW5nVG1tZT4yMDIxLTA3LTA5VDA40jAy0jM3WjwveGFkZXM
6U21nbm1uZ1RpbWU+PHhhZGVz01NpZ25pbmdDZxJ0aWZpY2F0ZVYyPjx4YWR1czpDZxJ0Pjx4YWR1czpDZxJ0R
G1nZXN0PjxkczpEaWd1c3RNZXRob2QgQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGV
uYyNzaGE1MTIiLz48ZHM6RG1nZXN0VmFsdWU+MTR3TWpERnpmc3FrZFpVc25QSDAvaCtabzhyejhERXdptXEyW
U8xd05UZnBsTDN3a1E3RTBsMn1aUf1kZXFHs1TeENUbHpwMTFStkVVZUxLTXNDZVE9PTwvZHM6RG1nZXN0VmF
sdWU+PC94YWR1czpDZxJ0Rg1nZXN0Pjx4YWR1czpJc3N1ZXJTZXJpYWxWMj5NRDR3TktReU1EQXhHekFaQmd0V
kJBTU1FbEp2YjNSVFpXeG1VmxxuYm1Wa1JtRnJaVEVSTUE4R0ExVUVdz3dJUkZ0VExYUmxjM1FDQmk3V0Z0ZTd
Wdz09PC94YWR1czpJc3N1ZXJTZXJpYWxWMj48L3hhZGVz0kN1cnQ+PC94YWR1czpTaWduaW5nQ2VydG1maWNhd
GVWMj48L3hhZGVz01NpZ251ZFNpZ25hdHVyZVByb3B1cnRpZXM+PC94YWR1czpTaWduZWRQcm9wZXJ0aWVzPjw
veGFkZXM6UXVhbGlmeWluZ1Byb3B1cnRpZXM+PC9kczpPYmp1Y3Q+PC9kczpTaWduYXR1cmU+PC94YWR1czpDb
3VudGvyU21nbmF0dXJ1PjwveGFkZXM6VW5zaWduZWRTaWduYXR1cmVQcm9wZXJ0aWVzPjwveGFkZXM6VW5zaWd
uZWRQcm9wZXJ0aWVzPjwveGFkZXM6UXVhbGlmeWluZ1Byb3B1cnRpZXM+PC9kczpPYmp1Y3Q+PC9kczpTaWduY
XR1cmU+",

    "digestAlgorithm" : null,
    "name" : "xades-detached-counter-signed-xades-baseline-b.xml"
}

```

REST server signature service

This service also exposed 4 methods :

Rest server signing service

```
// Instantiate a RestSignatureTokenConnection
RestSignatureTokenConnection remoteToken = new RestSignatureTokenConnectionImpl();

// Retrieves available keys on server side
List<RemoteKeyEntry> keys = remoteToken.getKeys();

String alias = keys.get(0).getAlias();

// Retrieves a key on the server side by its alias
RemoteKeyEntry key = remoteToken.getKey(alias);

DSSDocument documentToSign = new InMemoryDocument("Hello world!".getBytes());

// Create a toBeSigned DTO
ToBeSignedDTO toBeSigned = new ToBeSignedDTO(DSSUtils.toByteArray(documentToSign));

// Signs the document with a given Digest Algorithm and alias for a key to use
// Signs the digest value with the given key
SignatureValueDTO signatureValue = remoteToken.sign(toBeSigned,
DigestAlgorithm.SHA256, alias);

// Or alternatively we can sign the document by providing digest only

// Prepare digestDTO.
// NOTE: the used Digest algorithm must be the same!
DigestDTO digestDTO = new DigestDTO(DigestAlgorithm.SHA256,
DSSUtils.digest(DigestAlgorithm.SHA256, documentToSign));

// Signs the digest
SignatureValueDTO signatureValueFromDigest = remoteToken.signDigest(digestDTO, alias);
```

Get keys

This method allows retrieving of all available keys on the server side (PKCS#11, PKCS#12, HSM,...). All keys will have an alias, a signing certificate and its chain. The alias will be used in following steps.

Request

```
GET /services/rest/server-signing/keys HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Host: localhost:8080
```

Response

```
HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:38 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 2189
```

```
[ {
  "alias" : "certificate",
  "encryptionAlgo" : "RSA",
  "certificate" : {
    "encodedCertificate" :
    "MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMELJvb3RTZWxmU21nbmVkr
    mFrZTERMA8GA1UECgwIRFNTLXR1c3QwHhcNMTCwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
    DDApTaWduZXJGYWtLMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
    MI3kZhtnipn+iiZH9ax8F1fE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
    pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
    sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC71Zg7PUZUTrdegABTUzYCR
    J1kWBRPm4qo0LN405c94QD45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
    wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmu8p6yhAi40kDN2b1
    dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUi x00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
    9d1L0He3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
    D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6ky0I0Bvw14PWtsWG10Qd0SRvIBBrP4adCnGT
    gjgjk9LTc08B8FKrr+81HGuc0bp4lIUToiUkGILXsiEeEg9WAqm+Xqo"
  },
  "certificateChain" : [ {
    "encodedCertificate" :
    "MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMELJvb3RTZWxmU21nbmVkr
    mFrZTERMA8GA1UECgwIRFNTLXR1c3QwHhcNMTCwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
    DDApTaWduZXJGYWtLMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
    MI3kZhtnipn+iiZH9ax8F1fE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
    pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
    sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC71Zg7PUZUTrdegABTUzYCR
    J1kWBRPm4qo0LN405c94QD45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
    wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmu8p6yhAi40kDN2b1
    dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUi x00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
    9d1L0He3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
    D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6ky0I0Bvw14PWtsWG10Qd0SRvIBBrP4adCnGT
    gjgjk9LTc08B8FKrr+81HGuc0bp4lIUToiUkGILXsiEeEg9WAqm+Xqo"
  } ]
}
```

Get key

This method allows retrieving a key information for a given alias.

Request

```
GET /services/rest/server-signing/key/certificate HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Host: localhost:8080
```

Response

```
HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:38 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 2185

{
  "alias" : "certificate",
  "encryptionAlgo" : "RSA",
  "certificate" : {
    "encodedCertificate" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMME1Jvb3RTZWxmU21nbmVkR
mFrZTERMA8GA1UECgwIRFNTLXR1c3QwHhcNMTCwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtLMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZH9ax8F1fE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC71Zg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QD45a5kTgowHzEgLnaQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmu8p6yhAi40kDN2b1
dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUi x00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1L0He3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6ky0I0Bvw14PWtsWG10Qd0SRvIBBrP4adCnGT
gjgjk9LTc08B8FKrr+81HGuc0bp4lIUToiUkGILXsiEeEg9WAqm+Xqo"
  },
  "certificateChain" : [ {
    "encodedCertificate" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMME1Jvb3RTZWxmU21nbmVkR
mFrZTERMA8GA1UECgwIRFNTLXR1c3QwHhcNMTCwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtLMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZH9ax8F1fE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC71Zg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QD45a5kTgowHzEgLnaQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmu8p6yhAi40kDN2b1
dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUi x00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1L0He3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6ky0I0Bvw14PWtsWG10Qd0SRvIBBrP4adCnGT
gjgjk9LTc08B8FKrr+81HGuc0bp4lIUToiUkGILXsiEeEg9WAqm+Xqo"
  } ]
}
```

Sign

This method allows signing of given digests with a server side certificate.

Request

```
POST /services/rest/server-signing/sign/certificate/SHA256 HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 388

{
  "bytes" :
"MYIBDzAYBqkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMTA3MDkwODAyMzhaMCsGC
SqGSIb3DQEJNDEeMBwwCwYJYIZIAWUDBAIBoQ0GCSqGSIb3DQEBCwUAMC8GCSqGSIb3DQEJBDEiBCCRwMc5mPB
mFV+40G6EJ4IRFGZi1MfgmPhGZ9VWLfVD2DB3BgsqhkiG9w0BCRACLzFoMGYwZDBiBCAC8+vKAWMnQl08gJ0nS
Y3UG7AxbX5rBmlgEV3hVVidnDA+MDSkMjAwMRswGQYDVQQDBJSb290U2VsZ1NpZ25lZEZha2UxETAPBgNVBAo
MCERTUy10ZXN0AgYu1hTXu1c="
}
```

Response

```
HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:38 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 395

{
  "algorithm" : "RSA_SHA256",
  "value" :
"1X91IFs5lYzbdyJiGRm8k+DmPejpo2SCoZJmVGX3+1E1hBxDkrtsckjGkqS94W3ZCK0giC1CMXSj/eghcBbTJ
J27w2QRsxnqNn5i8obn5qTUef0v+9v1cDNL5S10AZx7Lq8D2QazK8nj5H2RVeWczc2VBWhCHm6NKodA0/oGuvQ
v/+r43rb/WVBj8zj+fH2MK3EALVj3x6l/lgHTrlk7cTKbMBA7AB/o1F0XH1gPyrti8F5LALSq6gkjcSC7yPhpJ
STQ01J769V0XYFZR4YA3sJXMZZY1QKRWWU/PT5ZGR/NVv6gjxsKxdixybjhV05tYjiLzd5VGd5k+D+4gI8Sw=
"
}
```

REST validation service

DSS provides also a module for documents validation.

Validate a document

This service allows a signature validation (all formats/types) against a validation policy.

Request

```
POST /services/rest/validation/validateSignature HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 7404

{
  "signedDocument" : {
    "bytes" :
"PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGlubz0iVVRLTgiPz48ZHM6U21nbmF0dXJ1IHhtbG5z0mRzPSJod
HRw0i8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIiBJZD0iaWQtYWZkZTc4MjQzNjQ20GRkNzR1ZWIx0DF
mN2N1MTEwZTEiPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpeM0aW9uTWV0aG9kIEFsZ29yaXRobT0ia
HR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXjhLWMxNG4jIi8+PGRz01NpZ25hdHVyZU1ldGhvZCBBbGd
vcm10aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3J1I3JzYS1zaGEyNTYiLz48ZHM6U
mVmZXJ1bmN1IE1kPSJyLWlkLTEiIFR5cGU9IiIgVVJJPSJzYW1wbGUueG1sIj48ZHM6RGlnZXN0TWV0aG9kIEF
sZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hhMjU2Ii8+PGRz0kRpZ2VzdFhb
HV1PmtjREhPWmp3WmhWZnVEaHV0Q2VDRVJSbV1wVEg0Smo0Um1mV1ZpMzFROWc9PC9kczpEaWd1c3RWYWx1ZT4
8L2Rz01J1ZmVyZW5jZT48ZHM6UmVmZXJ1bmN1IFR5cGU9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDE5MDMjU21nb
mVkuHJvcGVydG1lcIgVVJJPSIjeGFkZXMaWQtYWZkZTc4MjQzNjQ20GRkNzR1ZWIx0DFmN2N1MTEwZTEiPjx
kczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRw0i8vd3d3LnczLm9yZy8yMDAxLzEwL
3htbC1leGMtYzE0b1miLz48L2Rz01RyYW5zZm9ybXM+PGRz0kRpZ2VzdE11dGhvZCBBbGdvcm10aG09Imh0dHA
6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPjxkczpEaWd1c3RWYWx1ZT5EenR3T1RtUm9PQ
W02L2xNSThSeW01eFpQek12TF1Eem4vZWJZWWtQc3I0PTtwZHM6RGlnZXN0VmFsdWU+PC9kczpSZWZ1cmVuY2U
+PC9kczpTaWduZWRJbmZvPjxkczpTaWduYXR1cmVWYWx1ZSBJZD0idmFsdWUtaWQtYWZkZTc4MjQzNjQ20GRkN
zR1ZWIx0DFmN2N1MTEwZTEiP11BN3NFTnQzTjh1ZkxGTW5LcjM2cjbQcXpNaVkjUTBzKytJR1RFVUMwc3BheFV
2MGRIWk0wZC95bjNrcExKTG9Va0k0TTNmBGo1V0du0DNrZjA1QnFNMWtoc1g2MUDkemFGVFBHcG03YWtSUUtod
m9IMjV5eXFUWVhFU2xCY20wNG16aUtoTE16WmpVZng0L0IxWk15c3Y1cE1CZ0oycjJvaTzqTG9wOXd3M2d1NGM
0WUpvYUsrU1hrNmh5VE5PY044UGpHZTYzV11PVE5WUFFGdmp0EJjud2crYTBiQnV3RCs4TjZmd21nQ2RXNWEvN
ERKVWUvSjhNYjcwWkk4UG9Pem5HRGZpK1RQYm1JZVztQ2jsNW1Vb1VnM1EveF1sdUpmTGgzdUdRQVhLQnZGNDV
vRE1IU1Z1Zm50L0QvV310QUNsVVZEb1FTeXd1bW5qUHBxRjh1Zz09PC9kczpTaWduYXR1cmVWYWx1ZT48ZHM6S
2V5SW5mbz48ZHM6WDUwOURhdGE+PGRz0lg1MD1DZXJ0aWZpY2F0ZT5NSU1EMURDQ0FyeWdBd01CQWdJQkNqQ5
CZ2txaGtpRz13MEJBUXNGQURCTk1SQXdEZ11EV1FRRERBZG5iMj1rTFd0aE1Sa3dGd11EV1FRS0RCQk9iM2RwY
m1Fz1UyOXNkWFJwYjI1ek1SRXdEd11EV1FRTERBaFFTMGt0VkvWVFZERUxNQWtHQTfVRUJoTUNUR1V3SGhjTk1
UWXhNREkyTURjMU5ETXhXaGNOTVRnd09ESTJNRGMxTkRNeFdqQ1BNuk13RUFZRFZRUUREQWxuYjI5a0xYVnpaW
E14R1RBWEJnT1ZCQW9NRUU1dmQybhVZU0JUYjJ4MWRhbHzibk14RVRBUEJnT1ZCQXNNQ0ZCTFNTMVSVk5VTVF
zd0NRWURWUWFHRXdKTVZUQ0NBu013RFFZSktrWk1odmNOQVFFQkJRQURnZ0VQQRDQ0FRb0NnZ0VCQUxSQ1VJU
VpidzNuU2RMcCtCOWN6RUNncFpra1E1eFY0Zz1NLzd3bGc5N29DQ2Y3VUVoOUJBMWQrellqc3p2K0jKMWjKw1B
nYW4yMTQ0QXZnc29HSmZiN1VJeVZNGdrbFvNShwxYXJVdm9u1k1drS25zZUZ1UU9mSn1qU0ZVRE13bnV2cDBoe
mNKWEhYUm1MZG15aCtuKzZOTUgbw201dFZvU2ZRcnRCVm1DTGVTtvZ6dUQ1RVBqMG1JUmN40TFwTDM4ZTNGT1R
XN05hR1pMZWV6dUZ1Ui9xN3o5M2xMa3zaNFZBTU5HR0x2SVhPWWVSQLpNeVBocEJaNEwzQThJM0VFbEtXSC8xT
HdpaVhUVFNHMXNNN1d2TVRWYmYydmJkNDduW1JRTQJtU3B0R2pRb3VPQUVYzmvWV1Vxe1DZ2hRQ0hSR090dVN
```

MRy9IZnFGSGI0aldnMENBd0VBQWFQnZEQ0J1VEFPQmdOVkhROEJBZjhFQkFNQ0JrQXdWWNHQ0NzR0FRVUZCd0VCQkhzd2VUQTCZ2dyQmdFRkJRY3dBWV10YUhSMGNEb3ZMM1J6Y3k1dWIzZHBibUV1YkhVdmNHdHBMV1poWTNSdmNua3ZiMk56Y0M5bmIy0WtMV05oTUR3R0NDc0dBUVVGQnpBQ2hQm9kSFJ3T2k4dIpITnpMbTV2ZDjsdV1TNXNkUz13YTJrdFptRmpkRz15ZVM5amNuUXZamj12WkMxa11TNWpjblF3SFFZRFZSME9CQ11FRk4ycEhELzdQZWZtQ1Q4b1gy0VpoV3kvT0hKMU1BMEdDU3FHU01iM0RRRUJd1VBQTRJQkFRQkszV9MaERJ1dLb0ZycmhoV3phZGR0azZYUXRjd1JvT1BWU3NpL2dPcnpzE03MEEzMXhJVh3WWZMaHBvVkExeG83b3ZIBGRwT6xocXk5bzV3aDI4Mn1DcHFCVUF0Z3JTa0RHb2crSzdtDZnVXBBybF1pWnVHWnJ0ZjYM2ZIuJVC3g0Wk0zdElqNndWZWNERVVxSVNGzkZUMkVzbTBRWFVuZ01LRk1s0TVY2Z10dzJ3eFh1T3pVZURkNERJUHJ2K21XNXBvQVdyNk10c1YrSDJWUStaTC9rQm53V0hqu1RPYUdGaXNxWFkvYUgvMVB0Q1hBKzE1K11JV2VtSkJtdjNrRGFGek9YQUV0Uj1aSThsWU9KYXJuWTdBeS9hTjZi0XVHZmZyYm8vaFZBY0w0V0RkaGtiQk4zbTh3K2c3NkxvQVh0ZUV1dTA0Q8weExae1VCPC9kcZpYNTA5Q2VydG1maWNhdGU+PGRz01g1MD1DZJ0aWZpY2F0ZT5NSU1ENmpDQ0F0S2dBd01CQWdJQkJEQU5CZ2txaGtpRz13MEJBUXNGQURCTk1SQxdEZ11EV1FRRERBZH1iMjkTFd0aE1Sa3dGd11EV1FRS0RCQk9iM2RwYm1FZ1Uy0XNkWFJwYjI1ek1SRXdEd11EV1FRTERBaFFTMGt0VkvWVVFZERUxNQWtHQTFRVUJ0TUNUR1V3SGhjTk1UWXhNREkyTURjMU5ETXdXaGNOTVRnd09ESTJNRGMxTkRnd1dqQk5NUkF3RGdZRFZRUUxEQWhRUzBrdFZFV1RWREVMUFrR0ExVUVCaE1DVEZVd2dnRW1NQTBHQ1NxR1NJyjNEUUVCVFVQUE0SUJE0F3Z2dFS0FvSUJBUUNiYmw1c0tCQ2pTQjhUTWRhY31teC9XZk9qTVcxZ21Ja1ZKU1ky0EpiT1drQ1ZtdHpnBw10Z2hmc1BRUGx1ZXUwRFRhbGJEa3JTU31oQ3Z6elBTR1B3Q0ZPYWhGL243aFFhMUYzVWFU3hUS3JGQzVvT3dkTHp4S1JPM1dqVnRJR1JTWDJrdjFGZ1VwUXk1RX15K3JzZ1N6SjU5ZFU1WlpkV3BkYUR1RhhWVn1EZXiZRU15Q2JHny81SD1NRDRZdXp0cGVUR1dtTTZjVVNUMDc5N1hEbGJFeFNUVEdRWEZKQT1rQ0NzeT1EWG5KYThejBGRThmbWN2UUh1VTzr0VFiChpHak1kM0RXbEU2bm83VWRDwUQxSDA0K3VzQnA1aGhDckFCNjcwTmRvVHJ0V61HTkFGdDRKVDB2aXRqS0hx0utFSWQ2TGHkY20yVGc5M2REY1dGdEFnTUJBQUDqZ2RRd2dkRXdEZ11EV1IwUEFRSC9CQVFEQWd1QU1FRUdBMVVkSHdRnk1EZ3d0cUEwb0RLR01HaDBkSEE2Thk5a2MzTXVibTkzYVc1aExteDFMM0JyYVMxbV1XTjBim0o1TDJ0eWJD0X1iMjkTFd0aExtTn1iREJNQmdnckJnRUZCUWNCQVFSQ1ENhdQQV1JS3dZQkJRVUhnQuHTUdoMGR1QTZMeT1rYzNNdWJtOTNhVzVoTG14MUwzQnJhUzFtWvdomG1zSjVMMk55ZEM5eWiyOTBMV05oTG10eWREQWRCZ05WSFE0RUZnUVUrMnRGcXBOZTNHMjNZUjh5cUJaSw1WV1Mzd1V3RHdZRFZSMFRUUgvQkFVd0F3RUIvekFOQmdrcWhraUc5dzBCQVFzRkFBT0NBUUVBRSt0dWQnVhHT002RkVaSFdUYzgrYm16LzZCMFhRWE41njRLV0JCaGNo0Wk1R2FkanFwU3N1dmTuK3R1THE1bTZDTG8zZTRsWDJkSjdoc1BBdn1hTHFPSXB6ZzQ5VEDkaWIxbk9CMk83Nct5QWhUOHY5R1p0SDFFQ0hYeF1zdX1TR01LdmQrTDVJakpUaXMzbGw0d1U4Rkh6eVj5TT1JUW53W1I1MDZqRmNKZUdsT2d5WmgrVUxXb1JORUV3cU44RFRGMkQwWg9nWUJzckN4Q0JqMFbwYUpGcnV2RvFxcFV1dV1nMTRSMURKRmFoTHdxV1LTT0Q1Z1BobUEwSFI0ejNHRjNqSFN6MGk5a1hTVE9zVWNka3ZVSnkwd1PbnVqc1VFa2czSDZXZzNsejhUdzNjYzdWMU51YitNQzVLNfp2WCs1U115dTArZXi3YkZzY01yWVp3PT08L2Rz01g1MD1DZJ0aWZpY2F0ZT48L2Rz01g1MD1EYXrhPjwvZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWR1czpRdWfsaWZ5aW5nUHJvcGVydG11cyB4bWxuczp4YWR1cz0iaHR0cDovL3VyaS51dHNpLm9yZy8wMTkwMy92MS4zLj1j1iBUYXJnZQ9IiNpZC1hZmR1NzgyNDM2NDY4ZGQ3NGV1YjE4MWY3Y2UxMTB1MSI+PHhhZGVz01NpZ251ZFByb3B1cnRpZXMgSWQ9InhhZGVzLw1kLWFmZGU3OD10MzY0NjhkZDc0ZWV1MTgxZjdjZTExMGUxIj48eGFkZXM6U21nbmVku21nbmF0dXJ1UHJvcGVydG11cz48eGFkZXM6U21nbmluZ1RpbWU+MjAxNy0wOS0y0FQxMTow0TowNFo8L3hhZGVz01NpZ25pbmduaW11Pjx4YWR1czpTaWduaW5nQ2VydG1maWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlzXN0TWW0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc21nI3NoYTEiLz48ZHM6RGlzXN0VmfsdWU+YytWb2hnMGpJY1o0VVFV2VnbENnMG9HT1dzPTwvZHM6RGlzXN0VmfsdWU+PC94YWR1czpDZJ0RGlzXN0Pjx4YWR1czpJc3N1ZJXTXJpYwXWMj5NR113WFSUE1FMHhFREFPQmd0VkjBTU1CMmR2YjJrdFkyRXhHVEFYQmd0VKjBb01FRTV2ZDjsdV1TQ1RiMngxZEsdmJuTxhFVEFQmd0VkjBc01DRkJMU1MxVVJWt1VNUXN3Q1FZRFZRUUdFd0pNv1FJQkNnPT08L3hhZGVz0k1z3V1c1n1cm1hbFYYpjwveGFkZXM6Q2VydD48L3hhZGVz01NpZ25pbmdDZXJ0aWZpY2F0ZVYyPjwveGFkZXM6U21nbmVku21nbmF0dXJ1UHJvcGVydG11cz48eGFkZXM6U21nbmVkrGF0YU9iamVjdFByb3B1cnRpZXM+PHhhZGVz0kRhdGFpYmp1Y3RQb3JtYXQgT2JqZWN0UmVmZJ1bmN1PSIjci1pZC0xIj48eGFkZXM6T1tZVR5cGU+dGV4dC94bWw8L3hhZGVz0k1pbWVUeXb1PjwveGFkZXM6RGF0YU9iamVjdEZvcm1hdD48L3hhZGVz01NpZ251ZERhdGFpYmp1Y3RQcm9wZXJ0aWVzPjwveGFkZXM6U21nbmVkuUHJvcGVydG11cz48L3hhZGVz01F1YwpxZnlpbmdQcm9wZXJ0aWVzPjwvZHM6T2JqZWN0PjwvZHM6U21nbmF0dXJ1Pg==",
 "digestAlgorithm" : null,
 "name" : "xades-detached.xml"
 },

```

"originalDocuments" : [ {
    "bytes" :
"77u/PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGlubZ0iVVRGLTgiPz4NCjxoOnRhYmxlIHhtbG5z0mg9Imh0d
HA6Ly93d3cudzMu3JnL1RSL2h0bWw0LyI+DQoJP6dHI+DQoJCTxoOnRkPkhlbGxvPC9o0nRkPg0KCQk8aDp
0ZD5Xb3JsZDwvaDp0ZD4NCgk8L2g6dHI+DQo8L2g6d6FibGU+",
    "digestAlgorithm" : null,
    "name" : "sample.xml"
} ],
"policy" : null,
"tokenExtractionStrategy" : "NONE",
"signatureId" : null
}

```

Response

```

HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:37 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 50391

{
    "validationReportDataHandler" :
"PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGlubZ0iVVRGLTgiIHN0YW5kYWxvbmU9InllcyI/Pgo8VmFsaWRhd
Glvb1JlcG9ydCB4bWxucz0iaHR0cDovL3VyaS51dHNpLm9yZy8xOTEwMi92MS4yLjEjIiB4bWxuczpuZ9Imh
0dHA6Ly93d3cudzMu3JnLzIwMDAvMDkveG1sZHNpZyMiIHhtbG5z0m5zM0iaHR0cDovL3VyaS51dHNpLm9yZ
y8wMTkwMy92MS4zLjIjIiB4bWxuczpuZ9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDIyMzEvdjIjIj4KICAgIDx
TaWduYXR1cmVWYWxpZGF0aW9uUmVwb3J0PgogICAgICAgIDxTaWduYXR1cmVJZGVudGlmaWVYIG1kPSJTLUEyN
DFCQ0ExNzI3MUQ2MDNCMzI3MTBFNTAzNTM20EZBRkJDNThGNj1DQ0M2NKE0MzJFQzcZ0EM5QjRDMjc1M0EiPgo
gICAgICAgICAgICA8RG1nZXN0QWxnQW5kVmFsdWU+CiAgICAgICAgICAgICA8bnMy0kRpZ2VzdE1ldGhvZ
CBBbGdvcm10aG09Imh0dHA6Ly93d3cudzMu3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPgogICAgICAgICA
gICAgICAgPG5zMjpEaWd1c3RWYwX1ZT4rakxLZXZ2MDhaTWdZL2ZSbnZjN0RneTlaSnBGUktmM01ZTWUrK2t60
FBNPTwvbnMy0kRpZ2VzdFZhbHV1PgogICAgICAgICA8L0RpZ2VzdEFsZ0FuZFZhbHV1PgogICAgICAgICA
gICA8bnMy0lNpZ25hdHvYzVZhbHV1P11BN3NFTnQzTjh1ZkxGTW5Lcjm2cjbQcXpNaVkzUTBzKytJR1RFVUmc
3BheFV2MGRIWk0wZC95bjNrcExKTG9Va0k0TTNmbGo1V0duODNrZja1QnFNMWtoc1g2MUDKemFGVFBHcG03Ywt
SUUtdm9IMjV5eXFUWVhFU2xCY20wNG16aUtoTE16WmpVZng0L0IxWk15c3Y1cE1CZ0oycjJvaTzqTG9wOXd3M
2d1NGM0WUpvYUsrU1hrNmh5VE5PY044UGpHZTYzV11PVE5WUFFGdmpbOEJud2crYTBiQnV3RCs4TjZmd2lnQ2R
XNWEvNERKVWUvSjhNYjcwkk4UG9Pem5HRGZpK1RQYm1JZVztQ2jsNW1Vb1VnM1EveF1sdUpmTGgzdUdRQVhLQ
nZGNDVvRE1I1Z1Zm50L0QvV310QUNsVVZEB1FTeXd1bW5qUHBxRjh1Zz09PC9uczI6U21nbmF0dXJ1VmFsdWU
+CiAgICAgICAgICAgIDxIYXNoT25seT5mYWxzZTwvSGFzaE9ubHk+CiAgICAgICAgIDxEQU1kZW50aWZpZXi+aWQtYWZkZTc4MjQzNjQ20GR
kNzR1ZWIxODFmN2N1MTEwZTE8L0RBSWR1bnRpZm11cj4KICAgICAgICA8L1NpZ25hdHVyZU1kZW50aWZpZXi+C

```


mVyzW5jZT0iQy02RjM1REUzOTY1Qj1BNj1CQzM2NjFEMUEzNTVCMEFFnjA5MDdBreI3NDFDQzE5MTFFRkQwRjNCRTcyRDZBNkU5Ii8+CiAgICAgICAgICAgICA8L0N1cnRpZm1jYXR1Q2hhaW4+CiAgICAgICAgICAgICA8Q3J5cHRvSW5mb3JtYXRpb24+CiAgICAgICAgICAgICA9PfZhbG1kYXRpb25PYmp1Y3RjZCBWT1J1ZmVzW5jZT0iUy1BMjQxQkNBMTcyNzFENjAzQjMyNzEwRTUwMzUzNjhGQUZCQzU4RjY5Q0NDNjZBNDMyRUM3MzhDOUI0QzI3NTNBIi8+CiAgICAgICAgICAgICA9PEFsZ29yaXRobT5odHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNyc2Etc2hhMjU2PC9BbGdvcml0aG0+CiAgICAgICAgICAgICA9PfN1Y3VzUFSZ29yaXRobT50cnV1PC9TZWN1cmVBbGdvcml0aG0+CiAgICAgICAgICAgICA9PfE5vdEFmdGVyPjIwMjItMTItMzFUMjM6MDA6MDBaPC90b3RBZnR1c4KICAgICAgICAgICA9IDwvQ3J5cHRvSW5mb3JtYXRpb24+CiAgICAgICAgICA9IDwvQXNzb2NpYXR1ZFzhBGlkYXRpb25SXBvcnREYXRhPgogICAgICA9IDwvU21nbmF0dXJ1VmFsaWRhdG1vb1N0YXR1cz4KICAgIDwvU21nbmF0dXJ1VmFsaWRhdG1vb1J1cG9ydD4KICAgIDxTaWduYXR1cmVWYxpZGF0aW9uT2JqZWN0cz4KICAgICA8VmFsaWRhdG1vbk9iamVjdCBpZD0iQy02RjM1REUzOTY1Qj1BNj1CQzM2NjFEMUEzNTVCMEFFnjA5MDdBreI3NDFDQzE5MTFFRkQwRjNCRTcyRDZBNkU5Ij4KICAgICAgICAgICA9Pf95cGU+dXJu0mV0c2k6MDE5MTAyOnZhbG1kYXRpb25PYmp1Y3Q6Y2VydG1maWNhdGU8L09iamVjdFR5cGU+CiAgICAgICAgICA9IDxWYWxpZGF0aW9uT2JqZWN0UmVwcmVzZW50YXRpb24+CiAgICAgICAgICAgICA8RGlnZXN0QWxnQW5kVmFsdWU+CiAgICAgICAgICAgICA9PG5zMjpEaWd1c3RNZXRob2QgQWxb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGVuYyNzaGEyNTYiLz4KICAgICA9ICAgICAgICA8bnMyOkRpZ2VzdFZhBHV1PmJ6WGVPV1c1cHB2RFpoMGFOVnNLNWdrSHjZEJ6QmtSNz1EenZuTFwdwWs9PC9uczI6RGlnZXN0VmFsdWU+CiAgICAgICAgICA9ICA8L0RpZ2VzdEFsZ0FuZFzhbHV1PgogICAgICA9ICA8L1ZhBGlkYXRpb25PYmp1Y3RSZXByZXN1bnRhdG1vbj4KICAgICAgICAgICA9PfRt4KICAgICAgICAgICA9IDxQT0VUaW11PjIwMjEtMDctMD1UMDg6MDI6MzdaPC9QT0VUaW11PgogICAgICA9ICAgICAgICA9Pf5cGVpZ1Byb29mPgogICAgICA9ICA8L1BPRT4KICAgICA8L1ZhBGlkYXRpb25PYmp1Y3Q+CiAgICAgICAgPfZhBGlkYXRpb25PYmp1Y3QgaWQ9IKMtRjBGRjBCNDUxNEQzMTYzMDRGmjgxN0RCQTBCRKIwNURFREI50DUyN0MwRTQ3QzczRThEOEZERkUxNkRGMjY3RSI+CiAgICAgICAgICA9IDxPYmp1Y3RUeXB1PnVybjp1dHNpOjAxOTEwMjp2YWxpZGF0aW9uT2JqZWN0UmN1cnRpZm1jYXR1PC9PYmp1Y3RUeXB1PgogICAgICAgICA8VmFsaWRhdG1vbk9iamVjdFJ1cHJ1c2VudGF0aW9uPgogICAgICAgICAgPERpZ2VzdEFsZ0FuZFzhbHV1PgogICAgICA9ICAgICAgICAgICA9IDxuczI6RGlnZXN0TW0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjaWMS8wNC94bWx1bmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICA9PG5zMjpEaWd1c3RWYwx1ZT44UDhMu1JUVEZqQ1BLQmZib0wrd1h1MjVoU2ZBNUh4ejZ0ajkvaGjSm40PTwvbnMyOkRpZ2VzdFZhBHV1PgogICAgICA9ICAgICAgPC9EaWd1c3RBbGdBbmRWYwx1ZT4KICAgICAgICA9PC9WYxpZGF0aW9uT2JqZWN0UmVwcmVzZW50YXRpb24+CiAgICAgICAgICA9IDxQT0U+CiAgICAgICAgICA9ICA8UE9FVG1tZT4yMDIxLTA3LTAVD40jAy0jM3WjwvUE9FVG1tZT4KICAgICAgICA9IDxUeXB1T2ZQcm9vZj51cm46ZXrzaTowMTkxMDI6cG91dHlwZTp2YWxpZGF0aW9uPC9UeXB1T2ZQcm9vZj4KICAgICAgICA9PC9QT0U+CiAgICAgICAgPC9WYxpZGF0aW9uT2JqZWN0PpgogICAgICA9IDxWYWxpZGF0aW9uT2JqZWN0IG1kPSJELUM10EM4MEE4MDUzMEUwRjM0OUJDMzJERUY1MDI4MEQ3NEM0QjdfQkYyNTI4MDQ0MDE4MTE2N0EYrjFBMEIzMUQipgogICAgICAgICA8T2JqZWN0VHlwZT51cm46ZXrzaTowMTkxMDI6dmFsaWRhdG1vbk9iamVjdDpzaWduZREYXRhPC9PYmp1Y3RUeXB1PgogICAgICAgICA8VmFsaWRhdG1vbk9iamVjdFJ1cHJ1c2VudGF0aW9uPgogICAgICAgICAgICA9PERpZ2VzdEFsZ0FuZFzhbHV1PgogICAgICAgICAgICA9IDxuczI6RGlnZXN0TW0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjaWMS8wNC94bWx1bmMjc2hhMjU2Ii8+CiAgICAgICAgICAgICA9PG5zMjpEaWd1c3RWYwx1ZT5rY0RIT1pqd1poVmZ1RGh1aEN1Q0VSUm1ZcFRINEpqNFJtZ1ZwaTMxUTlnPTwvbnMyOkRpZ2VzdFZhBHV1PgogICAgICAgICAgICA9PC9EaWd1c3RBbGdBbmRWYwx1ZT4KICAgICAgICA9PC9WYxpZGF0aW9uT2JqZWN0UmVwcmVzZW50YXRpb24+CiAgICAgICAgICA9IDxQT0U+CiAgICAgICAgICA9ICAgICA8UE9FVG1tZT4yMDIxLTA3LTAVD40jAy0jM3WjwvUE9FVG1tZT4KICAgICAgICAgICA9PC9QT0U+CiAgICAgICAgPC9WYxpZGF0aW9uT2JqZWN0PpgogICAgICA9PC9TaWduYXR1cmVWYxpZGF0aW9uT2JqZWN0cz4KPC9WYxpZGF0aW9uUmVwb3J0Pgo=",

```
"DiagnosticData" : {  
  "DocumentName" : "xades-detached.xml",  
  "ValidationDate" : "2021-07-09T08:02:37Z",  
  "ContainerInfo" : null,  
  "Signature" : [ {
```

```

    "Id" : "S-A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
    "DAIdentifier" : "id-afde782436468dd74eeb181f7ce110e1",
    "SignatureFilename" : "xades-detached.xml",
    "ErrorMessage" : null,
    "ClaimedSigningTime" : "2017-09-28T11:09:04Z",
    "SignatureFormat" : "XAdES-BASELINE-B",
    "StructuralValidation" : {
        "Message" : [ ],
        "valid" : true
    },
    "DigestMatcher" : [ {
        "DataFound" : true,
        "DataIntact" : true,
        "DigestMethod" : "SHA256",
        "DigestValue" : "kcDH0ZjwZhVfuDuhCeCERRmYpTH4Jj4RmfVVi31Q9g=",
        "match" : null,
        "type" : "REFERENCE",
        "name" : "r-id-1",
        "duplicated" : null
    }, {
        "DataFound" : true,
        "DataIntact" : true,
        "DigestMethod" : "SHA256",
        "DigestValue" : "DztwNTmRo0Am6/lMI8Rym5xZPzIvLYDzn/ebYYkPsr4=",
        "match" : null,
        "type" : "SIGNED_PROPERTIES",
        "name" : "#xades-id-afde782436468dd74eeb181f7ce110e1",
        "duplicated" : null
    } ],
    "BasicSignature" : {
        "EncryptionAlgoUsedToSignThisToken" : "RSA",
        "KeyLengthUsedToSignThisToken" : "2048",
        "DigestAlgoUsedToSignThisToken" : "SHA256",
        "MaskGenerationFunctionUsedToSignThisToken" : null,
        "SignatureIntact" : true,
        "SignatureValid" : true
    },
    "SigningCertificate" : {
        "PublicKey" : null,
        "Certificate" : "C-
F0FF0B4514D316304F2817DBA0BFB05DEDDB98527C0E47C73E8D8FDFE16DF267E"
    },
    "ChainItem" : [ {
        "Certificate" : "C-
F0FF0B4514D316304F2817DBA0BFB05DEDDB98527C0E47C73E8D8FDFE16DF267E"
    }, {
        "Certificate" : "C-
6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9"
    } ],
    "ContentType" : null,
    "MimeType" : "text/xml",

```

```

    "ContentIdentifier" : null,
    "ContentHints" : null,
    "SignatureProductionPlace" : null,
    "CommitmentTypeIndication" : [ ],
    "SignerRole" : [ ],
    "Policy" : null,
    "SignaturePolicyStore" : null,
    "SignerInfo" : null,
    "PDFRevision" : null,
    "SignerDocumentRepresentations" : {
        "HashOnly" : false,
        "DocHashOnly" : false
    },
    "FoundCertificates" : {
        "RelatedCertificate" : [ {
            "Origin" : [ "KEY_INFO" ],
            "CertificateRef" : [ {
                "Origin" : "SIGNING_CERTIFICATE",
                "IssuerSerial" : {
                    "value" :
                        "match" : true
                },
                "DigestAlgoAndValue" : {
                    "DigestMethod" : "SHA1",
                    "DigestValue" : "c+Vohg0jIcZ4UQSWeglCg0oGNWs=",
                    "match" : true
                },
                "SerialInfo" : null
            } ],
            "Certificate" : "C-
F0FF0B4514D316304F2817DBA0BFB05DEDB98527C0E47C73E8D8FDFE16DF267E"
        }, {
            "Origin" : [ "KEY_INFO" ],
            "CertificateRef" : [ ],
            "Certificate" : "C-
6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9"
        } ],
        "OrphanCertificate" : [ ]
    },
    "FoundRevocations" : {
        "RelatedRevocation" : [ ],
        "OrphanRevocation" : [ ]
    },
    "FoundTimestamp" : [ ],
    "SignatureScope" : [ {
        "Scope" : "FULL",
        "Name" : "sample.xml",
        "Description" : "Full document",
        "Transformation" : null,
    }]
}

```

```

    "SignerData" : "D-
C58C80A80530E0F349BC32DEF50280D74C4B7EBF25280440181167A2F1A0B31D"
    } ],
    "SignatureDigestReference" : {
        "CanonicalizationMethod" : "http://www.w3.org/2001/10/xml-exc-c14n#",
        "DigestMethod" : "SHA256",
        "DigestValue" : "SXLCmUDMsYRI6Fz6pek8zrxrZbkyyZOIFVzmJJuWPm4="
    },
    "DataToBeSignedRepresentation" : {
        "DigestMethod" : "SHA256",
        "DigestValue" : "+jLKevv08ZMgY/fRnvc7Dgy9ZJpFRKf3MYMe++kz8PM=",
        "match" : null
    },
    "SignatureValue" :
"YA7sENT3N8ufLFMnKr36r0PqzMiY3Q0s++IGTEUC0spaxUv0dHZM0d/yn3kpLJLoUkI4M3f1j5WGn83kf05Bq
M1khsX61GJzaFTPGpm7akRQKhvoH25yyqTYXES1Bcm04iziKhLMzzjUfx4/B1ZIysv5pIBgJ2r2o16jLop9ww3
ge4c4YJoaK+SXk6hyTNOcN8PjGe63WYOTNVPQFvja8Bnwg+a0bBuwD+8N6fwigCdW5a/4DJUe/J8Mb70ZI8Po0
znGDfi+TPbiIeVmCb15mUoUg2Q/xYluJfLh3uGQAXKBvF45oDIHRVefnN/D/WytAC1UVDoQSywemnjPpqF8eg=
",
        "CounterSignature" : null,
        "Parent" : null,
        "Duplicated" : null
    } ],
    "Certificate" : [ {
        "Id" : "C-6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9",
        "SubjectDistinguishedName" : [ {
            "value" : "c=lu,ou=pk1-test,o=nowina solutions,cn=good-ca",
            "Format" : "CANONICAL"
        }, {
            "value" : "C=LU,OU=PKI-TEST,O=Nowina Solutions,CN=good-ca",
            "Format" : "RFC2253"
        } ],
        "IssuerDistinguishedName" : [ {
            "value" : "c=lu,ou=pk1-test,o=nowina solutions,cn=root-ca",
            "Format" : "CANONICAL"
        }, {
            "value" : "C=LU,OU=PKI-TEST,O=Nowina Solutions,CN=root-ca",
            "Format" : "RFC2253"
        } ],
        "SerialNumber" : 4,
        "SubjectSerialNumber" : null,
        "CommonName" : "good-ca",
        "Locality" : null,
        "State" : null,
        "CountryName" : "LU",
        "OrganizationIdentifier" : null,
        "OrganizationName" : "Nowina Solutions",
        "OrganizationalUnit" : "PKI-TEST",
        "Title" : null,
        "GivenName" : null,
        "Surname" : null,
    }
]

```

```

    "Pseudonym" : null,
    "Email" : null,
    "subjectAlternativeName" : null,
    "aiaUrl" : [ "http://dss.nowina.lu/pki-factory/crt/root-ca.crt" ],
    "crlUrl" : [ "http://dss.nowina.lu/pki-factory/crl/root-ca.crl" ],
    "ocspServerUrl" : [ ],
    "Source" : [ "SIGNATURE" ],
    "NotAfter" : "2018-08-26T07:54:30Z",
    "NotBefore" : "2016-10-26T07:54:30Z",
    "PublicKeySize" : 2048,
    "PublicKeyEncryptionAlgo" : "RSA",
    "EntityKey" : "PK-
4B0F8D40371579A6397E2DE77E4F6E9F51E676B1B1C9A2C5FB043DAF56F42C50",
    "KeyUsage" : [ "digitalSignature" ],
    "extendedKeyUsages0id" : [ ],
    "IdPkixOcspNoCheck" : false,
    "BasicSignature" : {
        "EncryptionAlgoUsedToSignThisToken" : "RSA",
        "KeyLengthUsedToSignThisToken" : "?",
        "DigestAlgoUsedToSignThisToken" : "SHA256",
        "MaskGenerationFunctionUsedToSignThisToken" : null,
        "SignatureIntact" : false,
        "SignatureValid" : false
    },
    "SigningCertificate" : null,
    "ChainItem" : [ ],
    "Trusted" : false,
    "SelfSigned" : false,
    "certificatePolicy" : [ ],
    "QcStatements" : null,
    "TrustedServiceProvider" : [ ],
    "CertificateRevocation" : null,
    "Base64Encoded" : null,
    "DigestAlgoAndValue" : {
        "DigestMethod" : "SHA256",
        "DigestValue" : "bzXe0WW5ppvDZh0aNVsK5gkHrbdBzBkR79DzvnLWpuk=",
        "match" : null
    }
}, {
    "Id" : "C-F0FF0B4514D316304F2817DBA0BFB05DEDB98527C0E47C73E8D8FDFE16DF267E",
    "SubjectDistinguishedName" : [ {
        "value" : "c=lu,ou=pk-test,o=nowina solutions,cn=good-user",
        "Format" : "CANONICAL"
    }, {
        "value" : "C=LU,OU=PKI-TEST,O=Nowina Solutions,CN=good-user",
        "Format" : "RFC2253"
    } ],
    "IssuerDistinguishedName" : [ {
        "value" : "c=lu,ou=pk-test,o=nowina solutions,cn=good-ca",
        "Format" : "CANONICAL"
    }, {

```

```

    "value" : "C=LU,OU=PKI-TEST,O=Nowina Solutions,CN=good-ca",
    "Format" : "RFC2253"
  } ],
  "SerialNumber" : 10,
  "SubjectSerialNumber" : null,
  "CommonName" : "good-user",
  "Locality" : null,
  "State" : null,
  "CountryName" : "LU",
  "OrganizationIdentifier" : null,
  "OrganizationName" : "Nowina Solutions",
  "OrganizationalUnit" : "PKI-TEST",
  "Title" : null,
  "GivenName" : null,
  "Surname" : null,
  "Pseudonym" : null,
  "Email" : null,
  "subjectAlternativeName" : null,
  "aiaUrl" : [ "http://dss.nowina.lu/pki-factory/crt/good-ca.crt" ],
  "crlUrl" : [ ],
  "ocspServerUrl" : [ "http://dss.nowina.lu/pki-factory/ocsp/good-ca" ],
  "Source" : [ "SIGNATURE" ],
  "NotAfter" : "2018-08-26T07:54:31Z",
  "NotBefore" : "2016-10-26T07:54:31Z",
  "PublicKeySize" : 2048,
  "PublicKeyEncryptionAlgo" : "RSA",
  "EntityKey" : "PK-
DF0FFC7C7CFE69DA1465D07B837F6A241D76761DBE89FF53E5C8B8EF26FDB8D4",
  "KeyUsage" : [ "nonRepudiation" ],
  "extendedKeyUsages0id" : [ ],
  "IdPkixOcspNoCheck" : false,
  "BasicSignature" : {
    "EncryptionAlgoUsedToSignThisToken" : "RSA",
    "KeyLengthUsedToSignThisToken" : "2048",
    "DigestAlgoUsedToSignThisToken" : "SHA256",
    "MaskGenerationFunctionUsedToSignThisToken" : null,
    "SignatureIntact" : true,
    "SignatureValid" : true
  },
  "SigningCertificate" : {
    "PublicKey" : null,
    "Certificate" : "C-
6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9"
  },
  "ChainItem" : [ {
    "Certificate" : "C-
6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9"
  }],
  "Trusted" : false,
  "SelfSigned" : false,
  "certificatePolicy" : [ ],

```

```

    "QcStatements" : null,
    "TrustedServiceProvider" : [ ],
    "CertificateRevocation" : null,
    "Base64Encoded" : null,
    "DigestAlgoAndValue" : {
        "DigestMethod" : "SHA256",
        "DigestValue" : "8P8LRRTTFjBPKBfb0L+wXe25hSfA5Hxz6Nj9/hbfJn4=",
        "match" : null
    }
},
"Revocation" : [ ],
"Timestamp" : [ ],
"OrphanTokens" : null,
"SignerData" : [ {
    "Id" : "D-C58C80A80530E0F349BC32DEF50280D74C4B7EBF25280440181167A2F1A0B31D",
    "ReferencedName" : "sample.xml",
    "DigestAlgoAndValue" : {
        "DigestMethod" : "SHA256",
        "DigestValue" : "kcDHOZjwZhVfuDhuhCeCERRmYpTH4Jj4RmfVVi31Q9g=",
        "match" : null
    },
    "Parent" : null
} ],
"TrustedList" : [ ]
},
"SimpleReport" : {
    "ValidationPolicy" : {
        "PolicyName" : "QES AdESQC TL based",
        "PolicyDescription" : "Validate electronic signatures and indicates whether they are Advanced electronic Signatures (AdES), AdES supported by a Qualified Certificate (AdES/QC) or a\n\t\tQualified electronic Signature (QES). All certificates and their related chains supporting the signatures are validated against the EU Member State Trusted Lists (this includes\n\t\tthe signer's certificate and certificates used to validate certificate validity status services - CRLs, OCSP, and time-stamps).\n\t"
    },
    "DocumentName" : "xades-detached.xml",
    "ValidSignaturesCount" : 0,
    "SignaturesCount" : 1,
    "ContainerType" : null,
    "signatureOrTimestamp" : [ {
        "Signature" : {
            "SigningTime" : "2017-09-28T11:09:04Z",
            "BestSignatureTime" : "2021-07-09T08:02:37Z",
            "SignedBy" : "good-user",
            "SignatureLevel" : {
                "value" : "N/A",
                "description" : "Not applicable"
            },
            "SignatureScope" : [ {
                "value" : "Full document",
                "name" : "sample.xml",
            }]
        }
    }]
}

```

```

        "scope" : "FULL"
    } ],
    "Timestamps" : null,
    "Filename" : null,
    "CertificateChain" : {
        "Certificate" : [ {
            "id" : "C-
F0FF0B4514D316304F2817DBA0BFB05DEDB98527C0E47C73E8D8FDFE16DF267E",
            "qualifiedName" : "good-user"
        }, {
            "id" : "C-
6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9",
            "qualifiedName" : "good-ca"
        } ]
    },
    "Indication" : "INDETERMINATE",
    "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
    "AdESValidationDetails" : {
        "Error" : [ {
            "value" : "The certificate chain for signature is not trusted, it does not
contain a trust anchor.",
            "Key" : "BBB_XCV_CCCBB_SIG_ANS"
        } ],
        "Warning" : [ ],
        "Info" : [ ]
    },
    "QualificationDetails" : {
        "Error" : [ {
            "value" : "Unable to build a certificate chain up to a trusted list!",
            "Key" : "QUAL_CERT_TRUSTED_LIST_REACHED_ANS"
        } ],
        "Warning" : [ {
            "value" : "The signature/seal is an INDETERMINATE AdES digital
signature!",
            "Key" : "QUAL_IS_ADES_IND"
        } ],
        "Info" : [ ]
    },
    "Id" : "S-A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
    "CounterSignature" : null,
    "ParentId" : null,
    "SignatureFormat" : "XAdES-BASELINE-B",
    "ExtensionPeriodMin" : null,
    "ExtensionPeriodMax" : null
    }
},
"Semantic" : null,
"ValidationTime" : "2021-07-09T08:02:37Z"
},
"DetailedReport" : {
    "signatureOrTimestampOrCertificate" : [ {

```

```

"Signature" : {
  "ValidationProcessBasicSignature" : {
    "Constraint" : [ {
      "Name" : {
        "value" : "Is the result of the 'Format Checking' building block
conclusive?",
        "Key" : "BSV_IFCRC"
      },
      "Status" : "OK",
      "Error" : null,
      "Warning" : null,
      "Info" : null,
      "AdditionalInfo" : null,
      "Id" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A-FC",
      "BlockType" : null
    }, {
      "Name" : {
        "value" : "Is the result of the 'Identification of Signing Certificate'
building block conclusive?",
        "Key" : "BSV_IISCRC"
      },
      "Status" : "OK",
      "Error" : null,
      "Warning" : null,
      "Info" : null,
      "AdditionalInfo" : null,
      "Id" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A-ISC",
      "BlockType" : null
    }, {
      "Name" : {
        "value" : "Is the result of the 'Validation Context Initialization'
building block conclusive?",
        "Key" : "BSV_IVCIRC"
      },
      "Status" : "OK",
      "Error" : null,
      "Warning" : null,
      "Info" : null,
      "AdditionalInfo" : null,
      "Id" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A-VCI",
      "BlockType" : null
    }, {
      "Name" : {
        "value" : "Is the result of the 'X.509 Certificate Validation' building
block conclusive?",
        "Key" : "BSV_IXCVRC"
      },
      "Status" : "WARNING",
    }
  }
}

```

```

    "Error" : null,
    "Warning" : {
        "value" : "The result of the 'X.509 Certificate Validation' building
block is not conclusive!",
        "Key" : "BSV_IIXCVRANS"
    },
    "Info" : null,
    "AdditionalInfo" : null,
    "Id" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A-XCV",
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Is the signing certificate not revoked at validation time?",
        "Key" : "BSV_ISCRAVTC"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : null,
    "Id" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A-XCV",
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Is the validation time in the validity range of the signing
certificate?",
        "Key" : "BSV_IVTAVRSC"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : null,
    "Id" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A-XCV",
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Is the result of the 'Cryptographic Verification' building
block conclusive?",
        "Key" : "BSV_ICVRC"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : null,
    "Id" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A-CV",

```

```

        "BlockType" : null
    }, {
        "Name" : {
            "value" : "Is the result of the Basic Validation Process conclusive?",
            "Key" : "ADEST_ROBVPIIC"
        },
        "Status" : "NOT OK",
        "Error" : {
            "value" : "The result of the Basic validation process is not conclusive!",
            "Key" : "ADEST_ROBVPIIC_ANS"
        },
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : "Basic Signature Validation process failed with INDETERMINATE/NO_CERTIFICATE_CHAIN_FOUND indication",
        "Id" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
        "BlockType" : null
    } ],
    "Conclusion" : {
        "Indication" : "INDETERMINATE",
        "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
        "Errors" : [ {
            "value" : "The certificate chain for signature is not trusted, it does not contain a trust anchor.",
            "Key" : "BBB_XCV_CCCBB_SIG_ANS"
        } ],
        "Warnings" : [ ],
        "Infos" : [ ]
    },
    "Title" : "Validation Process for Basic Signatures",
    "ProofOfExistence" : {
        "Time" : "2021-07-09T08:02:37Z",
        "TimestampId" : null
    }
},
"Timestamp" : [ ],
"ValidationProcessLongTermData" : {
    "Constraint" : [ {
        "Name" : {
            "value" : "Is the result of the Basic Validation Process acceptable?",
            "Key" : "LTV_ABSV"
        },
        "Status" : "NOT OK",
        "Error" : {
            "value" : "The result of the Basic validation process is not acceptable to continue the process!",
            "Key" : "LTV_ABSV_ANS"
        },
        "Warning" : null,
    } ]
}

```

```

        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    } ],
    "Conclusion" : {
        "Indication" : "INDETERMINATE",
        "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
        "Errors" : [ {
            "value" : "The certificate chain for signature is not trusted, it does
not contain a trust anchor.",
            "Key" : "BBB_XCV_CCCBB_SIG_ANS"
        } ],
        "Warnings" : [ ],
        "Infos" : [ ]
    },
    "Title" : "Validation Process for Signatures with Time and Signatures with
Long-Term Validation Data",
    "ProofOfExistence" : {
        "Time" : "2021-07-09T08:02:37Z",
        "TimestampId" : null
    }
},
"ValidationProcessArchivalData" : {
    "Constraint" : [ {
        "Name" : {
            "value" : "Is the result of the LTV validation process acceptable?",
            "Key" : "ARCH_LTVV"
        },
        "Status" : "NOT OK",
        "Error" : {
            "value" : "The result of the LTV validation process is not acceptable to
continue the process!",
            "Key" : "ARCH_LTVV_ANS"
        },
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    } ],
    "Conclusion" : {
        "Indication" : "INDETERMINATE",
        "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
        "Errors" : [ {
            "value" : "The certificate chain for signature is not trusted, it does
not contain a trust anchor.",
            "Key" : "BBB_XCV_CCCBB_SIG_ANS"
        } ],
        "Warnings" : [ ],
        "Infos" : [ ]
    }
}

```

```

    },
    "Title" : "Validation Process for Signatures with Archival Data",
    "ProofOfExistence" : [
        "Time" : "2021-07-09T08:02:37Z",
        "TimestampId" : null
    ]
},
"ValidationSignatureQualification" : {
    "ValidationCertificateQualification" : [ ],
    "Constraint" : [ {
        "Name" : {
            "value" : "Is the signature/seal an acceptable AdES digital signature
(ETSI EN 319 102-1)?",
            "Key" : "QUAL_IS_ADES"
        },
        "Status" : "WARNING",
        "Error" : null,
        "Warning" : {
            "value" : "The signature/seal is an INDETERMINATE AdES digital
signature!",
            "Key" : "QUAL_IS_ADES_IND"
        },
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    }, {
        "Name" : {
            "value" : "Has a trusted list been reached for the certificate chain?",
            "Key" : "QUAL_CERT_TRUSTED_LIST_REACHED"
        },
        "Status" : "NOT OK",
        "Error" : {
            "value" : "Unable to build a certificate chain up to a trusted list!",
            "Key" : "QUAL_CERT_TRUSTED_LIST_REACHED_ANS"
        },
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    } ],
    "Conclusion" : {
        "Indication" : "FAILED",
        "SubIndication" : null,
        "Errors" : [ {
            "value" : "Unable to build a certificate chain up to a trusted list!",
            "Key" : "QUAL_CERT_TRUSTED_LIST_REACHED_ANS"
        } ],
        "Warnings" : [ {
            "value" : "The signature/seal is an INDETERMINATE AdES digital
        }
    }
}

```

```

signature!" ,
    "Key" : "QUAL_IS_ADES_IND"
} ],
"Infos" : [ ]
},
"Title" : "Signature Qualification",
"SignatureQualification" : "N/A"
},
"Conclusion" : {
    "Indication" : "INDETERMINATE",
    "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
    "Errors" : null,
    "Warnings" : null,
    "Infos" : null
},
"Id" : "S-A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
"CounterSignature" : null
}
},
"BasicBuildingBlocks" : [ {
    "FC" : {
        "Constraint" : [ {
            "Name" : {
                "value" : "Does the signature format correspond to an expected format?",
                "Key" : "BBB_FC_IEFF"
            },
            "Status" : "OK",
            "Error" : null,
            "Warning" : null,
            "Info" : null,
            "AdditionalInfo" : null,
            "Id" : null,
            "BlockType" : null
        }, {
            "Name" : {
                "value" : "Is the signature identification not ambiguous?",
                "Key" : "BBB_FC_ISD"
            },
            "Status" : "OK",
            "Error" : null,
            "Warning" : null,
            "Info" : null,
            "AdditionalInfo" : null,
            "Id" : null,
            "BlockType" : null
        }, {
            "Name" : {
                "value" : "Is the signed references identification not ambiguous?",
                "Key" : "BBB_FC_ISRIA"
            },
            "Status" : "OK",

```

```

        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    } ],
    "Conclusion" : {
        "Indication" : "PASSED",
        "SubIndication" : null,
        "Errors" : null,
        "Warnings" : [ ],
        "Infos" : [ ]
    },
    "Title" : "Format Checking"
},
"ISC" : {
    "CertificateChain" : {
        "ChainItem" : [ {
            "Source" : "SIGNATURE",
            "Id" : "C-
F0FF0B4514D316304F2817DBA0BFB05DEDDB98527C0E47C73E8D8FDFE16DF267E"
        }, {
            "Source" : "SIGNATURE",
            "Id" : "C-
6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9"
        } ]
    },
    "Constraint" : [ {
        "Name" : {
            "value" : "Is there an identified candidate for the signing certificate?",
            "Key" : "BBB_ICS_ISCI"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    }, {
        "Name" : {
            "value" : "Is the signed attribute: 'cert-digest' of the certificate
present?",
            "Key" : "BBB_ICS_ISACDP"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
    }
]
}

```

```

        "Id" : null,
        "BlockType" : null
    }, {
        "Name" : {
            "value" : "Does the certificate digest value match a digest value found in
the certificate reference(s)?",
            "Key" : "BBB_ICS_ICDVV"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    }, {
        "Name" : {
            "value" : "Are the issuer distinguished name and the serial number
equal?",
            "Key" : "BBB_ICS_AIDNASNE"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    } ],
    "Conclusion" : {
        "Indication" : "PASSED",
        "SubIndication" : null,
        "Errors" : null,
        "Warnings" : [ ],
        "Infos" : [ ]
    },
    "Title" : "Identification of the Signing Certificate"
},
"VCI" : {
    "Constraint" : [ {
        "Name" : {
            "value" : "Is the signature policy known?",
            "Key" : "BBB_VCI_ISPK"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    } ]
}

```

```

    },
    "Conclusion" : {
        "Indication" : "PASSED",
        "SubIndication" : null,
        "Errors" : null,
        "Warnings" : [ ],
        "Infos" : [ ]
    },
    "Title" : "Validation Context Initialization"
},
"XCV" : {
    "SubXCV" : [ ],
    "Constraint" : [ {
        "Name" : {
            "value" : "Can the certificate chain be built till a trust anchor?",
            "Key" : "BBB_XCV_CCCBB"
        },
        "Status" : "NOT OK",
        "Error" : {
            "value" : "The certificate chain for signature is not trusted, it does not
contain a trust anchor.",
            "Key" : "BBB_XCV_CCCBB_SIG_ANS"
        },
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    } ],
    "Conclusion" : {
        "Indication" : "INDETERMINATE",
        "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
        "Errors" : [ {
            "value" : "The certificate chain for signature is not trusted, it does not
contain a trust anchor.",
            "Key" : "BBB_XCV_CCCBB_SIG_ANS"
        } ],
        "Warnings" : [ ],
        "Infos" : [ ]
    },
    "Title" : "X509 Certificate Validation"
},
"CV" : {
    "Constraint" : [ {
        "Name" : {
            "value" : "Has the reference data object been found?",
            "Key" : "BBB_CV_IRDOF"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
    } ]
}

```

```

    "Info" : null,
    "AdditionalInfo" : "Reference : r-id-1",
    "Id" : null,
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Is the reference data object intact?",
        "Key" : "BBB_CV_IRDOI"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : "Reference : r-id-1",
    "Id" : null,
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Has the reference data object been found?",
        "Key" : "BBB_CV_IRDOF"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : "Reference : #xades-id-afde782436468dd74eeb181f7ce110e1",
    "Id" : null,
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Is the reference data object intact?",
        "Key" : "BBB_CV_IRDOI"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : "Reference : #xades-id-afde782436468dd74eeb181f7ce110e1",
    "Id" : null,
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Is the signature intact?",
        "Key" : "BBB_CV_ISI"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : null,
    "Id" : null,

```

```

        "BlockType" : null
    } ],
    "Conclusion" : {
        "Indication" : "PASSED",
        "SubIndication" : null,
        "Errors" : null,
        "Warnings" : [ ],
        "Infos" : [ ]
    },
    "Title" : "Cryptographic Verification"
},
"SAV" : {
    "CryptographicValidation" : {
        "Algorithm" : {
            "Name" : "RSA with SHA256",
            "Uri" : "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256",
            "KeyLength" : "2048"
        },
        "NotAfter" : "2022-12-31T23:00:00Z",
        "ValidationTime" : "2021-07-09T08:02:37Z",
        "ConcernedMaterial" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
        "secure" : true
    },
    "Constraint" : [ {
        "Name" : {
            "value" : "Is the structure of the signature valid?",
            "Key" : "BBB_SAV_ISSV"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    }, {
        "Name" : {
            "value" : "Is the signed attribute: 'signing-certificate' present?",
            "Key" : "BBB_ICS_ISASCP"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
    }, {
        "Name" : {
            "value" : "Is the signed attribute: 'signing-certificate' present only
        }
    }
]
}

```

```

once?", {
    "Key" : "BBB_ICS_ISASCPU"
},
"Status" : "OK",
"Error" : null,
"Warning" : null,
"Info" : null,
"AdditionalInfo" : null,
"Id" : null,
"BlockType" : null
}, {
    "Name" : {
        "value" : "Does the 'Signing Certificate' attribute contain references
only to the certificate chain?",
        "Key" : "BBB_SAV_DSCACRCC"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : null,
    "Id" : null,
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Is the signed qualifying property: 'signing-time' present?",
        "Key" : "BBB_SAV_ISQPSTP"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : null,
    "Id" : null,
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Is the signed qualifying property: 'message-digest' or
'SignedProperties' present?",
        "Key" : "BBB_SAV_ISQPM DOSPP"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : null,
    "Id" : null,
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Are cryptographic constraints met for the signature creation?",
```

```

        "Key" : "ACCM"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : "Signature algorithm RSA with SHA256 with key size 2048
at validation time : 2021-07-09 08:02",
    "Id" : null,
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Are cryptographic constraints met for the object reference?",
        "Key" : "ACCM"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : "Digest algorithm SHA256 at validation time : 2021-07-09
08:02 for object reference with name : r-id-1",
    "Id" : null,
    "BlockType" : null
}, {
    "Name" : {
        "value" : "Are cryptographic constraints met for the signed properties?",
        "Key" : "ACCM"
    },
    "Status" : "OK",
    "Error" : null,
    "Warning" : null,
    "Info" : null,
    "AdditionalInfo" : "Digest algorithm SHA256 at validation time : 2021-07-09
08:02 for signed properties with name : #xades-id-afde782436468dd74eeb181f7ce110e1",
    "Id" : null,
    "BlockType" : null
} ],
"Conclusion" : {
    "Indication" : "PASSED",
    "SubIndication" : null,
    "Errors" : null,
    "Warnings" : [ ],
    "Infos" : [ ]
},
"Title" : "Signature Acceptance Validation"
},
"PSV" : null,
"PCV" : null,
"VTS" : null,
"CertificateChain" : {
    "ChainItem" : [ {

```

```

    "Source" : "SIGNATURE",
    "Id" : "C-F0FF0B4514D316304F2817DBA0BFB05DEDB98527C0E47C73E8D8FDFE16DF267E"
  },
  {
    "Source" : "SIGNATURE",
    "Id" : "C-6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9"
  }
},
"Conclusion" : {
  "Indication" : "INDETERMINATE",
  "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
  "Errors" : [ {
    "value" : "The certificate chain for signature is not trusted, it does not
contain a trust anchor.",
    "Key" : "BBB_XCV_CCCBB_SIG_ANS"
  }],
  "Warnings" : [ ],
  "Infos" : [ ]
},
"Id" : "S-A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
"Type" : "SIGNATURE"
},
"TLAnalysis" : [ ],
"Semantic" : null,
"ValidationTime" : "2021-07-09T08:02:37Z"
}
}

```

Retrieve original document(s)

This service returns the signed data for a given signature.

Request

```

POST /services/rest/validation/getOriginalDocuments HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 8953

{
  "signedDocument" : {
    "bytes" :
"PD94bWwgdmVyc2lvbj0iMS4wIiB1bmNvZGlubz0iVVRGLTgiPz48ZHM6U21nbmF0dXJlIHhtbG5z0mRzPSJod
HRw0i8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIiBJZD0iaWQtZWEwNTE3Y2jN2Y1ND11YTN1njg
10DY3YWM5NWUiPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpeM0aW9uTWV0aG9kIEFsZ29yaXRobT0ia
HR0cDovL3d3dy53My5vcmcvVFIvMjAwMS9SRUMteG1sLWMxNG4tMjAwMTAzMTUiLz48ZHM6U21nbmF0dXJlTWV
0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxkc2lnLW1vcmUjcnNhLXNoYTI1N
iIvPjxkczpSWZlcmVuY2UgSWQ9InItaWQtMSIgVHlwZT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWx
kc2lnI09iamVjdCIgVVJJPSIjby1pZC0xIj48ZHM6VHJhbnNmb3Jtcz48ZHM6VHJhbnNmb3JtIEFsZ29yaXRob
T0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI2Jhc2U2NCIvPjwvZHM6VHJhbnNmb3Jtcz48ZHM

```

6RG1nZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hhMjU2Ii8+PGRz0kRpZ2VzdFZhbHV1PkxQSk51bCt3b3c0bTZEc3F4Ym5pbmhzV0hsd2ZwMEp1Y3dRe1lwT0xtQ1E9PC9kczpEaWd1c3RWYWx1ZT48L2Rz01J1ZmVyZw5jZT48ZHM6UmVmZxJ1bmN1IFR5cGU9Imh0dHA6Ly91cmkuZXrzaS5vcmcvMDE5MDMjU2lnbmVkuUHJvcGVydG11cyIgVVJJPSIjeGFkZXMaWQtZWEwMGewNTE3Y2jN2Y1ND11YTN1Njg10DY3YWM5NWUiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRw0i8vd3dLnczLm9yZy9UUi8yMDAxL1JFQy14bWwtYzE0b10yMDAxMDMxNSIvPjwvZHM6VHJhbnNmb3Jtcz48ZHM6Rg1nZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWx1bmMjc2hhMjU2Ii8+PGRz0kRpZ2VzdFZhbHV1PnpUZ1c5bjFUOWwwThg2T1hGNUFUM1Btb3FLOWFhbUpIZ1BdaX10Z1JNeVU9PC9kczpEaWd1c3RWYWx1ZT48L2Rz01J1ZmVyZw5jZT48L2Rz01NpZ251ZE1uZm8+PGRz01NpZ25hdHVyZVZhbHV1IE1kPSJ2Yw1ZS1pZC11YTeWYTA1MTdjYmM3jU00WvhM2U20DU4NjdhYzk1ZSI+Qy9FZnJvWmdGTkJvak40ZnpJd2UzVTR1bDQ5S2xBbmhKSmI3e1c0T2MxNWsweWpPSkZDcm9jY2JGV0U1eU52R3cyVHpxYVo0SVFYRjBKR11GM2I1rNW5saG1EcTJAcHBJYnN10WY5M005cGU4cTVHaFBjWkRDV1FmNnp2TnNvUHRPyktzL04vWj1zODVVcmY3UGd1SWtVZ1I3eUJuaW5waGhJNVpuRHZuSnZsQ1RNRE5tYm4yM1BYS1Yyc1IxWFpsa1BLWUsVGHyOFdnS1crUU9VREdTeGVSTmgyUWJPT11hR1FsUVJ6Vkhqb0c3emppVHM3UkdrR1ZVNGh3Q0pieW9TZThkd20zbkUxenVuQmp2TkRWenVqZVFyZkhTsJNrQUsxbS9odkt4Tnh1NRX4bkNmd0ZkN1dDanpDWXBaE9sSmI3MDJvV3RYam56azBHbkFZQndnPT08L2Rz01NpZ25hdHVyZVZhbHV1PjxkczpLZx1JbmZvPjxkczpYNTA5RGF0YT48ZHM6WDUwOUN1cnRpZm1jYXR1P1JsuQxRENDQXJ5Z0F3SUJBZ01CQ2pBTkJna3Foa21HOXcwQkFRc0ZBREJ0TVJBD0RnWURWUVFEREFkbmIyOWtMV05oTVJrd0Z3WURWUVFLREJCT2IzZHBibUVnVTI5c2RYUnBiMjV6TVJFd0R3WURWUVFMREFoUVMwa3RWRVZUVkRFT1Ba0dBMVVFQmhNQ1RGVXdIaGNOTVRjeE1ERTVNREG5TXpFd1doY05NVgt3T0RFNU1EY31NeKv3V2pCUE1SSxdFQV1EV1FRRERBbG5iMj1rTFhWe1pYSxhHEFYQmd0VkJbb01FRTV2ZDjsdV1TQ1rMngxZEdsdmJuTxhFVEFQQmd0VkJBc01DRkJMU1MxVVJWT1VNUXN3Q1FZRFZRUdFd0pNV1RDQ0FTSxdEUV1KS29aSWh2Y05UUVCQ1FBRGdnRVBBRENDQVFvQ2dnRUJBSi9SVHV5WVRvR0RpbUZGR21STDhsN31yRDZ1WDh1bkYzZkFmMGVTc1pSRGZpS1R3Rj1PT3RVclc10U5EmNyQyt5aXBWS11HR1NbCTjxS0NuR1BMcXZ0V0ZTbCtnZnJtWnNuN2tjUUJWd1FqQj1JS1pHMmc0VW55VUxCa3JQMF1vcW1pTmRuM0kzNHE2a01BV3hXUnprUC9CaFAXdWVvnjBnUnh1V01HUGEyeVZabnhKbDFUOEplSGkvSmpoN2tQSTgwR3V4UXJDQkg0eGJRWngvU1FpV2pJdDdwcm1WZ0crR0hoNHFpb0JGWWp20GdQM2ZMdExrM3ZGRHptRWFPeHRMMWRHQxdwcG83R0x1VXRvnkp4QXYwZm52eTZVT055QW4rK3V00Hfr0FU4W1M0Sk9nRnpKd1ZLT2N0UXBUVGptbitMbG9uMW1ncmtiSEptN21xYwJrQ0F3RUFByU9CdkRDQnVUQU9CZ05WSFE4QkFm0EVCQU1DQmtBd2dZy0dDQ3NHQFVFRkJ3RUJCSHN3ZVRBNUJnZ3JCZ0VGQ1Fj0FZWXRhSF1wY0RvdwyUnpjeTV1YjNkcGJtRXViSFV2Y0d0cExXWmhZM1J2Y25rdmIyTnpjQzluYjI5a0xXTmhNRhdHQ0NzR0FRVUZCekFDaGpCb2RIUndPaTh2Wkh0ekxtNXZkMmx1WVM1c2RT0XdhMmt0Wm1GamRHOX11Uz1qY25Rd1oy0XZaQzFqWVm1amNuUXdIUV1EV1IwT0JcwUVGtY9QL21xRW92UW4zYXByN1VTT0tTUzdSTU5TUEwR0NTcUdTSWIzRFFFQkN3VUFBNE1CQVFBR1JMSjdKa1R1Q21RT1p1STBTalcySnAwTFA2S11pNWNOYzR6SW4wT1kwQ05UukZzejdrR1c2QW9FVGhoSjkzMmd5NmxSK0ZMR3BwS1NRNvZtUDBLd2JZV3g2MGFUSFJTbmttramRvZn1rYStoNitSbk1mRn13NG9pZGvxdTBFWHBMFhtVFNQN3hPNi9PN2Ezzk9kM01Duy9Udm4wQ11mNV1TOFJuZxd4MFBZk5hb3czYuhDMTEyQkFTMUFNZVVtc2x5QVBYMUNGZ2dtK25aUEgwcnuVUL0NQVko5W1R6VFcyM1hLa11iaytHTVFE0GxRR1RwYTbzVnU1K2Z3Zm1JZ28xZ1NqY20raXhKN04raDvtVXFZcE1Ydkp1TnJLUWwvSjA3RURWWmlrRWvnL2NQTK2TE1XOXU5ckxqdU1rZwpQytETFUxRkpCZEpd3FJS2NBakE8L2Rz01g1MD1DZxJ0aWzP2F0ZT48ZHM6WDUwOUN1cnRpZm1jYXR1P1JSUQ2akNDQXRLZ0F3SUJBZ01CQkRBTkJna3Foa21HOXcwQkFRc0ZBREJ0TVJBD0RnWURWUVFEREFkeWIyOTBMV05oTVJrd0Z3WURWUVFLREJCT2IzZHBibUVnVTI5c2RYUnBiMjV6TVJFd0R3WURWUVFMREFoUVMwa3RWRVZUVkRFT1Ba0dBMVVFQmhNQ1RGVXdIaGNOTVRjeE1ERTVNREG5TWpVeFdoY05NVgt3T0RFNU1EY31Na1V4V2pCtk1SQxdEZ11EV1FRRERBZG5iMj1rTFd0aE1Sa3dGd11EV1FRS0RCQk9iM2RwYm1Fz1Uy0XNkWFJwYjI1ek1SRxdEd11EV1FRTERBaFFTNGt0VkvWVFZERuXNQwthQTFVRUJ0tUNUR1V3Z2dFaU1BMEddu3Fhu01iM0RRRUJBUVVBQTRJQkR3QXdnZ0VLQW9JQkFRQ2U4bjJoTDJrKzRck1XUDJ6UmMqkhBK1RGRDVTz1FrNwlna3p1RzI4UDZSSXAxZ1QwMhdDQzK3MVRndkt1Z0xyTmx5REducEFzQ2k1UDZndXd3dDk3NFhKSGJotitZc0xJa2g3djRYbVVQSFpDcEpLS1hScCs1bThpS002cGJGSS8r0E9KQ0jYADmxY3pHTFlnRUFnQ0ZkVt5WXY5Yt1Z2Z1FJVKQ3bko3aUFRV0xoSHJ6S1lwSkQ00Et2Wk1HMVJDNDhZNjhtNjFDZEdzenRVTHVHV1I10Go5Zm5qanVRSTRITWVmY11jK1pWRWR1dUp0bWp1M3h4UkE1aGhIYkczahN1NhpjSVJLd1pBT0hGcGJNVnZWDVSZk9GTE9rNkt6W1R0NzFUSzVmBk5Wn1lVShc30XJXU29yRkxrRzRMVUxTR2d5bH11TVVUdHd5R25GeVpuQWdNQkFBR2pznZFF3Z2RFd0RnWURWUjBQQVFI0JBUURBZ2VBTUVFR0ExVWRId1E2TURnd05xQTbVREtHTUdoMGR1QTZMe1rYzNNdWJt0TNhVzVoTG14MUwzQnJhUzFtWVdOMG1zSjVMMk55YkM5eWIy0TBMV05oTG10eWJEqk1CZ2dyQmdFrkJRY0JBUVJBTUQ0d1BBWULLd1LCQ1FVSE1BS0dNR2gwZEhBNkx50WtjM011Ym05M2FXNWhMbXgxTDNCcmFTMW1ZV04wYjNKNUwyTn1kQz15YjI5MExXTmhMb

```

U55ZERBZEJnT1ZIUTRFmdRVUhGUXMweWRjUDFSUH1vWXJ2bExHUjFaYksxZ3dEd11EV1IwVEFRSC9CQVV3QXd
FQi96QU5CZ2txaGtpRz13MEJBUXNGQUFPQ0FRRUFIM0hkZkpQYkhPQ3BjRXBteHzaRi9VMjcreTB3VFd6aUo0a
3Z1Rnp5YmNMcjJyRwt3UkpldDBPaEZBM1BTSXFFZx5S1lpb3BEd0V80GQxSXA4L3k5Tk1kYU9VWUvP2KRTZzk
wMWNNnVnhxR1FFRHJadUpWdEljQnh3MzBiNWFPMUE1V0FRRzhCMVhaNjI1K0Ni1RNQ110K0xoRHFZRWJhK1FXW
mdBR3BzWDFOS281Tmx1K0wySm1Vdng5Qj1xu95YkxZSWxWbmxuSGk3bFRJNDBjMjNTM2hTYVp6Z31BdUFWR2N
TKzZFS1dSc0dYNXJtaUE1MUN1TUhoMEtCdXR1L0FkczV0b0RteW93bH1hYU5vZHBtc2NivWxIK0hneG1MVWRYN
0tJND1abWRGSWtzUDB2Q1VvWF1iWF1TekdmYmt2VGZ5SjQ5NXJzcktaTcreWg0N1E9PTwvZHM6WDUwOUN1cnR
pZm1jYXR1PjxkczpYNTA5Q2VydG1maWnhdGU+TULJRFZ6Q0NBaitnQXdJQkFnSUJBVEFOQmdrcWhraUc5dzBCQ
VEwRkFEQk5NUkF3RGdZRFZRUUREQWR5YjI5MExXTmhNUmt3RndZRFZRUUteQkJPYjNkcGJtRWdVMj1zZfhsGI
yNXpNUkV3RhdZRFZRUUxEQWhRUzBrdFZV1LWREVMUfR0ExVUVCaE1DVEZVd0hoY05NVGN3T1RFNU1EY31Na
1F4V2hjTk1Ua3dPVEU1TURjeU1qUXhXakJOTVJBd0RnWURWUVFEREfkeWiY0TBMV05oTVJrd0Z3WURWUVFLREJ
CT2IzZHBibUVnVT15c2RYUnBiMjV6TVJFd0R3WURWUVFMRFoUVMwa3RWRVZUVkRFT1Ba0dBMVVFQmhNQ1RGV
XdnZ0VpTUEwR0NTcUdTSW1zRFFFQkFRVUFBN1CRhdBd2dnRutBb01CQVFEUmc0SDRvbEhveDF1Nz1jVUpRSTV
1S1tETGtYVn12ZkExNW1WZGVJY3ZhS0xrUmdoYWLheTRsbmRJWTVGRjR0TVkwRWI2aW45Z1B2Vz1nZytPMy9BM
HFUchc00XA5Z0FSdXE0SjzjNGFUZC8zUmdVem8wNHRXb1JkbUg3Tm5Nc3ZKcmhHcGRvc1pnejd5Sm1HUVVWRjQ
4bfkzT0Vld3dCWUQz0GJER01UZG9jdGdrY2F6bThFVGf6M0hwQm9yRi9GM09nZ3JPNu0c0S1dtNGFuT1BvYUdZM
WZaR3ZJQ0RTNCt1e1kNytoB80Sjky2hwUkRuMj14djdra3g5VVCQWVSYjZ3YzVhTkxmdGx2aEF4S0U
2bk5Dbk0wYXbVQmRCRGVuY3IzZk9SWL0cmxxd1NsNkg2T3pseHFddW9Qk5P2tra3hvRzRabHVmWGHQV2JBZ
01CQUFH1fQkFNQTRHQTfVZER3RUIvd1FFQXdJQkJqWRCZ05WSFE0RUZnUVVZVB6YmV6NXNiY0pIcys50E1
HU2haaEw3UEF3RhdZRFZSMFRBUUgvQkFVd0F3RUIvekF0QmdrcWhraUc5dzBCQVEwRkFBT0NBUVBVG1QYVpTb
0dMNFg5UTfM0Xh0a0NCYjZUQj1TUeWzVVCky9wUUVReXR5Rys5c1FRkY4aGvMvJb6bGdGOUzqM1VwbWwyM0h
1dnZRQXk1YmE4dGxxWStMdE52THBRb1pHcXZEUDN0nkF1RDNONTQwNFNzd2FpT2tPL1gySmVZZz13RDN4RU9na
kNSTVdyTU1FSWhxb1pOZXFN1dLS1pKL3RHT01vSEjxSHFXYVZmbGpqVmNUNnA0enI4bzB0MX15T3AzN1NqVS9
LOHBNdFg0YU1PU05uUlpTdn12a3F5Ly9pNH1FbmFRNnMvVks1eUgzyStXcENiTnpLQ0xmbTEzMS8rVUdZV1FOV
G1zWURYUUtGwmkwcnzo0Gt0dFFDeVEzYXVzQUZMdWsy0FmSUszVgtIZm11ZnFZSXZsBEMzc1ZZQuo2TU91Ww1
XWGPtd1RrK25pWFBPT08L2Rz01g1MD1DZXJ0aWZpY2F0ZT48L2Rz01g1MD1EYXRhPjwvZHM6S2V5SW5mbz48Z
HM6T2JqZWN0Pjx4YWR1czpRdWFsaWZ5aW5nUHJvcGVydG11cyB4bWxuczp4YWR1cz0iaHR0cDovL3VyaS51dHN
pLm9yYz8wMTkwMy92MS4zLj1j1iBUYXJnZXQ9IiNpZC11YTeWYTA1MTdjYmM3zjU00WVhM2U20DU4NjdhYzk1Z
SI+PHhhZGVz01NpZ251ZFByb3B1cnRpZXmgSWQ9InhhZGVzLW1kLWVhMTBhMDUxN2NiYzdmNTQ5ZWEzZTY4NTg
2N2Fj0TV1Ij48eGfkZXM6U21nbmVku21nbmF0dXJ1UHJvcGVydG11cz48eGfkZXM6U21nbmluZ1RpbWU+MjAx0
C0wOS0yN1QxMTo10Do0M1o8L3hhZGVz01NpZ25pbmdUaW1Pjx4YWR1czpTaWduaW5nQ2VydG1maWnhdGVWMj4
8eGfkZXM6Q2VydD48eGfkZXM6Q2VydERpZ2VzdD48ZHM6RGlzXN0TWW0aG9kIEFsZ29yaXRobT0iaHR0cDovL
3d3dy53My5vcmcvMjAwMC8wOS94bWxkc21nI3NoYTEiLz48ZHM6RGlzXN0VmFsdWU+aE5yb1k4cjFDQjU5Nmp
HQ1BnUmdaRnZSRGJjPTwvZHM6RGlzXN0VmFsdWU+PC94YWR1czpDZXJ0RG1zXN0Pjx4YWR1czpJc3N1ZJXTZ
XJpYWhWmj5NR113VWFsue1FMHhFREFPQmd0VkJBTU1CMmR2YjJRdFkyRXhHVEFYQmd0VkJBb01FRTV2ZDjsdV1
TQ1RiMngxZEdsdmJuTXhFVEFQmd0VkJBc01DRkJMU1MxVVJWt1VNUXN3Q1FZRFZRUUdF0pNv1FJQkNnPT08L
3hhZGVz0klzc3Vlc1Nlcm1hbFYyPjwveGfkZXM6Q2VydD48L3hhZGVz01NpZ25pbmdDZXJ0aWZpY2F0ZVYyPjw
veGfkZXM6U21nbmVku21nbmF0dXJ1UHJvcGVydG11cz48eGfkZXM6U21nbmVkRGF0YU9iamVjdFBByb3B1cnRpZ
XM+PHhhZGVz0kRhdGFPYmp1Y3Rg3JtYXQgT2JqZWN0UmVmZJ1bmN1PSIjci1pZC0xIj48eGfkZXM6TW1tZVR
5c6U+dGV4dC9wbGFBjwveGfkZXM6T1tZVR5cGU+PC94YWR1czpEYXRhT2JqZWN0Rm9ybWF0PjwveGfkZXM6U
21nbmVkRGF0YU9iamVjdFBByb3B1cnRpZXM+PC94YWR1czpTaWduZWRQcm9wZJ0aWVzPjwveGfkZXM6UxVhbg1
meWluZ1Byb3B1cnRpZXM+PC9kczpPYmp1Y3Q+PGRz0k9iamVjdCBjZD0iby1pZC0xIj5hR1ZzYkc4PTwvZHM6T
2JqZWN0PjwvZHM6U21nbmF0dXJ1Pg==",
    "digestAlgorithm" : null,
    "name" : "hello-signed-xades.xml"
},
"originalDocuments" : null,
"policy" : null,
"tokenExtractionStrategy" : "NONE",
"signatureId" : null
}

```

Response

```
HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:38 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 83
```

```
[ {
  "bytes" : "aGVsbG8=",
  "digestAlgorithm" : null,
  "name" : "o-id-1"
} ]
```

REST certificate validation service

Validate a certificate

This service allows a certificate validation (provided in a binary format).

Request

```
POST /services/rest/certificate-validation/validateCertificate HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 2195

{
  "certificate" : {
    "encodedCertificate" :
      "MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIB3DQEBCwUAMDAxGzAZBgNVBAMME1Jvb3RTZWxmU2lnbmVkr
      mFrZTERMA8GA1UECgwIRFNTLXR1c3QwHhcNMTCwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
      DDApTaWduZXJGYWt1MREwDwYDVQQKDAhEU1MtdGVzdDCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
      MI3kZhtnipn+iiZHZ9ax8F1fE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
      pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
      sStkYXzdULqpwz4JEXW9vz64eTbde4vQJ6pjHgarJf1gQNEc2XzhmI/prXLysWNqC71Zg7PUZUTrdegABTuzYCR
      J1kWBRPm4qo0LN405c94QD45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
      wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIB3DQEBCwUAA4IBAQCK6LGA01TR+rM8p6yhAi40kDN2b1
      dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUi x00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
      9d1LOHe3WKBNB6GZALT1ewjh7hSbWjft1mcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
      D7UT93Nuw3xcV8ViftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyO10Bvw14PWtsWG10QdOSRvIBBrP4adCnGT
      gjgjk9LTc08B8FKrr+81H6uc0bp4lIUToiUkGILXsiEeEg9WAqm+Xq0"
  },
  "certificateChain" : [ {
    "encodedCertificate" :
      "MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIB3DQEBCwUAMDAxGzAZBgNVBAMME1Jvb3RTZWxmU2lnbmVkr
      mFrZTERMA8GA1UECgwIRFNTLXR1c3QwHhcNMTCwNjA4MTEyNjAxWhcNNDCwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
      DDApTaWduZXJGYWt1MREwDwYDVQQKDAhEU1MtdGVzdDCCASiWdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
      MI3kZhtnipn+iiZHZ9ax8F1fE50w/cFwBTfAEb3R1ZQUp6/BQnBt70o0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
      pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrt01Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
      sStkYXzdULqpwz4JEXW9vz64eTbde4vQJ6pjHgarJf1gQNEc2XzhmI/prXLysWNqC71Zg7PUZUTrdegABTuzYCR
      J1kWBRPm4qo0LN405c94QD45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
      wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIB3DQEBCwUAA4IBAQCK6LGA01TR+rM8p6yhAi40kDN2b1
      dbIL818iCMYopLCxx8xqq3ubZC0xqh1X2j6pgWzarb0b/MUi x00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
      9d1LOHe3WKBNB6GZALT1ewjh7hSbWjft1mcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
      D7UT93Nuw3xcV8ViftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyO10Bvw14PWtsWG10QdOSRvIBBrP4adCnGT
      gjgjk9LTc08B8FKrr+81H6uc0bp4lIUToiUkGILXsiEeEg9WAqm+Xq0"
  } ],
  "validationTime" : null,
  "tokenExtractionStrategy" : "NONE"
}
```

Response

```
HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:39 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
```

Pragma: no-cache
Expires: 0
X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 6803

```
{  
  "diagnosticData" : {  
    "DocumentName" : null,  
    "ValidationDate" : "2021-07-09T08:02:39Z",  
    "ContainerInfo" : null,  
    "Signature" : null,  
    "Certificate" : [ {  
      "Id" : "C-02F3EBCA0163274253BC809D27498DD41BB0316D7E6B066960115DE155589D9C",  
      "SubjectDistinguishedName" : [ {  
        "value" : "o=dss-test,cn=signerfake",  
        "Format" : "CANONICAL"  
      }, {  
        "value" : "O=DSS-test,CN=SignerFake",  
        "Format" : "RFC2253"  
      } ],  
      "IssuerDistinguishedName" : [ {  
        "value" : "o=dss-test,cn=rootselfsignedfake",  
        "Format" : "CANONICAL"  
      }, {  
        "value" : "O=DSS-test,CN=RootSelfSignedFake",  
        "Format" : "RFC2253"  
      } ],  
      "SerialNumber" : 51497007561559,  
      "SubjectSerialNumber" : null,  
      "CommonName" : "SignerFake",  
      "Locality" : null,  
      "State" : null,  
      "CountryName" : null,  
      "OrganizationIdentifier" : null,  
      "OrganizationName" : "DSS-test",  
      "OrganizationalUnit" : null,  
      "Title" : null,  
      "GivenName" : null,  
      "Surname" : null,  
      "Pseudonym" : null,  
      "Email" : null,  
      "subjectAlternativeName" : null,  
      "aiaUrl" : [ ],  
      "crlUrl" : [ ],  
      "ocspServerUrl" : [ ],  
      "Source" : [ "OTHER" ],  
      "SignatureAlgorithm" : "SHA256WithRSA",  
      "Signature" : "MIIBIjANBgkqhkiG9w0BAQEFAASCAQD...  
    } ]  
  } }  
}
```

```

    "NotAfter" : "2047-07-04T07:57:24Z",
    "NotBefore" : "2017-06-08T11:26:01Z",
    "PublicKeySize" : 2048,
    "PublicKeyEncryptionAlgo" : "RSA",
    "EntityKey" : "PK-
3CFCA257859E202BCC83864D02B267B08A997C357AB98D923BBC63F00607C7B6",
    "KeyUsage" : [ "keyCertSign", "crlSign" ],
    "extendedKeyUsagesOid" : [ ],
    "IdPkixOcspNoCheck" : false,
    "BasicSignature" : {
        "EncryptionAlgoUsedToSignThisToken" : "RSA",
        "KeyLengthUsedToSignThisToken" : "?",
        "DigestAlgoUsedToSignThisToken" : "SHA256",
        "MaskGenerationFunctionUsedToSignThisToken" : null,
        "SignatureIntact" : null,
        "SignatureValid" : null
    },
    "SigningCertificate" : null,
    "ChainItem" : [ ],
    "Trusted" : false,
    "SelfSigned" : false,
    "certificatePolicy" : [ ],
    "QcStatements" : null,
    "TrustedServiceProvider" : [ ],
    "CertificateRevocation" : [ ],
    "Base64Encoded" : null,
    "DigestAlgoAndValue" : {
        "DigestMethod" : "SHA256",
        "DigestValue" : "AvPrygFjJ0JTvICdJ0mN1BuwMW1+awZpYBFd4VVYnZw=",
        "match" : null
    }
} ],
"Revocation" : [ ],
"Timestamp" : null,
"OrphanTokens" : null,
"SignerData" : null,
"TrustedList" : [ ]
},
"simpleCertificateReport" : {
    "ChainItem" : [ {
        "id" : "C-02F3EBCA0163274253BC809D27498DD41BB0316D7E6B066960115DE155589D9C",
        "subject" : {
            "commonName" : "SignerFake",
            "surname" : null,
            "givenName" : null,
            "pseudonym" : null,
            "organizationName" : "DSS-test",
            "organizationUnit" : null,
            "email" : null,
            "locality" : null,
            "state" : null,
        }
    } ]
}

```

```

    "country" : null
  },
  "issuerId" : null,
  "notBefore" : "2017-06-08T11:26:01Z",
  "notAfter" : "2047-07-04T07:57:24Z",
  "keyUsage" : [ "keyCertSign", "crlSign" ],
  "extendedKeyUsage" : null,
  "ocspUrl" : null,
  "crlUrl" : null,
  "aiaUrl" : null,
  "cpsUrl" : null,
  "pdsUrl" : null,
  "qualificationAtIssuance" : "N/A",
  "qualificationAtValidation" : "N/A",
  "revocation" : {
    "productionDate" : null,
    "revocationDate" : null,
    "revocationReason" : null
  },
  "trustAnchor" : null,
  "Indication" : "INDETERMINATE",
  "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND"
} ],
"ValidationTime" : "2021-07-09T08:02:39Z"
},
"detailedReport" : {
  "signatureOrTimestampOrCertificate" : [ {
    "Certificate" : {
      "ValidationCertificateQualification" : [ ],
      "Constraint" : [ {
        "Name" : {
          "value" : "Is the result of the Basic Building Block conclusive?",
          "Key" : "BBB_ACCEPT"
        },
        "Status" : "WARNING",
        "Error" : null,
        "Warning" : {
          "value" : "The result of the Basic Building Block is not conclusive!",
          "Key" : "BBB_ACCEPT_ANS"
        },
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null,
        "BlockType" : null
      } ],
      "Conclusion" : {
        "Indication" : "INDETERMINATE",
        "SubIndication" : null,
        "Errors" : [ ],
        "Warnings" : [ {
          "value" : "The result of the Basic Building Block is not conclusive!",
          "Key" : "BBB_ACCEPT_W"
        } ]
      }
    }
  }
}

```

```

        "Key" : "BBB_ACCEPT_ANS"
    } ],
    "Infos" : null
},
"Title" : "Certificate Qualification",
"Id" : "C-02F3EBCA0163274253BC809D27498DD41BB0316D7E6B066960115DE155589D9C"
}
} ],
"BasicBuildingBlocks" : [ {
    "FC" : null,
    "ISC" : null,
    "VCI" : null,
    "XCV" : {
        "SubXCV" : [ ],
        "Constraint" : [ {
            "Name" : {
                "value" : "Can the certificate chain be built till a trust anchor?",
                "Key" : "BBB_XCV_CCCBB"
            },
            "Status" : "NOT OK",
            "Error" : {
                "value" : "The certificate chain is not trusted, it does not contain a
trust anchor.",
                "Key" : "BBB_XCV_CCCBB_ANS"
            },
            "Warning" : null,
            "Info" : null,
            "AdditionalInfo" : null,
            "Id" : null,
            "BlockType" : null
        } ],
        "Conclusion" : {
            "Indication" : "INDETERMINATE",
            "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
            "Errors" : [ {
                "value" : "The certificate chain is not trusted, it does not contain a
trust anchor.",
                "Key" : "BBB_XCV_CCCBB_ANS"
            } ],
            "Warnings" : [ ],
            "Infos" : [ ]
        },
        "Title" : "X509 Certificate Validation"
    },
    "CV" : null,
    "SAV" : null,
    "PSV" : null,
    "PCV" : null,
    "VTS" : null,
    "CertificateChain" : null,
    "Conclusion" : {

```

```

    "Indication" : "INDETERMINATE",
    "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
    "Errors" : [ {
        "value" : "The certificate chain is not trusted, it does not contain a trust
anchor.",
        "Key" : "BBB_XCV_CCCBB_ANS"
    } ],
    "Warnings" : [ ],
    "Infos" : [ ]
},
"Id" : "C-02F3EBCA0163274253BC809D27498DD41BB0316D7E6B066960115DE155589D9C",
"Type" : "CERTIFICATE"
} ],
"TLAnalysis" : [ ],
"Semantic" : null,
"ValidationTime" : null
}
}
}

```

REST remote timestamp service

Get Timestamp Response

This service allows a remote timestamp creation. The method takes as an input the digest to be timestamped and digest algorithm that has been used for the digest value computation. The output of the method is the generated timestamp's binaries.

Request

```

POST /services/rest/timestamp-service/getTimestampResponse HTTP/1.1
Accept: application/json, application/javascript, text/javascript, text/json
Content-Type: application/json; charset=UTF-8
Host: localhost:8080
Content-Length: 91

{
    "algorithm" : "SHA256",
    "value" : "Z0yIygCya0W6GjVnihtTFtIS9PNmskdyMlNKiuyjfzw="
}

```

Response

```

HTTP/1.1 200 OK
Date: Fri, 09 Jul 2021 08:02:37 GMT
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0

```

X-Frame-Options: DENY
Server: ESIG-DSS
Content-Type: application/json
Transfer-Encoding: chunked
Keep-Alive: timeout=20
Connection: keep-alive
Content-Length: 3287

```
{  
  "binaries" :  
  "MIIJigYJKoZIhvcNAQcCoIIJezCCCXcCAQMxDTALBglghkgBZQMEAgEwcAYLKOZIhvcNAQkQAQSgYQRfMF0CA  
  QEGAYoDBDAvMASGCWCGSAFlAwQCAQQgZ0yIygCya0W6GjVnihtTfTis9PnmskdyM1NKiuyjfzwCEQDT4Uk/A+K  
  2tSneQgHCLZ7HGA8yMDIxMDcwOTA4MDIzN1qgggWFMIIFgTCCA2mgAwIBAgIUGTcivXeyQZ1vqc77RH9+3LWD7  
  oowDQYJKoZIhvcNAQELBQAwUTELMAkGA1UEBhMCQkUxFTATBgnVBAoMDERTUyBURVNUIFBLSTERMA8GA1UECww  
  IVEVTVCBQS0kxGDAWBgnVBAAMD3N1bGytc21nbmVkLXRzYTAeFw0yMDEyMTYwNzAzMzZaFw0zMDEyMTQwNzAzM  
  zZaMFExCzAJBgnVBAyTAKJFMRUwEwYDVQQKDAxEU1MgVEVTVCBQS0kxETAPBgnVBAsMCFRFU1QgUEtJMRgwFgY  
  DVQQDDA9zZWxmLXNpZ251ZC10c2EwggIiMA0GCSqGSIB3DQEBAQUAA4ICDwAwggIKAoICAQDG01Qy0zCXb7u/8  
  4JqrQJi4RpqPrVSFn0QbKvFfuffjIhesqgbVrkYFUmQZNhIEKvXW/BPUdMLMdXgQsElV1nQrMGg8875GFBD8H  
  Cui+bAAIp72Hn390UmD2Z7KFV9+3MEpLqTuJwaKwQLcgci0fUjizbzk0knPI/oS678dGR4v02cGDWf+h2uXE2v  
  NdulCgC7K7ZVoeUJ4FkzNURhhWQTPGSrlrmF2kDmzMFMJniKaLrAcHfFphsOUpWKqGxPR24svCDbuifXxge4e  
  4ndE570yl+qaKODnbU2afOLMA/UAkzsBnBR9otwGTTvnIymjZPikhvimbSAosXkU6iY/3P4+buwCQKgA2pV/8tV  
  EG3u/z+U77pIr4QoT+0zEh+aFaKJ0Vby3Q5Bb0N1f1IKpV7skf30WSe/7aKFR1dC2YUcenb3RUnQ2y/nFKTwm1  
  NFkAXHy8Lt8Wz+q0NiBURzW7JJQmZ/hmESMuj9yZ+stmlJUvc00gPA1SH4sepE2zS3HJQEitxzBHPiaDrIb2ab  
  oemKBT1J0MQL+ej/ReBsobJixuqsZ4rrTHptcSFLvXF+tjcYh+5jk8uNFkcZPV5f199jw3pB7E19CE0hYQKQYZ  
  /LxLaQSFx+58n3Gp2zHRurrfz+7I0JYytLdd7Cnzb+RbTzTdSgpiiyhwu1Bx104SAUMQIDAQABo1EwTzAJBgnVH  
  RMEAjAAMAsGA1UdDwQEAvIHgDAWBgnVHSUBAf8EDDAKBggRgEFBQcDCDAdbgNVHQ4EFgQUx/Bc6x+I3yHJnKM  
  Tunz62C1TlogwDQYJKoZIhvcNAQELBQADggIBAHLtUgoBdF0VNmgZ4p6FwmxLhCwZz/RKZ5WrwntQMt/AmPr4V  
  61vftyXSfRWC/UfWhrFnQnrJakublbp1stpctrA9v41qtaJDbFWD1nyXL/jYpbc7TPdMTUuh+S4Ymrld1Usy7nc  
  TgKaRwAVav8N0l0AErLSpBd31ANBk0ScPB40vzYd0g7rUK+Tsj8BOXYR38gLaojreBeRRFIhuXkgCL+i/00ZV  
  10dNMkqgYtbP3bPLYUm5Rkk+LX/iWETCudXcBLHncN4hd2s2yBmMULTG41YaDFwL/VWOBrlSoa3tqIwp18RFKB  
  jMU8z1V10Ua+YtW9DMe6xNyZrvJgCR3ZS5kb1XLk1Frts0LMnLhh7qq1nwONpUCVE00ZiqC13M2qMDx14jWgtZ  
  E5xSUSGio+H1sjQj9imCWLuhcRnTwEpYEhziYhshrvC3v6QGejD13RjxcWZG/RSo0905DMjWoBFVLSZLFjShNt  
  4zwuWKh0gtQEeW3Ya8WdgB5Y1rg1cGzxLKWxJY3u8c56Iogk1S6s2U10PBnsPVF7FyySF52B90h0mSier9iye  
  rngUz20NxWRjv0138AAh9e9Ko55LAtX15ak2jPnfYUxyxkGVdLz0ECWZ8jC7FQnSKxpZ4hwSCLMof+5+I8dCho  
  DoGA/kaAWHh/TsP1DQwjeJTdTQrtXuTAugMYIDzjCCA2ICAQewaTBRMQswCQYDVQQGEwJCRTEVMBMGA1UECgwMR  
  FNTIFRFU1QgUEtJMRewDwYDVQLDAhURVNUIFBLSTEYMBGA1UEAwwPc2VsZi1zaWduZWQtdHNhAhQzNyK9d7J  
  BnW+pzvtEf37ctYpuijALBglghkgBZQMEAgGggdEwGgYJKoZIhvcNAQkDMQ0GCyqGSIB3DQEJEAEEBwGCSqGS  
  Ib3DQEJBTEPfw0yMTA3MDkw0DAyMzdaMCsGCSqGSIB3DQEJNDEeMBwwCwYJYIZIAWUDBAIBoQ0GCSqGSIB3DQE  
  BCwUAMC8GCSqGSIB3DQEJBDEiBCAjBu34G/2LynxVyx8if16iKX3DJm56Gx/2HGqAd0jYVTA3Bgsqhkig9w0BC  
  RAeLzEoMCYwJDAiBCA2jS+vUa9rFQTeeWd0a4TS85RBVv57qvKm9s1gWiAXjANBgkqhkiG9w0BAQsFAASCAgC  
  M6kTPoFj2CqGubKvIBS4MmrWIb2JeXbHVWVjnsBS1NZj1TXG8V/bNGPn60pXK2SzwsGmgMu8QkACTup83nGM4  
  vv22wlREHFqgYo0BvEMbIwKyT1Ve6aTD2KSvwwPwnCuimDS481/SBWPd7HKZzyfc/7SceAXzB0aiecFe2C8q6  
  f648K9yp6np3sjJwZxhZzDdIMGWgsu7X+K00z69CpXnUbRZsbCwJc/g+K/36uknX+bR1LhaLoSH7MssRAD/awD  
  W0cnA1bfPfeH63eQDa0cgjvD/HR0HKZrla/BTZfeMgj16wbHJ+Nwb10Wnsy1RdnYRoVeM3zPyPat1Gomj4+5gC  
  S1A/NenU03Bu1taBeR5qzE9asR1RCedgigA9vld2cifh22fUDgnhnbEEmkaenACaG4c9HnHINQoU95+oIgoeTZ  
  d+AFOTXx9BSHAM43Ku9c4fUxzM2cAU69uLisr0T7tuMxEm/I+3q17186u8k/C1gnAm6wc2yRpF+T1VU24TAUyo  
  SabHJ5vsCZt+ItPP8WU6jwjtjB19f/1Md97IqiLvkVypxGqZVi9t8sY/9e00By+o49xDraclw19DZp8G1+pxhz3  
  KGOPW5D8JNMn7NQVCm+GfDIxxZuCMwhxmAWvuIufpy1T00pZ0C9q47NPvy91ZnzRsPkSzuxynEvPgNqyQ=="  
 }
```