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 . . . 43
- BASELINE-T . . . 43
- BASELINE-LT and -LTA . . . 44

Signature Validation . . . 44
- Validation Process . . . 44
- Validation Result Materials . . . 49
- Customised Validation Policy . . . 90

CAdES signature (CMS) . . . 97

PAdES signature (PDF) . . . 98
- PAdES Visible Signature . . . 101
- Shadow attack detection . . . 108

JAdES signature (JWS) . . . 108

JWS Serialization type . . . 109
JAdES Signature Packaging . . . 110
Base64Url encoding . . . 111
ASiC signature (containers) . . . 112
Counter signatures . . . 115
Various parameters . . . 116
Signature policy . . . 116
Signature Policy Store . . . 121
Trust anchor inclusion policy . . . 122
Timestamps . . . 122
Timestamp creation . . . 122
Timestamp validation . . . 123
Available implementations of DSSDocument . . . 128
Management of signature tokens . . . 129
PKCS#11 . . . 129
PKCS#12 . . . 130
MS CAPI . . . 130
Other Implementations . . . 131
Management of certificates sources . . . 131
Management of CRL and OCSP sources . . . 132
Repository Revocation Source . . . 133
Other implementations of CRL and OCSP Sources . . . 135
CertificateVerifier configuration . . . 137
Trust Anchor(s) configuration . . . 139
Trust store initialization . . . 139
Trusted List Certificate Source . . . 140
TLValidationJob . . . 141
TLSource and LOTLSource . . . 141
DSSFileLoader . . . 143
The SynchronizationStrategy . . . 144
The CacheCleaner . . . 145
Alerting from TL Loading . . . 145
Executor Service . . . 146
Complete configuration for the European LOTL . . . 147
The TL / LOTL refresh . . . 149
TLValidationJobSummary . . . 151
TSP Sources . . . 152
Time-stamp policy . . . 153
Composite TSP Source . . . 153
Supported algorithms . . . 154
Implementation management with ServiceLoader . . . 155

Document Validation Factory . . . 156
Signature Policy Validator . . . 157
Multi-threading . . . 157
Resource sharing . . . 157
Caching . . . 157
XML Securities . . . 158
JAXB modules . . . 159
Report stylesheets . . . 161
Alerts . . . 162
I18N (Internationalization) . . . 162
Additional features . . . 163
Certificate validation . . . 163
SSL Certificate validation (QWAC) . . . 164
Extract the signed data from a signature . . . 166
REST and SOAP Services . . . 166
REST signature service . . . 167
REST server signature service . . . 281
REST validation service . . . 287
REST certificate validation service . . . 313
REST remote timestamp service . . . 319

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 5.8 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	draft 0.0.6
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
- TL Browser
- 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 9 and higher (tested up to Java 15) for the build is required. For usage Java 8 is a mimimum 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-xmldsig;
- 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 asciidoctor
```

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 15) 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 ad 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 availble 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 15, the following profile can be used from `dss-demo-bundle` module:

```
mvn clean install -P java15
```

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-xmldsig

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)

specs-jades

ETSI TS 119 182-1 v.0.0.6 JSON Schemas for JAdES

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 doen'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 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 Serialization	✓	✓	✓	✓	✗	✗	✗	✗	✗
		Flattened JSON Serialization	✓	✓	✓	✓	✗	✗	✗	✓	✗
		JSON Serialization	✓	✓	✓	✓	✗	✗	✓	✓	✗
	Detached	Compact Serialization	✓	✓	✓	✓	✓	SigD only	✗	✗	✗
		Flattened JSON Serialization	✓	✓	✓	✓	✓	SigD only	✗	✓	✗
		JSON Serialization	✓	✓	✓	✓	✓	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_example.xml

```
<?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<CommitmentType>();
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>2FeANjXziO9x2877SfclRlRVjlE=</ds:DigestValue>
                </xades:CertDigest>

<xades:IssuerSerialV2>MD4wNKQyMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkRmFrZTERMA8GA1UECgwI
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:EncapsulatedTimeStam
p>
        </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">
            MIAGCSqGSIb3DQEHAqCAMIACAQMxCzAJBgUrDgMCGg
            UAMIAGCyqGSIb3DQEJEAEEoIAkgARMMEoCAQEGBoIS
            ...
    </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");

LOTLSource lotlSource = new LOTLSource();
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>
            MIIGzAoBAKCCBsUwggbBBgkr...
        </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.setXmldsigNamespace(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 xmlDocumentValidator = new XMLDocumentValidator(xmlDocument);
xmlDocumentValidator.setCertificateVerifier(certificateVerifier);

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

Reports reports = xmlDocumentValidator.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,...). 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();
// optionally the certificate chain can be provided
List<CertificateToken> certificateChain =
Arrays.asList(privateKeyEntry.getCertificateChain());

// 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 the packaging, the references, the
canononicalization,...
TrustedListSignatureParametersBuilder builder = new
TrustedListSignatureParametersBuilder(signingCertificate, certificateChain,
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:

XAdES Unsigned Signature Properties

```
<UnsignedSignatureProperties>
    <SignatureTimeStamp Id="time-stamp-b16a2552-b218-4231-8982-40057525fbb5">
        <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.

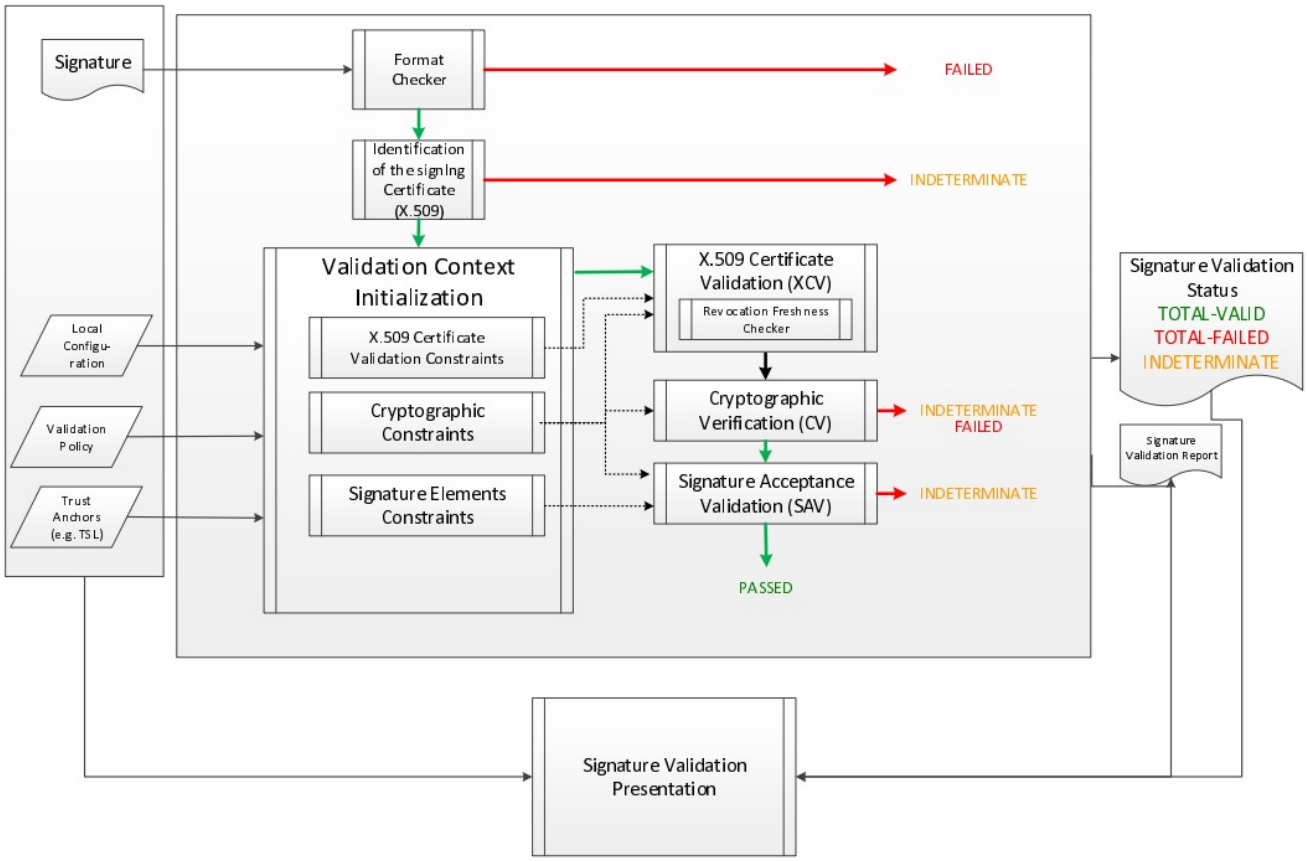


Figure 1. Signature Validation Process

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.setDataLoader(new CommonsDataLoader());

// 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.loadCertificateFromBase64EncodedStrin
g(

"MIIC9TCCAd2gAwIBAgIBAjANBgkqhkiG9w0BAQUFADArMQswCQYDVQQGEwJBQTEMMAoGA1UEChMDRFNTMQ4wD
AYDVQQDEwVJQ0EgQTAeFw0xMzEyMDIxNzMzMTBaFw0xNTEyMDIxNzMzMTBaMDAxCzAJBgNVBAYTAkFBMQwwCgY
DVQQKEwNEU1MxEzARBgNVBAMTCnVzZXIgQSBSU0EwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJUHHAphm
SDdQ1t62tppK+dLTANsE2nAj+HCpasS3ohlBsrhteRsvTAbrDyIzCmTYWu/nVI4TGvbzBESwV/QitlkoMLpYFw
32MIBf2DLmECzGJ3vm5haw6u8S9quR1h8Vu7QWd+5KMabZuR+j91RiSuoY0xS2ZQxJw1vhvW9hRYjAgMBAAGjg
aIwgZ8wCQYDVR0TBAIwADAdBgNVHQ4EFgQU9ESnTWfwg13c3LQZzqqwibY5WVYwUwYDVR0jBEwwSoAUIO1CDsB
SUcEoFZxKaWf1PAL1U+uhL6QtMCsxDDAKBgNVBAoTA0RTUzELMAkGA1UEBhMCQUExDjAMBgNVBAMTBVJDQSBBg
gEBMAsGA1UdDwQEAwIHgDARBgNVHSAECjAIMAYGBFUdIAAwDQYJKoZIhvcNAQEFBQADggEBAGnhhnoyVUhDnr/
BSbZ/uWfSuwzFPG+2V9K6WxdIaaXOORFGIdFwGlAwA/Qzpq9snfBxuTkAykxq0uEDhHTj0qXxWRjQ+Dop/Drmc
coF/zDvgGusyY1YXaABd/kc3IYt7ns7z3tpiqIz4A7a/UHplBRXfqjyaZurZuJQRaSdxh6CNhdEUiUBxkbb1Sd
MjuOgjzSDjcDjcegjvDquMKdDetvtu2Qh4ConBBo3fUImwiFRWnbudS5H2HE18ikC7gY/QIuNr7USf1PNyUgcG
2g31cMtemj7UTBHZ2V/jPf7ZXqwfnVSaYkNvM3weAI6R3PI0STjdxN6a9qjt9xld40YEdw="));
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

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<SimpleReport ValidationTime="2020-12-14T11:03:21"
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>
    <DocumentName>EmptyPage-signed-pades-baseline-b.pdf</DocumentName>
    <ValidSignaturesCount>1</ValidSignaturesCount>
    <SignaturesCount>1</SignaturesCount>
    <Signature SignatureFormat="PAdES-BASELINE-B" ExtensionPeriodMin="2020-12-
14T11:04:21" ExtensionPeriodMax="2027-01-21T23:59:59" Id="S-
08113A9BAB65F6271F837FF4992635CC725B49D27B1AED0D714EAD428BE98C6E">
        <CertificateChain>
            <Certificate>
                <id>C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F</id>
                <qualifiedName>Pierrick Vandenbroucke (Signature)</qualifiedName>
            </Certificate>
            <Certificate>
                <id>C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA</id>
                <qualifiedName>Citizen CA</qualifiedName>
            </Certificate>
            <Certificate>
                <id>C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F</id>
                <qualifiedName>Belgium Root CA4</qualifiedName>
            </Certificate>
        </CertificateChain>
        <Indication>TOTAL_PASSED</Indication>
        <Warnings>The organization name is missing in the trusted
certificate!</Warnings>
        <SigningTime>2019-08-27T14:06:11</SigningTime>
        <BestSignatureTime>2020-12-14T11:03:21</BestSignatureTime>
        <SignedBy>Pierrick Vandenbroucke (Signature)</SignedBy>
        <SignatureLevel description="Qualified Electronic
Signature">QESig</SignatureLevel>
        <SignatureScope name="Full PDF" 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

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DetailedReport ValidationTime="2020-12-14T11:03:21"
xmlns="http://dss.esig.europa.eu/validation/detailed-report">
    <Signature Id="S-
08113A9BAB65F6271F837FF4992635CC725B49D27B1AED0D714EAD428BE98C6E">
        <ValidationProcessBasicSignature Title="Validation Process for Basic
Signatures">
            <Constraint Id="S-
08113A9BAB65F6271F837FF4992635CC725B49D27B1AED0D714EAD428BE98C6E">
                <Name NameId="ADEST_ROBVPIIC">Is the result of the Basic Validation
Process conclusive?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
            <ProofOfExistence>
                <Time>2020-12-14T11:03:21</Time>
            </ProofOfExistence>
        </ValidationProcessBasicSignature>
        <ValidationProcessLongTermData Title="Validation Process for Signatures with
Time and Signatures with Long-Term Validation Data">
            <Constraint>
                <Name NameId="LTV_ABSV">Is the result of the Basic Validation Process
acceptable?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint Id="R-
83B6DB1D1C8369D90161C212DEBCE29E2608D9B117356B8AA9E531721CB1F5BF">
                <Name NameId="ADEST_RORPIIC">Is the result of the revocation data
validation process acceptable?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_XCV_IRDC">Is the revocation data consistent?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Revocation R-
```

```
83B6DB1D1C8369D90161C212DEBCE29E2608D9B117356B8AA9E531721CB1F5BF thisUpdate 2020-12-14
11:03 is in the certificate validity range : 2017-01-25 22:12 - 2027-01-21
23:59</AdditionalInfo>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_XCV_IARDPFC">Is an acceptable revocation data
present for the certificate?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Certificate Id = C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F</AdditionalInfo>
            </Constraint>
            <Constraint Id="R-
46B3B0FDA4A56A2601A20B31701AF2B1D0DCAEA8A903CD1346FBA80F49DFF1BD">
                <Name NameId="ADEST_RORPIIC">Is the result of the revocation data
validation process acceptable?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_XCV_IRDC">Is the revocation data consistent?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Revocation R-
46B3B0FDA4A56A2601A20B31701AF2B1D0DCAEA8A903CD1346FBA80F49DFF1BD thisUpdate 2020-07-01
11:00 is in the certificate validity range : 2015-11-25 10:00 - 2027-07-25
10:00</AdditionalInfo>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_XCV_IARDPFC">Is an acceptable revocation data
present for the certificate?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Certificate Id = C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA</AdditionalInfo>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_SAV_ISVA">Is the signature acceptable?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
            <ProofOfExistence>
                <Time>2020-12-14T11:03:21</Time>
            </ProofOfExistence>
        </ValidationProcessLongTermData>
        <ValidationProcessArchivalData Title="Validation Process for Signatures with
Archival Data">
            <Constraint>
                <Name NameId="ARCH_LTVV">Is the result of the LTV validation process
acceptable?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
```

```
                <Indication>PASSED</Indication>
            </Conclusion>
            <ProofOfExistence>
                <Time>2020-12-14T11:03:21</Time>
            </ProofOfExistence>
        </ValidationProcessArchivalData>
        <ValidationSignatureQualification SignatureQualification="QESig"
Title="Signature Qualification">
            <Constraint>
                <Name NameId="QUAL_IS_ADES">Is the signature/seal an acceptable AdES
digital signature (ETSI EN 319 102-1)?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="QUAL_CERT_TRUSTED_LIST_REACHED">Has a trusted list been
reached for the certificate chain?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint Id="LOTL-
EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B">
                <Name NameId="QUAL_LIST_OF_TRUSTED_LISTS_ACCEPT">Is the list of
trusted lists acceptable?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Trusted List : https://ec.europa.eu/tools/lotl/eu-
lotl.xml</AdditionalInfo>
            </Constraint>
            <Constraint Id="TL-
61C0487109BE27255C19CFF26D8F56BEA621E7F381A7B4CBE7FB4750BD477BF9">
                <Name NameId="QUAL_TRUSTED_LIST_ACCEPT">Is the trusted list
acceptable?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Trusted List : https://tsl.belgium.be/tsl-
be.xml</AdditionalInfo>
            </Constraint>
            <Constraint>
                <Name NameId="QUAL_VALID_TRUSTED_LIST_PRESENT">Has been an acceptable
trusted list found?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="QUAL_QC_AT_ST">Is the certificate qualified at (best)
signing time?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="QUAL_FOR_SIGN_AT_ST">Is the certificate for eSig at
(best) signing time?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="QUAL_QC_AT_CC">Is the certificate qualified at issuance
```

```
time?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="QUAL_QSCD_AT_ST">Does the private key reside in a QSCD
at (best) signing time?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
            <ValidationCertificateQualification Id="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F" DateTime="2017-01-
25T22:12:12" ValidationTime="CERTIFICATE_ISSUANCE_TIME" CertificateQualification="QC
for eSig with QSCD" Title="Certificate Qualification at certificate issuance time">
                <Constraint>
                    <Name NameId="QUAL_HAS_CAQC">Is the certificate related to a
CA/QC?</Name>
                    <Status>OK</Status>
                </Constraint>
                ...
                <Conclusion>
                    <Indication>PASSED</Indication>
                </Conclusion>
            </ValidationCertificateQualification>
            <ValidationCertificateQualification Id="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F" DateTime="2020-12-
14T11:03:21" ValidationTime="BEST_SIGNATURE_TIME" CertificateQualification="QC for
eSig with QSCD" Title="Certificate Qualification at best signature time">
                <Constraint>
                    <Name NameId="QUAL_HAS_CAQC">Is the certificate related to a
CA/QC?</Name>
                    <Status>OK</Status>
                </Constraint>
                ...
                <Conclusion>
                    <Indication>PASSED</Indication>
                </Conclusion>
            </ValidationCertificateQualification>
        </ValidationSignatureQualification>
        <Conclusion>
            <Indication>TOTAL_PASSED</Indication>
        </Conclusion>
    </Signature>
    <BasicBuildingBlocks Id="R-
46B3B0FDA4A56A2601A20B31701AF2B1D0DCAEA8A903CD1346FBA80F49DFF1BD" Type="REVOCATION">
        <ISC Title="Identification of the Signing Certificate">
            ...
        </ISC>
        <XCV Title="X509 Certificate Validation">
            ...
```

```
        </XCV>
        <CV Title="Cryptographic Verification">
            ...
        </CV>
        <SAV ValidationTime="2020-12-14T11:03:21" Title="Signature Acceptance
Validation">
            ...
        </SAV>
        <CertificateChain>
            <ChainItem Id="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                <Source>TRUSTED_LIST</Source>
            </ChainItem>
        </CertificateChain>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
    </BasicBuildingBlocks>
    <BasicBuildingBlocks Id="R-
83B6DB1D1C8369D90161C212DEBCE29E2608D9B117356B8AA9E531721CB1F5BF" Type="REVOCATION">
        <ISC Title="Identification of the Signing Certificate">
            ...
        </ISC>
        <XCV Title="X509 Certificate Validation">
            ...
        </XCV>
        <CV Title="Cryptographic Verification">
            ...
        </CV>
        <SAV ValidationTime="2020-12-14T11:03:21" Title="Signature Acceptance
Validation">
            ...
        </SAV>
        <CertificateChain>
            <ChainItem Id="C-
B336CAA1F3C4930E4EF9C803C12877A004991EE9206C0D4AD3891688C1E478FF">
                <Source>OCSP_RESPONSE</Source>
            </ChainItem>
            <ChainItem Id="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA">
                <Source>SIGNATURE</Source>
            </ChainItem>
            <ChainItem Id="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                <Source>TRUSTED_LIST</Source>
            </ChainItem>
        </CertificateChain>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
    </BasicBuildingBlocks>
```

```
    <BasicBuildingBlocks Id="S-
08113A9BAB65F6271F837FF4992635CC725B49D27B1AED0D714EAD428BE98C6E" Type="SIGNATURE">
        <FC Title="Format Checking">
            <Constraint>
                <Name NameId="BBB_FC_IEFF">Does the signature format correspond to an
expected format?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_FC_ISD">Is the signature identification not
ambiguous?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_FC_ISRIA">Is the signed references identification
not ambiguous?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_FC_IOSIP">Is only one SignerInfo present?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_FC_DSFREAP">Do signed and final revisions contain
equal amount of pages?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_FC_IAOD">Is no element overlapping detected in the
PDF?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_FC_IVDBSFR">Is there no visual difference between
signed and final revisions in the PDF?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
        </FC>
        <ISC Title="Identification of the Signing Certificate">
            <Constraint>
                <Name NameId="BBB_ICS_ISCI">Is there an identified candidate for the
signing certificate?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_ICS_ISASCP">Is the signed attribute: 'signing-
certificate' present?</Name>
                <Status>OK</Status>
```

```
            </Constraint>
            <Constraint>
                <Name NameId="BBB_ICS_ISASCPU">Is the signed attribute: 'signing-
certificate' present only once?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_ICS_ISACDP">Is the signed attribute: 'cert-digest'
of the certificate present?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_ICS_ICDVV">Does the certificate digest value match a
digest value found in the certificate reference(s)?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_ICS_AIDNASNE">Are the issuer distinguished name and
the serial number equal?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
            <CertificateChain>
                <ChainItem Id="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F">
                    <Source>SIGNATURE</Source>
                </ChainItem>
                <ChainItem Id="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA">
                    <Source>SIGNATURE</Source>
                </ChainItem>
                <ChainItem Id="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                    <Source>TRUSTED_LIST</Source>
                </ChainItem>
            </CertificateChain>
        </ISC>
        <VCI Title="Validation Context Initialization">
            <Constraint>
                <Name NameId="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 NameId="BBB_XCV_CCCBB">Can the certificate chain be built till a
```

```
trust anchor?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint Id="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F">
                <Name NameId="BBB_XCV_SUB">Is the certificate validation
conclusive?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint Id="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA">
                <Name NameId="BBB_XCV_SUB">Is the certificate validation
conclusive?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint Id="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                <Name NameId="BBB_XCV_SUB">Is the certificate validation
conclusive?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
            <SubXCV Id="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F" TrustAnchor="false"
SelfSigned="false" Title="Certificate">
                <Constraint>
                    <Name NameId="QUAL_UNIQUE_CERT">Is the certificate unique?</Name>
                    <Status>OK</Status>
                </Constraint>
                ...
                <Conclusion>
                    <Indication>PASSED</Indication>
                </Conclusion>
                <RAC Id="R-
83B6DB1D1C8369D90161C212DEBCE29E2608D9B117356B8AA9E531721CB1F5BF" Title="Revocation
Acceptance Checker">
                    <Constraint>
                        <Name NameId="BBB_XCV_ISCUKN">Is the revocation status
known?</Name>
                        <Status>OK</Status>
                    </Constraint>
                    ...
                    <Conclusion>
                        <Indication>PASSED</Indication>
                    </Conclusion>
                    <RevocationProductionDate>2020-12-
14T11:03:21</RevocationProductionDate>
                </RAC>
                <RFC Id="R-
```

```
83B6DB1D1C8369D90161C212DEBCE29E2608D9B117356B8AA9E531721CB1F5BF" Title="Revocation
Freshness Checker">
                    <Constraint>
                        <Name NameId="BBB_XCV_IARDPFC">Is an acceptable revocation
data present for the certificate?</Name>
                        <Status>OK</Status>
                    </Constraint>
                    ...
                    <Conclusion>
                        <Indication>PASSED</Indication>
                    </Conclusion>
                </RFC>
            </SubXCV>
            <SubXCV Id="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA" TrustAnchor="false"
SelfSigned="false" Title="Certificate">
                ...
            </SubXCV>
            <SubXCV Id="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F" TrustAnchor="true"
SelfSigned="true" Title="Certificate">
                <Conclusion>
                    <Indication>PASSED</Indication>
                </Conclusion>
                <EquivalentCertificate>C-
C3FBF37259AF0954EEEA4282DD1C7226A54E7150F7C29A2C495BA34DBFE09CA0</EquivalentCertificat
e>
            </SubXCV>
        </XCV>
        <CV Title="Cryptographic Verification">
            <Constraint>
                <Name NameId="BBB_CV_IRDOF">Has the reference data object been
found?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Reference : MESSAGE_DIGEST</AdditionalInfo>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_CV_IRDOI">Is the reference data object
intact?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Reference : MESSAGE_DIGEST</AdditionalInfo>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_CV_ISI">Is the signature intact?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Id = S-
08113A9BAB65F6271F837FF4992635CC725B49D27B1AED0D714EAD428BE98C6E</AdditionalInfo>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
```

```
        </CV>
        <SAV ValidationTime="2020-12-14T11:03:21" Title="Signature Acceptance
Validation">
            <Constraint>
                <Name NameId="BBB_SAV_ISSV">Is the structure of the signature
valid?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_SAV_DSCACRCC">Does Signing Certificate attribute
contains references only to the certificate chain?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_SAV_ISQPSTP">Is the signed qualifying property:
'signing-time' present?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_SAV_ISQPMDOSPP">Is the signed qualifying property:
'message-digest' or 'SignedProperties' present?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="ACCM">Are cryptographic constraints met for the
signature creation?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Validation time : 2020-12-14 11:03</AdditionalInfo>
            </Constraint>
            <Constraint>
                <Name NameId="ACCM">Are cryptographic constraints met for the message
digest?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Validation time : 2020-12-14 11:03 for message
digest</AdditionalInfo>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
            <CryptographicInfo>
                <Algorithm>http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256</Algorithm>
                <KeyLength>2048</KeyLength>
                <Secure>true</Secure>
                <NotAfter>2022-12-31T23:00:00</NotAfter>
                <ConcernedMaterial>S-
08113A9BAB65F6271F837FF4992635CC725B49D27B1AED0D714EAD428BE98C6E</ConcernedMaterial>
            </CryptographicInfo>
        </SAV>
        <CertificateChain>
            <ChainItem Id="C-
```

```
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F">
                <Source>SIGNATURE</Source>
            </ChainItem>
            <ChainItem Id="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA">
                <Source>SIGNATURE</Source>
            </ChainItem>
            <ChainItem Id="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                <Source>TRUSTED_LIST</Source>
            </ChainItem>
        </CertificateChain>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
    </BasicBuildingBlocks>
    <TLAnalysis CountryCode="EU" URL="https://ec.europa.eu/tools/lotl/eu-lotl.xml"
Id="LOTL-EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B" Title="List
Of Trusted Lists EU">
        <Constraint>
            <Name NameId="QUAL_TL_FRESH">Is the trusted list fresh?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_EXP">Is the trusted list not expired?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_VERSION">Does the trusted list have the expected
version?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_WS">Is the trusted list well signed?</Name>
            <Status>OK</Status>
        </Constraint>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
    </TLAnalysis>
    <TLAnalysis CountryCode="BE" URL="https://tsl.belgium.be/tsl-be.xml" Id="TL-
61C0487109BE27255C19CFF26D8F56BEA621E7F381A7B4CBE7FB4750BD477BF9" Title="Trusted List
BE">
        <Constraint>
            <Name NameId="QUAL_TL_FRESH">Is the trusted list fresh?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_EXP">Is the trusted list not expired?</Name>
            <Status>OK</Status>
        </Constraint>
```

```
        <Constraint>
            <Name NameId="QUAL_TL_VERSION">Does the trusted list have the expected
version?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_WS">Is the trusted list well signed?</Name>
            <Status>OK</Status>
        </Constraint>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
    </TLAnalysis>
</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

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DiagnosticData xmlns="http://dss.esig.europa.eu/validation/diagnostic">
    <DocumentName>EmptyPage-signed-pades-baseline-b.pdf</DocumentName>
    <ValidationDate>2020-12-14T11:03:21</ValidationDate>
    <Signatures>
        <Signature Id="S-
08113A9BAB65F6271F837FF4992635CC725B49D27B1AED0D714EAD428BE98C6E">
            <SignatureFilename>EmptyPage-signed-pades-baseline-
b.pdf</SignatureFilename>
            <ClaimedSigningTime>2019-08-27T14:06:11</ClaimedSigningTime>
            <SignatureFormat>PAdES-BASELINE-B</SignatureFormat>
            <StructuralValidation valid="true"/>
            <DigestMatchers>
                <DigestMatcher type="MESSAGE_DIGEST">
                    <DigestMethod>SHA256</DigestMethod>

<DigestValue>SGEPVFOj/zskv8+nlzixt+PbLxWE9SS67rkpd0V5Wi4=</DigestValue>
                    <DataFound>true</DataFound>
                    <DataIntact>true</DataIntact>
                </DigestMatcher>
            </DigestMatchers>
            <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
                <KeyLengthUsedToSignThisToken>2048</KeyLengthUsedToSignThisToken>
                <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
                <SignatureIntact>true</SignatureIntact>
                <SignatureValid>true</SignatureValid>
            </BasicSignature>
            <SigningCertificate Certificate="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F"/>
            <CertificateChain>
                <ChainItem Certificate="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F"/>
                <ChainItem Certificate="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA"/>
```

```
                <ChainItem Certificate="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            </CertificateChain>
            <ContentType>1.2.840.113549.1.7.1</ContentType>
            <CommitmentTypeIndications/>
            <Policy>
                <Id>IMPLICIT_POLICY</Id>
                <ZeroHash>false</ZeroHash>
                <Asn1Processable>false</Asn1Processable>
                <Identified>false</Identified>
                <Status>false</Status>
                <DigestAlgorithmsEqual>false</DigestAlgorithmsEqual>
            </Policy>
            <SignerInformationStore>
                <SignerInfo Current="true">
                    <IssuerName>SERIALNUMBER=201631, CN=Citizen CA, C=BE</IssuerName>

<SerialNumber>212676479325593460004449038464688276 73</SerialNumber>
                </SignerInfo>
            </SignerInformationStore>
            <PDFRevision>
                <SignatureFieldName>Signature1</SignatureFieldName>
                <PDFSignatureDictionary>
                    <SignerName>Pierrick Vandenbroucke (Signature)
70a3cb70f0f4d6513fb12cf0691965c58c7e7679</SignerName>
                    <Type>Sig</Type>
                    <Filter>Adobe.PPKLite</Filter>
                    <SubFilter>ETSI.CAdES.detached</SubFilter>
                    <SignatureByteRange>0 5340 43230 342</SignatureByteRange>
                </PDFSignatureDictionary>
            </PDFRevision>
            <SignerDocumentRepresentations HashOnly="false" DocHashOnly="false"/>
            <FoundCertificates>
                <RelatedCertificate Certificate="C-
C3FBF37259AF0954EEEA4282DD1C7226A54E7150F7C29A2C495BA34DBFE09CA0">
                    <Origin>SIGNED_DATA</Origin>
                </RelatedCertificate>
                <RelatedCertificate Certificate="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F">
                    <Origin>SIGNED_DATA</Origin>
                    <CertificateRef>
                        <Origin>SIGNING_CERTIFICATE</Origin>
                        <IssuerSerial
match="true">MEswN6Q1MDMxCzAJBgNVBAYTAkJFMRMwEQYDVQQDEwpDaXRpemVuIENBMQ8wDQYDVQQFEwYyM
DE2MzECEBAAAAAAAJKLPMkehMM6uhk=</IssuerSerial>
                        <DigestAlgoAndValue match="true">
                            <DigestMethod>SHA256</DigestMethod>

<DigestValue>T6spAncn5Y5FGO0LauVU0FXwWz2Rl+DRayACjSJ9Gp8=</DigestValue>
                        </DigestAlgoAndValue>
                    </CertificateRef>
```

```
                </RelatedCertificate>
                <RelatedCertificate Certificate="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA">
                    <Origin>SIGNED_DATA</Origin>
                </RelatedCertificate>
            </FoundCertificates>
            <FoundRevocations/>
            <FoundTimestamps/>
            <SignatureScopes>
                <SignatureScope SignerData="D-
BD598965790CA5673910D64EFEDA798485364C66B6F4E7C49D23E1FF8BAFBBE8">
                    <Scope>FULL</Scope>
                    <Name>Full PDF</Name>
                    <Description>Full document</Description>
                </SignatureScope>
            </SignatureScopes>
            <SignatureDigestReference>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>uZKN338SfkpJkk+eM3bEDx3UgdcPW41o7liEALTjb9M=</DigestValue>
            </SignatureDigestReference>
            <DataToBeSignedRepresentation>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>e3CCNW+o+NudYnfnQfN5oPXdaV5eSmYYwFwBb3TvDLs=</DigestValue>
            </DataToBeSignedRepresentation>

<SignatureValue>m2sMUIIfYKHDVr1IMTyVufWJcwuxwhWjGvkF/xa/rOAKieZCe4LZPa0/uwzgwM4QAbPstd
y4gHSQzCF0R6/fT9hv639kQS3TyZedw1raMeDj9mQOwKOlMl1OIxEI7jSf7xP6n62s0wQAhTJlARnOY1G5vppz
iVKb1vPED27HPBB4Yljn8j6hse+EJ0bwxAN1gwufbxZBvjHYgz/U/9EHafa1oGPcoIBrXvoUdzVX76sVE3nlDv
X4psEU4eq7paIZA7AWGSfWk8/k98pPqFcP2VYJaAju9GI+uZNMfRgPdOvGPxTjUBYiEyr3satod+cMQGiAzie8
0n0ovQrfn7ebcA==</SignatureValue>
        </Signature>
    </Signatures>
    <UsedCertificates>
        <Certificate Id="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA">
            <SubjectDistinguishedName
Format="CANONICAL">2.5.4.5=#1306323031363331,cn=citizen
ca,c=be</SubjectDistinguishedName>
            <SubjectDistinguishedName
Format="RFC2253">2.5.4.5=#1306323031363331,CN=Citizen
CA,C=BE</SubjectDistinguishedName>
            <IssuerDistinguishedName Format="CANONICAL">cn=belgium root
ca4,c=be</IssuerDistinguishedName>
            <IssuerDistinguishedName Format="RFC2253">CN=Belgium Root
CA4,C=BE</IssuerDistinguishedName>
            <SerialNumber>311326114055543504637452593660348150 60</SerialNumber>
            <SubjectSerialNumber>201631</SubjectSerialNumber>
            <CommonName>Citizen CA</CommonName>
            <CountryName>BE</CountryName>
```

```
            <AuthorityInformationAccessUrls/>
            <CRLDistributionPoints>
                <crlUrl>http://crl.eid.belgium.be/belgium4.crl</crlUrl>
            </CRLDistributionPoints>
            <OCSPAccessUrls/>
            <Sources>
                <Source>SIGNATURE</Source>
                <Source>OCSP_RESPONSE</Source>
            </Sources>
            <NotAfter>2027-07-25T10:00:00</NotAfter>
            <NotBefore>2015-11-25T10:00:00</NotBefore>
            <PublicKeySize>4096</PublicKeySize>
            <PublicKeyEncryptionAlgo>RSA</PublicKeyEncryptionAlgo>
            <EntityKey>PK-
490F57DD2D327C98CFE3D09E8E54C0B30DBBFD3669681C1013F79F4A6ADD09AC</EntityKey>
            <KeyUsageBits>
                <KeyUsage>keyCertSign</KeyUsage>
                <KeyUsage>crlSign</KeyUsage>
            </KeyUsageBits>
            <ExtendedKeyUsages/>
            <IdPkixOcspNoCheck>false</IdPkixOcspNoCheck>
            <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
                <KeyLengthUsedToSignThisToken>4096</KeyLengthUsedToSignThisToken>
                <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
                <SignatureIntact>true</SignatureIntact>
                <SignatureValid>true</SignatureValid>
            </BasicSignature>
            <SigningCertificate Certificate="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            <CertificateChain>
                <ChainItem Certificate="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            </CertificateChain>
            <Trusted>false</Trusted>
            <SelfSigned>false</SelfSigned>
            <CertificatePolicies>
                <certificatePolicy
cpsUrl="http://repository.eid.belgium.be">2.16.56.12.1.1.2</certificatePolicy>
            </CertificatePolicies>
            <QCStatementIds/>
            <QCTypes/>
            <TrustedServiceProviders>
                <TrustedServiceProvider TL="TL-
61C0487109BE27255C19CFF26D8F56BEA621E7F381A7B4CBE7FB4750BD477BF9" LOTL="LOTL-
EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B">
                    <TSPNames>
                        <TSPName lang="en">Certipost n.v./s.a.</TSPName>
                    </TSPNames>
                    <TSPTradeNames>
```

```
                        <TSPTradeName lang="en">Certipost s.a./n.v.</TSPTradeName>
                    </TSPTradeNames>
                    <TSPRegistrationIdentifiers>
                        <TSPRegistrationIdentifier>VATBE-
0475396406</TSPRegistrationIdentifier>
                    </TSPRegistrationIdentifiers>
                    <TrustedServices>
                        <TrustedService ServiceDigitalIdentifier="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                            <ServiceNames>
                                <ServiceName lang="en">CN=Belgium Root CA4,
C=BE</ServiceName>
                            </ServiceNames>

<ServiceType>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</ServiceType>

<Status>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted</Status>
                            <StartDate>2016-06-30T22:00:00</StartDate>
                            <AdditionalServiceInfoUris>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/RootCA-
QC</AdditionalServiceInfoUri>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSigna
tures</AdditionalServiceInfoUri>
                            </AdditionalServiceInfoUris>
                        </TrustedService>
                        <TrustedService ServiceDigitalIdentifier="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                            <ServiceNames>
                                <ServiceName lang="en">CN=Belgium Root CA4,
C=BE</ServiceName>
                            </ServiceNames>

<ServiceType>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</ServiceType>

<Status>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/undersupervision</Status>
                            <StartDate>2013-06-26T12:00:00</StartDate>
                            <EndDate>2016-06-30T22:00:00</EndDate>
                            <AdditionalServiceInfoUris>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/RootCA-
QC</AdditionalServiceInfoUri>
                            </AdditionalServiceInfoUris>
                        </TrustedService>
                    </TrustedServices>
                </TrustedServiceProvider>
            </TrustedServiceProviders>
            <Revocations>
                <CertificateRevocation Revocation="R-
46B3B0FDA4A56A2601A20B31701AF2B1D0DCAEA8A903CD1346FBA80F49DFF1BD">
```

```
                    <Status>GOOD</Status>
                </CertificateRevocation>
            </Revocations>
            <DigestAlgoAndValue>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>KT0Lo6MeXYKo4/rhJwmTL/3aREI+D3M/sB7xI+c+tNo=</DigestValue>
            </DigestAlgoAndValue>
        </Certificate>
        <Certificate Id="C-
4FAB29027727E58E4518ED0B6AE554D055F05B3D9197E0D16B20028D227D1A9F">
            <SubjectDistinguishedName
Format="CANONICAL">2.5.4.5=#130b383730313237333037333338,2.5.4.42=#130d506965727269636b2
05061636f,2.5.4.4=#130d56616e64656e62726f75636b65,cn=pierrick vandenbroucke
(signature),c=be</SubjectDistinguishedName>
            <SubjectDistinguishedName
Format="RFC2253">2.5.4.5=#130b383730313237333037333338,2.5.4.42=#130d506965727269636b205
061636f,2.5.4.4=#130d56616e64656e62726f75636b65,CN=Pierrick Vandenbroucke
(Signature),C=BE</SubjectDistinguishedName>
            <IssuerDistinguishedName
Format="CANONICAL">2.5.4.5=#1306323031363331,cn=citizen
ca,c=be</IssuerDistinguishedName>
            <IssuerDistinguishedName
Format="RFC2253">2.5.4.5=#1306323031363331,CN=Citizen
CA,C=BE</IssuerDistinguishedName>
            <SerialNumber>212676479325593460004449038464688276 73</SerialNumber>
            <SubjectSerialNumber>87012730738</SubjectSerialNumber>
            <CommonName>Pierrick Vandenbroucke (Signature)</CommonName>
            <CountryName>BE</CountryName>
            <GivenName>Pierrick Paco</GivenName>
            <Surname>Vandenbroucke</Surname>
            <AuthorityInformationAccessUrls>
                <aiaUrl>http://certs.eid.belgium.be/belgiumrs4.crt</aiaUrl>
            </AuthorityInformationAccessUrls>
            <CRLDistributionPoints>
                <crlUrl>http://crl.eid.belgium.be/eidc201631.crl</crlUrl>
            </CRLDistributionPoints>
            <OCSPAccessUrls>
                <ocspServerUrl>http://ocsp.eid.belgium.be/2</ocspServerUrl>
            </OCSPAccessUrls>
            <Sources>
                <Source>SIGNATURE</Source>
            </Sources>
            <NotAfter>2027-01-21T23:59:59</NotAfter>
            <NotBefore>2017-01-25T22:12:12</NotBefore>
            <PublicKeySize>2048</PublicKeySize>
            <PublicKeyEncryptionAlgo>RSA</PublicKeyEncryptionAlgo>
            <EntityKey>PK-
89D4C291843414CFB5E13D00B17CC57CD50E9FEAB6230AE5535C239AFB595681</EntityKey>
            <KeyUsageBits>
                <KeyUsage>nonRepudiation</KeyUsage>
```

```
            </KeyUsageBits>
            <ExtendedKeyUsages/>
            <IdPkixOcspNoCheck>false</IdPkixOcspNoCheck>
            <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
                <KeyLengthUsedToSignThisToken>4096</KeyLengthUsedToSignThisToken>
                <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
                <SignatureIntact>true</SignatureIntact>
                <SignatureValid>true</SignatureValid>
            </BasicSignature>
            <SigningCertificate Certificate="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA"/>
            <CertificateChain>
                <ChainItem Certificate="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA"/>
                <ChainItem Certificate="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            </CertificateChain>
            <Trusted>false</Trusted>
            <SelfSigned>false</SelfSigned>
            <CertificatePolicies>
                <certificatePolicy
cpsUrl="http://repository.eid.belgium.be">2.16.56.12.1.1.2.1</certificatePolicy>
            </CertificatePolicies>
            <QCStatementIds>
                <qcStatementOid Description="qc-
compliance">0.4.0.1862.1.1</qcStatementOid>
                <qcStatementOid Description="qc-sscd">0.4.0.1862.1.4</qcStatementOid>
            </QCStatementIds>
            <QCTypes/>
            <TrustedServiceProviders>
                <TrustedServiceProvider TL="TL-
61C0487109BE27255C19CFF26D8F56BEA621E7F381A7B4CBE7FB4750BD477BF9" LOTL="LOTL-
EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B">
                    <TSPNames>
                        <TSPName lang="en">Certipost n.v./s.a.</TSPName>
                    </TSPNames>
                    <TSPTradeNames>
                        <TSPTradeName lang="en">Certipost s.a./n.v.</TSPTradeName>
                    </TSPTradeNames>
                    <TSPRegistrationIdentifiers>
                        <TSPRegistrationIdentifier>VATBE-
0475396406</TSPRegistrationIdentifier>
                    </TSPRegistrationIdentifiers>
                    <TrustedServices>
                        <TrustedService ServiceDigitalIdentifier="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                            <ServiceNames>
                                <ServiceName lang="en">CN=Belgium Root CA4,
C=BE</ServiceName>
```

```
                            </ServiceNames>

<ServiceType>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</ServiceType>

<Status>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted</Status>
                            <StartDate>2016-06-30T22:00:00</StartDate>
                            <CapturedQualifiers>

<Qualifier>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/QCQSCDStatusAsInCert</Qu
alifier>
                            </CapturedQualifiers>
                            <AdditionalServiceInfoUris>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/RootCA-
QC</AdditionalServiceInfoUri>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSigna
tures</AdditionalServiceInfoUri>
                            </AdditionalServiceInfoUris>
                        </TrustedService>
                    </TrustedServices>
                </TrustedServiceProvider>
            </TrustedServiceProviders>
            <Revocations>
                <CertificateRevocation Revocation="R-
83B6DB1D1C8369D90161C212DEBCE29E2608D9B117356B8AA9E531721CB1F5BF">
                    <Status>GOOD</Status>
                </CertificateRevocation>
            </Revocations>
            <DigestAlgoAndValue>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>T6spAncn5Y5FGO0LauVU0FXwWz2Rl+DRayACjSJ9Gp8=</DigestValue>
            </DigestAlgoAndValue>
        </Certificate>
        <Certificate Id="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
            <SubjectDistinguishedName Format="CANONICAL">cn=belgium root
ca4,c=be</SubjectDistinguishedName>
            <SubjectDistinguishedName Format="RFC2253">CN=Belgium Root
CA4,C=BE</SubjectDistinguishedName>
            <IssuerDistinguishedName Format="CANONICAL">cn=belgium root
ca4,c=be</IssuerDistinguishedName>
            <IssuerDistinguishedName Format="RFC2253">CN=Belgium Root
CA4,C=BE</IssuerDistinguishedName>
            <SerialNumber>5706940941790920504</SerialNumber>
            <CommonName>Belgium Root CA4</CommonName>
            <CountryName>BE</CountryName>
            <AuthorityInformationAccessUrls/>
            <CRLDistributionPoints/>
            <OCSPAccessUrls/>
```

```
            <Sources>
                <Source>SIGNATURE</Source>
                <Source>OCSP_RESPONSE</Source>
                <Source>TRUSTED_LIST</Source>
            </Sources>
            <NotAfter>2032-10-22T12:00:00</NotAfter>
            <NotBefore>2013-06-26T12:00:00</NotBefore>
            <PublicKeySize>4096</PublicKeySize>
            <PublicKeyEncryptionAlgo>RSA</PublicKeyEncryptionAlgo>
            <EntityKey>PK-
B1538A9071705A92D938E506B738DA0CAFD1AE66AEC2EEDDC75479D433BD8776</EntityKey>
            <KeyUsageBits>
                <KeyUsage>keyCertSign</KeyUsage>
                <KeyUsage>crlSign</KeyUsage>
            </KeyUsageBits>
            <ExtendedKeyUsages/>
            <IdPkixOcspNoCheck>false</IdPkixOcspNoCheck>
            <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
                <KeyLengthUsedToSignThisToken>4096</KeyLengthUsedToSignThisToken>
                <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
                <SignatureIntact>true</SignatureIntact>
                <SignatureValid>true</SignatureValid>
            </BasicSignature>
            <CertificateChain/>
            <Trusted>true</Trusted>
            <SelfSigned>true</SelfSigned>
            <CertificatePolicies>
                <certificatePolicy
cpsUrl="http://repository.eid.belgium.be">2.16.56.12.1.1</certificatePolicy>
            </CertificatePolicies>
            <QCStatementIds/>
            <QCTypes/>
            <TrustedServiceProviders>
                <TrustedServiceProvider TL="TL-
61C0487109BE27255C19CFF26D8F56BEA621E7F381A7B4CBE7FB4750BD477BF9" LOTL="LOTL-
EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B">
                    <TSPNames>
                        <TSPName lang="en">Certipost n.v./s.a.</TSPName>
                    </TSPNames>
                    <TSPTradeNames>
                        <TSPTradeName lang="en">Certipost s.a./n.v.</TSPTradeName>
                    </TSPTradeNames>
                    <TSPRegistrationIdentifiers>
                        <TSPRegistrationIdentifier>VATBE-
0475396406</TSPRegistrationIdentifier>
                    </TSPRegistrationIdentifiers>
                    <TrustedServices>
                        <TrustedService ServiceDigitalIdentifier="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
```

```
                            <ServiceNames>
                                <ServiceName lang="en">CN=Belgium Root CA4,
C=BE</ServiceName>
                            </ServiceNames>

<ServiceType>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</ServiceType>

<Status>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted</Status>
                            <StartDate>2016-06-30T22:00:00</StartDate>
                            <AdditionalServiceInfoUris>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/RootCA-
QC</AdditionalServiceInfoUri>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSigna
tures</AdditionalServiceInfoUri>
                            </AdditionalServiceInfoUris>
                        </TrustedService>
                        <TrustedService ServiceDigitalIdentifier="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                            <ServiceNames>
                                <ServiceName lang="en">CN=Belgium Root CA4,
C=BE</ServiceName>
                            </ServiceNames>

<ServiceType>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</ServiceType>

<Status>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/undersupervision</Status>
                            <StartDate>2013-06-26T12:00:00</StartDate>
                            <EndDate>2016-06-30T22:00:00</EndDate>
                            <AdditionalServiceInfoUris>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/RootCA-
QC</AdditionalServiceInfoUri>
                            </AdditionalServiceInfoUris>
                        </TrustedService>
                    </TrustedServices>
                </TrustedServiceProvider>
            </TrustedServiceProviders>
            <DigestAlgoAndValue>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>cC3VwaCTzwqdcfrdm/mnxYV9iftztxboZyKLPCvrlo8=</DigestValue>
            </DigestAlgoAndValue>
        </Certificate>
        <Certificate Id="C-
B336CAA1F3C4930E4EF9C803C12877A004991EE9206C0D4AD3891688C1E478FF">
            <SubjectDistinguishedName Format="CANONICAL">c=be,cn=belgium ocsp
responder</SubjectDistinguishedName>
            <SubjectDistinguishedName Format="RFC2253">C=BE,CN=Belgium OCSP
Responder</SubjectDistinguishedName>
```

```
            <IssuerDistinguishedName
Format="CANONICAL">2.5.4.5=#1306323031363331,cn=citizen
ca,c=be</IssuerDistinguishedName>
            <IssuerDistinguishedName
Format="RFC2253">2.5.4.5=#1306323031363331,CN=Citizen
CA,C=BE</IssuerDistinguishedName>
            <SerialNumber>48357032784600921550098 66</SerialNumber>
            <CommonName>Belgium OCSP Responder</CommonName>
            <CountryName>BE</CountryName>
            <AuthorityInformationAccessUrls/>
            <CRLDistributionPoints/>
            <OCSPAccessUrls/>
            <Sources>
                <Source>OCSP_RESPONSE</Source>
            </Sources>
            <NotAfter>2021-01-29T11:00:00</NotAfter>
            <NotBefore>2019-12-10T11:00:00</NotBefore>
            <PublicKeySize>2048</PublicKeySize>
            <PublicKeyEncryptionAlgo>RSA</PublicKeyEncryptionAlgo>
            <EntityKey>PK-
FB98E97110AA655D971E0788C666FF32A45606153C8977C738F2005ADED0C59B</EntityKey>
            <KeyUsageBits>
                <KeyUsage>digitalSignature</KeyUsage>
            </KeyUsageBits>
            <ExtendedKeyUsages>
                <extendedKeyUsagesOid
Description="ocspSigning">1.3.6.1.5.5.7.3.9</extendedKeyUsagesOid>
            </ExtendedKeyUsages>
            <IdPkixOcspNoCheck>true</IdPkixOcspNoCheck>
            <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
                <KeyLengthUsedToSignThisToken>4096</KeyLengthUsedToSignThisToken>
                <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
                <SignatureIntact>true</SignatureIntact>
                <SignatureValid>true</SignatureValid>
            </BasicSignature>
            <SigningCertificate Certificate="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA"/>
            <CertificateChain>
                <ChainItem Certificate="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA"/>
                <ChainItem Certificate="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            </CertificateChain>
            <Trusted>false</Trusted>
            <SelfSigned>false</SelfSigned>
            <CertificatePolicies/>
            <QCStatementIds/>
            <QCTypes/>
            <TrustedServiceProviders>
```

```
                <TrustedServiceProvider TL="TL-
61C0487109BE27255C19CFF26D8F56BEA621E7F381A7B4CBE7FB4750BD477BF9" LOTL="LOTL-
EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B">
                    <TSPNames>
                        <TSPName lang="en">Certipost n.v./s.a.</TSPName>
                    </TSPNames>
                    <TSPTradeNames>
                        <TSPTradeName lang="en">Certipost s.a./n.v.</TSPTradeName>
                    </TSPTradeNames>
                    <TSPRegistrationIdentifiers>
                        <TSPRegistrationIdentifier>VATBE-
0475396406</TSPRegistrationIdentifier>
                    </TSPRegistrationIdentifiers>
                    <TrustedServices>
                        <TrustedService ServiceDigitalIdentifier="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                            <ServiceNames>
                                <ServiceName lang="en">CN=Belgium Root CA4,
C=BE</ServiceName>
                            </ServiceNames>

<ServiceType>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</ServiceType>

<Status>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted</Status>
                            <StartDate>2016-06-30T22:00:00</StartDate>
                            <AdditionalServiceInfoUris>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/RootCA-
QC</AdditionalServiceInfoUri>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSigna
tures</AdditionalServiceInfoUri>
                            </AdditionalServiceInfoUris>
                        </TrustedService>
                    </TrustedServices>
                </TrustedServiceProvider>
            </TrustedServiceProviders>
            <DigestAlgoAndValue>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>szbKofPEkw5O+cgDwSh3oASZHukgbA1K04kWiMHkeP8=</DigestValue>
            </DigestAlgoAndValue>
        </Certificate>
        <Certificate Id="C-
C3FBF37259AF0954EEEA4282DD1C7226A54E7150F7C29A2C495BA34DBFE09CA0">
            <SubjectDistinguishedName Format="CANONICAL">cn=belgium root
ca4,c=be</SubjectDistinguishedName>
            <SubjectDistinguishedName Format="RFC2253">CN=Belgium Root
CA4,C=BE</SubjectDistinguishedName>
            <IssuerDistinguishedName Format="CANONICAL">cn=belgium root
ca4,c=be</IssuerDistinguishedName>
```

```
            <IssuerDistinguishedName Format="RFC2253">CN=Belgium Root
CA4,C=BE</IssuerDistinguishedName>
            <SerialNumber>5706940941790920504</SerialNumber>
            <CommonName>Belgium Root CA4</CommonName>
            <CountryName>BE</CountryName>
            <AuthorityInformationAccessUrls/>
            <CRLDistributionPoints/>
            <OCSPAccessUrls/>
            <Sources>
                <Source>SIGNATURE</Source>
                <Source>OCSP_RESPONSE</Source>
                <Source>TRUSTED_LIST</Source>
            </Sources>
            <NotAfter>2028-01-28T12:00:00</NotAfter>
            <NotBefore>2013-06-26T12:00:00</NotBefore>
            <PublicKeySize>4096</PublicKeySize>
            <PublicKeyEncryptionAlgo>RSA</PublicKeyEncryptionAlgo>
            <EntityKey>PK-
B1538A9071705A92D938E506B738DA0CAFD1AE66AEC2EEDDC75479D433BD8776</EntityKey>
            <KeyUsageBits>
                <KeyUsage>keyCertSign</KeyUsage>
                <KeyUsage>crlSign</KeyUsage>
            </KeyUsageBits>
            <ExtendedKeyUsages/>
            <IdPkixOcspNoCheck>false</IdPkixOcspNoCheck>
            <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
                <KeyLengthUsedToSignThisToken>4096</KeyLengthUsedToSignThisToken>
                <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
                <SignatureIntact>true</SignatureIntact>
                <SignatureValid>true</SignatureValid>
            </BasicSignature>
            <CertificateChain/>
            <Trusted>true</Trusted>
            <SelfSigned>true</SelfSigned>
            <CertificatePolicies>
                <certificatePolicy
cpsUrl="http://repository.eid.belgium.be">2.16.56.12.1.1</certificatePolicy>
            </CertificatePolicies>
            <QCStatementIds/>
            <QCTypes/>
            <TrustedServiceProviders>
                <TrustedServiceProvider TL="TL-
61C0487109BE27255C19CFF26D8F56BEA621E7F381A7B4CBE7FB4750BD477BF9" LOTL="LOTL-
EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B">
                    <TSPNames>
                        <TSPName lang="en">Certipost n.v./s.a.</TSPName>
                    </TSPNames>
                    <TSPTradeNames>
                        <TSPTradeName lang="en">Certipost s.a./n.v.</TSPTradeName>
```

```
                    </TSPTradeNames>
                    <TSPRegistrationIdentifiers>
                        <TSPRegistrationIdentifier>VATBE-
0475396406</TSPRegistrationIdentifier>
                    </TSPRegistrationIdentifiers>
                    <TrustedServices>
                        <TrustedService ServiceDigitalIdentifier="C-
C3FBF37259AF0954EEEA4282DD1C7226A54E7150F7C29A2C495BA34DBFE09CA0">
                            <ServiceNames>
                                <ServiceName lang="en">CN=Belgium Root CA4,
C=BE</ServiceName>
                            </ServiceNames>

<ServiceType>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</ServiceType>

<Status>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted</Status>
                            <StartDate>2016-06-30T22:00:00</StartDate>
                            <AdditionalServiceInfoUris>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/RootCA-
QC</AdditionalServiceInfoUri>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/ForeSigna
tures</AdditionalServiceInfoUri>
                            </AdditionalServiceInfoUris>
                        </TrustedService>
                        <TrustedService ServiceDigitalIdentifier="C-
C3FBF37259AF0954EEEA4282DD1C7226A54E7150F7C29A2C495BA34DBFE09CA0">
                            <ServiceNames>
                                <ServiceName lang="en">CN=Belgium Root CA4,
C=BE</ServiceName>
                            </ServiceNames>

<ServiceType>http://uri.etsi.org/TrstSvc/Svctype/CA/QC</ServiceType>

<Status>http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/undersupervision</Status>
                            <StartDate>2013-06-26T12:00:00</StartDate>
                            <EndDate>2016-06-30T22:00:00</EndDate>
                            <AdditionalServiceInfoUris>

<AdditionalServiceInfoUri>http://uri.etsi.org/TrstSvc/TrustedList/SvcInfoExt/RootCA-
QC</AdditionalServiceInfoUri>
                            </AdditionalServiceInfoUris>
                        </TrustedService>
                    </TrustedServices>
                </TrustedServiceProvider>
            </TrustedServiceProviders>
            <DigestAlgoAndValue>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>w/vzclmvCVTu6kKC3RxyJqVOcVD3wposSVujTb/gnKA=</DigestValue>
```

```
            </DigestAlgoAndValue>
        </Certificate>
    </UsedCertificates>
    <UsedRevocations>
        <Revocation Id="R-
46B3B0FDA4A56A2601A20B31701AF2B1D0DCAEA8A903CD1346FBA80F49DFF1BD">
            <Origin>EXTERNAL</Origin>
            <Type>CRL</Type>
            <SourceAddress>http://crl.eid.belgium.be/belgium4.crl</SourceAddress>
            <ProductionDate>2020-07-01T11:00:00</ProductionDate>
            <ThisUpdate>2020-07-01T11:00:00</ThisUpdate>
            <NextUpdate>2021-01-31T11:00:00</NextUpdate>
            <CertHashExtensionPresent>false</CertHashExtensionPresent>
            <CertHashExtensionMatch>false</CertHashExtensionMatch>
            <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
                <KeyLengthUsedToSignThisToken>4096</KeyLengthUsedToSignThisToken>
                <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
                <SignatureIntact>true</SignatureIntact>
                <SignatureValid>true</SignatureValid>
            </BasicSignature>
            <SigningCertificate Certificate="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            <CertificateChain>
                <ChainItem Certificate="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            </CertificateChain>
            <DigestAlgoAndValue>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>RrOw/aSlaiYBogsxcBrysdDcrqipA80TRvuoD0nf8b0=</DigestValue>
            </DigestAlgoAndValue>
        </Revocation>
        <Revocation Id="R-
83B6DB1D1C8369D90161C212DEBCE29E2608D9B117356B8AA9E531721CB1F5BF">
            <Origin>EXTERNAL</Origin>
            <Type>OCSP</Type>
            <SourceAddress>http://ocsp.eid.belgium.be/2</SourceAddress>
            <ProductionDate>2020-12-14T11:03:21</ProductionDate>
            <ThisUpdate>2020-12-14T11:03:21</ThisUpdate>
            <NextUpdate>2020-12-14T11:04:21</NextUpdate>
            <CertHashExtensionPresent>false</CertHashExtensionPresent>
            <CertHashExtensionMatch>false</CertHashExtensionMatch>
            <BasicSignature>

<EncryptionAlgoUsedToSignThisToken>RSA</EncryptionAlgoUsedToSignThisToken>
                <KeyLengthUsedToSignThisToken>2048</KeyLengthUsedToSignThisToken>
                <DigestAlgoUsedToSignThisToken>SHA256</DigestAlgoUsedToSignThisToken>
                <SignatureIntact>true</SignatureIntact>
                <SignatureValid>true</SignatureValid>
```

```
            </BasicSignature>
            <SigningCertificate Certificate="C-
B336CAA1F3C4930E4EF9C803C12877A004991EE9206C0D4AD3891688C1E478FF"/>
            <CertificateChain>
                <ChainItem Certificate="C-
B336CAA1F3C4930E4EF9C803C12877A004991EE9206C0D4AD3891688C1E478FF"/>
                <ChainItem Certificate="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA"/>
                <ChainItem Certificate="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F"/>
            </CertificateChain>
            <FoundCertificates>
                <RelatedCertificate Certificate="C-
702DD5C1A093CF0A9D71FADD9BF9A7C5857D89FB73B716E867228B3C2BEB968F">
                    <Origin>BASIC_OCSP_RESP</Origin>
                </RelatedCertificate>
                <RelatedCertificate Certificate="C-
B336CAA1F3C4930E4EF9C803C12877A004991EE9206C0D4AD3891688C1E478FF">
                    <Origin>BASIC_OCSP_RESP</Origin>
                    <CertificateRef>
                        <Origin>SIGNING_CERTIFICATE</Origin>
                        <SerialInfo>
                            <IssuerName>C=BE, CN=Belgium OCSP Responder</IssuerName>
                        </SerialInfo>
                    </CertificateRef>
                </RelatedCertificate>
                <RelatedCertificate Certificate="C-
293D0BA3A31E5D82A8E3FAE12709932FFDDA44423E0F733FB01EF123E73EB4DA">
                    <Origin>BASIC_OCSP_RESP</Origin>
                </RelatedCertificate>
            </FoundCertificates>
            <DigestAlgoAndValue>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>g7bbHRyDadkBYcIS3rziniYI2bEXNWuKqeUxchyx9b8=</DigestValue>
            </DigestAlgoAndValue>
        </Revocation>
    </UsedRevocations>
    <UsedTimestamps/>
    <OriginalDocuments>
        <SignerData Id="D-
BD598965790CA5673910D64EFEDA798485364C66B6F4E7C49D23E1FF8BAFBBE8">
            <ReferencedName>Full PDF</ReferencedName>
            <DigestAlgoAndValue>
                <DigestMethod>SHA256</DigestMethod>

<DigestValue>XC6PrKORnL59qX5UJCibBUw/kPLQNtqQx81H+vdlPXw=</DigestValue>
            </DigestAlgoAndValue>
        </SignerData>
    </OriginalDocuments>
    <TrustedLists>
```

```
        <TrustedList Id="LOTL-
EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B" LOTL="true">
            <CountryCode>EU</CountryCode>
            <Url>https://ec.europa.eu/tools/lotl/eu-lotl.xml</Url>
            <SequenceNumber>277</SequenceNumber>
            <Version>5</Version>
            <LastLoading>2020-12-14T11:02:18</LastLoading>
            <IssueDate>2020-11-24T14:00:00</IssueDate>
            <NextUpdate>2021-05-24T00:00:00</NextUpdate>
            <WellSigned>true</WellSigned>
        </TrustedList>
        <TrustedList Id="TL-
61C0487109BE27255C19CFF26D8F56BEA621E7F381A7B4CBE7FB4750BD477BF9">
            <CountryCode>BE</CountryCode>
            <Url>https://tsl.belgium.be/tsl-be.xml</Url>
            <SequenceNumber>49</SequenceNumber>
            <Version>5</Version>
            <LastLoading>2020-12-14T11:02:18</LastLoading>
            <IssueDate>2020-10-20T00:00:00</IssueDate>
            <NextUpdate>2021-04-15T00:00:00</NextUpdate>
            <WellSigned>true</WellSigned>
        </TrustedList>
    </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)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<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/rOAKieZCe4LZPa0/uwzgwM4QAb
Pstdy4gHSQzCF0R6/fT9hv639kQS3TyZedw1raMeDj9mQOwKOlMl1OIxEI7jSf7xP6n62s0wQAhTJlARnOY1G5
vppziVKb1vPED27HPBB4Yljn8j6hse+EJ0bwxAN1gwufbxZBvjHYgz/U/9EHafa1oGPcoIBrXvoUdzVX76sVE3
nlDvX4psEU4eq7paIZA7AWGSfWk8/k98pPqFcP2VYJaAju9GI+uZNMfRgPdOvGPxTjUBYiEyr3satod+cMQGiA
zie80n0ovQrfn7ebcA==</ns2:SignatureValue>
            <HashOnly>false</HashOnly>
            <DocHashOnly>false</DocHashOnly>
        </SignatureIdentifier>
        <ValidationConstraintsEvaluationReport>
            <ValidationConstraint>

<ValidationConstraintIdentifier>urn:cef:dss:bbb:formatChecking</ValidationConstraintId
entifier>
                <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</Valid
ationConstraintIdentifier>
                <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</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>
            <ValidationConstraint>

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

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

<ValidationConstraintIdentifier>urn:cef:dss:bbb:validationTimeSliding</ValidationConst
raintIdentifier>
```

```
                <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>XC6PrKORnL59qX5UJCibBUw/kPLQNtqQx81H+vdlPXw=</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>SGEPVFOj/zskv8+nlzixt+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>
                    <ValidationObjectId 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>
                <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>T6spAncn5Y5FGO0LauVU0FXwWz2Rl+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>szbKofPEkw5O+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/vzclmvCVTu6kKC3RxyJqVOcVD3wposSVujTb/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>RrOw/aSlaiYBogsxcBrysdDcrqipA80TRvuoD0nf8b0=</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-
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-
BD598965790CA5673910D64EFEDA798485364C66B6F4E7C49D23E1FF8BAFBBE8">
            <ObjectType>urn:etsi:019102:validationObject:signedData</ObjectType>
            <ValidationObjectRepresentation>
                <DigestAlgAndValue>
                    <ns2:DigestMethod
Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<ns2:DigestValue>XC6PrKORnL59qX5UJCibBUw/kPLQNtqQx81H+vdlPXw=</ns2:DigestValue>
                </DigestAlgAndValue>
            </ValidationObjectRepresentation>
```

```
            <POE>
                <POETime>2020-12-14T11:03:21Z</POETime>
                <TypeOfProof>urn:etsi:019102:poetype:validation</TypeOfProof>
            </POE>
        </ValidationObject>
    </SignatureValidationObjects>
</ValidationReport>
```

Customised Validation Policy

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

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" />
            <PdfVisualDifference Level="WARN" />
            <SigningCertificate>
                <Recognition Level="FAIL" />
                <Signature Level="FAIL" />
                <NotExpired Level="FAIL" />
                <AuthorityInfoAccessPresent Level="WARN" />
                <RevocationInfoAccessPresent Level="WARN" />
                <RevocationDataAvailable Level="FAIL" />
                <RevocationDataNextUpdatePresent 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" />
                <RevocationDataNextUpdatePresent Level="WARN" />
                <RevocationDataFreshness Level="FAIL" />
                <NotRevoked Level="FAIL" />
                <NotOnHold Level="FAIL" />
                <Cryptographic />
            </CACertificate>
            <Cryptographic />
```

```
        </BasicSignatureConstraints>
        <SignedAttributes>
            <SigningCertificatePresent Level="FAIL" />
            <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" />
            <SigningCertificate>
                <Recognition Level="FAIL" />
                <Signature Level="FAIL" />
                <NotExpired Level="FAIL" />
                <AuthorityInfoAccessPresent Level="WARN" />
                <RevocationInfoAccessPresent Level="WARN" />
                <RevocationDataAvailable Level="FAIL" />
                <RevocationDataNextUpdatePresent Level="WARN" />
                <RevocationDataFreshness Level="FAIL" />
                <KeyUsage Level="WARN">
                    <Id>nonRepudiation</Id>
                </KeyUsage>
                <SerialNumberPresent Level="WARN" />
                <NotRevoked Level="FAIL" />
                <NotOnHold Level="FAIL" />
                <NotSelfSigned Level="WARN" />
                <UsePseudonym Level="INFORM" />
                <Cryptographic />
            </SigningCertificate>
```

```
            <CACertificate>
                <Signature Level="FAIL" />
                <NotExpired Level="FAIL" />
                <RevocationDataAvailable Level="FAIL" />
                <RevocationDataNextUpdatePresent Level="WARN" />
                <RevocationDataFreshness Level="FAIL" />
                <NotRevoked Level="FAIL" />
                <NotOnHold Level="FAIL" />
                <Cryptographic />
            </CACertificate>
            <Cryptographic />
        </BasicSignatureConstraints>
        <SignedAttributes>
            <SigningCertificatePresent Level="FAIL" />
            <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" />
                <RevocationDataNextUpdatePresent 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" />
                <RevocationDataNextUpdatePresent Level="WARN" />
                <RevocationDataFreshness Level="FAIL" />
                <NotRevoked Level="FAIL" />
                <NotOnHold Level="FAIL" />
                <Cryptographic />
            </CACertificate>
            <Cryptographic />
        </BasicSignatureConstraints>
        <SignedAttributes>
            <SigningCertificatePresent Level="WARN" />
            <!-- <UnicitySigningCertificate Level="WARN" /> RFC 5816 -->
            <CertDigestPresent Level="WARN" />
            <AllCertDigestsMatch Level="WARN" />
            <IssuerSerialMatch Level="WARN" />
        </SignedAttributes>
    </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" />
                <RevocationDataNextUpdatePresent Level="WARN" />
                <RevocationDataFreshness Level="FAIL" />
                <NotRevoked Level="FAIL" />
                <NotOnHold Level="FAIL" />
                <Cryptographic />
            </SigningCertificate>
            <CACertificate>
                <Signature Level="FAIL" />
                <NotExpired Level="FAIL" />
                <RevocationDataAvailable Level="WARN" />
                <RevocationDataNextUpdatePresent 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="160">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="160">DSA</Algo> <!-- ETSI TS 102 176-1
(Historical) V2.1.1 -->
            <Algo Date="2013" Size="192">DSA</Algo> <!-- ETSI TS 102 176-1
(Historical) V2.1.1 -->
            <Algo Date="2023" Size="224">DSA</Algo> <!-- ETSI 119 312 V1.3.1 -->
            <Algo Date="2026" Size="256">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:

Signing 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:

Signing 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.

Defining a Signature Policy

```
// 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',
'l', '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, 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.CEN
TER);

// 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&#176; clockwise. Coordinates'
origin begins from top right page corner);
   _ROTATE_180_ (Rotates a signature field for a 180&#176; clockwise. Coordinates'
origin begins from the bottom right page corner);
   _ROTATE_270_ (Rotates a signature field for a 270&#176; 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);

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.CEN
TER);
```

```
// 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&#176; clockwise. Coordinates'
origin begins from top right page corner);
   _ROTATE_180_ (Rotates a signature field for a 180&#176; clockwise. Coordinates'
origin begins from the bottom right page corner);
   _ROTATE_270_ (Rotates a signature field for a 270&#176; 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);

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);
// Set textParameters to a SignatureImageParameters object
imageParameters.setTextParameters(textParameters);
```

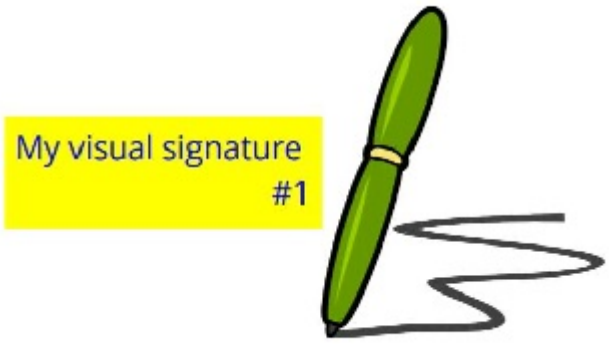
Text and image combination

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 Customised 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.

The implementation is based on a draft of the standard ETSI TS 119 182-1. Some modifications can occur in future releases.

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

Signing a file with JAdES

```
// 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(JWSSerializationType.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:

Sign multiple files within an ASiC-E container

```
// 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 pr 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 ObjectIdentier 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.

XAdES with explicit 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();

// 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 Signature Policy element

```
<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:

Add SignaturePolicyStore

```
// 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 (`ASiCEWithCAdESTimestampValidator`).

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

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<DetailedReport xmlns="http://dss.esig.europa.eu/validation/detailed-report">
    <Timestamp Id="T-
1F9F5B92B9DF5678CD57A7B47FEEADDE365017BD67E5E7468A5ADF14F6CC901F">
        <ValidationProcessTimestamp Type="CONTENT_TIMESTAMP" ProductionTime="2018-05-
23T13:18:29" Title="Validation Process for Timestamps">
            <Constraint Id="T-
1F9F5B92B9DF5678CD57A7B47FEEADDE365017BD67E5E7468A5ADF14F6CC901F">
                <Name NameId="ADEST_ROTVPIIC">Is the result of the timestamps
validation process conclusive?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
        </ValidationProcessTimestamp>
        <ValidationTimestampQualification TimestampQualification="QTSA"
Title="Timestamp Qualification">
            <Constraint>
                <Name NameId="QUAL_CERT_TRUSTED_LIST_REACHED">Has a trusted list been
reached for the certificate chain?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint Id="LOTL-
EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B">
                <Name NameId="QUAL_TRUSTED_LIST_ACCEPT">Is the trusted list
acceptable?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Trusted List : https://ec.europa.eu/tools/lotl/eu-
lotl.xml</AdditionalInfo>
            </Constraint>
            <Constraint Id="TL-
B66584674C05B72F6BB09FF13C632EB1BA72A0DD0DCF9A6822EF73374F6DFC48">
                <Name NameId="QUAL_TRUSTED_LIST_ACCEPT">Is the trusted list
acceptable?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Trusted List : https://www.nrca-ds.de/st/TSL-
XML.xml</AdditionalInfo>
            </Constraint>
            <Constraint>
```

```
                <Name NameId="QUAL_HAS_QTST">Is the certificate related to a
TSA/QTST?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="QUAL_HAS_GRANTED">Is the certificate related to a trust
service with a granted status?</Name>
                <Status>OK</Status>
            </Constraint>
            <Constraint>
                <Name NameId="QUAL_HAS_GRANTED_AT">Is the certificate related to a
trust service with a granted status at the production time?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
        </ValidationTimestampQualification>
    </Timestamp>
    <BasicBuildingBlocks Id="T-
1F9F5B92B9DF5678CD57A7B47FEEADDE365017BD67E5E7468A5ADF14F6CC901F" Type="TIMESTAMP">
        <ISC Title="Identification of the Signing Certificate">
            <Constraint>
                <Name NameId="BBB_ICS_ISCI">Is there an identified candidate for the
signing certificate?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
            <CertificateChain>
                <ChainItem Id="C-
9ACF4D4E37AD647B1310D55800234BBF17FFFE01CAF7E63ED7223B770CB0D2E9">
                    <Source>TRUSTED_LIST</Source>
                </ChainItem>
                <ChainItem Id="C-
5F522A439611DEC501F0468FF6A455D23DFBD08EEB33964A0C5EDE95593E1FA8">
                    <Source>TIMESTAMP</Source>
                </ChainItem>
                <ChainItem Id="C-
4D24807B9CAD5110F40ED79D934346D7C9B0290431DC9B11A40BBB86FCF2AEF6">
                    <Source>TIMESTAMP</Source>
                </ChainItem>
            </CertificateChain>
        </ISC>
        <XCV Title="X509 Certificate Validation">
            <Constraint>
                <Name NameId="BBB_XCV_CCCBB">Can the certificate chain be built till a
trust anchor?</Name>
                <Status>OK</Status>
            </Constraint>
```

```
            <Constraint Id="C-
9ACF4D4E37AD647B1310D55800234BBF17FFFE01CAF7E63ED7223B770CB0D2E9">
                <Name NameId="BBB_XCV_SUB">Is the certificate validation
conclusive?</Name>
                <Status>OK</Status>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
            <SubXCV Id="C-
9ACF4D4E37AD647B1310D55800234BBF17FFFE01CAF7E63ED7223B770CB0D2E9" TrustAnchor="true"
Title="Certificate">
                <Conclusion>
                    <Indication>PASSED</Indication>
                </Conclusion>
            </SubXCV>
        </XCV>
        <CV Title="Cryptographic Verification">
            <Constraint>
                <Name NameId="BBB_CV_IRDOF">Has the reference data object been
found?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Reference : MESSAGE_IMPRINT</AdditionalInfo>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_CV_IRDOI">Is the reference data object
intact?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Reference : MESSAGE_IMPRINT</AdditionalInfo>
            </Constraint>
            <Constraint>
                <Name NameId="BBB_CV_ISIT">Is timestampss signature intact?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Id = T-
1F9F5B92B9DF5678CD57A7B47FEEADDE365017BD67E5E7468A5ADF14F6CC901F</AdditionalInfo>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
            </Conclusion>
        </CV>
        <SAV ValidationTime="2020-01-23T13:09:12" Title="Signature Acceptance
Validation">
            <Constraint>
                <Name NameId="ATCCM">Are timestamp cryptographic constraints
met?</Name>
                <Status>OK</Status>
                <AdditionalInfo>Validation time : 2020-01-23 13:09 for token with ID :
[T-1F9F5B92B9DF5678CD57A7B47FEEADDE365017BD67E5E7468A5ADF14F6CC901F]</AdditionalInfo>
            </Constraint>
            <Conclusion>
                <Indication>PASSED</Indication>
```

```
            </Conclusion>
            <CryptographicInfo>
                <Algorithm>http://www.w3.org/2007/05/xmldsig-more#sha256-rsa-
MGF1</Algorithm>
                <KeyLength>2048</KeyLength>
                <Secure>true</Secure>
                <NotAfter>2022-12-31T23:00:00</NotAfter>
            </CryptographicInfo>
        </SAV>
        <CertificateChain>
            <ChainItem Id="C-
9ACF4D4E37AD647B1310D55800234BBF17FFFE01CAF7E63ED7223B770CB0D2E9">
                <Source>TRUSTED_LIST</Source>
            </ChainItem>
            <ChainItem Id="C-
5F522A439611DEC501F0468FF6A455D23DFBD08EEB33964A0C5EDE95593E1FA8">
                <Source>TIMESTAMP</Source>
            </ChainItem>
            <ChainItem Id="C-
4D24807B9CAD5110F40ED79D934346D7C9B0290431DC9B11A40BBB86FCF2AEF6">
                <Source>TIMESTAMP</Source>
            </ChainItem>
        </CertificateChain>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
    </BasicBuildingBlocks>
    <TLAnalysis CountryCode="EU" URL="https://ec.europa.eu/tools/lotl/eu-lotl.xml"
Id="LOTL-EC2AE37FE9A43B48B1CFE2A57EBEE2BD6373EDFF36537EB1BC905747ACBF4C3B" Title="List
Of Trusted Lists EU">
        <Constraint>
            <Name NameId="QUAL_TL_FRESH">Is the trusted list fresh?</Name>
            <Status>WARNING</Status>
            <Warning NameId="QUAL_TL_FRESH_ANS">The trusted list is not considered as
fresh!</Warning>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_EXP">Is the trusted list not expired?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_VERSION">Does the trusted list have the expected
version?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_WS">Is the trusted list well signed?</Name>
            <Status>OK</Status>
        </Constraint>
        <Conclusion>
            <Indication>PASSED</Indication>
```

```
            <Warnings NameId="QUAL_TL_FRESH_ANS">The trusted list is not considered as
fresh!</Warnings>
        </Conclusion>
    </TLAnalysis>
    <TLAnalysis CountryCode="DE" URL="https://www.nrca-ds.de/st/TSL-XML.xml" Id="TL-
B66584674C05B72F6BB09FF13C632EB1BA72A0DD0DCF9A6822EF73374F6DFC48" Title="Trusted List
DE">
        <Constraint>
            <Name NameId="QUAL_TL_FRESH">Is the trusted list fresh?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_EXP">Is the trusted list not expired?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_VERSION">Does the trusted list have the expected
version?</Name>
            <Status>OK</Status>
        </Constraint>
        <Constraint>
            <Name NameId="QUAL_TL_WS">Is the trusted list well signed?</Name>
            <Status>OK</Status>
        </Constraint>
        <Conclusion>
            <Indication>PASSED</Indication>
        </Conclusion>
    </TLAnalysis>
</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 `OCSPAndCRLRevocationSource` class whose role is to fetch revocation data by querying an OCSP server first and then a CRL server if no OCSP response could be retrieved. 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();

// Set dataSource for the repository
cacheCRLSource.setDataSource(dataSource);

// 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();
cacheOCSPSource.setDataSource(dataSource);
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 :
    - `CMSCRLSource` : Extracts CRLs and CRL references from a CMS Signed Data :
      - `CAdESCRLSource` : Sub-class of `CMSCRLSource` for a CAdES Signature;
      - `TimestampCRLSource`: Sub-class of `CMSCRLSource` 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);
    - `PAdESOCSPSource` : Extracts OCSP responses and OCSP references from a PAdES signature.
    - `XAdESOCSPSource` : 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());

// Aets 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();

        // This data loader is used to collect certificates from external resources
        // (AIA)
        cv.setDataLoader(dataLoader);

        // 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)

        // DSS throws an exception by default in case of missing revocation data
        // Default : ExceptionOnStatusAlert
        cv.setAlertOnMissingRevocationData(new ExceptionOnStatusAlert());

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

        // DSS interrupts by default the extension process if a revoked certificate is
        // present
        // Default : ExceptionOnStatusAlert
        cv.setAlertOnRevokedCertificate(new ExceptionOnStatusAlert());

        // DSS stops the extension process if an invalid timestamp is met
        // Default : ExceptionOnStatusAlert
        cv.setAlertOnInvalidTimestamp(new ExceptionOnStatusAlert());

        // DSS v5.5+ : logs a warning message 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
        cv.setAlertOnNoRevocationAfterBestSignatureTime(new
LogOnStatusAlert(Level.ERROR));

        // 5.7 : The below methods have been moved to DocumentValidator /
        // CertificateValidator
```

```
//      cv.setIncludeCertificateRevocationValues(true);
//      cv.setIncludeCertificateRevocationValues(true);
//      cv.setIncludeTimestampTokenValues(true);
```

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 it to use a KeyStore / TrustStore.

Trust anchor initialization from a Trust Store

```
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.loadCertificateFromBase64EncodedStrin
g(

"MIIC9TCCAd2gAwIBAgIBAjANBgkqhkiG9w0BAQUFADArMQswCQYDVQQGEwJBQTEMMAoGA1UEChMDRFNTMQ4wD
AYDVQQDEwVJQ0EgQTAeFw0xMzEyMDIxNzMzMTBaFw0xNTEyMDIxNzMzMTBaMDAxCzAJBgNVBAYTAkFBMQwwCgY
DVQQKEwNEU1MxEzARBgNVBAMTCnVzZXIgQSBSU0EwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAJUHHAphm
SDdQ1t62tppK+dLTANsE2nAj+HCpasS3ohlBsrhteRsvTAbrDyIzCmTYWu/nVI4TGvbzBESwV/QitlkoMLpYFw
32MIBf2DLmECzGJ3vm5haw6u8S9quR1h8Vu7QWd+5KMabZuR+j91RiSuoY0xS2ZQxJw1vhvW9hRYjAgMBAAGjg
aIwgZ8wCQYDVR0TBAIwADAdBgNVHQ4EFgQU9ESnTWfwg13c3LQZzqqwibY5WVYwUwYDVR0jBEwwSoAUIO1CDsB
SUcEoFZxKaWf1PAL1U+uhL6QtMCsxDDAKBgNVBAoTA0RTUzELMAkGA1UEBhMCQUExDjAMBgNVBAMTBVJDQSBBg
gEBMAsGA1UdDwQEAwIHgDARBgNVHSAECjAIMAYGBFUdIAAwDQYJKoZIhvcNAQEFBQADggEBAGnhhnoyVUhDnr/
BSbZ/uWfSuwzFPG+2V9K6WxdIaaXOORFGIdFwGlAwA/Qzpq9snfBxuTkAykxq0uEDhHTj0qXxWRjQ+Dop/Drmc
coF/zDvgGusyY1YXaABd/kc3IYt7ns7z3tpiqIz4A7a/UHplBRXfqjyaZurZuJQRaSdxh6CNhdEUiUBxkbb1Sd
MjuOgjzSDjcDjcegjvDquMKdDetvtu2Qh4ConBBo3fUImwiFRWnbudS5H2HE18ikC7gY/QIuNr7USf1PNyUgcG
2g31cMtemj7UTBHZ2V/jPf7ZXqwfnVSaYkNvM3weAI6R3PI0STjdxN6a9qjt9xld40YEdw="));

    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.

Trusted List Certificate Source

```
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 (LOTLSource)

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.

LOTLSource configuration

```
LOTLSource lotlSource = new LOTLSource();

// 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 europeanLOTLSource is configured with an
// OfficialJournalSchemeInformationURI
LOTLAlert officialJournalDesynchronizationAlert = new LOTLAlert(new
OJUrlChangeDetection(europeanLOTLSource()), 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(europeanLOTLSource()), databaseUpgrader);

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

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);
        CommonsDataLoader commonsDataLoader = new CommonsDataLoader();
        commonCertificateVerifier.setCrlSource(new OnlineCRLSource());
        commonCertificateVerifier.setOcspSource(new OnlineOCSPSource());
        commonCertificateVerifier.setDataLoader(commonsDataLoader);

        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 DSSException("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();
        LogTLSignatureErrorAlertHandler handler = new
LogTLSignatureErrorAlertHandler();
        return new TLAlert(signingDetection, handler);
    }

    public TLAlert tlExpirationDetection() {
        TLExpirationDetection expirationDetection = new TLExpirationDetection();
        LogTLExpirationAlertHandler handler = new LogTLExpirationAlertHandler();
        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.

Configuration of a CompositeTSPSource

```
// 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 DSSException 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	RIPEMD160
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 indepenedent 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.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` in `DomUtils`, which allows you to parse and create XML `Document` object, also can be configured with the methods:

DocumentBuilderFactory configuration

```
// allows enabling of a feature
DomUtils.enableFeature("http://xml.org/sax/features/external-general-entities");

// allows disabling of a feature
DomUtils.disableFeature("http://apache.org/xml/features/nonvalidating/load-external-dtd");
```

The classes `XmlDefinerUtils` and `DomUtils` are singletons, therefore all changes performed on the instances will applied on 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 bootstrap3Templates = DetailedReportXmlDefiner.getHtmlBootstrap3Templates();

// 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 for final reports generation:

- Bootstrap 3 XSLT;
- Bootstrap 4 XSLT.

By default, in DSS since version 5.6 the style sheet for Bootstrap 4 is used. 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):

Bootstrap 3 generation

```
String bootstrap3Report =
SimpleReportFacade.newFacade().generateHtmlBootstrap3Report(xmlSimpleReport);
```

Otherwise, in case if you need to customize the transformer, you can create a report by using an XmlDefiner:

Bootstrap 3 custom generation

```
try (Writer writer = new StringWriter()) {
    Transformer transformer =
SimpleCertificateReportXmlDefiner.getHtmlBootstrap3Templates().newTransformer();
    // specify custom parameters if needed
    transformer.transform(new StreamSource(new StringReader(simpleReport)), new
StreamResult(writer));
    String bootstrap3Report = 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`).

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 ("catched" 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.

Validate a certificate and retrieve its qualification level

```
// 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.setDataLoader(dataLoader); // 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);

original.save("target/original.xml");
```

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" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
```

```
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+XqO"
    },
    "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" : 1607960338233,
      "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: Mon, 14 Dec 2020 15:38:58 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" :
"PGRzOlNpZ25lZEluZm8geG1sbnM6ZHM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyMiPjxkc
zpDYW5vbmljYWxpemF0aW9uTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWw
tZXhjLWMxNG4jIj48L2RzOkNhbm9uaWNhbGl6YXRpb25NZXRob2Q+PGRzOlNpZ25hdHVyZU1ldGhvZCBBbGdvc
ml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiPjwvZHM6U2l
nbmF0dXJlTWV0aG9kPjxkczpSZWZlcmVuY2UgSWQ9InItaWQtMWRlNTcyY2JiMTk5ZWNhNDUyMWQ5NzI1YWE3Y
zNmYTctMSIgVHlwZT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI09iamVjdCIgVVJJPSIjby1
pZC0xZGU1NzJjYmIxOTllY2E0NTIxZDk3MjVhYTdjM2ZhNy0xIj48ZHM6VHJhbnNmb3Jtcz48ZHM6VHJhbnNmb
3JtIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI2Jhc2U2NCI+PC9kczpUcmF
uc2Zvcm0+PC9kczpUcmFuc2Zvcm1zPjxkczpEaWdlc3RNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczL
m9yZy8yMDAxLzA0L3htbGVuYyNzaGEyNTYiPjwvZHM6RGlnZXN0TWV0aG9kPjxkczpEaWdlc3RWYWx1ZT5HRit
Oc3lKeC9pWDFZYWI4azRzdUprTUc3REJPMmxHQUI5RjJTQ1k0R1drPTwvZHM6RGlnZXN0VmFsdWU+PC9kczpSZ
WZlcmVuY2U+PGRzOlJlZmVyZW5jZSBUeXBlPSJodHRwOi8vdXJpLmV0c2kub3JnLzAxOTAzI1NpZ25lZFByb3B
lcnRpZXMiIFVSST0iI3hhZGVzLWlkLTFkZTU3MmNiYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3Ij48ZHM6VHJhb
nNmb3Jtcz48ZHM6VHJhbnNmb3JtIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXh
jLWMxNG4jIj48L2RzOlRyYW5zZm9ybT48L2RzOlRyYW5zZm9ybXM+PGRzOkRpZ2VzdE1ldGhvZCBBbGdvcml0a
G09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiI+PC9kczpEaWdlc3RNZXRob2Q+PGR
zOkRpZ2VzdFZhbHVlPkhmUVliWEEvd2gxT0ZlL0dMYWNKemFSdWtlTmhIWkpOLzVTeHl6b3AwOWc9PC9kczpEa
Wdlc3RWYWx1ZT48L2RzOlJlZmVyZW5jZT48L2RzOlNpZ25lZEluZm8+"
}
```

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" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+XqO"
    },
    "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" : 1607960338233,
      "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" :
"nNdceaOC9YdkSPTiUb8PeC1PLHiN58PHVHqagKNksamPTJxn4UJnZebtCYADDt64NPMXfz93NC67UcF5zusvu
sxFK7/plQjnn0x/LG51ICwIgGkUAz636QwMugr5xO+LY9QQHkd2KBnsioR5+oMoZaVdVS47RcY2fmPe8vyonFD
+d4A0GxE3+KbR5DJRvx7mZ6nqYdhGLXYMf+Tpjah5S/EqVFkofbM9uXGft4iIua0fqmzKxQn58TCF69Q2/Mqdl
bbfhjyQkjOisX7Q/Ki7Jb/tiOib7F+XNb6xnuzdiOOh1wpO/ObYLQlTQSblcIo9AA7IUlig6q9ShLzWlSvrUA=
="
  },
  "toSignDocument" : {
    "bytes" : "SGVsbG8=",
    "digestAlgorithm" : null,
    "name" : "RemoteDocument"
  }
}
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2020 15:38:58 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" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+PGRzOlNpZ25hd
HVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyIgSWQ9ImlkLTFkZTU3MmN
iYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3Ij48ZHM6U2lnbmVkSW5mbz48ZHM6Q2Fub25pY2FsaXphdGlvbk1ld
GhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxkczpTaWd
uYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNyc
2Etc2hhMjU2Ii8+PGRzOlJlZmVyZW5jZSBJZD0ici1pZC0xZGU1NzJjYmIxOTllY2E0NTIxZDk3MjVhYTdjM2Z
hNy0xIiBUeXBlPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjT2JqZWN0IiBVUkk9IiNvLWlkL
TFkZTU3MmNiYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3LTEiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0
gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjYmFzZTY0Ii8+PC9kczpUcmFuc
2Zvcm1zPjxkczpEaWdlc3RNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGV
uYyNzaGEyNTYiLz48ZHM6RGlnZXN0VmFsdWU+R0YrTnN5SngvaVgxWWFiOGs0c3VKa01HN0RCTzJsR0FCOUYyU
0NZNEdXaz08L2RzOkRpZ2VzdFZhbHVlPjwvZHM6UmVmZXJlbmNlPjxkczpSZWZlcmVuY2UgVHlwZT0iaHR0cDo
vL3VyaS5ldHNpLm9yZy8wMTkwMyNTaWduZWRQcm9wZXJ0aWVzIiBVUkk9IiN4YWRlcy1pZC0xZGU1NzJjYmIxO
TllY2E0NTIxZDk3MjVhYTdjM2ZhNyI+PGRzOlRyYW5zZm9ybXM+PGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh
0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjwvZHM6VHJhbnNmb3Jtcz48ZHM6RGlnZ
XN0TWV0aG9kIEFsZ29yaXRob T0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+PGR
zOkRpZ2VzdFZhbHVlPkhmUVliWEEvd2gxT0ZlL0dMYWNKemFSdWtlTmhIWkpOLzVTeHl6b3AwOWc9PC9kczpEa
Wdlc3RWYWx1ZT48L2RzOlJlZmVyZW5jZT48L2RzOlNpZ25lZEluZm8+PGRzOlNpZ25hdHVyZVZhbHVlIElkPSJ
2YWx1ZS1pZC0xZGU1NzJjYmIxOTllY2E0NTIxZDk3MjVhYTdjM2ZhNyI+bk5kY2VhT0M5WWRrU1BUaVViOFBlQ
zFQTEhpTjU4UEhWSHFhZ0tOa3NhbVBUSnhuNFVKblplYnRDWUFERHQ2NE5QTVhmejkzTkM2N1VjRjV6dXN2dXN
4Rks3L3BsUWpubjB4L0xHNTFJQ3dJZ0drVUF6NjM2UXdNdWdyNXhPK0xZOVFRSGtkMktCbnNpb1I1K29Nb1phV
mRWUzQ3UmNZMmZtUGU4dnlvbkZEK2Q0QTBHeEUzK0tiUjVESlJ2eDdtWjZucVlkaEdMWFlNZitUcGphaDVTL0V
xVkZrb2ZiTTl1WEdmdDRpSXVhMGZxbXpLeFFuNThUQ0Y2OVEyL01xZGxiYmZoanlRa2pPaXNYN1EvS2k3SmIvd
GlPaWI3RitYTmI2eG51emRpT09oMXdwTy9PYllMUWxUUVNibGNJbzlBQTdJVWxpZzZxOVNoTHpXbFN2clVBPT0
8L2RzOlNpZ25hdHVyZVZhbHVlPjxkczpLZXlJbmZvPjxkczpYNTA5RGF0YT48ZHM6WDUwOUNlcnRpZmljYXRlP
k1JSUM2akNDQWRLZ0F3SUJBZ0lHTHRZVTE3dFhNQTBHQ1NxR1NJYjNEUUVCQ3dVQU1EQXhHekFaQmdOVkJBTU1
FbEp2YjNSVFpXeG1VMmxuYm1Wa1JtRnJaVEVSTUE4R0ExVUVDZ3dJUkZOVExYUmxjM1F3SGhjTk1UY3dOakE0T
VRFeU5qQXhXaGNOTkRjd056QTBNRGMxTnpJMFdqQW9NUk13RVFZRFZRUUREQXBUYVdkdVpYSkdZV3RsTVJFd0R
3WURWUVFLREFoRVUxTXRkR1Z6ZERDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBT
Ukza1podG5pcG4raWlaSFo5YXg4RmxmRTVPdy9jRndCVGZBRWIzUjFaUVVwNi9CUW5CdDdPbzBKV0J0Yzlxa3Y
3SlVEZGNCSlhQVjVRV1M1QXlNUEhwcVE3NUhpdGpzcS9GenU4ZUh0a0twRml6Y3hHYTlCWmRrUWpoNHJTcnRPM
UtqczBSZDVEUXRXU2drZVZDQ04wOWtOMFpzWjBFTlkrSXA4UXhTbXl6dHNTdGtZWGRVTHFwd3o0SkVYVzl2ejY
0ZVRiZGU0dlFKNnBqSEdhckpmMWdRTkVjMlh6aG1JL3ByWEx5c1dOcUM3bFpnN1BVWlVUcmRlZ0FCVFV6WUNSS
jFrV0JSUG00cW8wTE40MDVjOTRRUWQ0NWE1a1Rnb3dIekVnTG5BUUkyOHgwTTNBNTlUS0MraWVOYzZWRjFQc1R
McFV3N1BOSTJWc3RYNWpBdWFzQ0F3RUFBYU1TTUJBd0RnWURWUjBQQVFIL0JBUURBZ0VHTUEwR0NTcUdTSWIzR
FFFQkN3VUFBNElCQVFDSzZMR0EwMVRSK3JtVThwNnloQWk0T2tETjJiMWRiSUw4bDhpQ01Zb3BMQ3h4OHhxcTN
1YlpDT3hxaDFYMmo2cGdXemFyYjBiL01VaXgwMElvVXZOYkZPeEFXN1BCWklLRExubTZMc2NrUnhzMVUzMnNDO
WQxTE9IZTNXS0JOQjZHWkFMVDFld2poN2hTYldqZnRsbWNvdnErNmVWR0E1Y3ZmMnUvMitUa0treUhWL05SMzk
0blhyZHNkcHZ5Z3d5cEV0WGpldHpEN1VUOTNOdXczeGNWOFZJZnRJdkhmOUxqVTdoK1VqR21LWEc5YzE1ZVlyM
1N6VW12Nmt5T0kwQnZ3MTRQV3RzV0dsMFFkT1NSdklCQnJQNGFkQ25HVGdqZ2prOUxUY084QjhGS3JyKzhsSEd
1YzBicDRsSVVUb2lVa0dJTFhzaUVlRWc5V0FxbStYcU88L2RzOlg1MDlDZXJ0aWZpY2F0ZT48L2RzOlg1MDlEY
XRhPjwvZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWR
lcz0iaHR0cDovL3VyaS5ldHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9IiNpZC0xZGU1NzJjYmIxOTllY
2E0NTIxZDk3MjVhYTdjM2ZhNyI+PHhhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzLWlkLTFkZTU3MmN
iYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3Ij48eGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZ
XM6U2lnbmluZ1RpbWU+MjAyMC0xMi0xNFQxNTozODo1OFo8L3hhZGVzOlNpZ25pbmdUaW1lPjx4YWRlczpTaWd
```

```
uaW5nQ2VydGlmaWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlnZXN0TWV0aG9kI
EFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhNTEyIi8+PGRzOkRpZ2VzdFZ
hbHVlPjE0d01qREZ6ZnNxa2RaVXNuUEgwL2grWm84cno4REV3aU1xMllPMXdOVGZwbEwzd2tRN0UwbDJ5WlBZZ
GVxR0s5U3hDVGx6cDExUk5FVWVMS01zQ2VRPT08L2RzOkRpZ2VzdFZhbHVlPjwveGFkZXM6Q2VydERpZ2VzdD4
8eGFkZXM6SXNzdWVyU2VyaWFsVjI+TUQ0d05LUXlNREF4R3pBWkJnTlZCQU1NRWxKdmIzUlRaV3htVTJsbmJtV
mtSbUZyWlRFUk1BOEdBMVVFQ2d3SVJGTlRMWFJsYzNRQ0JpN1dGTmU3Vnc9PTweveGFkZXM6SXNzdWVyU2VyaWF
sVjI+PC94YWRlczpDZXJ0PjweveGFkZXM6U2lnbmluZ0NlcnRpZmljYXRlVjI+PC94YWRlczpTaWduZWRTaWduY
XR1cmVQcm9wZXJ0aWVzPjx4YWRlczpTaWduZWREYXRhT2JqZWN0UHJvcGVydGllcz48eGFkZXM6RGF0YU9iamV
jdEZvcm1hdCBPYmplY3RSZWZlcmVuY2U9IiNyLWlkLTFkZTU3MmNiYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3L
TEiPjx4YWRlczpNaW1lVHlwZT5hcHBsaWNhdGlvbi9vY3RldC1zdHJlYW08L3hhZGVzOk1pbWVUeXBlPjweveGF
kZXM6RGF0YU9iamVjdEZvcm1hdD48L3hhZGVzOlNpZ25lZERhdGFPYmplY3RQcm9wZXJ0aWVzPjweveGFkZXM6U
2lnbmVkUHJvcGVydGllcz48L3hhZGVzOlF1YWxpZnlpbmdQcm9wZXJ0aWVzPjwvZHM6T2JqZWN0PjxkczpPYmp
lY3QgSWQ9Im8taWQtMWRlNTcyY2JiMTk5ZWNhNDUyMWQ5NzI1YWE3YzNmYTctMSI+U0dWc2JHOD08L2RzOk9ia
mVjdD48L2RzOlNpZ25hdHVyZT4=",
  "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" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+PGRzOlNpZ25hd
HVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyIgSWQ9ImlkLTFkZTU3MmN
iYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3Ij48ZHM6U2lnbmVkSW5mbz48ZHM6Q2Fub25pY2FsaXphdGlvbk1ld
GhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxkczpTaWd
uYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNyc
2Etc2hhMjU2Ii8+PGRzOlJlZmVyZW5jZSBJZD0ici1pZC0xZGU1NzJjYmIxOTllY2E0NTIxZDk3MjVhYTdjM2Z
hNy0xIiBUeXBlPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjT2JqZWN0IiBVUkk9IiNvLWlkL
TFkZTU3MmNiYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3LTEiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0
gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjYmFzZTY0Ii8+PC9kczpUcmFuc
2Zvcm1zPjxkczpEaWdlc3RNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGV
uYyNzaGEyNTYiLz48ZHM6RGlnZXN0VmFsdWU+R0YrTnN5SngvaVgxWWFiOGs0c3VKa01HN0RCTzJsR0FCOUYyU
0NZNEdXaz08L2RzOkRpZ2VzdFZhbHVlPjwvZHM6UmVmZXJlbmNlPjxkczpSZWZlcmVuY2UgVHlwZT0iaHR0cDo
vL3VyaS5ldHNpLm9yZy8wMTkwMyNTaWduZWRQcm9wZXJ0aWVzIiBVUkk9IiN4YWRlcy1pZC0xZGU1NzJjYmIxO
TllY2E0NTIxZDk3MjVhYTdjM2ZhNyI+PGRzOlRyYW5zZm9ybXM+PGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh
0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjwvZHM6VHJhbnNmb3Jtcz48ZHM6RGlnZ
XN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+PGR
zOkRpZ2VzdFZhbHVlPkhmUVliWEEvd2gxT0ZlL0dMYWNKemFSdWtlTmhIWkpOLzVTeHl6b3AwOWc9PC9kczpEa
```

```
Wdlc3RWYWx1ZT48L2RzOlJlZmVyZW5jZT48L2RzOlNpZ25lZEluZm8+PGRzOlNpZ25hdHVyZVZhbHVlIElkPSJ
2YWx1ZS1pZC0xZGU1NzJjYmIxOTllY2E0NTIxZDk3MjVhYTdjM2ZhNyI+bk5kY2VhT0M5WWRrU1BUaVViOFBlQ
zFQTEhpTjU4UEhWSHFhZ0tOa3NhbVBUSnhuNFVKblplYnRDWUFERHQ2NE5QTVhmejkzTkM2N1VjRjV6dXN2dXN
4Rks3L3BsUWpubjB4L0xHNTFJQ3dJZ0drVUF6NjM2UXdNdWdyNXhPK0xZOVFRSGtkMktCbnNpb1I1K29Nb1phV
mRWUzQ3UmNZMmZtUGU4dnlvbkZEK2Q0QTBHeEUzK0tiUjVESlJ2eDdtWjZucVlkaEdMWFlNZitUcGphaDVTL0V
xVkZrb2ZiTTl1WEdmdDRpSXVhMGZxbXpLeFFuNThUQ0Y2OVEyL01xZGxiYmZoanlRa2pPaXNYN1EvS2k3SmIvd
GlPaWI3RitYTmI2eG51emRpT09oMXdwTy9PYllMUWxUUVNibGNJbzlBQTdJVWxpZzZxOVNoTHpXbFN2clVBPT0
8L2RzOlNpZ25hdHVyZVZhbHVlPjxkczpLZXlJbmZvPjxkczpYNTA5RGF0YT48ZHM6WDUwOUNlcnRpZmljYXRlP
k1JSUM2akNDQWRLZ0F3SUJBZ0lHTHRZVTE3dFhNQTBHQ1NxR1NJYjNEUUVCQ3dVQU1EQXhHekFaQmdOVkJBTU1
FbEp2YjNSVFpXeG1VMmxuYm1Wa1JtRnJaVEVSTUE4R0ExVUVDZ3dJUkZOVExYUmxjM1F3SGhjTk1UY3dOakE0T
VRFeU5qQXhXaGNOTkRjd056QTBNRGMxTnpJMFdqQW9NUk13RVFZRFZRUUREQXBUYVdkdVpYSkdZV3RsTVJFd0R
3WURWUVFLREFoRVUxTXRkR1Z6ZERDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBT
Ukza1podG5pcG4raWlaSFo5YXg4RmxmRTVPdy9jRndCVGZBRWIzUjFaUVVwNi9CUW5CdDdPbzBKV0J0Yzlxa3Y
3SlVEZGNCSlhQVjVRV1M1QXlNUEhwcVE3NUhpdGpzcS9GenU4ZUh0a0twRml6Y3hHYTlCWmRrUWpoNHJTcnRPM
UtqczBSZDVEUXRXU2drZVZDQ04wOWtOMFpzWjBFTlkrSXA4UXhTbXl6dHNTdGtZWGRVTHFwd3o0SkVYVzl2ejY
0ZVRiZGU0dlFKNnBqSEdhckpmMWdRTkVjMlh6aG1JL3ByWEx5c1dOcUM3bFpnN1BVWlVUcmRlZ0FCVFV6WUNSS
jFrV0JSUG00cW8wTE40MDVjOTRRUWQ0NWE1a1Rnb3dIekVnTG5BUUkyOHgwTTNBNTlUS0MraWVOYzZWRjFQc1R
McFV3N1BOSTJWc3RYNWpBdWFzQ0F3RUFBYU1TTUJBd0RnWURWUjBQQVFIL0JBUURBZ0VHTUEwR0NTcUdTSWIzR
FFFQkN3VUFBNElCQVFDSzZMR0EwMVRSK3JtVThwNnloQWk0T2tETjJiMWRiSUw4bDhpQ01Zb3BMQ3h4OHhxcTN
1YlpDT3hxaDFYMmo2cGdXemFyYjBiL01VaXgwMElvVXZOYkZPeEFXN1BCWklLRExubTZMc2NrUnhzMVUzMnNDO
WQxTE9IZTNXS0JOQjZHWkFMVDFld2poN2hTYldqZnRsbWNvdnErNmVWR0E1Y3ZmMnUvMitUa0treUhWL05SMzk
0blhyZHNkcHZ5Z3d5cEV0WGpldHpEN1VUOTNOdXczeGNWOFZJZnRJdkhmOUxqVTdoK1VqR21LWEc5YzE1ZVlyM
1N6VW12Nmt5T0kwQnZ3MTRQV3RzV0dsMFFkT1NSdklCQnJQNGFkQ25HVGdqZ2prOUxUY084QjhGS3JyKzhsSEd
1YzBicDRsSVVUb2lVa0dJTFhzaUVlRWc5V0FxbStYcU88L2RzOlg1MDlDZXJ0aWZpY2F0ZT48L2RzOlg1MDlEY
XRhPjwvZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWRlczpRdWFsaWZ5aW5nHJvcGVydGllcyB4bWxuczp4YWR
lcz0iaHR0cDovL3VyaS5ldHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9IiNpZC0xZGU1NzJjYmIxOTllY
2E0NTIxZDk3MjVhYTdjM2ZhNyI+PHhhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzLWlkLTFkZTU3MmN
iYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3Ij48eGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZ
XM6U2lnbmluZ1RpbWU+MjAyMC0xMi0xNFQxNTozODo1OFo8L3hhZGVzOlNpZ25pbmdUaW1lPjx4YWRlczpTaWd
uaW5nQ2VydGlmaWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlnZXN0TWV0aG9kI
EFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhNTEyIi8+PGRzOkRpZ2VzdFZ
hbHVlPjE0d01qREZ6ZnNxa2RaVXNuUEgwL2grWm84cno4REV3aU1xMllPMXdOVGZwbEwzd2tRN0UwbDJ5WlBZZ
GVxR0s5U3hDVGx6cDExUk5FVWVMS01zQ2VRPT08L2RzOkRpZ2VzdFZhbHVlPjwveGFkZXM6Q2VydERpZ2VzdD4
8eGFkZXM6SXNzdWVyU2VyaWFsVjI+TUQ0d05LUXlNREF4R3pBWkJnTlZCQU1NRWxKdmIzUlRaV3htVTJsbmJtV
mtSbUZyWlRFUk1BOEdBMVVFQ2d3SVJGTlRMWFJsYzNRQ0JpN1dGTmU3Vnc9PTweveGFkZXM6SXNzdWVyU2VyaWF
sVjI+PC94YWRlczpDZXJ0PjweveGFkZXM6U2lnbmluZ0NlcnRpZmljYXRlVjI+PC94YWRlczpTaWduZWRTaWduY
XR1cmVQcm9wZXJ0aWVzPjx4YWRlczpTaWduZWREYXRhT2JqZWN0UHJvcGVydGllcz48eGFkZXM6RGF0YU9iamV
jdEZvcm1hdCBPYmplY3RSZWZlcmVuY2U9IiNyLWlkLTFkZTU3MmNiYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3L
TEiPjx4YWRlczpNaW1lVHlwZT5hcHBsaWNhdGlvbi9vY3RldC1zdHJlYW08L3hhZGVzOk1pbWVUeXBlPjweveGF
kZXM6RGF0YU9iamVjdEZvcm1hdD48L3hhZGVzOlNpZ25lZERhdGFPYmplY3RQcm9wZXJ0aWVzPjweveGFkZXM6U
2lnbmVkUHJvcGVydGllcz48L3hhZGVzOlF1YWxpZnlpbmdQcm9wZXJ0aWVzPjwvZHM6T2JqZWN0PjxkczpPYmp
lY3QgSWQ9Im8taWQtMWRlNTcyY2JiMTk5ZWNhNDUyMWQ5NzI1YWE3YzNmYTctMSI+U0dWc2JHOD08L2RzOk9ia
mVjdD48L2RzOlNpZ25hdHVyZT4=",
    "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" : 1607960338295,
      "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: Mon, 14 Dec 2020 15:38:58 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: 8713

{
  "bytes" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+PGRzOlNpZ25hd
HVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyIgSWQ9ImlkLTFkZTU3MmN
iYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3Ij48ZHM6U2lnbmVkSW5mbz48ZHM6Q2Fub25pY2FsaXphdGlvbk1ld
GhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxkczpTaWd
uYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNyc
2Etc2hhMjU2Ii8+PGRzOlJlZmVyZW5jZSBJZD0ici1pZC0xZGU1NzJjYmIxOTllY2E0NTIxZDk3MjVhYTdjM2Z
hNy0xIiBUeXBlPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjT2JqZWN0IiBVUkk9IiNvLWlkL
TFkZTU3MmNiYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3LTEiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0
gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjYmFzZTY0Ii8+PC9kczpUcmFuc
2Zvcm1zPjxkczpEaWdlc3RNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGV
uYyNzaGEyNTYiLz48ZHM6RGlnZXN0VmFsdWU+R0YrTnN5SngvaVgxWWFiOGs0c3VKa01HN0RCTzJsR0FCOUYyU
0NZNEdXaz08L2RzOkRpZ2VzdFZhbHVlPjwvZHM6UmVmZXJlbmNlPjxkczpSZWZlcmVuY2UgVHlwZT0iaHR0cDo
vL3VyaS5ldHNpLm9yZy8wMTkwMyNTaWduZWRQcm9wZXJ0aWVzIiBVUkk9IiN4YWRlcy1pZC0xZGU1NzJjYmIxO
TllY2E0NTIxZDk3MjVhYTdjM2ZhNyI+PGRzOlRyYW5zZm9ybXM+PGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh
0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjwvZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+PGRzOkRpZ2VzdFZhbHVlPkhmUVliWEEvd2gxT0ZlL0dMYWNKemFSdWtlTmhIWkpOLzVTeHl6b3AwOWc9PC9kczpEaWdlc3RWYWx1ZT48L2RzOlJlZmVyZW5jZT48L2RzOlNpZ25lZEluZm8+PGRzOlNpZ25hdHVyZVZhbHVlPjxkczpSaWduYXR1cmVWYWx1ZS1pZC0xZGU1NzJjYmIxOTllY2E0NTIxZDk3MjVhYTdjM2ZhNyI+bk5kY2VhT0M5WWRrU1BUaVViOFBlQ
zFQTEhpTjU4UEhWSHFhZ0tOa3NhbVBUSnhuNFVKblplYnRDWUFERHQ2NE5QTVhmejkzTkM2N1VjRjV6dXN2dXN
4Rks3L3BsUWpubjB4L0xHNTFJQ3dJZ0drVUF6NjM2UXdNdWdyNXhPK0xZOVFRSGtkMktCbnNpb1I1K29Nb1phV
mRWUzQ3UmNZMmZtUGU4dnlvbkZEK2Q0QTBHeEUzK0tiUjVESlJ2eDdtWjZucVlkaEdMWFlNZitUcGphaDVTL0V
xVkZrb2ZiTTl1WEdmdDRpSXVhMGZxbXpLeFFuNThUQ0Y2OVEyL01xZGxiYmZoanlRa2pPaXNYN1EvS2k3SmIvd
GlPaWI3RitYTmI2eG51emRpT09oMXdwTy9PYllMUWxUUVNibGNJbzlBQTdJVWxpZzZxOVNoTHpXbFN2clVBPT0
8L2RzOlNpZ25hdHVyZVZhbHVlPjxkczpLZXlJbmZvPjxkczpYNTA5RGF0YT48ZHM6WDUwOUNlcnRpZmljYXRlP
k1JSUM2akNDQWRLZ0F3SUJBZ0lHTHRZVTE3dFhNQTBHQ1NxR1NJYjNEUUVCQ3dVQU1EQXhHekFaQmdOVkJBTU1
FbEp2YjNSVFpXeG1VMmxuYm1Wa1JtRnJaVEVSTUE4R0ExVUVDZ3dJUkZOVExYUmxjM1F3SGhjTk1UY3dOakE0T
VRFeU5qQXhXaGNOTkRjd056QTBNRGMxTnpJMFdqQW9NUk13RVFZRFZRUUREQXBUYVdkdVpYSkdZV3RsTVJFd0R
3WURWUVFLREFoRVUxTXRkR1Z6ZERDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBRGdnRVBBRENDQVFvQ2dnRUJBT
Ukza1podG5pcG4raWlaSFo5YXg4RmxmRTVPdy9jRndCVGZBRWIzUjFaUVVwNi9CUW5CdDdPbzBKV0J0Yzlxa3Y
3SlVEZGNCSlhQVjVRV1M1QXlNUEhwcVE3NUhpdGpzcS9GenU4ZUh0a0twRml6Y3hHYTlCWmRrUWpoNHJTcnRPM
UtqczBSZDVEUXRXU2drZVZDQ04wOWtOMFpzWjBFTlkrSXA4UXhTbXl6dHNTdGtZWGRVTHFwd3o0SkVYVzl2ejY
0ZVRiZGU0dlFKNnBqSEdhckpmMWdRTkVjMlh6aG1JL3ByWEx5c1dOcUM3bFpnN1BVWlVUcmRlZ0FCVFV6WUNSS
```

```
jFrV0JSUG00cW8wTE40MDVjOTRRUWQ0NWE1a1Rnb3dIekVnTG5BUUkyOHgwTTNBNTlUS0MraWVOYzZWRjFQc1R
McFV3N1BOSTJWc3RYNWpBdWFzQ0F3RUFBYU1TTUJBd0RnWURWUjBQQVFIL0JBUURBZ0VHTUEwR0NTcUdTSWIzR
FFFQkN3VUFBNElCQVFDSzZMR0EwMVRSK3JtVThwNnloQWk0T2tETjJiMWRiSUw4bDhpQ01Zb3BMQ3h4OHhxcTN
1YlpDT3hxaDFYMmo2cGdXemFyYjBiL01VaXgwMElvVXZOYkZPeEFXN1BCWklLRExubTZMc2NrUnhzMVUzMnNDO
WQxTE9IZTNXS0JOQjZHWkFMVDFld2poN2hTYldqZnRsbWNvdnErNmVWR0E1Y3ZmMnUvMitUa0treUhWL05SMzk
0blhyZHNkcHZ5Z3d5cEV0WGpldHpEN1VUOTNOdXczeGNWOFZJZnRJdkhmOUxqVTdoK1VqR21LWEc5YzE1ZVlyM
1N6VW12Nmt5T0kwQnZ3MTRQV3RzV0dsMFFkT1NSdklCQnJQNGFkQ25HVGdqZ2prOUxUY084QjhGS3JyKzhsSEd
1YzBicDRsSVVUb2lVa0dJTFhzaUVlRWc5V0FxbStYcU88L2RzOlg1MDlDZXJ0aWZpY2F0ZT48L2RzOlg1MDlEY
XRhPjwvZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWR
lcz0iaHR0cDovL3VyaS5ldHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9IiNpZC0xZGU1NzJjYmIxOTllY
2E0NTIxZDk3MjVhYTdjM2ZhNyI+PHhhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzLWlkLTFkZTU3MmN
iYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3Ij48eGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZ
XM6U2lnbmluZ1RpbWU+MjAyMC0xMi0xNFQxNTozODo1OFo8L3hhZGVzOlNpZ25pbmdUaW1lPjx4YWRlczpTaWd
uaW5nQ2VydGlmaWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlnZXN0TWV0aG9kI
EFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhNTEyIi8+PGRzOkRpZ2VzdFZ
hbHVlPjE0d01qREZ6ZnNxa2RaVXNuUEgwL2grWm84cno4REV3aU1xMllPMXdOVGZwbEwzd2tRN0UwbDJ5WlBZZ
GVxR0s5U3hDVGx6cDExUk5FVWVMS01zQ2VRPT08L2RzOkRpZ2VzdFZhbHVlPjwveGFkZXM6Q2VydERpZ2VzdD4
8eGFkZXM6SXNzdWVyU2VyaWFsVjI+TUQ0d05LUXlNREF4R3pBWkJnTlZCQU1NRWxKdmIzUlRaV3htVTJsbmJtV
mtSbUZyWlRFUk1BOEdBMVVFQ2d3SVJGTlRMWFJsYzNRQ0JpN1dGTmU3Vnc9PTwveGFkZXM6SXNzdWVyU2VyaWF
sVjI+PC94YWRlczpDZXJ0PjwveGFkZXM6U2lnbmluZ0NlcnRpZmljYXRlVjI+PC94YWRlczpTaWduZWRTaWduY
XR1cmVQcm9wZXJ0aWVzPjx4YWRlczpTaWduZWREYXRhT2JqZWN0UHJvcGVydGllcz48eGFkZXM6RGF0YU9iamV
jdEZvcm1hdCBPYmplY3RSZWZlcmVuY2U9IiNyLWlkLTFkZTU3MmNiYjE5OWVjYTQ1MjFkOTcyNWFhN2MzZmE3L
TEiPjx4YWRlczpNaW1lVHlwZT5hcHBsaWNhdGlvbi9vY3RldC1zdHJlYW08L3hhZGVzOk1pbWVUeXBlPjwveGF
kZXM6RGF0YU9iamVjdEZvcm1hdD48L3hhZGVzOlNpZ25lZERhdGFPYmplY3RQcm9wZXJ0aWVzPjwveGFkZXM6U
2lnbmVkUHJvcGVydGllcz48eGFkZXM6VW5zaWduZWRQcm9wZXJ0aWVzPjx4YWRlczpVbnNpZ25lZFNpZ25hdHV
yZVByb3BlcnRpZXM+PHhhZGVzOlNpZ25hdHVyZVRpbWVTdGFtcCBJZD0iVFMtOGU4MTY1ZDktNjlkZS00NTVjL
TgyMTUtOGJkYjViNThlYWIyIj48ZHM6Q2Fub25pY2FsaXphdGlvbk1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly9
3d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjx4YWRlczpFbmNhcHN1bGF0ZWRUaW1lU3RhbXAgS
WQ9IkVUUy04ZTgxNjVkOS02OWRlLTQ1NWMtODIxNS04YmRiNWI1OGVhYjIiPk1JSUdid1lKS29aSWh2Y05BUWN
Db0lJR1lEQ0NCbHdDQVFNeER6QU5CZ2xnaGtnQlpRTUVBZ0VGQURCeEJnc3Foa2lHOXcwQkNSQUJCS0JpQkdBd
1hnSUJBUVlES2dNRU1ERXdEUVlKWUlaSUFXVURCQUlCQlFBRUlJUVBnSFhoVDVlSGIxeHBhL2xwa1MxK0xENlF
ka0tIOHJTTkxRQXRzRVVKQWhCR2FyNk4zSHJwSEhCdS9uWnZMZnFBR0E4eU1ESXdNVEl4TkRFMU16ZzFPRnFnZ
2dOeU1JSURiakNDQWxhZ0F3SUJBZ0lCWkRBTkJna3Foa2lHOXcwQkFRc0ZBREJWTVJnd0ZnWURWUVFEREE5elp
XeG1MWE5wWjI1bFpDMTBjMkV4R1RBWEJnTlZCQW9NRUU1dmQybHVZU0JUYjJ4MWRHbHZibk14RVRBUEJnTlZCQ
XNNQ0ZCTFNTMVVSVk5VTVFzd0NRWURWUVFHRXdKTVZUQWVGdzB4T1RBM01URXdORFV5TVRsYUZ3MHlNVEExTVR
Fd05EVXlNVGxhTUZVeEdEQVdCZ05WQkFNTUQzTmxiR1l0YzJsbmJtVmtMWFJ6WVRFWk1CY0dBMVVFQ2d3UVRtO
TNhVzVoSUZOdmJIVjBhVzl1Y3pFUk1BOEdBMVVFQ3d3SVVFdEpMVlJGVTFReEN6QUpCZ05WQkFZVEFreFZNSUl
CSWpBTkJna3Foa2lHOXcwQkFRUZBQU9DQVE4QU1JSUJDZ0tDQVFFQXNQWmFnb2k0YlFjaDNKWnlQeUxLZEM2d
E9IQnhrR3ZVMEltZDJGTHh0SWxvVTVGSUpqQjlUS0k2OVFSMFY0QlE0WWNHa2tzYncrYTBIbFFDZWZpajZ5Snl
BY003NEZvRDNvWDBRdzlwbVRreU1qajFIbHBXMkxHRnAwMTNqV1JiRjdLd0YvOHBrYTVPR3M1ZUpYMVdUczd1Y
UQwd1FhUGZ3cEhqb2JXZkJtdi8wRDhIanVOUEV0NnNGZEwzay82WnR5SzV5UHRaYWRnNnM4dmdSd2xmZXVLSTF
zbktqRmdXR1VzQjR0UnVwS1F1S29tSDhXUVJucW1DMDZ3KzVoOTNkSHJSb1dhNHQ0d09rWkpTZ1p1V0ttcHZ5M
2lvYVVOc2pkS3FFalFNTldpSkJqbjZpc2xqMXZhU0lZV0NQY3VzbmhVY29lSUtlVXhzazg1dkhMb3IwUUlEQVF
BQm8wa3dSekFPQmdOVkhROEJBZjhFQkFNQ0I0QXdGZ1lEVlIwbEFRSC9CQXd3Q2dZSUt3WUJCUVVIQXdnd0hRW
URWUjBPQkJZRUZDQVkweUc3THpDdTd0TU5ncjMrMklwOFJNajVNQTBHQ1NxR1NJYjNEUUVCQ3dVQUE0SUJBUUN
YVElsTE12UGZqR2l2Q3FCN2dkZ2Fsb1piUkRHY1h6dkk0UUVEUnBmejc0UTFDNElMazVVekliSXZ0ZWJMY3FDQ
0FxVit6YmtUV2xpK3ZESFBmV05NdS8wdzkxWEdkdGJmQnU5UTFyaWhMWmdFOHExTC9mZzZ4VE1xUVRRa0JhRHB
MZVdHYVJTZ3puMVJsVVlveFFPR3pGaHQxdlg5OFg0UTZyak9uSzdQaGpJaXJ1NjBUZ3pFanB4dElSRGFMS3lXZ
3JRQXlFQUp5OGdmU1NraW1LZWN2SVhzbXkwMUFRRFN1MFhERk5ITnZjUy9ZbEg4Q0dxQlN5QVZNY29rYUoxWEw
wRVpiS2d3NDBnQVBjekFBajh2TWhoNFhObEJFeEVaMFg5Wm9TQWV2QnlVeVJ4VGZKdlVFN09KL09CWGRyanBic
mFmbG9zcm1zd1pVbTNrVGp3OE1ZSUNXekNDQWxjQ0FRRXdXakJWTVJnd0ZnWURWUVFEREE5elpXeG1MWE5wWjI
```

```
1bFpDMTBjMkV4R1RBWEJnTlZCQW9NRUU1dmQybHVZU0JUYjJ4MWRHbHZibk14RVRBUEJnTlZCQXNNQ0ZCTFNTM
VVSVk5VTVFzd0NRWURWUVFHRXdKTVZRSUJaREFOQmdsZ2hrZ0JaUU1FQWdFRkFLQ0IwekFhQmdrcWhraUc5dzB
CQ1FNeERRWUxLb1pJaHZjTkFRa1FBUVF3SEFZSktvWklodmNOQVFrRk1ROFhEVEl3TVRJeE5ERTFNemcxT0Zvd
0xRWUpLb1pJaHZjTkFRazBNU0F3SGpBTkJnbGdoa2dCWlFNRUFnRUZBS0VOQmdrcWhraUc5dzBCQVFzRkFEQXZ
CZ2txaGtpRzl3MEJDUVF4SWdRZzVibnArU3JqRGRseXU1ajB5Q1laWjV0Vm1Fb3U2WjBiRzh0N2VmQTdpZFF3T
ndZTEtvWklodmNOQVFrUUFpOHhLREFtTUNRd0lnUWc0WHVIaUM2dTlxMUFoS0U3Y3VKMkhYYXdIWFZFZVdIS0R
6L3FHcEl5ZnRNd0RRWUpLb1pJaHZjTkFRUxCUUFFZ2dFQVlabHVTcS9YWWJpWk16SWVidTFkbUxrTXE4TnkxS
y9tK3VyNTZqV0oxWVNGeGgrNU14ZEhEeEZUeGV0T1ZPUGRIbnJpVUlKcXJud2VtTHd6WDZzUktOdm9UTkcrd3J
meEx2OTdNWWNvTmtKV3JDR01VS1VGb1JvN3NrNnJKU05uYTl2NVFiaFlUT3ZZc0RzaFREQ2NhL3ZUQ2pWWGJyQ
XlURDBabWFvNzhjeXlkdTd0cWVyZGU5OVVpMzJnbUxaMGdFOW1SUGtJckw0UWJNUnpZaG01b3JMSDhsek5QdlF
PY0JVT0VjcnFkK1d0TEFPVEZ6M2hnajZteThld05UYmFDb1J4WnAvTGFaU3FKdURuMUx3MTV5N2VobWh4S05aM
nc1QVNnUUVaSGNpN1RIN1NmQlJxVTVhdFkzS3RxeWdNMXBlRnI2OElVc2lVYktONUNFTTh6dz09PC94YWRlczp
FbmNhcHN1bGF0ZWRUaW1lU3RhbXA+PC94YWRlczpTaWduYXR1cmVUaW1lU3RhbXA+PC94YWRlczpVbnNpZ25lZ
FNpZ25hdHVyZVByb3BlcnRpZXM+PC94YWRlczpVbnNpZ25lZFByb3BlcnRpZXM+PC94YWRlczpRdWFsaWZ5aW5
nUHJvcGVydGllcz48L2RzOk9iamVjdD48ZHM6T2JqZWN0IElkPSJvLWlkLTFkZTU3MmNiYjE5OWVjYTQ1MjFkO
TcyNWFhN2MzZmE3LTEiPlNHVnNiRzg9PC9kczpPYmplY3Q+PC9kczpTaWduYXR1cmU+",
  "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" :
"JVBERi0xLjQKJcOkw7zDtsOfCjIgMCBvYmoKPDwvTGVuZ3RoIDMgMCBSL0ZpbHRlci9GbGF0ZURlY29kZT4+C
nN0cmVhbQp4nGWTPWvEMAyG9/wKz4Wkki3ZDhhD7y43dCscdCjd+rEV2qV/v5LsHLmUgHAkRXr0WoEJ3e/w7cC
BnHjmybtMOGX38z4837kviaHEvGT8fA6Hy+DzxC6DF9flzd2f0aGcPl4K5colch2xxGMdvVpxnNXhoY5UOFXfH
cnXvKZRzwgloB6Dtwwwmy1Ai5aaLf0UW9w6pWBxDtE+jcvGScf6enkclsvwpBNOJDPgdoqU0xTbFH6dArBiAes
AQXCbEUQuwBArs76FAkmNBXMdkwZn9TyoOWgNGS6r+6SeVnDRdJ3fF4SaYkEERq/ZZxUIA9INs1BHU39Lnbxcj
1GHDTVJE+EZDXk0ZmzI1M+pB3ILzDUqrnJ1XnOfJM/LswgSKqZg4a2QvokpaFssnteVoA0Wdyz+p2RK69uNlPN
eyrFpGZqWejlXHYPyRQMUB+s9iJpeU7Qmas2dogYfd3tAvKLzBl21srvZ7IE0QeGONeseSBe944au2PAABzgaq
F+vu9WwTKHFDmsxElrbHSEdsXdLFttj0/+fMIWwrm/cra8gUWmGuqRtD5QDeyTb8Sq27YHKbH+lzUDKGfSzU/s
6tDU7yyBXuif3B+aa6fYKZW5kc3RyZWFtCmVuZG9iagoKMyAwIG9iago0NTkKZW5kb2JqCgo0IDAgb2JqCjw8L
```

```
0xlbmd0aCA1IDAgUj4+CnN0cmVhbQpQSwMEFAAACAAAJme5Sl7GMgwnAAAAJwAAAAgAAABtaW1ldHlwZWFwcGx
pY2F0aW9uL3ZuZC5vYXNpcy5vcGVuZG9jdW1lbnQudGV4dFBLAwQUAAAIAAAmZ7lKScopFokGAACJBgAAGAAAA
FRodW1ibmFpbHMvdGh1bWJuYWlsLnBuZ4lQTkcNChoKAAAADUlIRFIAAAC1AAABAAgDAAAAwv3H6QAAAblQTFR
FeEseflMoWrx6ZcGDbsWLdMeQd8iSesmUgFYshV00imM9jmlElHFOl3RSmHZUm3pYn4BgoYJjpolsqIxvqo5xr
JB1rpR5sJZ9sph//5F1/5N5s5qBuKGJvKaQv6mVs4H/u4//vJH/pLrRqL7Thc2dis+hjdCjktOnlNSqmdWtnde
wo9m1ptu4qty7msz/sMPXtMbZtsjau8zcsd/Aoc//o9D/rtb/sNf/s9j/vN3/tOHDueLGveTKw7CdxrShyLakz
Lur/5uD/5+I/6GK/6qV/62Z/7Ge/7Si/7yrw5z/xqD/yab/za3/z7D/0bT/1rv/2L7/1ci6/8Ky/8e5/8q9wM/
fwNDf2s7C3dLH3tPIxNLhydbjzNjlwN7/2sP/3sn/0Nvn093owOXMxujSyOjS0OvZ2O/fxOH/yuT/1uDq2+Ts0
Ob+1On+2uz/3fHk/83B/9XL/9jO/93V4c7/5NP/6dv/6uPc/+Pc4Ofv7+rl5Ovx4O//7eH/6O3z6/bv4/D+7PH
06vT+8e3o/+fh/+nk/+3p8Ob/8+z/9PHu9PTz8/X69Pr29fn9+/b0+PT///j3/v7+AAAA////LxzPJgAABItJR
EFUeNrt2I1T2mYcB3D30rPacU5bWZ3arU5Bypg1Oz2x1L4ggpuXZJReXWBxTDdCTBQlVJcLBVo1F5Lq70/uA1y
3219A3H3z5HnJ7wl3nzx5wpPLAF3HbQBqqKGGGuprpj6uEFVYpmzuKEt0dkYVOnqfOe/0nWSPzlhV6Z2ZybDi6
t/jPqrfJ5PHExOHL9KUSaeepyvJDMOlzyuZVPrZ+XEqk0snN55lcrlUljLZykQ6lU1Rrv/qyuHJxvphLk3H68k
X6eNcNpOlw6NnuVQm+e7k3deHbGNXsp5LUXoj+zzDuohdgU/n9dm5r+e1zRLZdrfV7lVsNy4to9dvVD/Gup2yL
9SXoi3yF6IgGiTLoshXt3ib52nHqskCb9cMEm1BrlYtWaZNqtFWzRIMoSbU+qzeFO/zhsjLIlk/b4nVTUOW2zz
VLFvg2/LmDokk3BdE8dva1ualyMb6J/4bEnijzzPkgmqyZVkyG72qcSkbtnUp1lhTFrdE26qxO2AbVcHgSb4QL
doRa4JMlmz5dpVp/9O6tH32NHqJvTfubsylg71woulS4s0fdBBLRDyXdYZ2ySWPVmN/uuSQ6x+1OzgbGlydC0f
c0GwkEg5F5mZdirqRUCQco8Hd0KfRz79/FH1wIzQfmXf9o15NRGM/hsNRGjwIReajj0JMHXsQm4vMx4ilSGI+s
Rr7LDS7t3rD8f3b01/+fnvKs0TlYrdZdn5nVZlIpbJTzPdOKBRMoiaL1YusNAtqJ9gsqn1VLxQkiSsU6lKR8py
0KEkPOVOSSDXb5TxXeClRgaQCJ6mLXkHK56mcr3P5hWJ+xuurWiJpgVnLp8xXmJFMTpWKKlOfqpxaZNdDUlPiH
krcjCMVpEU24tJ3HPvNy/7OEJNMp1k3PbPzN+jRadOsU3evM3S7zrpVaktc22y3qWmSp1K9rXaCfVUzaTezRaV
Teh9jvWXmP/OgF/b88DSurLT0Ukn3WooS11paSY9TvBVv6A2dVkhbbumNhlZSPF1r0UpDa2j6D/5Q7y8rbKNSv
BRXtE5LIVZprPYUpbS0oi3FWa10okvLLOQH9b7e0jRWXDXqrf0WG0udSg3lN03TidHZ6OsNRVdaeue4oWkNXfO
DeuSxQ6Ov3zrB7bEr9+3a07VutM5ykJwrch7T9pXvVpnXAzenp28FpmhoeHQoMH5n6NdXt16NTAbpl7Gbd4bvj
g7fDgw9CTh+U9+dDgaDX2zT9NTUl5NPJ8fo70++GhqfpnsjwbHx8cDoZPD2k8C9a/AVZ+1afntyrqUa3/mghhp
qqKGGGmqooYYaaqihhhpqqKGGGmqooYYaaqihhhpqqKGGGmqooYYaaqihhhpq
```

```
iRXtWPDNafNkEmVLlOgwPh8P8cDfnYhdGq9UqNNTG4LjhKyvBDFcSh4QRPZkMo3kUOt6cKDzVPs3rm1RU+ZaIy
dBghU9WVe53kyNivxuBJs6wmBwbhrm7vHfJ9OW9S3zZHKtsZE0ew49ANH8+PrWxIPKpc2neDlSxoOVkNy23L88
5b0zVAjZBjbnLxeI+tN8e9+Es+0FQRYTHHp9ljzGLG8R5PgQa8EUhcCCy12HquIV2elTzQyhIyYVqDEmnFyhAZ
9mkV6ZyNp5emupYdyJJBlnBnLsQUg0CHe0pOXwXdCrn+QVY9RbAlKFLIobJr1NnBaJFqHmatIElaYuq2DVlP+V
VAU7AVlEDSJ5LIqgmYWbE1h0NfpQx/gKV9V7haegUUEqYy9DGpUE1nKNcIlpAYPJy7Ul3qoWUd2po+T7/Gmoa0
rsH1Md6Hm/TXAbv3A6ZctgdUxwTlJCYyXdvbWVrhmf2Wxu5CZ54RtXsJ7LHBd5hQaNgBsXMMeeUHTfBP3HJ5b/
6nHY0CM/r/4n8if9TzX7DhRzV7PHUSmeeypKqGArZHqY0ETVxQoA+vTijZrp+yp8FISMOeqQXq/9E820lDTyzp
1GXOlzX+xRnPOYMNtWBCX3aiyd4orA/m6y7sFZ9xgHfrADakQIYoNYLnuPitYw7F7k9vgmmyQOVV0VDPw2/NV/
PGneUiuTXWPcbrwQMfcAWkNn7T7Nfya5iWIzaOS7yahaHY5WxHrdHCedZQlJcsfqA4TTXJu8ELjMaB463/kYlF
HUiFIUDiXZTKsG/EGgJGYce9Lu7+x8e8H0w0zscFH3GGsqb5SqNIfZTvj6AKsRLZQp6wZH+rkVkhhN+QGCtJAo
9b4LF/C7OB4nHHlFBR4mgASdIljiG9h9lXNCvXG9KmnX5eJZ5r52KT1mhY5mq9YR1QGcNMgM/DlRlyB6oUsykF
3ElFtjg3UHbkDQ/wpXieg4IQ5oQblkxK7OmThkztoJgOKzAEtFYOYruGLVtOU9AnAmktp2gokVCdJumD56+M85
IZyM0PRA3vJQ66sbNbti13SfeVJIADIVeVTN5HSxKVKRj1FghhWAydEm/Aj1alsqMMVzsKryDoawyAzE0PkpAx
Hz4vUGIKGhQ0BciCuPd4JwI+ldcXEj2VkSb4USixfyhbFB3Fjnq18xRatMc4f2nUyP0eYuR58GK2Ju8YbVY9CZ
vqBntT9+QfvkUtCvaKRBTqkYTAsHZWIYl4QLCtZCbYGlrAoUgaL6yY5mRwiw1YjhJYKGMoaZcMJrTxreJ+VBWR
awqq1CXG4AAQIGFv5wwLtBRQqFsFHoSaEIfojabuylVAtJtKt/i/n8n7v3II67e9cNRkBzTAukLDBeTpzFbVjL
rsVyRVPZ86RVeRvyQsldlWy50mugYhP0GAorhUuoAv3ZiJPihNzmM9LL5CyElUnxHVKbvonRCXprYn7BuqSC9E
iySYLSouMVjWEKt0LnVZtqpvg8EJ16Kj6qDgeaaFQ2bUug09hk+w8Afy8UfW54ch8y6VP5yLKD8AGSlbg7ul6Y
5aMe3XCl9PwN9Q7TUJHsejyHg4f8Ks/623CyC6SoK01VgdsBHeakWjRQacxhwdebvPwv0S9p9m8YvrUq1EbXQ1
S24UzTWiDt669JYhRz0bVqxq01qmF94AGoVjHrTcIz70xZd7VGb7hfSfCh1YG8tGT56yTXzydek7ouzcjQj799
MzUjTVWSE7jKlt6bF99NReoJu4yXunylMTKucUAw6wbr8pmCNXh6tk5F5j83R8RVjA6AQ+BtLdl2XrynZ5oHMP
t+Yo5bsUkzwuNedxXgZH2kG7ZiWgK4YTqtwlp+8yBazmjog/LcFwbTGzrOxoddWTo6iX+CI+/yKMUSNvrMx9Er
r/d+XqcZ8Xil7X3AC9b8tJegxMrInrGa3AOgBcKRpuqoc6fc3DMWzQVYXsVq0D61P4pKaThzWyuQdcRceeAtkC
JVdod8rhtT2WGrdZjCFtpofSIK2R1tOoR/24Bi0Dahjbk8EZHkDpAvI3Q2QLiD3N0C6gDzcAOkC8sMNkC4gb26
AdAF5vAHSBWR1A6TXmC3+jxDpknyYCq6IhFa5SOmuqu/BGgKqjwMp50p/DyEW1Q2/fYXcY1YR3fDbQScoG6TqJ
wVfxp4S9JuD1ud+3qP9nW4hKZIxA+mwgU69RqS1YGia0TOOfbs1V56rVfuCMoROraRFgZFU1TRaxML8jFLv9N6
zt9HWvnbra2jQSWPkCO50toPVxkdY3c5B5WOZR8EAU++0bSgHmuhfHS6j+WIR1TcBhuCue5ar+ZtRD+spAECFu
KD6l1P1UnOhBKYq6N84RMO3Db1hjdHJoKjvn8ZerHqPwDb+UI6fG1/0XVj7c4iaQZLSqbNYLOaL6LGdxL2XoS0
Bzw2/5okW0QAPTvWr1BBLm3+bQHJGm4M3Tv6spLJRYGPDjgtIYrc6D9+3Dz72dwgL8y/wX62H1tl5mxGsn1XMR
+hD4A2eKmoj8jQEa0KOZaOjma0e1JrOvoz4Nnuh62VCT304/Jvyd38BUEsHCLzDyaM6CAAAky4AAFBLAwQUAAg
ICAAmZ7lKAAAAAAAAAAAAAAAACAAAAG1ldGEueG1sjZPNjpswEMfvfQpE9wrmM4BFWKmHPW3VSk2lqpfIsWeJW
2Ij2yzpI/U5+mI1XynZ5NCj//Obmf8MQ/l4PjXOKyjNpdi6oR+4DggqGRf11v26e/Jy97F6V8qXF04BM0m7Ewj
jncAQx6YKjafQ1u2UwJJorrEgJ9DYUCxbEEsKXtN4bDQp54aLn1v3aEyLEer73u9jX6oahUVRoDG6oIxeuLZTz
UgxiqCBoYNGoR+ihR0c/q+pgV1bklJeGg34ZHpsFwVBgqb3QteKsebeAJaNkXVIDPFeOfTvXWcef7XwyK2W7Q4
2qnI0wwU3nDQeVUCMVNUTMX9+K+e7lo0p0V1kShxftrRnu0IVBWHmBakXpbswwUmBg9zfRHEQREkSz3WuM0pG8
Z3UNMNh7GdFkWabIi9KtGADf9/kKjBZA2YNi9pjnRobVp934cc4/TL7uAlfZ9FftAFdRW/oWZ7Yy3lqY0tow6k
z6oYcGvCo7ITZuvYrT/s7kfpGlIcfQM1btV2R4UVTpFakPS6BzRzopWKLFhWzSI8WpwbUpUy2FBJ29f2RG9Ato
bbNDZnkLpoHrEGAGlf6zA8KPo13g1I/8jM/enjmojvvv+Wb/SZxVsC+VXIYyx7kw4eON8xbtvivXomujhDd++G
rv1BLBwh4qlPx5wEAAC4EAABQSwMEFAAICAgAJme5SgAAAAAAAAAAAAAAAAwAAABzZXR0aW5ncy54bWy9Wlt32
jgQft9fkcN7SkJC2nASegwpbRoSOECbs30T9gBaZI2PJAf49zuSTZoSvE1ttE8hvsxIM983N/nq4zoWR0+gNEd
5XTt9d1I7AhlixOX8uvZt0jv+UPvY/usKZzMeQivCMI1BmmMNxtAj+ohel7qV3b6upUq2kGmuW5LFoFsmbGECc
vta6+XTLacsu7IWXC6vawtjkla9vlqt3q3O3qGa108vLy/r7u720RDljM/fqip7+qUqRHxWZF/IFuOUNU5Ozuv
Z/7WjfJEvTNOotbd22G6/fZUryP4ccwOxtc1Rftku7bpGKltPHFbPVqvte+/Xd77T84ECNsGktr1jNgndESjnt
fbJVf21iLeL7cPM+JD7yCOz2Cu40Tg5b1YT/gX4fLF/2afn56en5aSPF7gaQUQYg+6CyTnoHQ1TRAFM1tpGpVB
Ox63sKFxpuMcIiqTPmNBvFn8cs+SYywjWEL021n6AuXeIGmrzNpPfRjtL1UZxa2yL5UZ5VxZi7/1Z87K82CKmn
DUbJZHxnWs+FXB4rjixB2e2kzoqpIgl4Fkl2R00BuMi/p2U9N0PxHhConbBtkBlyHsVcMY2mJouijSWu5zOpZc
GxlZ6B3F5MFK/tkuPhQZVwdqb70sGIz0GAaGBqKfoQoml77n4MrgU3c7j1f4HKDe+PZtmF1LFDOXmP0mrQ4pgZ
qhQJ2QAD05z8m3AGDIviSRbP8nusHA5V5jK3SB9KCVdJGei8LWHQEo0zn176JMjvGQkdPI/K5YseOhr+S7M+vR
xj60fXpMzz8D1NwMlAdVTGI/BpLvZ5mBwn8DaDAULYYEigt1odQg1QZKIzZApNrdevWdq2UMVMzPBhzSegrOJp
82NwHYAUMiyCvInbDog4bSdOZce5H8i00Q9onEhRisIH6fKRR/r/EfyyjhmQnjQ8zPOWf9be90otnLR/vDKuoI
nCURDHppUFVO7goaeQGYMo2LqAWPneB9avskptUtLp2ZLEB96HMLGG003fOEsJ8lM4MpTpJ1YM/kK42OyvICvO
```

```
PXCQcu5Lkv0EFRItRelo4sLH7RAIUgLfIoTs+mCEJaKhzfYhDCbGywr2jsWxIF25Y796WFr98wsOkyD7f4Dwef
Szo4Ov7UgymrtwWxGxaoPLNzqPpuCuMnHX77SIahKVckgNdbSfUqq4m8OItI+E/jtXKKCHlfa9EnrLbUf0txKn
yq/JREzYEuuCcSJoN+eAkuHCq6lx6ArYISrOwAfVaNLGxFEN4wyIZGvEqQyJ9OKdSAjsopcaioOrb+7TISpcD2
GD09rGIgoJ8U90HqLm40qGJaU/UwHFVXW44SFfmA7Zk/wWeCUPQcQMuDSE7T0CKxXnmCCGSUPT5GR5rsVO/mp1
j790GxeND5U6b3z2cFo0vcVY12a/c/esmorRdhVlnyeYtMXTuyWtjPwsYHIdQIuROWECMzYMOUBRj2booLUYBb
WD68gCyI+E9Iecx1+G90FqQgpDnYxTqhxsidmxSPdkmMdG6O+Z6dxA9kVqH0MXe1OjE+P5wnDBj57WmkjrRskF
xy1LPh8cUwWpXrY5rGyteEdKBlozuQwldTbesyJdjADajD9h2LkEDW3mrwhm+BgeMiE59p6nE6NxXdPbKjICAn
FEAXuP3+Ezbk6Qcdc23R5aFFJ16c1oVEy0QcW+YkNqVKE8m2tZ/+OMVVh+YovEAJXjkfUV3eZDKFw8FSFp2gIw
hbOPoli61R/ZZ0z0u/mZZWyV7aL7RAw8dRUyTXpCeEHKCS8/gaq1YrTwJ3eZHUpKX0eOutb2efaR0GTf3uQA47
C5R1sdtUQcS7OO1wytXnb8B+TF8gaL5St5p8bYQ8kR0kV908gDCRlAAr/PsxFikaIu57Iyvv3jbPTZtnvQMxGg
H7AG5ixVPjwtO2psrqi+Myr5OLd7IrgOzaYbNPu4d18gw9ovqba8NnGAkk/crO4ZzJlwteAzqkcAWV2S8XAGGX
be6qY7ezDRwH4a7qigjZmMtpTzDrAlSxl+8iiEUUxlOIV1Q9lsy5L7CGGjf7ERT2QtqP83wxW7dwS1N4Dyz+Mg
vY0J9DPvcm2cntU9Kii6s1VjWVCVOHnC/VXnwfWiz6cbP8LUEsHCCh8lF80BgAAeikAAFBLAwQUAAAIAAAmZ7l
KAAAAAAAAAAAAAAAAHAAAAENvbmZpZ3VyYXRpb25zMi9hY2NlbGVyYXRvci9QSwMEFAAACAAAJme5SgAAAAAA
AAAAAAAABoAAABDb25maWd1cmF0aW9uczIvdG9vbHBhbmVsL1BLAwQUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAGAA
AAENvbmZpZ3VyYXRpb25zMi9mbG9hdGVyL1BLAwQUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAGAAAAENvbmZpZ3VyY
XRpb25zMi9tZW51YmFyL1BLAwQUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAHwAAAENvbmZpZ3VyYXRpb25zMi9pbWF
nZXMvQml0bWFwcy9QSwMEFAAACAAAJme5SgAAAAAAAAAAAAAAAAoAAABDb25maWd1cmF0aW9uczIvcG9wdXBtZ
W51L1BLAwQUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAHAAAAENvbmZpZ3VyYXRpb25zMi9wcm9ncmVzc2Jhci9QSwM
EFAAACAAAJme5SgAAAAAAAAAAAAAAABgAAABDb25maWd1cmF0aW9uczIvdG9vbGJhci9QSwMEFAAACAAAJme5S
gAAAAAAAAAAAAAAAAoAAABDb25maWd1cmF0aW9uczIvc3RhdHVzYmFyL1BLAwQUAAgICAAmZ7lKAAAAAAAAAAA
AAAAADAAAAG1hbmlmZXN0LnJkZs2TzW6DMBCE7zyFZc7YQC8FBXIoyrlqn8A1hlgFL/KaEt6+jpNWUaSq6p/U4
65GM9+OtJvtYRzIi7KowVQ0YyklykhotekrOrsuuaXbOtrYtisfmh3xaoOlnyq6d24qOV+WhS03DGzPs6IoeJr
zPE+8IsHVOHFIDMa0jggJHo1CafXkfBo5zuIJZldRdOugkHn3ID2L3TqpoLIKYbZSvYe2IJGBQI0JTMqEdIMcu
k5LxTOW81E5waHt4sdgvdODojxg8CuOz9jeiAym5V7gvbDuXIPffJVoeu5jenXTxfHfI5RgnDLuT+q7O3n/5/4
uz/8Z4q+0dkRsQM6jZ/qQ57TyH1VHr1BLBwi092jSBQEAAIMDAABQSwMEFAAICAgAJme5SgAAAAAAAAAAAAAA
BUAAABNRVRBLUlORi9tYW5pZmVzdC54bWytk0tqwzAQhvc5hdG2WGqzKsJOFoGeID2AKo8dgTQy1ijEt69skti
lGGLwbp7/fBpGxfHmbHaFLhiPJfvg7ywD1L4y2JTs+/yVf7LjYVc4haaGQPJhZKkPw9MtWexQehVMkKgcBEla+
haw8jo6QJJ/6+U46enNAPbssMumebWxkKf+rp+q62ht3iq6lEwsiUxhB5VROfUtlEy1rTVaUSoTV6z4CMznnJz
gRkysYThfovtBZWwQ9DB5i80Cg3GqATHkV03RHmngS3tcEB7IxZBepRuotxA2l3VAantWIEqHuT3tyWNtmtiNl
xH24sXzCREHFB4N13OFlau6x3hX1S8MTlVv9wcW4t+/PPwCUEsHCJu2OBwMAQAA0gMAAFBLAQIUABQAAAgAACZ
nuUpexjIMJwAAACcAAAAIAAAAAAAAAAAAAAAAAAAAAABtaW1ldHlwZVBLAQIUABQAAAgAACZnuUpJyikWiQYAA
IkGAAAYAAAAAAAAAAAAAAAAAE0AAABUaHVtYm5haWxzL3RodW1ibmFpbC5wbmdQSwECFAAUAAgICAAmZ7lKt2J
NpZQEAAAaGQAACwAAAAAAAAAAAAAAAMBwAAY29udGVudC54bWxQSwECFAAUAAgICAAmZ7lKvMPJozoIAACTL
gAACgAAAAAAAAAAAAAAADZCwAAc3R5bGVzLnhtbFBLAQIUABQACAgIACZnuUp4qlPx5wEAAC4EAAAIAAAAAAA
AAAAAAAAAEsUAABtZXRhLnhtbFBLAQIUABQACAgIACZnuUoofJRfNAYAAHopAAAMAAAAAAAAAAAAAAAAAGgWA
ABzZXR0aW5ncy54bWxQSwECFAAUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAHAAAAAAAAAAAAAAAAADWHAAAQ29uZml
ndXJhdGlvbnMyL2FjY2VsZXJhdG9yL1BLAQIUABQAAAgAACZnuUoAAAAAAAAAAAAAAAaAAAAAAAAAAAAAAAAA
BAdAABDb25maWd1cmF0aW9uczIvdG9vbHBhbmVsL1BLAQIUABQAAAgAACZnuUoAAAAAAAAAAAAAAAYAAAAAA
AAAAAAAAEgdAABDb25maWd1cmF0aW9uczIvZmxvYXRlci9QSwECFAAUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAG
AAAAAAAAAAAAAAAAB+HQAAQ29uZmlndXJhdGlvbnMyL21lbnViYXIvUEsBAhQAFAAACAAAJme5SgAAAAAAAA
AAAAAAB8AAAAAAAAAAAAAAAAAtB0AAENvbmZpZ3VyYXRpb25zMi9pbWFnZXMvQml0bWFwcy9QSwECFAAUAAAIA
AAmZ7lKAAAAAAAAAAAAAAAGgAAAAAAAAAAAAAAADxHQAAQ29uZmlndXJhdGlvbnMyL3BvcHVwbWVudS9QSwE
CFAAUAAAIAAAmZ7lKAAAAAAAAAAAAAAAHAAAAAAAAAAAAAAAApHgAAQ29uZmlndXJhdGlvbnMyL3Byb2dyZ
XNzYmFyL1BLAQIUABQAAAgAACZnuUoAAAAAAAAAAAAAAAYAAAAAAAAAAAAAAAGMeAABDb25maWd1cmF0aW9
uczIvdG9vbGJhci9QSwECFAAUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAGgAAAAAAAAAAAAAACZHgAAQ29uZmlnd
XJhdGlvbnMyL3N0YXR1c2Jhci9QSwECFAAUAAgICAAmZ7lKtPdo0gUBAACDAwAADAAAAAAAAAAAAAAAADRHgA
AbWFuaWZlc3QucmRmUEsBAhQAFAAICAgAJme5Spu2OBwMAQAA0gMAABUAAAAAAAAAAAAAAAAECAAE1FVEEtS
U5GL21hbmlmZXN0LnhtbFBLBQYAAAAAEQARAGUEAABfIQAAAAKZW5kc3RyZWFtCmVuZG9iagoKNSAwIG9iago
5NjkwCmVuZG9iagoKNyAwIG9iago8PC9MZW5ndGggOCAwIFIvRmlsdGVyL0ZsYXRlRGVjb2RlL0xlbmd0aDEgM
TE3NzY+PgpzdHJlYW0KeJzlem1UXNeR4K37Xn833a+/Hh9N06/1aL66oRENEkh8PBA0LSFLLQQKICMaQSPQByB
```

```
AKLbjCNv6sJC1ZmLFliNNonidbGJr143kZOU4G5NznMzJcZw4ux5PMrEdkvFMJicmVhwnm7UttHVfN1jy5OOcn
T1nf2yr37t161bVvbeqblXdRlMTRxLETGYIR5SBQ/3jSl11BSHk+4SAfWB6SrrZpEgILxJCxaHxfYeKwj/9DSH
cHwnRafYdvGuoZM/9DYSYkCUzfzjRP9j44blqQtbkI2LdMCJ2L9+lw/6d2M8fPjT1ya9pn4lifwb7HQfHBvq/X
rutHPuvYj94qP+T41/RzvGEyNgl0mj/ocS2xBPnsI9r0G8dH5ucGiT5NwkJHGTj4xOJ8d/3PKbH/kO4vinEAf5
jHzOCWtanHK/R6vQGo8mcYbEKNrvDSf5/+mjOEheJauqIlYyr79s+3GWSzdqbb9/+Xt568/3/m6vQp5rz5MvkW
XKW/IT0pgciJEZGyBHE3Pr5NvkRYtknRnrIU2T2z4i9TK7heIouTh4mj/8Zuhh5jFwlf3fbLDFyiNyDa/ka+Qm
sJd9DVxkj74Ke3Ee+g1LfRdwdf0oUteBrSAWHbsH+lFygZ8gW+hZ2HmcjNEQF8iK5CHtQ8hTu8+zqjmv/jdBT5
F587yTDZBph9aOp+/AfieHm73BX95It5H7SSA7ewvFN+AJnRPt1kC+gTr+t4kIrg7oot59+ndIbj2Dnb8g+fPo
B907Pco2kWWODZwlRWrq7Ojt2tu+Ibd92x9a2LZujrZGW5k1NjUpDfV3txg011evXVa0tD5WVBosKC/z58hqfN
8tpE6yWDJPRoNdpNTxHgQRb5EhcShbEk3yBHI2Wsr7cj4j+WxDxpISoyO00SSmukkm3UypIOfQxSiVFqaxSgiD
VktrSoNQiS8mXm2XpGvTs6EL4bLPcLSWXVPgOFeYL1E4Gdnw+5JBasoabpSTEpZZkZHp4tiXejPLmTcZN8qaEs
TRI5o0mBE0IJYvk8XkoqgcVoEUtG+Yp0WewaZOcv6V/MBnb0dXS7Pb5ukuDm5MWuVkdIptUkUntpqROFSmNsKW
TM9J8cGH2oWsC2RsPmAflwf47u5JcP/LOci2zs6eStkCyWG5OFt/9VhbuPJEMys0tyQCT2ta+Ok/bR1NCUuMXZ
Gn29wS3Iy+9fTumP43R+oXfEwZGUL2zsxFZiszGZ/uv3ZzZK0uCPDtvNs+Ot6CGSawLua7d/MYZdzLyUHdSiA/
DhvRmI+1tSceO3V1J6o9Iw/2IwW+D7Kt2+2zdKzSxPzdMUBGoDtSpz8c2fuaaQvZiJzmzoyvVl8he9xWihALdS
RpnIwsrI65ONjKzMrLKHpfRmm07u2aTvH/zoNyCOj7Tn5zZi/60n5lCFpKWP7h98qzdJtWEulVaCVe1eXBESmo
KUC3IdSsDegpjmRXUjuUPqWbJjRMU2OxSjYximJwWuSWe/k4PZ6EAqTSYjAZSpu/oSirNCCj9aRu1zJeHkKM/j
iYaaVbNlwzJ40mn3LRqT7aslpGdXSpLmi3p3JQk8YE0VzLU0sxmllpm482pJTBZ8o6u50j45uJ8peS+GiaVpLu
ZEYub0K8KWma7BoeS3rh7EE/akNTl9iWVbjRwt9yV6GaOhhoqXsTpfOqMSbqpo6ttp9y2o6erOr2Q1AATx/tbP
iZG7nKnxKDLJfV+vdRF3Vw3EgqIkCIIyE21+E7q/Hp8BFS4imWu2lQrdYGbrFDjMpLFUkuiOU3H+rcJ1TB32hR
dkaZlXZSzKer2dftSn9IgxWEpPTFy6JlSoytDnB8jAeIoilFRTJdZzOelLjkhd8vDUlKJdbG9MfWoWk4rQ9V52
lYdt/VuURaqifhweKXDlJmMBNy3KjfZqvZXu9GPDW9eGZZm9XLbzlkmXE4LJLjyzUnCXFiptrnV08/Osxzpx0O
MJ1o9z7PzisLO8jA7trPy5sFZeWdXrUqNEeRe991sLjtpg7aOptIgBrOmeRke3DGvwIM7e7qeE7CkerCj6woFu
ine1D2fj2Ndz0mYK1QsZViGZB2JdZikduzoVXr3cwohM+ooryLU/sA1ICpOv4IDMnCNpnDCCo4ijk/hFBXHPmi
lrGHUMcbvFmmQ2edT3cOz8W7m40REjeAXkiDXo3bk+nmgWnPSKCeakia5ieEbGL4hhdcyvA49A0QoDd49K7TIv
88qVVM3acbXoKYTK2AdKZsHEqq9ouP1SxXzWs3rtVc4iiCZ5xhaw9BXdFrDh7VXgOHDNp/N77P5mqm0nA/nl4c
1ne8/3cy/TFgluhGz+BmsudxkTinWu0DvBH0G6M2gN4o9u21QaWu2UZun2TPoedHD5XsqEeQ8124uXN1QF2WtU
lZYEl30AI15xj2XPK94eEN2T4wuoimy4gYDFUx6ri9Dw2WLNtrnJHbSULFUUVHREGZve01oCUK9SxW9fb2BwOE
J4fVe7Kwt39OLn3DVelthlc+lw/Wn2kyXD3E+bDdevkyzL9Ojl2nW5cs3/vXyjdOXq9z853OqqnJu/JZaWfvBX
ndVlZtW3XjJXZXSYezm21yE+w7xkhIypZSedsJ5B5gcZxxUdBe4qSErO6s46/EsXl8Q9ZpM3iAJQv1M8FLwepA
Lsh1v2hJlrZJZUhb1Q/RBEUQS8/u1Uixb0O6wiaRhqWFpdUu4nz29wg8qQsJSekO94HLm0XBFPV3vsnDymjJaV
bkuXJFHPQBrLNTlqyyjwLsaJnrzmprqczIbt3WVHvniYPAHL7Q9sLdm+bHqHVXZ8BlbIAo/sW8+ua9Oozdqq61
uMUP59Dfu+sO7RXs+P90OF0O77tm69Z5daqEFpBg3LmB9bSCnlCJNa4CAicCG3eQA1pUXCO8mu8nz5CXCs97T6
F2mF01gYpttiERZq+RWb4zOmYASk2CKmS6ZkqYFk3YOgesmTiUsLU8RmjOEqDZmQH+KcS5URUXDEgQCvYf7JgI
BZtjDE3t6D6Mq/CkrokVh7IvL2ZcuQSSSXVqaraErvr4N7XQF7WRCSz2k1B43njNSjRHO6C/oqVEPZ/gLPDXwc
Jyeo1RL0VmjxEd8ko8KvnJfzLfo41lP8XEbfWxhYuOW6Bd8MO4DxRf3zfgu+fi4D9Qhi78sKka1tphBcMe4lAE
htISrnpgI4HInhDdXjcfs56TymgLVaKIOjSUzq6VNyl157Rev/vjHr7/2j8/m1A1u3hKvFsXq+JbNg3U58NN3b
pLl3/7mw//5u/7HR9avH3m8f+/nDtTUHPhcykb05s91p/EMOukPlSs6K2Ra4J4M6MqAQAYMayCLggZAz2mg3WQ
29Wg1Tq1WkwHtPMffac5wms0ZFNotVksPR50c1rvQLliFOwngHRJOmkFnLjSvNx8184ZPWPdZ6TprxEp11kxro
ZUzmjmrlTNzWiJ+T4TnRfiqCBdEOC1CgbhLnBa518R/Eemz4osi/bKKvkeEIRE6RIiIYBCzxWKR+4XIFHnwJ29
Gz4lPivSkCO0i5IuVYrPIiSJQEd4V4S0RXhPhSZGJ4u4ST4sUBRTj+RELcJqT4tdFjUGEp34l/i+RisqXvhJ9S
fwpgz73t1GccoTNt0ukBeI6pOTYhFfnHomqE/tPnYk6RdCK8AcRXhWBSfuuyHWL0MawIrJwEyrHgbEUR2hoJPq
cCPeLAOMiDDKut0R6RrwgPi1y42x3gyJVRHCKbBfqbEpwUyR6RgRFjImURyyteY/p6yXWc4rnRM4pTqmbe1XUq
AxVzsyoWwyIbSKn5RS5KBriGrjtHOfmIIMDTjFnRYuxzciwWAQT0fbZ9ZzVbGacdiSWzOVmxcw5zUDZV/VUS1a
00txspmYe3cFBGsL46Q30HT58OGALp6JpRSgcQhxDYQTCcDoxwVyXIfAzMdGHTh1IOzPSqAQB/LeHddM07BNYd
fg0XviuLbOmLhQKsQF8TmUJgVOBF/9ks7Yc+lb5DSDjF9TobYCwIQU5l7dULT+5fKFqufkIJd+BBhgphTuh/Ef
wTf437/+Ou/jhoCaDhfAPt3NPfLiHm0d4JZadx7iQRQ5c5Y1Ar938sRIyWKMOL4zBMeAADK3EIlgky4LlFcuiR
au3eHP6cqiSA7scQw7q4LIo06VgMEcxOwnWmN1qsMTM6WjVEA4FhO+Fe+HwBG50qSLUi5kogEFbLsRzvpJ9cAe
Z9VyYng9syFWUjeIXl5uOHgW7ITPW25vPfWd5VJ9hN95oYhGNk7JLjzjWBvMwpm3BmPZLjMM5uItPKVVDRdNF9
LweDPrTenqRh7M8mHnQ24ncmhkgAWjFRwnMBBYCnBSIqwAfUMNxoCzqjm7XgCYzluNyxERSGDMKMlZFO9T4FRa
+x4JYg5qDAmrQFV5H51iNYn4LyJKtsh6DlphZxtUxk7CtgRrH1q23lYG8Rkv/R/CB7uVj4f1PjIUnqygGlc9D8
```

```
9TyH5e9/ub4xtr9/pLR8IljEXk9/PzI8w+0mE2mwNpy63tZpe8/l10KL4/MdRdmCvSXesNraDe8yXMnNFFiJPV
KiWCOmWnMPG5Omq+beTKqCBogGkUT01zSJDUavUY3Y9QSzSSXpVol52VAYxzOWeo9jKlDgybw2zRV/jCdBPuNE
DiWfwunKmMs0/OR/qo31Xjag/M9rNlK1pBjSqc/DzTZn82merM9115r32rnT1uhhAenkxrGctasyckn+Uo+VfL
j+ZfyF/P5Sl+zj/rKscSj1JDpe9gOxC7Yqd1unjyVCZla6puG1NJ6l8Kh3l6bvQZVHcajksr1tprQ2nKCuYIdl
GKoqtdUVRawRGEF2VYPqHWX08LrfNzDH76U+OqpxCbf9Mm8jetCdrlp6yO7Xn8jEJs6Nz9Irzyy57P3Tc+c673
/QYPVYXwSqD37v/6n9ofuu/fkY7vRn3zLW7kkngUfKcfKLTESuitEtR44bjtno1obHDedM1HOBHotHoo1UUuFU
gGkYqaC1iAQqxivmKt4peJ6hSYFcNsrICDyua3EJ2DmfMXH632+3FieuyzmEEsKd/AGgcQ4K0uPYfWkLFWAekQ
qKnpVF1vJk2kfc6QPCZY2uFmWMwvzOA+oBwjUvKnV2fJQF+vOHwA9ddU0bSnoemhvuHL4b0fCh8PsJD+5rBylg
2sa92wsP1RQMhQ+/kluKLt0vd3jMtff87XpyeceiJhMZq8v17CcFQplcduG5naX2IQbNr3hp8wH7OgDv+QuEzd
8S1kz7XzUSTXuE266P+funM/lcJocyJT0GVGRvY44wH7t5uJVBG3YKhkIGDNmM6gBq2BDekTPRsoR4BkLx15uB
9FaNLocndNBMiwas8vsRsihRdjS4oATDnBcu/mvykMFxdEtGpjVAKfJQtgF7cjTZnZh5nZhTkeWVovGabFoXFt
yICfHiXIzULCGJ05CjcTjUfoHo7zH6aGp8nvQc9zzJFbjr3re8hgYPh+RDPUsIt/yvOcx1jBspWfKc07F6irx9
SoO8Grh7vGlCvdKa2Y0hoW74FE81LHXAhaLjpgFs2Tm9GaHK4ezxLQZbidvzBSsRKPjzDHOiKEyvJSJ1bs9swZ
b5gXhcOjwSv3OSqaA8P09vRV4LGrYg4PhU7fmBwj09R7GwCScWlhIPfoFffqNuSMQUFMUFo747e0lvaly0ZRyn
nQCcaca4O4aXd5170+W71v+L4egavn6GDx179deuQ/aDy7/sclVWpoJdyzPYyvAefgbjMju5XdBwNa1/FT67sM
PY90lkEXFqNG5dNTA6e3oCzNKeXwwqrWL9gJ7xL7LPmTXvmeHf7HDq3YYsk/bv2z/uv27dk2HHdg4fc0OKlNNY
0uUDdNme4edVtgBJDs4WQCBGqe90j5oP2d/1f6W/T27DlF03A6KPWYft3N2ZhZ/UVRtczxqq+Q6MqO0J2ZexJw
PGX1Yp/UZY3rQ67m40c4ifahXSJ1EDPSpQltN930s5acTNxrjB2vLMR8DXqIwuDvCXDqH0aM/BP29n8+rb7R/8
M94marV6HM+GLZoIXP5WOomlarFA/iyYyx1EA+5qOwkW0zGC8anjdyvjR8Y6XEjGLNbTc6Ak7Y5dzsvOD9w8qy
30fm083nnr51awanU1EWdXt7r9GLN5IU5L14UvZe8Se+Cl59DgHrTdwi1zXKrrSLgXUKz08rnxDxWZ3YscyVDs
xvFyk6F19UK/carveq1AtRAw65ULLXlUe6ja8Y9trwiUSzMs9nyCkWxKM9mZPeOExDgf3YrFqk+2LGSt9neCbu
HYM4W8L74sCKYZFS+NctKLVyeI4/msfJjg94UJVGdlClRQQoqQSDBmSCtEYJzQaoE49iZCyaDC8HFoE5SuwtBP
sfU+rMSKGH7dODdpcQS84s5RqNmh0ewxbBoV+MrS+MQCrCrcTrAYrzFxIL1yJ5eggZNXz/ScVTN4qgB3UqGUXV
RSJf9Lf0bs9avq7CXHAzPfurGmdMQAoyvpfdtW3i58uCXDpcPxHsK4PrQmV1+3mDW38jU63/Ml2WVLicda6uqs
uTAr94++sLJqMmebVX9IYo6Ocx9GzWyjjym+A4UgDszkEktYr1I7ZLJGvXYS+3UbIcMGwAPHAubHoMtinPqc43
rWrXVM9XQVw1KNSCwttVZyPTgNVqihYXbneAsKFgTiOXmknXhHUariPdJ15oYGkCtaTD3YHZlN2yWcbFYW2L3t
NXyhsUL9QUs77CapnDlll3PN0BVKvFgGq6qB4fOwrmcrN6BHymjsdIjy8sOazjat6G5tzorb93mzr7ysxZfdUn
5Xv+a6sYzrz2wcVd17sPNAxXct7M2DLTdOJFdusdaJGeVtO2rrd9dXyjqgX+kpKUiN8d15GWLazmPp46yWH3Sm
4VxxnnzbVrK30eyySeUGlqtt0V5LTzjhgU3NLi3u6nR0srFnHEndTp1hBM4ieP0HG+OGRSDJWrQmawu2w6iFnc
N4R8EllaiLh5/TL8Ta8t7Axq8ldrkqgYIu8Iu2eYU2c5cFg62xfvuuTfR8A//sLHcv9lrXbuxyTmxjz5SWvj3f
99x41hjk1HbaHRajal7aAhz5s+xppDJ158jejRciUGI8hhyrKxitgqtJtOvTVQ0+Z/0A+93+l/0v+Xna570v+e
nfmbFrEB5NOAHpx8W/HjM/TDun/HP+Tk2eNW7JqoSlTrEKMlpncEAQIlX8EoIvOJd9Gr1XikmW3MEkzdm8biyC
XG2a5jhwyvXG3Yg2E8IFWoIYDd0lm+E11khElAdANI1R7pUT9VdtxTu0Jx7R0dH/rqeRv/E8oF7d3Tm1tdW2Y8
t7z98GMxcXMjHc920v+3GoywK0K6dfVq9ib/h+CgmADmN/v8exsON5J+VJ5zefG+llzM5IAAbgeYAGCtzKunxE
JwsA1dZVRltLAE+y5lF9cUOMNrBZLKAWWsUjdSa582jQl6euWdtHamD6sW663WU1L1SR8vrFGy4oh4XGsVV7lJ
cMRevdw0VQUcRPBCE6SB0BAeD1B8EMQinBdgi3C1QsxAs4nN0fesA1jnkvhziBa83h8d0raaJMFammLJX73WqC
4XUA4SR9E1MJnt6WdmqlnDpKg7UUi6cmcex3zfw+Gir0tcGm1zIbgnqjx/a1WBzplF2NO+8M7h1fHN+3cCnT3x
6oK526qsHB660NcrBmdiWA5E1dQPHThwbqKuZ/M9H6j+5v8cHI1/LCvgcxdG9G6J7GkvLqncd6912rLs8x7b8q
yelEml9W6BxV10wtHH3A/HeRw/UmJ05GSlfbcBg9JTmCeKDZ5QMgzZbW6zl9Bihb8jMyfb88YPoGRkq5WZ5UOa
Oy6/Kb8nvyfy4DE5EdSCSZ68p+Vl1QGuS3TL9/nUZXlRJOZWXjXNPrvCm6BmoUacwJq9GVbaLatd8/kL0ggxT8
nGZqoi1p89Gn5aBsR2XObcMvAzvyfC8DEyOigrIFJEHGME5mVO55hLD0bYV2qfl52V6ToaAvJtROmXKMC/JHIP
ZNqZkzYYPZHgW10gvyZAvsw1PqeK0ggyUyCDJ5XJMnpHn5KS8KF+X9YIsYXdB5rMyMnJbuVTFP8Mq/lxfzOsiO
TEu22qPGfpYPWiAVQdi+VeNPBV4M2ZJqU/9rWDlt4PUaVz5ySCg/oig/oy2QqJi2Cl1yFXrVxzJteJIHmBxSz2
oP3/iicCOI5tLI7lrS4WCXDmYY3z//ZeW+TNc19rCpv1fPFRt0r98j9HkbRyMXOz48A++0lJfyifwsf16yy/Wn
++z1v6eeFN/z//vP/6P+z/6k/TyVt1p9Br2x36aRiGfzrfcQj6xSgQf+8u3i75NmjV/RzZyZ0mM1pBinpBtjF1
3llD6lNrfgs+z+PTg48PHzv8T2Yi0AYQJ8kURdiIc4ifJaeRB/yVB8gTkwjT8Bn7D3c8t8nv4R/H+e177lK5W9
w19s/5NQ4XhReODps+oK3KR2vSaKSbEELvl8iZtDeFUbC7sWl13fHUPgJTxNEyJjoynYY64ydE0zCPNZ9Kwhlj
IE2lYS6wkmYZ15G7yQhrWEyfUpGEDscAdadiEa9i9+j9bymBFfgYZgy+lYQupp06cHXgD9hZoexoGInH2NEyJh
atIwxxZxylpmEea6TSsIbnco2lYS/K4K2lYR97jXknDelLEv5iGDSSXfzsNm0i1Rp+GzeROzYr8DPKm5mIatpB
Pae/eNDZ+18TIvuEpqWigWKooL18vtScGpWj/VFDaPDpQJjUePCipBJPSRGIyMTGdGCyTtm5uamlv7Ni8fZs0M
in1S1MT/YOJQ/0TB6Sxodv5t47sTUz0T42MjUo7ExMjQ+2JfUcO9k80Tg4kRgcTE1Kp9HGKj/d3JSYmWWdtWfn
```

```
6ssqPRj9O/FcWgqvfNzI5lZhA5Mio1Fm2s0yK9U8lRqek/tFBqWOVcfvQ0MhAQkUOJCam+pF4bGoYl7r/yMTI5
ODIAJttsmx1B5vGJsbH0kuaSkwnpDv6p6YSk2Ojw1NT4xtCoaNHj5b1p4kHkLZsYOxQ6C+NTd01nhhMTI7sG8W
dlw1PHTq4FRc0OokLP6LOiKu5VWuRsVE0zsEUTVCaTCQkJn4S5Q8lBnFp4xNj+xMDU2VjE/tCR0cOjIRS8kZG9
4U+EsOkpOf593GTTWQMz+BdZIKMkH1kmEwRiRSRAVKMbQUpx3/rEWonCTKIbZT0I0UQoc1kFKnKEGL/w+Ygth9
JmFR7CWwT2E6rvIxyK3I1kRaU1kg6EN5OtiF2RKXvx2cKqfuRNkEOYTtBDiBujAz9xfm3Iv9edR42MoL0ozi6U
8WMIC/j3EeO4AqZxEacawAxo+osE0hZqq7rL8v4a+O7VGhydWQtrovprYxU/knevyb536eRlO73qVKmVNkpyhF
VdidS7FSpYion08WUOtuoStXxJ2bcjjMOIT/T3EeUA6rsKeynJI8hPJzW6n7U+IS6gkGVb2Vvkzjzv7UB88EJ9
MKxj2mJrW5anfMOFT+l+hQbG1Z742QDZp0Q5g32rwxpbpc8kJZbpkKHkPL/lG8KT8i4qseEaud9SJuyeZkq8xD
619a0hkZVv2caOnLLHlO6+XO+FlHb1Mk5eJscZlnWMt6V1U+m1z+kzpPS2ji+x1DvCVXbZSp2n7rHEbThCEK3r
o9ZbF8a9/HVrKzl9v38v5ybSxURN30445/4zBvi3wIdq8DV9wvAK92weAN+eAOkG3DsA4h9ADPvzr1Lf3u92Pv
M9Reu0+3v9L3zzDtc+TtgfQf0ZElYii3Fl8aXLi1pjda3wUx+DbZ/Wqz2/iz8Rueb4dc7yRtQG3tj5o3kGxwri
Xve0JsibwDX+ToneoUFaaF8YXxhZuGVhcWF6wv6mW/NfYv+t2+GvNZver9JvVe3Xz12lYt/Baxf8X6Fxi7EL9C
5i2C96L0Yush97vEy7+Oted7HHi30Lj56/VH1j0FVj2bYIn2fhWOfefgzdPzkzMm5k9zMibkT9JnpF6bpZKzYO
zYa8I62lnizw1mdujDXqeVuqj9SNe/1F0XifYq3D4l295R7e1qLvY6wvVODi+WR0Mp51b/6jXEPcy9wOn17LM+
7A5/F2PUYtW73bg9tV38f6W/zoaAt41tmtnCbI8XeaGu119rqbQ21/rD1Z63vtGr7WuEL+I08E3khwimR4lBEi
eT5IrlRd6cYdnUKYWsnBdIJYdIZst60Uqu1z3rMyn6wJ3RGBA1cg7n5jp2BQNs13c32tqQ+tjsJDyb9O9lb2dG
T1D6YJJ09u7vmAf5D94mzZ0mTpy1ZsbMrGfd0tyUHEVAYMIOA4JkXSVP35OQU+6k2AIEAgkfwTQJHELVnMoUkg
ZVhEpiEyUkyOQkBNqaCiCGTAYZmGMYDyLlnkrAXGw2oVAyanMza878BOUzSmgplbmRzdHJlYW0KZW5kb2JqCgo
4IDAgb2JqCjc3NzEKZW5kb2JqCgo5IDAgb2JqCjw8L1R5cGUvRm9udERlc2NyaXB0b3IvRm9udE5hbWUvRkFBQ
UFBK0xpYmVyYXRpb25TZXJpZgovRmxhZ3MgNAovRm9udEJCb3hbLTE3NiAtMzAzIDEwMDUgOTgxXS9JdGFsaWN
BbmdsZSAwCi9Bc2NlbnQgODkxCi9EZXNjZW50IC0yMTYKL0NhcEhlaWdodCA5ODEKL1N0ZW1WIDgwCi9Gb250R
mlsZTIgNyAwIFIKPj4KZW5kb2JqCgoxMCAwIG9iago8PC9MZW5ndGggMzE1L0ZpbHRlci9GbGF0ZURlY29kZT4
+CnN0cmVhbQp4nF2STW6DMBCF95zCy3QRYZMQEgkhpSRILPqj0h4A7CG1VIxlyILb1zOmrdQF6BvNm/HjmbisL
7XRc/zqRtnAzHptlINpvDsJrIObNpFImNJyXit6y6G1Uexnm2WaYahNP+Z5FL/53jS7hW3OauzgIYpfnAKnzY1
tPsrG183d2i8YwMyMR0XBFPR+z1Nrn9sBYpra1sq39bxs/cif4H2xwBKqRbAiRwWTbSW41twgyjkvWF5VRQRG/
eslPIx0vfxsnZcKL+V8fyw8J8SHFHkXuETeB66QU+KEIx+I0ww5I84S5GPQ75FPxDuBfA5Mmsegpz1lYPJwCX7
o3GvYc0KuAuOs4IFxpwj+MzxLrP6vyMF/ukNe/R8okPXLMRq8u5/Imbw75+OmC6acMWFtgP3+BHa0OEbPN3Qjm
gkKZW5kc3RyZWFtCmVuZG9iagoKMTEgMCBvYmoKPDwvVHlwZS9Gb250L1N1YnR5cGUvVHJ1ZVR5cGUvQmFzZUZ
vbnQvRkFBQUFBK0xpYmVyYXRpb25TZXJpZgovRmlyc3RDaGFyIDAKL0xhc3RDaGFyIDIwCi9XaWR0aHNbMzY1I
DcyMiA0NDMgMjc3IDUwMCAyNTAgOTQzIDMzMyA1MDAgNTAwIDUwMCA1MDAgNTAwIDYxMCAyNzcgNTAwCjQ0MyA
yNzcgNTAwIDU1NiAzMzMgXQovRm9udERlc2NyaXB0b3IgOSAwIFIKL1RvVW5pY29kZSAxMCAwIFIKPj4KZW5kb
2JqCgoxMiAwIG9iago8PC9MZW5ndGggMTMgMCBSL0ZpbHRlci9GbGF0ZURlY29kZS9MZW5ndGgxIDEzOTkyPj4
Kc3RyZWFtCnic5Xt5lGRXed/d71uqXq3vveq1qqum603P0lt1Vc2MRt3SjGa0gxCSLCAgekYtzcAsYhaEIITNY
kfIZjGWg0jYsTFgGwIBTCDYgBEkQMD2AXwi4pjNHLE6BiR15/fdV9IBzDn2Sc7JP+nuqXrLfffd5ft+3+/33Tv
nzpzfYD57PpOsfPTk+m0Lc7Mtxti/ZYxXbz1xxy0/ufxFn8bx+xibv//YxvrNb/j3r1libPFulBkcw4XG1vcFz
j+D823HTp571o/1dBXn38P5JSdOH10/cIo9ibGlRbp/cv1Zt+0TV+Jw6Tp8tE6tn9w496wv0/1zjOnKbafPnlt
mH9pC1S+l+/jH8Us/BRwaOhdSaWM9PwgLxahUZv/f/RQZ29oSr5M+22BPY6fYWfZM9mzM30vYK9nd7A3sXvZmD
FPELf6MnefZPF/lQzoX09xYMy+yeTEY4hruTfN0uLIq6GxV0hn9Sfq0q7yHq4N5MexNSxu5i3XLB2bYsYN6RjX
HwyytdbM4TdJ53k+7y1nSTYedum1yKoHHO3gvyqeDYQ1XxKsqs4VCUgxUOp+OV3hR737i7sAKwaWpT0+MRd3Zg
meFEqZorA2ltr6SVtpGVRYL2qtuqxbjehxpXy/vuG7RFoyUWkq5dHguDIU1QiZjodi83QhlPCHeL7goezIU6kK
jOW8o7uHvISn0E4SIjJae0GLMk3jOoAkDabzDz71EcK9oC/sOXTqG13hZb99gviiFF+rCvsc8cVd9e92reBrly
/Wo6I8tNiZ2715u6aIxodFRKVJ+1b9fm9nLZmvddqvoTyxcsafQmmkV5VjRePxyvP1mJdQNwsjwRult3i2M91a
ptdWJlj6Xn+fCaCXVKSXQcTmBDgqYvFAsxrw/xH/Kv8BugEcdYcfZbewO9jz2YvYK9tvsd9mb2FvY77M/Yh9gH
2IfZZ9lX2BfYWy2HgnMXUZm4D6TFBNNc91PMO2DZLhMV/gwn/pI2kWL7yQ3idgkzorqmGy9yjMzHKzQ9A5XMOm
DZbIXVGnN0qLt9QftzNAdm5uVpaJZN0vqTU5VdvE06lp1J1Rlz5mmK7qynNk0kzAeOkX7cD1yDZrnw6xtOvxzp
WQiaKxdce2uoOpXC0r61khP+SYs1isqtFyXa6ViGCbheFaWXHBdMKau4sU9B+cExnH52sNrrWiqpmFsRQy3Vpv
/WmNUz3MuxKtQnifaN9rTAsMNKxHK8lhyDROyVERJzvUMZkmFQpCByga+xnGRozBXnGPyxLzVZTyBsjxsTifSV
IvlnUfo/eY+qubd5cm0ZquzDVXxil5hfGqmJpQWwnR6O7LURFEEw/KLkVTSRAGHeepQl8aqc4f3LycKXSrVdvR
rfqp1WAw9rWGKUk0KoU0bZaX5NIylwIc6ULDGQ9IYsQbT4eIo+oeOXQUrLxpYoBCzaCKs7APoiuDzirqEXvJDc
Ce4Q8zFPmE2PyGcPXIbj01Gflk0Lvwunpb6Fk6YvBf2uMmb4n5mgc5/zf6G/S37DvsB+wn7OdvEuPi8wqd5h2c
ck8jX+CF+BX8sv54/hR/ht/AT/Bn83/AX8ZfzV/PX8zfyt/P38D/mH+Yf45/k9/Gv8K/xb/Bv8x/wTWFEKKoiE
```

```
ZOiLbaLedEXe4BvK4Nhr25iCStNarCrJM0Rj/4k2Z6tmymepINev7u0mCVDlI/zm10YWcTj5YTwLbZdmxh7AWF
kD/cHKNwxFl4xydNON3NVE4r1BkM8nybLae4UhJz0DF2EubZtAk8zXbrlrJc+eyjU6zv8paL9HIf7+SnMvJcmd
Wo3NUQmabc+7K4AQocdONcaBi9d6fYtYWruaMlgja8MM/gOILWXLQ96QFyTrfLlZNjuZg5rcaVOTwD6exiFrI5
qUSJzz6fdFTggudZ+XJHJ0mKT2taJaQRTaoVNuk0U7NboMiqqp9koUpglNMJI8uQUV9CybGXQ7xAiwEHTbpvwH
mOT0fA20Xg0dDkbJkMzdK+0PQSL3rDfsTE93+52suGKG9562pO2s7KGwe2ihTQ8ie1Q9EhM3M2/LXVimNTtIj0
SpzVc6tAcxO1Mpl24GpWrubLDQb8HJMK/Tt+2484AVZuMj8Hi+evIzO+CB3AlCnA8EVUCLbUXcG44XB9eYsl0l
aeUJu83cGYJ9wEoFxBFlEIxeIjwDD3gWQln4kYqbgSeAlgAXJw34bEQHwI1aWEKCEkKJ9IvKFuiwIPoAt/3atJ
XJSsCpUJdb9ccbknl160Uyh8bTxEBne8SMH0ffupzAjeLs20EOwlhEzm0kJK7MMYjRf3CDwqOSxAlraucLq3ik
zoe0Big9wodk9RnuDSXysNXSaMOvBBDFaMUHYm6Ky5odLgk+kUhyXIrNL1SKKWm0Bo+ASzUVNLS41wg1qIo6pS
G02u0UUbHVJLAFnja1ErTMX4sITaacSNGU9JoU4+55e7VUlnqXAFwjzaiTdSwQ+4OlTLX4UPKgjRykQbCLtN8G
WA7P0d134WCH8Tgo+zDn1ZW++KwDrQf2lBHU1FyYaM4NdkgzoE5l1bRRKKnRZ8jkINbWMuF73uex6XVpoR21CJ
btP4EYgVmnSYSRmE9rmAKgG2ECAxERVgvREMVjEYExApgMdOtSYXKEQ3QeAQf2wz92Kja0kovvuat18DFE2GUq
s3VPFgE7AdlMGNAcC0ffhW6CxCl2Z/HtNxJwyj4+zEnZzG1Wj4GQ9jKh6RqDsISZc+4GUTXv40hmEaF5zEfXF1
qiFFgDv+Kghum8Vp6CV80FDsOS/Fuah1/CoU5zMYtOPMxOJvfgtmpEG8xO7gcTQOqq9EcP9VNH5Bd8AM05u8mb
+KrznA02ayLnw3f45e5OCWvkvo3cXNI3QG/ULmBbX6U7LZGvskjFPTNMzEvEgNQhq3pFLHbgyeJqntM8BUe5yY
kNt9GVvOjWoCwuvnv4Fqy+HfS8+RQemSUUuxzIYvNbW1t/SP/mhiwC9jF7FL2GHCpJ7J1doydYM9gt7PnOhb92
+x3wKbeCjb1fvaf2WfYf2VfQnz7GruffYt9j/0Y0c1DABnnLd7lO/ki2rGfH2BsOE/EathdFYTXFxB+E+pP8zz
smCSlEEGY5eKUoz8cgD8NbOw7HjVIQYoocBAmZkOQou4CRRpHo9Ke7aRDh6RxTqSS5V7XsWyAP06ArfUOBZgL+
LCbZGlm4h4gesVSEKSokaS2nVHcQ+SQCAJ4YpDYYT8bcTabhyI57FBQGgwtETjEQ7o+HCwjkKRDuwguD9iHHlg
VeG0G4K8jZqxxHExzVGtRuwsZwOwEjTZLKV7xCBvsOomA19WTOHVBNNtbaFQ0jMCv1sXE3O5KFAZyFt43d8mUi
oMgKehKtSxNmfycwFBUbjpUyWanfWn53WRPnRpmvE53OAexgamJaYCFr2HSyvPosjC+VbBDThamVKDIxj7ijNE
ZnxC3OhveTx+JIFsXElVqUxbGoZRIJPk7fMEI66MAqiGk0CrRuqgs8LuvREk4wkUO4gHZJ6nJF5XHKj6f7JSEt
y5cY/AoYIBiC9m3yK2Z0BJER6gazkLqTBX3I3gLYERbYOQwShrNRn15ftZWVhabUtYMosv0xduTfr9X88Kg6Jd
roY3jqgIzFGtPjiS3IIoV/jqCUDNPL5rRAsj19xgWIFgHrX2sgZ4W+s/RKCNv0/x9aM97iBeK/fg44zqvIPvEf
RiuLrzJCOOQ9w9x+zPUAfUPIqBCpmBFpIoUHszzFQH0c50+4xv6adoqkFZ7qxZlwh79JMF9sReYJnZBZO1p8G3
91fH3Y3oVYUWRIglgCN+YWs6fiBeqb+CWuBo3mjSKhG67aKDxd4OmcAn/niYnFw9AGxsWsNeO1PA72B+w90IV/
Sf2Z+y/sb9kX2X/g32bPcD+F3uYMx7wGL68Df7Ygx9fxC+DaoJddub5bL+ekheDa+TMMldIzldyghdDYOeyCp7
dNtY9J0dSi1R2MiA5baYJEiLQInDAR2S2092QzQP4+aDvRNmwl6ROi2n47zCBg2ZwOvgeKBrIT2rJrd1b8GCai
6cOEbw052gRB93rDSCjYhLdI4G/lnO8lJR7zmWJKo20Gzz6kWdtjSRe3TrKO8CFbFW6MiQOM1ekNqB6iQ12qBE
4Bgsl0mrEcYezfPMawX9SnrG6JAt+VNwGKaICq2wAJROoMI4ai2NczeyYqHralsx4twvti7ipZG1mMkWgxUwX0
jLRKl0fj72kMVbQ3PjCVY9g2qiHcvnQtG75OoiK0PBA+AnhBVbLsJFWFfQTImzaffjl0okbEtouOgnobT0lSbU
pKPgg1DWEJlRZVcbYMNQac0tBD/gxkLBis1yABtcyuAOPe+AzwhOHA6Jbnj4qKQcgykYf4i6aglhdEFTBfTSam
9ZAWiD3Dfih8uMqBSQ0ivMmUZiecjSMgEfqAF0Hwfk4op3yvuoBbA5WZsfrE4XsYKf12Dbe5o1PTRXwhnC4d0F
5ZU97YTS9vT/b2B2NHbr8EtBauqi9cqW0c7mXXP/B68uD4Tz4io5q5VJQ3NbtRs3V/Xsa0sRrhw40xHjz8O8/Q
SBt5gyQC+I1BOPUk5120L3yisu3cfipFs1vobt/hC6aTyiubxDGe6rh6qm4+CDAz4NULvcDxGQQJlUs7qF4j76
ONyd13HxhGE6H8mUgxxgDfZcpmMYrJDfmneT4NCVgWxiNa415AZ4HcBgVbn5WhSHmAmc6VAXlXaKISPlCNP4CI
1V2xIffpIz3KQwlJWzOC/M4kt+CsoJVaNEH+Zf459ll7Cp2LbuRPZkdZbciqt/OnsNeyF7M7mKvARq8kb2FvdN
F6owczhk+PpeHiJ7k3E7mUdZrWsAN4MoUuB9xebL77g7EfMQ/eqq9wEnruDiL4E1+ZEkRkSwiH0UcxBMdOzCZJ
F1Xp5yaC6YQmUPrtJBBmM0ooNIDiLjkX+OUkPIxxkZ7kUc9B8n0qvVK2C5QpLImnh2fu3KpGHtVL6zUkvL4fDY
dQmcUECY9EidK+/V6jT/DmD7lpcx2RYqEBvgEUXhxvRAHOY8JPB3XJtJFyIu5wMMVipeiAPN1IN6DdvFJz/DJU
qjjCeuDEFfmdi025q7ZEcZhrjaiWhHTbayXGD9ZvKiFqqam6yrctefCidrC/HaoJ4EoCY+DILhMqBLFwbokz0S
8Vc8jeSG+pcQ3UNcppzW0IFLLt0OcYAQuwOcamY85r8VrOYLlzzjVhlarn/PAZ9Wtza2fwyzuZxeyQ+wKdg27n
j2dnWbn2bPZb7KXOgu4B/GAsmTvY4wDl4Haj4C7dIwMHyXuMqQc9oESlCnLEVuPaJnskLwlLDVpRgC46vCeKFI
WQR4PB/UFThre1Aith4Nehu/Y3chqOdGjRJftgTIlOCb4pmknQAXXImHb5z+Mu60JX9swICeoL7e1MYFfGkuSY
nGseN3E3v6OoOFXs97eiaTXW6hpgCymqQwZ5Fmz+cmcZyCuCzlF4k7cS6TjlZV9Ky1fjk1PGRCEgpjeOeYVZj7
kWDpc8wISeRc69YqBBxW5k55XbvRzhh5wC8SAQE1IW4AR0M99sJlaUphptyOv4j/2nkNhc3rMBqT5gnqwzStOb
itve9L6TY/bF4alqBpZf7I5FdRazU6sIvlTTbpOb74OtlajBniY0TV/176LJk1jvC4o66XkzgsP7d/+EejNP6A
im88h4jZGsoXk5gtwna8Tl6jQ9XXgK5fPKZqgh8ByF2kioBSMhbGU2a1NRPwvsgXWZ/vY1Q4p1tnNQIozQIrns
ZeD+9/L3ua4/4fYR9gnwP+/CsSgFETE2xlxfE6kN3LZS/zlgLHiPpPUsegMs79s046xvRRokBE2pHUTgxHHefK
HonAKSu1SKKDEFgHe1F2yKHZIU0PQX+n2OyU+6K6RVrAgzMmyyzm1u/Q4TmoDSrxn4OYEZGmHEmKDOuh2d5CZG
mrLEsropF3K/GTUmjz49/qOUeCk/ZDy66Zcq9oqHJmTpSgRlmumYgPgSbHSqWgvqHqtPXXEAC2/4BKFVeCvl1C
```

```
aBPwMAwugqhFAUCqBcxx2gCtl/CoH25iUIwEBjHcdntYOSmBNZYhksFvSw4iv2iCIGhLfcPwiqXHS15CcXASUq
SDCTkKSy4p2BgtT9OlJfmZuWKUrYXdxMFGYAVRBaIaBH7WqyqYLDa+kUTkvhDuOLTavej2eLgvedZkKYuBkRZQ
UEa/L45PmJ9HFCbzx03qVeifU9rxnCXepEe9LuP+yPCuCXj0j4E5lc77541wTO8zmR6jsb5CW1kBfs/kpz3uj1
qkiQvzHmlTtPkJ2dCsogtmEGnFsG7Nsij3Av8B2sxVo1EOIZb/B/hU7AoV6ip1jd7DnI5bdDWZ7j4tl72F/wj4
IXvsp9jkw26+zv2XfZT9kP0WDAl4FrwUfHuZKU1LE6jrGmS8IAd+WKcpJ64yW0G1pkYwU0EYSDiZu2lbmhkbpy
H7m0qouo2gdjI3SknmaftDGm2DydTvsDsi+02wncU9U5TKTvSxZGTFS0pX5QZrnOLsIpRQJKfxl0KdwnzqFTxT
pu4IZ2oVnY5dGRCuku9rklIVMHuG5I79yuhTfVH1+g67bmOjqyHuM/fvS7uXFKsKRoVwcBGEggFW+FZ5RtST1Z
/ozYWDqcV3vnuTloSOQZLVFaWSuF19KJuh+fRFQtCIroFOXW5TcJeUoe+hyIQgrhKviJRKSUsbOWXJl5VKGUoR
aFp2oDCQpHh1oSwi7AwSS6CunNQmX/aMlL+05+cNdVu6AVk6Gcl/JCGYECyyRF8FNVdGK0CNWJTNBAli7zBOIA
AcjdTlKdXESmXpB0dIV3hJNT02GIICFhd586IV4ptk/0K53Ou2arJUm3eJH0XVbFl3qjr/MEVv4HJcbZPd7YeA
mdYr5RtewZbci0qTwfiOunoRba/VCXeC7XVwfx/Wh830llzTht0/1OdcKaKDv5fJJ5HhdGv8TkvjLZA40XUoXo
tFbDIQSQHI1dcjomESnll9041MNkyQOgyRpFPZpErIfRkltxZvkEk2o5Gxia2vzo/w7iAnPA094BXu14wlvBk9
8L/sA+4/sk+w+9gX2ZfbX7L+z/wn/eoBtYkaLgLkx3uYZgoojD5awtZOtwPZi53Lwgm6+jIYT0nqJcaa5SrqKG
GYK9HYW3c/X39qOQqKCHmlGuFTiyImrnLzXCb+uW2XgdQvvcCn9nluMTdqphY8QBV3lPVq2oKWF7vKQvHUFEYZ
cIU1W8ny/k7L5atwwk04doro4Ba/FUzkZGfZlngTqrWQu24M2d/FKuQPeuJInotA5uUwEBj3I2hbRbJhTaAprb
4vqpaJVkF9E+2idrBiFBZpSbQplQ7rHVyD0kPseEBCyQsqJmbqv07FyMewc7JIcAJkBiYG8MQWbZ/2kpqrKIai
DLtanaqV0grLhBQgZiMOx1Pen+ATCxCRlYjSgnqqhZ6T4HMhLYLiYJdCFxhC8GPoQcyFaRLlWDUuWJzSJQyN3k
9iEtRtdIS818BvbFJQ8OSHE9dFssy4KWbdlbAEC9sWOmBphlCf1MyhlKSU82kQ5uxb4jtK9q2uT6IJfbpUlqoU
vSQ/9s/W9q/uTykwFw2SbM+MABBFMTE6G1VrZ7t67NsYL8XihIiFGSyD24NhTIXQpuG8F6rVaK/FQ6V3d+kJta
q45XiqMucyScMGThLPw+DJI2wxASZtX4caT3SIn/YyrPEMLFaVLlBHS3m20SiGWceUKrTwX1hZJ3KIYOHBBf13
oq7mmHL9Sl9PaCD+iUAFolil66fRs3fcL1Uo0CaTafBB4pvXLTWSVF2wewiP2MloxeLwOjYVpXIp2MDa59eDWN
6HdHoJiO44o9yz2XPYC8LB8LwOx9fey/8A+zD7GPs/+kn2NfYP9HbzwB+zH7GfsIfJExnRO2dtkn4B+OWL0iEU
dl7bIHg18FAjJ9WDzFOt6cLR4lNuBHKBkTewyO5R0TV3OkrwDXkG23l/M3LqYJWbVzYxbl8vg1m51fOjckdjWc
tqhZbTcvyztirBmCX6SlXhviJopIZPly4aJW3Uj5yYpCQKZ1YcdtGONUw2IyG4ZsAsaN8wQuJLUZjEhSAexOU7
z1sT83TM7m42SgvjSNK6qGNEikFCTM9OYKQsTDQLPm15YKYWNkBakypWS6Ax2duud3XEb7jLWnDJhozC1NF6q0
soyuF+lHFKkoSXwgHA8Ihtockl+1qGAVDd8yrEjWu12q+CbHxypyRlaEgM/w1zP0wKQFYsUFEO3dEScsKo8E/L
QbcHwSN9JO6dxBjtV8FjRCyWtq1B4k2qXA/JVyrfSYhvCB4cOhreoXZJWxFLZXBzUSQt4Vd8Lg8DCTs10qwFWJ
nEO9+8eXtvTKE1TXDStbTNWx+3txYlrdpDxF9I4CYxf9Sf3N3mlMYb26PLsjl0Ngi0irzcAoTTR3sd7tDUDTk6
MVixZsdvTysX1vZYW36bz5Sx1FbyHVnz47QrcVtsewvoC8d+itU/WRhS9MRsFFEZpX8oOQWIW5cANlNhG/reKv
pyVfMnxyEukfBdKnFL2IHeEWmze4xbFwBfjrQe2HuIPgy9eCab4hBFPPAsl83x2J2LZb0HNEE98F/tDp2c+xv6
csh88T35QACKzrz+yHrFKH5FI0jwb+UjGMxc1tFAbqe4sBZLBMnBrOFpx7dYhLWI428DlEhFEEB/c0i6dkQNlV
E+fUv6Dvst/dCWq6HdscgElSFKQvAV6NYJZYnJDX3aszq089OPBkLjoYK6p24vw292xA5AiLaZ55QLkZkMWfOG
bdNovxY2Z8bQ4tu+iAy3iAS5fglGDOo4BV978gWtuuJEfR/TB0/cBkTDCoSUMpCUqQcpD6ElH3mhPDjEKn1KRX
PRcImYWLiEmgOKwLFoYKAJrjd2uXQAQlOaCsUBwyFjRYgWfGLv82uuu3FcqwM41oqFXjCu1XTs6PohmpcSrSV0
W9x++JOZRrREmBw70ffSN6EshKuHDejWPcvwHL1hM7VdhJT6/Rwe0asGvEflSh7zDmQnJKtEHB5UlB+9cHCCWd
if15Plu0SFwpGrzfqV0V0A7kbW+mvjrnSh3E0zrFlzqwsl8se62tDDWZGzry/y74v7R7rLXw5be7rLpf8I+zj7
L/gv7iuNFlE//EftHMCMJIlgCN5rg2/gc7/E9/GJ+mLItox1pGYRo23Ty9DmtSJnuSEcDo7NIlrjboQHoBeo54
dB5dDtF37qNAI5CdVYGQOERwXp0Q4XL6Siyu0jGTpA7/LaQHtkQatqC7KRZbtfzDsk7Jl6jbI6FfqGo4HZ8NHl
3kLq8Nm02IMGfkqJ2+y6cxbezGuhSXgl0Ssf1BcCdKxxKpbvcT3dlASKlQytv9Mywu0QSHceojNyJpIxzI+gd0
jHdpcUF2tniGjGkpTL+ru6ErPs2KEtnfqJAbIaAopIVRZlW4KUNrJzqTibQJo5bS7J2CASAD+VZdS20oUeP2wC
GYdQ4UNYzpaqG0KjOTNGep3KceIXpydRA81cqYFYQ32HoVWeTdHCOjGcmX0hT39PGgqYo2uWgp5QoaWF7tPqPQ
mW3cg7lIsDYKrn2F1P4LLp1u5CYvZVubcy5CyjWaB1ZVHxpdM76LyTQvw56QjXQBcrGwx5Re0nmq8igcm4hC7e
4nIBnotMEvqGTBC+KVq9+3MzESs1lCDzQSgJvxCbfKyXFYhJ6szvmivHicG12u7AeGirdQjdiR9mzRUgDlIQ8o
LiH+oLYhzIUxfFG4peny1fedVFltlRdWtoZ6bBcL/jZ8nA58yjMVKtV368a7b8FbVtDr66miPHwzyhPAW9Flco
uyMzkronAM5nvuaLBMqT1RL4Jqwe5VqAMm1XdAv9TF1E2v2XtlW5p//v5UuEefD41X/vavIEk010Uduw3SWiV0
Fc39NaCO9PrHkfPPJ6KE1ogGCUerYwa4zVIyNGSWeL2cn1f3ADm9RLEjdey33Mr4O903OtP2afcfsK/Yn/Dvun
WzX7GtrgPr85zXm7Nt8lHuXK3q8o5e76XFJdWnEHTTkK30ux2pXadViDHJ/I2SsULVzBPVriE+CAZPqLvnZpKE
GNcoKBsalrvEkeb5/GAGFrPpV7JS2sgSqOEPXlXVjdu8Qtf06NdrzVoHdvJo5KTTVl9aTHtjjIJA9fCJF0e0F1
KREiQQCJiy1m+T6zEM0C/1eNTRRHphSt3cF5vTo6XyuNJDdbje8rCUIOxxJ+YHDPCLxSAzFqrSiEMOyFQUlsTk
palTYikT+r1sts2U9CU4lJe0a92psdcYIIv2mi8TB4CGtYqa10ZK0GcfxXMytgLhDjkIg0PwYqIM5AktgXpF6R
jK7QPBRJL0y6fGVhiSHm7Y9bqijZejawikzBIKBaKJwHttcED0y641CQPXW7hHnguLb+hsZfSYleh1D3YIQmnG
```

```
pNTgR8W61UNziY9NdHb0bS07GZkFIXcgyVHlkeK6Bxt5oC1G/AoUvxw3KhWU7boeVOene/toN0/cbEwOZtvLgk
1ZFKEUCdpPxa4ww4fMuzrFLJCNPHVhrsYxg+Hvme9q3D9x5TG3nyKlQ3u1oZO0JdQvycJDluk4zyFkH9e2D0kO
1u5O70dcqxNIZSyCQdpKTkQe6j76uHvqHcQps171SCoeoqcxd/6+dbD4jj/IptnffCux41Wm57GbmMMgFk3iGG
inS+YunTxKDUcW0rtmmlFcUYjNAH1h3mBTr69Ot/5Os3jfMcc7DCDHE8yBCSnD1La15HE4smeqhnhzV94qFmZi
KauuvrS5XqxeuRQpV1pHyiVwoKHtvvlRmXu4Pjm7bSdmAPtKNtFu2LnWhUpC43iRBy0HexQrsiIMoxoD+Ws1Ju
NJdJeo/2vXzdiop3Mpp16aOpea1q09t8wfMyzKAPDJ2qGKLqu7V1bW5kR6XL3Mw52t/xareJBnlZ39tey1lRod
/b3PJFEJm9i2mFK7yRbeIyktGkzT+beQJyZKK3b749/9Q9c/2c3lfb/A/Plt+nyF3/3pu89sgV+64HNj7pVfsw
GE6OLnFb+t8wv7JTnv7JzvsLvRhn6nw2vxhntkv+drS137Uc4fge+JVAwYhba0vIf4N79Ww/yF209wL+J+3+Bf
w9u/dzV9JC4RN9r77J3+S8PXhneW3ggemf5WNXUBvGO5OPpbe7NFbGX9sj+mnbQzxQ/8uj13qNlOPrTGx0LtGP
/6FiyBjs4OlYo86TRscHxzaNjy7axk6NjD2e/NTr2cfyG0XGA2t80Og7B7r706P+uaLIfjo6LaFt5dByxJt89O
q4yn1+FlsCycHYLv350zFmVf2R0LFjEPz86lmyRf210rFhVhKNjg+Op0bFll4mF0bHHInH76NjH8Z2j44DdIl4
zOg7ZqvjZ6LjA4C6j4yLbJ8+PjiMM13sPnr7tjjPHbz12rrX96FxreXGp3zpyR+vomfPPPn5qY7518YkTLXf7b
OvMxtmNM8/cuHn+6LHTR0+f2Di3fu3GredPrJ8ZFd7XevTGPlfRo6f/9KB1/GxrvXXuzPrNGyfXzzy9dfqWR18
5+h59/R8079i5c7ed3bewcPvtt88f2Ti2furoxvypjXMLo4f+mdvXbJw5e/rU+onW9Wc3Wt3WwdMnT26cOXo8v
/DPPPto92CCp91/OjjDjrNbITXPsRbbDuibw/cyW2RLAMQWROgd+DyKUrT0SimdDUBli13MTuC39QtPn3VnG/j
ewPcz8XkzSh7FvdP4PI3SGyi1Dmm7gSfO43wd5X655n3uXb/6xL5faNE/vfsvudJC/dQ+OjqHN9Ii4QYcjFrwd
Fw7zW75Nb385fNfPvt/M3p05xzecxZjsIDf293vPOrdwL111HfU1XjK9XPhV970f/f0Na4lZ9HPUyhLrb3eta7
FuvhH/T+J3w03h8d/qYTDuBzbP4DB+DU//xvo2dLfCmVuZHN0cmVhbQplbmRvYmoKCjEzIDAgb2JqCjg5MjQKZ
W5kb2JqCgoxNCAwIG9iago8PC9UeXBlL0ZvbnREZXNjcmlwdG9yL0ZvbnROYW1lL0dBQUFBQStjaG9jb2xldGE
KL0ZsYWdzIDQKL0ZvbnRCQm94Wy0xNDkgLTQ5MiAxMTMzIDc4OV0vSXRhbGljQW5nbGUgMAovQXNjZW50IDc1M
AovRGVzY2VudCAtMjUwCi9DYXBIZWlnaHQgNzg5Ci9TdGVtViA4MAovRm9udEZpbGUyIDEyIDAgUgo+PgplbmR
vYmoKCjE1IDAgb2JqCjw8L0xlbmd0aCAyOTYvRmlsdGVyL0ZsYXRlRGVjb2RlPj4Kc3RyZWFtCnicXZHLbsQgD
EX3fAXL6WIUyDwrRZGmmUbKog817Qck4EyRGoIIs8jfF5tpK3UBOsb3WtjOqubcWBOyVz+pFgIfjNUe5unqFfA
eLsYymXNtVLhFdKuxcyyL3naZA4yNHaaiYNlbzM3BL3x10lMPdyx78Rq8sRe++qjaGLdX575gBBu4YGXJNQyxz
lPnnrsRMnKtGx3TJizraPkTvC8OeE6xTF9Rk4bZdQp8Zy/ACiFKXtR1ycDqfzm5T5Z+UJ+dj1IZpUJsj2XknHi
/Q94krpC3iWvkHXEukPfEuwPygfiQIx+Tfot8T7yRyKfEpHlIeqpTJaY/nJN3g/yYmN7rpMGaUqR3SQ3eOsFWc
Rc/I+Tq6n0cHy2M5oYTMxb471Ld5NBG5xugzJDCCmVuZHN0cmVhbQplbmRvYmoKCjE2IDAgb2JqCjw8L1R5cGU
vRm9udC9TdWJ0eXBlL1RydWVUeXBlL0Jhc2VGb250L0dBQUFBQStjaG9jb2xldGEKL0ZpcnN0Q2hhciAwCi9MY
XN0Q2hhciAxNgovV2lkdGhzWzQwMCA2NjggMzk5IDE2MiA0MDcgNDAwIDc1NCA0MjMgNTE1IDI3MCAyNDMgNDk
2IDQ4MCAzODcgNDg2IDQ1Nwo1MDcgXQovRm9udERlc2NyaXB0b3IgMTQgMCBSCi9Ub1VuaWNvZGUgMTUgMCBSC
j4+CmVuZG9iagoKMTcgMCBvYmoKPDwvTGVuZ3RoIDE4IDAgUi9GaWx0ZXIvRmxhdGVEZWNvZGUvTGVuZ3RoMSA
xMDAzNj4+CnN0cmVhbQp4nOVaeXAb13l/Ab/b3dxEgBxkARJkOKDwEMUQIAExJsySVGiRPMQw0MCFMkySCwI2LiKg
zQzie3USetwmsRJ0ybjHH8knaaZjJtdiG5zdDJKHGeaiVtPnQwbe5pJ3ZmmVdLU7SSZ1G1F9XtvFyApKUebyV8
FjN3vffu9933f7zveW8rFfElEJvQk4tHEejqam+juIAihlxDC9vXNInlB/xgH9N8D72w8t5H+9onyPyHEtcNP3
Ehtxz+Rbv5LhITfQ8g0lxCjMffeA4MIWb8D8gMJYMzeucUjZLPCuD2RLj52TLfTBuNTML6Qyq5Hv96YcML47TD
uTEcfy/2tZgDD+I9gTDLRtPjXn32Jzv8aQnVv5rKFYhAt3wHRm/R5Li/mEukPR2BM7VuCH4Yv/ZiA1NIxxwsar
U5vMNaYzJZaqw39v/twJu4vuFcBnWvwfRCdQm9FK2gRiagNuVATmkQL6DKaQ0E0hiJoCE2haXQBjaAGNID8qBc
50MMS8knIMSudXAxLD25GJOQZb5S03vDpCOO9I0K+LWGHv7FHwj7ymmTy9kicb3YpfM4TcfdIvC/ZSKSJxbBbm
oj0SIKPTnV73G8Lf9f1VxEXyIVvu34UcXncksYblqY3I+xBJALraXzmq1d6JK1PPo6fBu3k6atXXRKCZXQ+uZ2
xJqosvc9uI8OBHsngI++gSr4GyxCJ75jxEEnofFBCi+EdcSdKKDHkcrsjrh02WlJGVKFRsc7qsrphxRofeYW5Y
/KRgKTzXg0Tct4zHX2EhElsTVmCypmpZlBNdsj5nemoZ4fseJg6D11cmgBJ8I8ypAmRDmCOhWk6vdfodrvI3g7
AAJNmwJpV1TY3E6v1ecieqtxDwrPLLreEI+EdcGjGs+MhOzM7niidoEyhtx7JSsNgB7tt1AFK2O9yYIfePNFHH
j7sCZ3q8IETO79LYXsw5tnRSWQxPOa6CU/qfDfQBJ44cwbPft6K1hG7UuHVML0uhT1rYL3njAtu2HMGkJ9YCpc
RQVPrZ8qYYLhJZF1qElsquup9EnABF7j00FQVkOdOibvNE+RBXciLAuj9qBzgkXe3W0CDglc6Edj1KVRnYLdLQ
K1AOQMyEW5hqTcgndjb9WnQGDB9VrkFe2WX5pbkssqd2CsZg3IHjPRBqcMqO+AZghGyyj3Yu+tlk+Q+4HZ02uw
SGZYcNlmoHR5GsrPLZr8hGFwt7c7h3j5cp9PSbyv29J8aHKDf/lNdnZ7jOm29BxihoLOBr/f0a2xuW4fb5uYeP
VtTP2CpuTJhcIxYzaePdTxqF1Z3HXWPD7XduLL/Ivfa7W3u1O2XOP3SlNHR1+psDAxOjBvs/W5Xy/Qxi9h07Fv
13HJgX5NI7PGf/e/VaR66J/S4Rajrd0JV16BxVIYGCf6FygIgJelDu1rMC2avrKWomAISvydz4KsmKHFWuQag0
AVlo+aWbMZeBB65bR5bCMwN2Txc+/57rl3DW/Cb29/lXt1/EY/e7gJV6NidW/hnnAs5IZ5zqAzt1StrhFtljcn
ovTGh0Rm85ToMvFbh1m4jNtWZvbuNAvIKXiy5A5J1T8ZgQWOwjK1UHiODFyqaqu9wNzgbQgODIQDVc7yrs18Fs
cOtwop/tv84N+v2xT0rDQnfwNPjXQ+dXv1WHbf/JO4a6wysdtf9oORzcBvbXnvm3Pon5vv+q/nEJX/LbAfDaQy
```

```
uzwBOPHKjMmxk3jIPIGFJCEgcM4r9NMwUBwDxzAr36u0umKD6/B/gcy10yiFU1lOfG8FnRB21UnChpkx7ch24V
qfkmwCLuWEx2doIaWSiCQO+1Ncd+OakmVIHLgeVzMHGjwy3xocXnhlzJwfnr3U6FsiJK131S22c651nrnz6fNf
W1PU/ne6Yv9o78Z5eR7h38gN+KxhQixD/LJhpQiVUFqhlmA+Fyhpwrqw31IRCISyZA7IFoj3Q/MXTX3kjiOq9R
knvt0i6mzInvKmR+Jtf/Gr0Xz/P+Ngv85xe4m5aZKMRnhlu8gCXzuj3+/GfwR5KN1C/8pEmmgErt6cGe/iQC9P
/uNfjf/Kp+JcS7/lk7I9/J8Fdvv0ZAFHiFulPiQEeYjj2o7KWxkAHCJbNFEZEYbQGJO2erNHc2jVoUCcUoo1Ci
LQAoYFCCKlRT5O0foDlBh7aaG9dmR8ZMVtMdfgFrt/7lv1/40zXr1kgtmrcfgzY2NFxdF2Nuo5C5ILg2SlCtVS
1g6r2BKSaPdkJEXRa5TZAyxbcxUoLwdZdToNcQHEB2Gu8cpsTDNJBS3C4bHbZxg9T22ye/sOZ2+C08XdH+McPj
lyqMf32WGdk+OKjA7+/2t083eFf7mq84Ka5dm5Zj3PnN56dDeBFXLv/k6XQ9NsGXEuhs0+FnIjlcBwub+VehFq
fQuUa6o2eg5bI4Ro9lBknoDZaZibmCdR1ucZIa6wGG6ApWHf1GmQDH/SBSr17+lnbGoR6x/zH48eO1XPcSBi/O
W52NpzXam+PUp3H4XIHdLahLCpbKXhmAI/jVBRdHFXhgjJmRYD25DaAsM0qOwEnC5DaYNlpoTJOK5S6m6KHbPY
yNrcODw9LTltZb6yjlMUu1QKiLjMgqq+jiDoGBmnJaGtp2HXuejfYOo4HBpV2UFcf1P/D9sRJ/Aexa416Hpf2T
+iC3vWr1/GYp+cpE375p3hxcOZdSwPBIXvD5SenhoZODa1c6Gr4FE0NyI2xO52cDXLRCn0sjMpm6lkTOGRuosa
awaEypsmpoRlioxniZu5Zwaf64C7P8lOyWuVWcMkA1U67mNwE5oMfks0m8eCCk6VqpfA7u2yOu1JibH7s0SdoP
iysjX8QMuHcce/Kyebznte471+8hrMXWDbM77/BfXoleO4dp5rfEjr7RH8zjYsRLq9wJsiFTlTW0L6PQ6ycoP/
THGBBlnUagJM3sAR1eLp0Hgcfcg6GcOvz0afeff0Ln7v++OMPQa79+ze+gWsw/vKXUXVPGYa6MaCJQ72SbiO7O
sTxsKPoKCDGSuukQcYsEbDGQK8GCHVNtZVC6OD73DV8a7+IryobytKtJUVX751bXJr7Rzh1PofK3UjpBGXUTVd
DPRCFWl7tto21lNfooKkWYjneAqp7yJ5NDgDRYpUFum3THkx5DojJKaXr/csnX0C0u1kkg1XS3pSJ4U3JfRMGN
/QGrYP2SS09uHi0Fpt9GO1q9QbiZi0OH6Kh3WEkN0LySkaAs0ONYAB3dqk7lRMirBwAIMJtuOGg33eFKAtf+kS
jkDGZnn8fTkxekcZdOIzT7q5Et32gqfvPH2vRzcw9/dyFbi68HByNWa1zHoHv7Xrq7UPHs31nsq36B1zHh1rMN
X3uaLNr7YHxxOkrmpi3Se1zXA3kcgA9AO9vZQvNhrrQrkE5IPUGyxxNYi9g6GVR8rJyHQ9IjXvyKOA1apUCFDJ
2EAowNEetFDzZ3hkM7uqUQ9EEjEep+3hYPhWAGj7WOzRMD0ZeAxyMtPaWfnYwUsvW1uAMDtYr3mt1alccHMWDo
xzU8Wmsbuo6R5CO2FGKYtSv+6bk4DivUe9pOG/G+PJbM17Hgu9dmH1mavi32BwbNg4vh/3O5msPtTimrPi3PoO
LZsv+SycwN24Y1+nm8KynZ629zoYNrXpTw4SJmzUY9r8y3Nr6kMlS21ZPJmuFypkAf5edZWZQ2UlRqwmxTb2sh
cuuTUBTtKE2BiTnntwAuxKnVH2DVTbBYbFGwaWJFpqNbgkcdb+yOTXQnapyIrR5xuYH5uqdyZOO5dEG10pH/Qp
nCq9ahGnSDUXx6HSjYd47vP89pSaOw771TbCrD30JlXtoTQgQO6GHxk7og5qop/FsAV5LPeW1NNN4BgOSbU8mE
LxqMIlV1kPUuoHstkpNlN0MdHNAboLqCCnV8caHv9quVIfFKpluyu3Cm1LnTRjcgG3V4ZXarTc62juhTOB6UCZ
leEir5YbJ3N7BaqVKsUrp7RtsOOhztEqqJ7p6qJP7VUn/2wy9rSdXWowDZsvCVpt1JTj1AY8xrIktObVnaq2Lm
07dlf6Zr1rwx87VmyAB/Q5BcLSGWk9eO9kkutzLuK+2tqcBeC3TDc0P+5o/adAreCL8sZHab95++cb12rGfcm3
8P1P2K9+JfP/g3fhOJ/8svG0gpK+yYB7/rjtDKHaYc+Rjp+8n3N+gRa4eHUN7aIxrgPt/wrzPoFo8B+MAOsb5U
Bx+x4Eew3+HjNxlkP8o6uXa4FkT8H5E91j0IfQSPoGn8Yfg+wPuOLfDfY3v5Z/hv8z/RPhDTVDzuLZTtcAOJ31
BeaGHMQfv53DTPAOWU26Qe6hq53zVZgxP51WaQzq0qtI8akZXVVoAmU2V1oCWx1VaC/xnVFoPMz6i0jXwXvZRl
TYL30BfV2kLate8H+SwwIEuq+YGo3mgTZoXGK1h/JcZrWX87zFax+gfMlrP6J8x2gCrPqvlVRojq/bjKs0hi/Y
5lebRKe2XVFoAmTdUWoPatXdUWousOo9K65FWF1LpGnQJznMKbTZ8UPd+lbagCzY3o43UHluS0TXUfluJ0SbGf
zejLYz+AKOtQBttip0OoO22TzO6jsl8gdH1QLfYXmR0A+O/zuhGRv+I0U3UR9ubjHZRvXYdo1upjN3J6DZGK36
5mUyA0ccZf5zR7XQd+yyjOxk/wuguoOvsIqNPAq21bzO6h8k/wehhRr93Kpvbzic3EkUS7O0N+ug1xK797DrEr
sP02tfLrkFyfmGVTOdFMZ7NFMk6XPLJtVIxmy/4KXc5miksiRulVDQPgiOkwiML2Xw6mhpha1SYl8R8IZnNkN5
TfUF/MDgwVBWHuYliMVcYCQQK0c1oJhNN+DcyJX82vxHI5bOPiOvFQiCuWhFYSSQLYEo6VyqKecIMA0Yumi+Sb
Lxq8DTw/YTMsId0Lilk48WtaF4cIdvZElmPZkhejCULikciSRZJNBMLZPMknY0l49uUUcrEQEUxIRJQlS7Q9em
A6jgvZsR8NEUWS2up5DqZS66LmYJIomAJ5RQSYoysbTNxag5ZVrWT6SysGi0CEj4iJuF5nmyqyIQqCtTVfASsO
REtUoPzJJujk7rBym2SihYP5vmNRoZJBYoDp2IkmWELJrI58CEBS4FXW8lUiqyJpFQQ46WUj4AkuTyzcuHi6gq
ZXIiQy5NLS5MLK5FRkCwmsvBU3BSVdZLpXCoJy4In+WimuE0Nnj+3NHUB5CfPzMzNrESozdMzKwvnlpfJ9MUlM
kkWJ5dWZqZW5yaXyOLq0uLF5XMQmGVR/GVYxlksALKYWIwmUwXwMwKRK4BJqRhJRDdFiOC6mNwEg6KQEbntXyV
AqWxmgzkGkipoNE/iJJMt+kgBzDIS+NCEhHzc2tqqpmJKWaIQMBonC6CxkBPXk6BBfGxdzCkBTcaV5MqLECAQi
WXXS2kRwrKVSK4nKOKFA7U+mnBETK+B/VUmha+UiaYgvsDOZfN0YTXzqjHOFLPMz8ryvkMPY1nQAb7Q7EsWC2I
qDrkOitmEvFgopYpJQKBqGSwFubCe3WQK1ZT9+QD6Ccu1qs8H+hLZLciTPFi3GU0lYwwASNUsS3EApED92EpsH
7GcpFk/+t9ZMKOgrNbpITwZN7oN1hXFjArqgaXgKSukSrmpyaJMpUVAp9MSpd5kQeVGlJYQTItloXtU9cKISmx
BkVcf+iBJU2JRvFtnoQhrMEfj+Wz6iH7/fVKMJnFKjBcDG7kUPE+nhNPCA8KEMCb0CqNCv3AaLcBrfB6lURSlg
FpD29iMRPQIyqBb8Dt4No+SqAhfEejn+ef5F3iZ/xwvoymQyaFtkEuiDZQACQKvbr3wDSJflQ4dovsP0UOH6OE
q3Qe/AzoI1HmwcxXu06BHhG8ctGaYrnWVovrXUAkoanMB+auyy2B/BjhLQG+ARArGeXXFkUNrVuTIEUxGDtlxt
+QloKmuJLOBgNQpJueHXxANgHf3rq7oTTAsczAegZe0ANyjcNSiEhm4JmCFDaBKcKeWbIBEDu5ZiIsIHhdBPgA
YHMUigFZgZpJ5QFFJw5wSi1geOAeIKRI5hgIdZ+HZvQhPq/J+4BF4HTqYWdFLYETnFtEWW0tkWG0Dr8QsiDJMK
```

```
D/GZh6OkchWo2tSqRjYnmVWpuFOpeOwTkWixCQUL4rgocjuIotQoWp/5UnFj/MwyjApGkWCFmGdNaCSYBn9QzW
9iywiIrNCwaQiU2CrxYC3xiyprD5d9X35Lt8JQ0yxNQp8JSdoFovMj4TqweZdORO6x4OjtvnYczrzBFu3gnCe8
XNVTd0qltTaFJO8nz4/MsL3IE/uzor7RSrGnmYOWZhgmkWVE1Xn0+sW3FMM7zX2vMR8iLOq8zFuicldhoxaQRf
QRYjVCozpPzVGGH8S6nSJjVeAM6quWWQ6lbki8+mwPUmW6zRuirVKTPIMjyJDREF4Hp2DtadAr7L+JDoDdswxW
yJVnKfZeAFkl+FLxxdhFrWRZhG1bwUkpsDyOTai3FW4L4LcMsxSKmZZzZNfLy/jh+pCyTJaCUWYS3EuqPGMqDV
XUFFKMRQSrKeIag2uszzcVBGKqj0idwidX7eCUiyXNg5FTFnzaKZV+gnVSPeXIsuLgoqWkT0lzHqlQyr9cYt97
+2KqSNW0K5I8ZhkGqNs3RzzPan6IKLHmHzuSIUmmTUHnYviFVW7VJRhngVuCaIgqtWyxfxaZz6WVN3389ZX7XB
Ud5pVRey+kpXsK7EdIKXWryKdY88qFh/teffWcYbtgQfxvNt638+ZSeUUP5S4VHpfku03dN+Pq329pMa8oiHP5
pWY1Uk1B+7FTLFK6Qs09zYPeXi0y/5fMpDm1UFfuzfO9/OPVsuW2k/yKnabTFdS7eLioa6aPdTFlQwpVONB82H
7F2BOK/jgfPSbxGDmSC4f3U/vn58HslEmRbErsjWPZur9MFVierAj3b27He0sh7VWdoKK9souWolNVvVyQ41Dr
Kotxq4U+3v9VZ5V1thSd/J7Z/rUTppi3VT8pX4WWM8tshquRDTOTmTpX+C//1fsYpVOnGJ7JT3JbbD9TJmfBur
wifSAVjL1fs9eZyd25eSVZTmUZjmt+Bw7MucijNOsQ2ZYt7sCNm6jR/EZGN1fb0o4JvQJs8J5eJuYFYaPaKW5+
nNmqdiJrEPAKVcYhzeRi/AucomtoZyAYyx6m9Ra7IDzbuHIW0iJYVtAm8d+CONJtsuXWDyihiHDAwY//E4Y5g0
B5W+Zyl9yF+Ewfp/P5/Gdd0v4vWhW0i+GZYzfF5Gn6f/hI1kjs1LdEhBPRlolrfdqGEkCOSfx3rNY0iiEpCXnv
og0T9xBnFfmpze9ZyNSnReh/wGKk3dsCmVuZHN0cmVhbQplbmRvYmoKCjE4IDAgb2JqCjU0OTEKZW5kb2JqCgo
xOSAwIG9iago8PC9UeXBlL0ZvbnREZXNjcmlwdG9yL0ZvbnROYW1lL0RBQUFBQStGcmVlU2FucwovRmxhZ3MgN
AovRm9udEJCb3hbLTExNjYgLTYzOCAyMjU5IDEwNTBdL0l0YWxpY0FuZ2xlIDAKL0FzY2VudCA5MDAKL0Rlc2N
lbnQgLTIwMAovQ2FwSGVpZ2h0IDEwNTAKL1N0ZW1WIDgwCi9Gb250RmlsZTIgMTcgMCBSCj4+CmVuZG9iagoKM
jAgMCBvYmoKPDwvTGVuZ3RoIDMwMi9GaWx0ZXIvRmxhdGVEZWNvZGU+PgpzdHJlYW0KeJxdkctqwzAQRff6Ci3
TRbBsJ3EKxpA6MXjRB3X6AY40TgW1LGRl4b+vZpS20IXEGebOZR5J3R5bo33y5ibZgeeDNsrBPN2cBH6BqzYsz
bjS0t8j+uXYW5aE2m6ZPYytGaayZMl7yM3eLXx1UNMFHljy6hQ4ba589VF3Ie5u1n7BCMZzwaqKKxiCz3NvX/o
REqpatyqktV/WoeRPcF4s8IziNLYiJwWz7SW43lyBlUJUvGyaioFR/3JpEUsug/zsXZCmQSrEZl8Fzoh3W+Q8c
o28idwgb4kzgbwj3hbIBXGRIe+jfoP8SJynyIfIpHmKevKpI1MPx9jPDvkU/XPkJnqiTyoin5Bj/0VOw96nwrH
xLj/r5PLmXFglHY92iNvTBvjvge1ksYzeN+4VkxgKZW5kc3RyZWFtCmVuZG9iagoKMjEgMCBvYmoKPDwvVHlwZ
S9Gb250L1N1YnR5cGUvVHJ1ZVR5cGUvQmFzZUZvbnQvREFBQUFBK0ZyZWVTYW5zCi9GaXJzdENoYXIgMAovTGF
zdENoYXIgMTcKL1dpZHRoc1s4MDAgNzIxIDUzMyAyMTQgNTM0IDI1MCA5MzcgMzMyIDU1OSA1NTYgNTU2IDU1O
SA0NzYgNTk5IDY2NyA1NDMKNTM3IDQ5MyBdCi9Gb250RGVzY3JpcHRvciAxOSAwIFIKL1RvVW5pY29kZSAyMCA
wIFIKPj4KZW5kb2JqCgoyMiAwIG9iago8PC9MZW5ndGggMjMgMCBSL0ZpbHRlci9GbGF0ZURlY29kZS9MZW5nd
GgxIDIyMjY4Pj4Kc3RyZWFtCnic7XsLfBTVufiZ+Wb2cTZZdkMWCOQxCQYILkkM8lbZPDYQCElMlkC0tNlkN2Q
hya67mwREDOoFxNKLVQlqrVKLqGgtWqoBqdWWahW47b9i/7W9Xmulf+u/kXJ7aW+LMNzvnJnNbkKgCPi4v3uzZ
OfMme9879d8QDTc6SdJZB0B4mpu94Zc184sIoQcIkRIae6KKrPHtP4I17/FX7UltLy9u7vIRwjgPXlqedvqlsd
mvtNFiLSZkBGmVr/X17t78wxCRt6Hz6e34sZUtcOI94iPXNHaHl01x/DzL+H9cbz3tAWbveqEX2cQklqG98527
6rQO+IiCe9DeK90eNv9nT8MV+L93YRc814oGIneQG47Q8j1UfY8FPaHWo5Lb+M90pNbcU/AD/tJwqWB3YsgyQa
jyUwtScnWETZ7yshU8j/pR5hG+shB/LxCdpGHhJ1414LbN+HOdvE5sp504s6PhYPCJnEK7u0kx8mbCLmRHIRdE
hEWkKm4S8jbskhOCB6yB3HMElKFWUaDRKQqaY9UK/VJH0iHyQwpIh2WGqWIMBUelevlnfg7C34ippDXSRbpE94
lEbIPPoSpsF8qk6zkXTgMu8jvkQraG2lsITvIGuQlVQiSHnGNWIs7r8mHyYP4CeLzw8LDwpvI3T7hDvIWuR8kc
T55WHgL5TpI/kruAI/Yg345VWxB/l9DXIfx/IMkIhH5LYESVbwS95B7pNXEvzNgivwW/xwnPUjZQ3YY+gypxvF
IhWlsp/Bjod9wL9lO3oQvwU3wG2G9NF56QppPtmgagEayBXE/yM4YWoTVKDv7rGHYxW6pUdhFPpQajU2I+ydMI
qS5R6xFiVrIfvztNthQpjnCetiEnLKnGeSwcYFUgOcRg3EtSk1IEKaRFbhaQ54hz5Ep0Eu2ICYur2GG/Fc8+ZD
0Hsq8Rfia+FdyGMpIHmmRjqGuCfp5LyEvGA2yBKJAnIptt5hb4dvtun6p8tOG7CnOIbeKzajsJjW7k1crfWfO1
CyVxskNu+X03ZBr2i3ljn/vXA/fm+JcWLNU2X3aXaZjdTeW4V7dUlyyO9zGfXcZf8aI7pZz8U9F426luVW5y3b
X+Nl32fyzp6DaSIvaK7XIOzAbGclYV5L0MTF8LJjkHlEiBQeO9F9FbEf6j/QXjrRn23Oz7dktEjkVgXGnfq/2G
q1/+3PYkIfx/zrmgjnyW8RCnK5U01bxOxK5jRqMkC7PNAvpREqyneovwj9XkYJrjuKi8LmaZEFYNnIqIp1qH4/
f419/Q/y3N944nfOG/Nbph0TfySvF107PYvzddOZ9qQdtnErGkZDrCuIQzBtMd8qOJwV5b5Lw4pi9KX1Jm9PHO
USTw0QWiikj3OlI7UT/AXvKLGT+6In+o7Zj+DlxDDcKXXlzM0IZ2zN+nnE8Q55L5gpzxbmOueNkp7HAVGB20iA
JCkEx6AiOMy+7SVgmOLIzhalF02c4rMJ4hdhtZGoRMeYL43MMRqnn1HNJh19Y8VpT889XqifU14S8U78TjH3iY
3c+uNcqfvmGl167+upnJjuFmQIVRgql6jsHtu155mGWMV/Br9WGVNR5zgtkmyiYSLmETKN6+gtdyTbZJdfIjXJ
IPi4buJbGv9KHgfL3/pg+OlEfFjKaLHCNNOxNIXuT+lI2jzGnjLgeUhzuMVx8TXTbsULX+Llpa8gaQ4+xx9Rj7
qE9ljVJPck91p4RPbYe+5qU7WnH0+woaY7BkToKRZ129YSJRUzS8TnsW4zc9/RTW+99+ul7jwsp6rHj/67+SbD
Dux+8/voHf/jpax8+pP5U7Vc/QuFnoYypwkyWAAnWFuNO9IckYY2rQh5rkM3ULI2lZhhLLVQcK4gWCzXYjSajb
Jdkk8ko2kFMQmg7lpESKotgAPJdiynJQs0mLZgsRpJsO3Jo9CzmQv1FRaNnFaK7G23yR0abSf+V41dt2ZDzHLG
```

```
in7n+bBANskgJflLoJPkKqtDrxOvkq2khrRQXySXURRvEFeJKeTltpGvEHvEWuUdeR3vFrXKGkZhFExDJIBOZG
ASjZCImo5mYJUqTiHUsOCSHKS3JZlWkbFkxKEbFNN58Bc21KFbFeo04G6ZJU+VC03TzLMvcpEJrOSkXFoguyY0
2LjGUGEtMLpPLXEYXJbmsLutSsd7UkFRjbRGXg1dqkhsNjcZGk8/soz5LN+kS1oiroFuKyqsNq43dppBpVVJPU
o91g7gR7pQ2yevNd1m2WLdJj1i/a71xGUHHMQvsjzDeLIwvOyTMFma9z74Oq5tU9Sfqj1T5rY9TpGPs9+SVsu3
kcW67fZjTF6B/2UmhK9WAerdbYKO1z7zfSA0oenmK7dQBHsnoXgeOHGJBtadm5CMjRfQhHhgODOipjuswbEaNh
gVZFc6HHt+7d/a+9SPz02FPiv3gS6efkxqfaWmWZRLz5xswFmwkncx0pY3dS6ype2XTZmufsA1GS8QkzrOnWNw
Z3KuLGOH+oycO9NsOHCt8vjFzXeb2TEDKLI1MwxjNEXmMjhJivowMwaN9fbOfveXgGXLm4C3Pnn7t8XvueeKJe
+55HJ4Xv/z3/id8XqFMMOGnzKs6Dn7wwUH8RT28hz3WDejDlLhcjhLYLonb5duMZLvZlGVIB5IlWGxHFu4e4Vm
6F4FdMxv6D2gBXHSCpUt0zoYcYc8IGCGJy2Zk2+VpuZjsHNmqsEB9QPC/ISw4tWOXFJnfN//kW7u43rFySAuwM
qaTh10T08aOgzHpdlkidlmWSmzfst+XvD316xJ2EMRGRYGmj7aBAXWycLfDs3D3KM+NC3enem5EXuDMyzMbMG+
//DLPf5yfE/0DwSJ/JOxOt9lHz0LuXEWLpXq53nizdLPcNW5jmhH7izRpLBaa9CjpMnSOjYyLpt9ONqTdPvb2c
benP0GeGGdHt8pFMaZNJzOuEzT1GqcxW0uOVIPRQLCpeeVUZUT42lTvosc3fOXNVTcfWfoHIdV9Y5p6YteuXd3
C12e3b6vo7i0pPXRV0R9+9KXHQhnqH7kf7FPrpR3od8wPrnONi/lBn3Wz8APYn4E+MI97QznzhKIiLcMdjbmCy
6z5wm8zJWFZ7oAHYgYTx+dMEDijWn4TIuiNz645RM6cObTmWXEmesPj7PeJ088Y6C6fV92v/g0/+73CH2POwO0
zB+2DHQexCKWuctmOwkp2kIzsIkuCKIBdxDRlR0hqN1OBXSwUU5zZjgmuhBolATPHDzELaSvRZEhiDkQ9zInQf
Db2Zec2NDB/YgZla82eB+yx1HeuzDf4npn3fipJdKzkoBPotdJVdLG0xLiUttAu4WapyxilX5Nupw9Ij0jbjPf
Qu+lO4Unpu9Jjxm/T7TSdYguPOduC2U12mMda8mCCnGuebFGSMX3ADPlqI8tnhckVUC67zQssruQGUi80iA2wR
K43NBjrTfXmBktNcjB5ldCT/A3hPuNTwg7j7uSfJb+bfCa5AJ3ILLKkxPOT5FNXCrveVvep+94WvqeG3xbyhDy
p8fS7p18R+tT54gJxlHqTsIX5COszdmNMmjA75bscZKv5NmGrzSTaKJHTkotIullK4XnCPgt/mPej6z/XOFLgG
YLX7+zcbH7NE4R7TwjThCz1PfWgWiI8Ijwn9Kqtao3qlQs+7hbGYDJzCqN3qtvUdeqtai/z0TO/Uet53baQEaT
MlW4RjcT6UpJxo/wDsj/pWZvJJhuqkwVTEim32U6hrY7O4j1IP6/EGI/Yf1hsdpe9xt5oD9llXttTDVcK01iuQ
pfFOv/t75dfFajs65MaP978y5cf8j5gmPQhj481mCenYHxQkkv2Y4bIsow2W8mTow17rXZlQ9a+9L3j++ybRye
R0TAm2WyyZIEp1T0BtXHoCIaLnavjwNETp5CVV7Efss+yz2L9UEdhRmFmYVahUphdmDN3oivDlenKcimubFdOT
UZNZk1WjVKTXZNTMzE0cX3GxsyNWRuVjdnrc+6euH3i8YmZsaOxQ7EDjZmNWY1KY3YoM5QVUkLZ6zLXZa1T1mW
PWZbQaVwrzNCT9oRpV0+fmp2YTkaJL7379G3BB/b29c3df+fTB09/LIiPb2t83uN/6Yb/OC5ObVnTFHl7T17l6
dt2tXhfefQHL6f0fDU/f9fEiadYrNrYuyn6iYG84JqI8WmQMTrBwC4S4HPRjvFZApLwXWLA7oLIsmC0HTkwWku
YwwYY9hI1JtZLNBGYDPOkefINsBbuAMx5RtEkmQ0OIVUcK42VJ5MJwgQxT8qTcw2KaSaZKkwVr5GukWcY5hO34
BYrpAp5ngHjxdAiBqSAfDPW9S5xtbRa7jSsM91PthnyMD6ytcotLjj96pvC28Kvf3H6NazWo6UPT16JvvAM+sJ
S9IWR2BXPcaWDmYBVMGy02jdTYX9qXxLaP9VSJWJb6WDhMEur10eLMFnaUzCB7Ak6XnZgwR6JZVMrlAP1c4LwT
N8tt2z9Tl9f6bOdr/xU+JmwT9x52vvIIy/tENd8vP3plubj8ATT8Ubk4VGsVxPJra5rkpNEq0XMzMo0mUUjFbO
yMkuoJTNLcgjE8a3U+8ZstUtbyX25X7dvnpRJLVnjjCRnXJp1ijEtNWeS7V8P9J/qP8rCY5bWsR/FaPnrMdur9
lius36ES/2ChhiBwbvs+ay8grzqPMAeXWs7UkdlCZmCI1XKzpkwcRpv29GjCjCOp119BXqUND9y6CuPfa97582
/+7/qO+oHK/60bk1/+Dv7Nz645ndvCKP/Evi1vOMnM6av62r2Z6Vd+fbzb/+2sODn7vI7b+24JWvMlJefevXoB
L0/knxc99NdaXHN9yXtpwJqvIrFXnkqq1GsROH7zoDSGx0/Y0oXsDpp9SjeKEk+pvSn9+4t+V7nK6+KO05/SXz
4kYdf2nF6I7ZKft+fkG7wzPvwGtKdSD7Q9V2nq7surm6yQdgkpW5wbBqz1y7tze2L67t2nMlqNKXmuCex/u3II
H3329S/HLP95VjKYH3H1E0mMr9vT6fplvSkfHxJclqcSXPMc+gcy5wki0IU4QpxEp1kmTyyILXAMXnUpMxJWXl
KXvYVEzfQDZYNSRuSU1gFFUUDNVggCZLBCiPABmkwFsZBupRhnliQNzfvK3k9eevy7s7bnnc8bwyGwE1DDWsYf
7Zhp6Py4KtVT9ywaVPTfXMPPPafv7rhx20tr3pv3+x/yvXU/b/9l5Y90txnJk3yeFwV2dbJD2x66Pnx41+aNq3
h+oU1uSOu2Hr7w09navUdc2zDA3ev/+j0V0Zc8xeSZeKznF/8KnlrbK7Dsr9xp4ENs0zxYQ++6bSrGYnjnyHjI
Id0mLQYJPK61E9uEmeRV6Tf4/WreO5DUgYLyD45ldwkjybv4W8vrvdJ+8kcfKt+XZx15jfSB2SNvJrY5BTyDD7
biNd90pV8TpFDatAXfyVMFiaLJtElbhYPQBncJ9VLvdIReYX8a0O64X7DEeM/cY4cZBLmDcIriY08wCSQHOIov
LJZ0FjhugG+7x+QQcA6d7++FhHu2/oacP9xfS3h+jl9LZMk8gN9bcA69bq+ZtX6TX1tIRnkqL5OTvmmEJseWsn
VIx/W1zZiGfkLfW0n0sh3kKIgmZGhwpHv6WuBjHLY9bVITI6J+hpwv0BfS7h262uZjHHcqK8NJNUR0dcmkuPYq
K8tZLbjSX2dnDvb8b6+tpLWORn62kZGzVmvr+3ENOcbpcHQ6nBgeWtUmdScpxQVFk5VmlYrJYFoJBr2e9udSkV
Hc75S3Nam1DKoiFLrj/jDXX5fPj3r6HR21OPtal8R7FiulHhbz3GwzL/CW9+pNLd6O5b7I4o37FcCHUqos6kt0
Kz4gu3eQEcMps7bESkJBlcm3CYs6/3hSCDYoRTlT52mbScAtAQ7kGoUhWiNRkOzCwp8uN/VmR8Jdoab/S3B8HJ
/foc/Ws7BGA9MigHBlUkRv19p8rcFu/PylQvgOF+Z17Y61BpRAu2hYDjq9ykt4WC7Uhz2d+msxGhwDXVqGkokQ
2mcOkrmVTTWBtRMp5z3h55tkAu2pTKEciBCvUo07PX5273hlUqwZSgWSmv84fZAhKs/EFFa/WE/0loe9nag6E6
UHcXCY6gx1LNTiQYVb8dqJYQGwwPBpihqLIAq8CrNyDRFyGirP6an5uZgewjBGUC0FbGjlv0dEdReDldJTh4i8
yneSCTYHPAiPeoLNne2+zui3ijjpyXQhkaaxDDyA0pdsCXajerPyeOchP2hcNDX2eznaHwBFCzQ1Bn1Mx7ooAN
ONHNzW6ePcdIdiLYGO6PITHtAJ8QohDVVItrOCMIzcZxKu59JTbmDRFqdCTScjGZBMKxE/GgHhA4gq7r4Q0gz5
```

```
hBtiCk6SjXVcULdrehYZx1gZmjpDHcgQT8/6AsqkaBTiXQ2rfA3R9kOk68l2IbOxgRqDnb4AkyOyGxKPYjO2xT
s8nMJNC/iDAw4QUcwimaIaLvMKqG4B2jPlEirt62NNvl1rSEbGCXeQXIGO9Avwkp7MOwfVmwlujrkb/EioXyNq
cFP272rMVrwuC/QEmCO5m2LouvhApF6fT4uuaY6FqDeMPLV2eYNU0bI548ElndwNpZrsYqHmId6mxFJhJ2I8RM
ZSomhpEiAK8zbNjwC/UyMjzg2ZK+jbbUSSHBzysQJ+9nfWHFYtogwRTK7xMLDjz7nD/ND3cGwL6LkDMRhDqMde
0BzWNjmcJWhZSr1eGnyYyQxrJ1oA6aTrmBggDH/qihGjOINhTC8vE1tfvZAkx0xswWNG6XVG1VavRHE6O8YpBP
mdXHv9imdHT6d4TirlDOnSXg+q0aCbSyqudmYkbxKG8seGCsxwJC3eaV3OQqGcdgRpMxVP5lTDSKFCQtZ9Le1M
Kbmu5Xy6iqPUldd7llSXOtWKuqUmtrq+ooyd5mSU1yH9zlOZUmFZ371Yo+CELXFVZ4GpbpcKa5qUBZWVJU5Fff
Smlp3XR2trlUqFtVUVrhxr6KqtHJxWUXVPKUEz1VVe5TKikUVHkTqqeZHdVQV7jqGbJG7tnQ+3haXVFRWeBqct
LzCU4U4kblapVipKa71VJQuriyuVWoW19ZU17kRRxmiraqoKq9FKu5FbhQCEZVW1zTUVsyb73HiIQ9uOqmntrj
Mvai4dqFTQWTVKHKtwkHykUvEobjr2eG6+cWVlUpJhafOU+suXsRgmXbmVVUvctPy6sVVZcWeiuoqpcSNohSXV
Lo13lCU0sriikVOpax4UfE8Jk6MCAPTxImrg7ID89xV7triSqdSV+MurWAL1GNFrbvUwyFR96iJSs5uaXVVnfv
6xbiBcDESTrpkvpuTQAGK8U8p54yLX4XiMjye6lrPACtLKurcTqW4tqKOWaS8thrZZfasLucesBj1yYxXpfPLb
MT2zvYOhGKndQHL3MWViLCOsYEbdBAsepd7VbM/FGW+rQe3lhp5GtVyp5N7rZYE0IXndWDgant8iWUJI4tXHS2
7xQs2K8dOLfXy9IHejZVIS72+Lj9mwAhLJcEwDbJk0h2I8EjHEtge1GqeEvG2ITE8xaKIQ2Gu9LbhscgAm4MCi
saKYSgcwCPd4UAUk4ni7cTdcOBmvQyH9TLFJVDiEjAq8eSg8R/2R0JYpQJd/rbV+QgbZrWMcxLowF6tXRedq68
5OjvWKkSV5Ry5Lxil2NHlK5TyjuuSW6cL7WUvTx9EtT5IuZg+iMb7IOUi+yB6dh+kJ/lmjikSqxnDNKjxhoVeS
q+kxHol+sXolahmh0+tV6JawF5Sr0QvY69E472ScpG9Eh3UF1xEr0TP1SspF94r0YReKTF8B7VLWM8xSVyudon
q7ZJySe0SHcQuf2+83C0T7Qgql9wy0cvaMlG9ZVIuvmWiQ1sm5WJaJjpsy6R8kpaJeorrFy2oZmwXz7+o7ojGJ
b+U7ojGuiPlUrojmtgdKRfVHdFhuyPlUroj5qyDAmWg8aHnbHyUT9D40PM3PsoFND6UNz6De4d/3NBEY/Au3jT
QfLzkn3dyVdAdWBkoCGAGWZUfag0V6GlsyOSMlJIgCZHVJEwCZDlpJVGikEmkmeThtYgU4mcqrpoQQiElCBMlE
fwNEz/xknbixN0K0oHw+bgqJm34UUjtAK4Iv/Pj1Y9nuvDbh5D0AqhOH6DqQUpdSIv9s7UOhGZ8ePHMJ6NYhqs
VeK6edCJEM8J6OTY/P+HlEimIpQO/QwjThHgDCKfg+SBS9/JnQ/HUcSwR5CiIn5XneDr8bj3nMIJ4g5xqEfI5l
UwbBD08hhZ+QpM1qluCyR5FzmeTAvz4dPguhM9HuCBewyiNn58Nc7nzEYcfz5QnYIvpIWaLsy3OnjHd+rl9/Ki
lIOlGWGaNy6NjhmkePlmNMK38ZACfhTjfUW5PpoEwP8E8gGHtGqKVoXLEfahzkA+dSxr2j6mGk12zmRdXiVo72
5spmXIJH3pBEXL543J4e8dlDuATyldRvsO8rJ3reiXuBdEC/4gXJlkNx9fOscW9P8B5auXP/LpcyzmVDt3qTt3
umrU0apqPaf7s5HwFufU7+PmQHmEahSBijeo+FtC9wMtxaJqmOs4o52KoPzVzOOaHGvYYBgat8a75sp/Hq+Z7O
QleksMtx876+DXC+WrGM15dPsqjoBk9tJ1jifInMf204KpNj6RJAzzGKbC8wviPov9q3s8oxnXCdkI8anxIoZm
fjnHj4xJEua814dMof6rRoOeh4NSjuRk56+RYNJ10cx9o5Vknqmumne8lShSTITzIKzVuO7kOnQnWYet2bk/N1
jQhg0TwtPMccjgH5CzgGUThmLV40HAHdK0Otv75pY5pTuM2NODRUc5X3OviEnVzfbRfEIVYNLTwrN2hS+hPoOj
j34yGk1+ZJlYgRDPHp8HE7Mf8uE3PbDELNXPaPs5xQOd0No9Oj86dFzEGeWaI2yAxF8U1cHYm6ED4qB4NkUGws
ViJaywxBySeU7jMXs455bl5sK9p2tBqifc89gzyKqfotm/n13j+uBBbRHklYpXTq0uUP0hT5zvLdLJary0adab
zFs6jT/ekNu6n4YEdjVOmU1+CzRO9LlZBvbwiBnjOaON3dEAiH+eU2asjQRvLB9VVjVIsh3q592i+G6MxVD+Rf
yhTjEuqSxD3MC+30YVzMJjOUH0Mx5tTt3cbPxc4RzanA9YJ8zzr5Xkljje2ExnwyFi8DK0efj3P+bkUMUrdXCo
fP58zTD3MGZB76AmKz2LVNifBy7SYqRxSX5p4vAcTeO3U4yDmJ134NDCMxvxkFddzhx7JIfxo1cvLM6p/4ESi3
TWeYzt02Ehp5Rle4deIzqOfe9K5/CSW64bL3T5eCTq43RP1NZxWaYLmEm14sbEa4VkzVqvj0RaLJNY5tA30HmH
9xGCMIe7RK/F7uW4xrR4yr6IDWfXTzFTnlqpJj5GoXg9bBjQ1n7g5nWpShXeMTjXeecgS7CNr+bMK3FOwj6vFJ
/V4V4a7ZdwuxfwJe57Do3EJrhnGarKY49Jw1OI3w92AOwy3wu/Z3UKEr0Jc7KybLOU03IitDjmrxjXDvQh3K/H
q1uHYiVLcWYz3bD2PsC5Uo1eFpzw8dtg5xovGqQf341QHc1XBKcY4W4R3tYh/vv60GHFXcHyMfyfvj9i6SudT0
1wtx850xDAznKXIUSW/Y7uL8VqDcHVcn8VcZo3bKi5DOT7XZHFzDjRLaByV4rUGaTOIeciXh2uBUfLokE5uRyZ
PGT/PqC7kUBpn1bqV2TqOJV/XpcYH03/9AOU6Ln8lfhQuvwd3PNw2xYg/hjfmO/M4BsY35dpYzOUr5nqo5hRKO
BzTItNn5YDH1SZYpZTri9mNcV7GKRVzjdQNK0kMW6J1hvMOOkBhHpfPzTVVyaHrUI9uhK8Y2NH8sYLLWqrrWsO
p+b3mE5UJ2i3lMjLLXo9U3bpPFXPdDZaC2WkJ5z8uhWaBYv27NEFncetX6daN8ePhlD3DaGUJj0U3hyrmtq4bi
JFyHr+LdM4XD3hYPAcs1v2zeoCzwfqNxVEM7kJyh4YrRnuwBcu4P1XqHNYNaEODoOfBq+UuN9a1Zv6eEx3I24M
rd2LXGO9GE/tOZ0KuTewEtCw8j8O2D4GL72pvS1rNir/rJPZuw71hx96OtV4+1vXGuw8td2vvRIldr4/351oPG
BnoSoK8DwwOdCbd/Gm8pof02Ulw0Hseo+zltd85QCtWi+K4tL7Sy7sFRi0yjDbPXaHoWW+GIV7vNSrdfB3VOxM
mX6cOy/ZvHvI2HJv/nG0DZVgbxGQZrnNI1H+Y2zukv0sFuIZZP5mv4w2T2HtZXCdMA9pcrX2I1ePex7DNJkOnC
kwHyxM493FdU6LN6BhNyvNVbMb1+U+dLvdc9os0D6KD5kFDO69Pbx5Eh50HKZ/xPIhe0DxocCffnMBTfNYRg7y
wCepwExb6uc2VlLPmSvR/50oJc6X4hOG/51yJDqqwn99ciQ7ztvZFmCvRYedKcYk+m7kSPc+84LOZK1HySedK8
b91upxzpXi8DZ4rnav6nnu6pL2fa53EF226RMng6dLw043PZrpEz6NdJUGDX+wpE+U+dnY389lPmegXeMpEh0y
Z4u+6n+WUif7DKZPymU2Z6CeYMimf2pSJch3UI9YFnFtN28X4/LObHdFhbf55zY7oWbMj5XObHdFzzo7iM6BPf
3ZEP8Hs6Hx4P93ZUSyznruinD3xoRcx8Umc0lzOiQ+9pInP2e9sFzfxoQkTn/PNHS7HhCZ6Fn4XiU8aKKfD7vI
```

```
v4d9cFXC9rMTfAs6bj3dN+bx/DeHe4G7s/P/mjP+fZe3/cN9KbiDD/PSJ61xnPlbhZCr8PRf+VgT/2Qt/tcJfV
Dihwn/kwp+t8O+9cDwX/nRXsfwnFY71wke90H8S/ngS/r8KH86GP5TAByr8vyL4/dE6+fe9cBQBj9bB+78rkN8
/Cb8rgPdU+K0K7xbBv6XCO73wryr8JgV+vRbefhF+pcIvEfyXa+GtI/Pkt9bCkXnw5i/GyW+q8Itx8H9U+LkKP
1PhX1Q43AuHDmbKh1Q4mAlvFMHrKry63i6/mg4/GQUHVPixCj9S4RUVXlbhhyq8pMIPVNivwosq7LPD3g258l4
V+l54Ue5T4YXnl8kvvAgvrJOe/36u/Pwy1xl43iV9Pxf2qPC9XnhOhWdV2K3Cd1V4xgffscLTT+XKT/vgqV0p8
lO5sCsFnkSmnzwJT6jwuAo7VXgsBXao8O1HrfK3i+BRK3zLB9sRZHsvPKLCw99Mkh9W4ZtJ8NA30uSHfPCNB23
yN9LgQRs8QOF+Fbb1JsvbVOhNhq14aGsv3HevVb5vEtxrhXtOwtfvflH+ugp3b1km3/0i3L1O2vLPufKWZbDFJ
f1zLnxNhc1fzZc3q/DVfLgLxbyrGDbdaZE3pcKdFtiIGxt9sAE1tSEX1tvhn1S443a7fIcKt9vhNhXWqdCjguv
MrWvXyreqsHYt3OKDNR6HvCYXblZhtQqrrNCdBF0UOlWInoTISQifhJtOQkiFoAodKrRlw0oVVthL5BV1EFChd
S0sx5sWFfwq+FRoVqFJBe9saDwJX06CZSrcqMINKjQspXLDSVhKYcmoNHlJEdSrsBgpLy4BjwPqBJtcNwZqU+H
6BSPl61WosUC1ClWLbHKVCotsUKnCQnyyUIUFFTZ5wUioyEiWK2wwPxnmqVDeC+5eKFOhVJwil56EkheheCG4V
JirwnXXpsjXpcK114yQr02Ba+Yky9e4zoyAOckwW4VZKsyckSrPPAkzptvkGakwfZpFnm6DaRa4OhOmJkPRVRa
5SIWrLFBYYJELk6HAAvlTzHK+DaaYwVkEV07Ola/0weS8FHlyLuSlwKSJufKkYpiYCxNyLfKEEZBrgStUGK9Cz
gjIRjmzU0DxQdZJyEQRMn2QkQzpqMF0FcadhLElkIY3aSqM8cFo1NRoFUbhoVFp4FAhVYWRKqQgQIoKdpTVXgK
2tTDCB1YVkpNGyckqJCF00iiwqEBtYFbBhGAmFYypYPCBhA8l9AAH4C6oIOK9OAUEGxAVhD7Bt/5rwpX/HX7I5
83AeX8y/gvbpQKHCmVuZHN0cmVhbQplbmRvYmoKCjIzIDAgb2JqCjkwMjMKZW5kb2JqCgoyNCAwIG9iago8PC9
UeXBlL0ZvbnREZXNjcmlwdG9yL0ZvbnROYW1lL0NBQUFBQStEZWphVnVTYW5zCi9GbGFncyA0Ci9Gb250QkJve
FstMTAyMCAtNDYyIDE3OTIgMTIzMl0vSXRhbGljQW5nbGUgMAovQXNjZW50IDkyOAovRGVzY2VudCAtMjM1Ci9
DYXBIZWlnaHQgMTIzMgovU3RlbVYgODAKL0ZvbnRGaWxlMiAyMiAwIFIKPj4KZW5kb2JqCgoyNSAwIG9iago8P
C9MZW5ndGggMzE2L0ZpbHRlci9GbGF0ZURlY29kZT4+CnN0cmVhbQp4nF2STW6DMBCF95zCy3QRYSCEVkJIKQk
Si/6otAcg9pBaKsYyZMHt65lJW6kL0Dea98aPMXHdHltrlvjVT6qDRQzGag/zdPUKxBkuxkZJKrRRy62itxp7F
8XB263zAmNrh6kso/gt9ObFr2Jz0NMZ7qL4xWvwxl7E5qPuQt1dnfuCEewiZFRVQsMQ5jz17rkfISbXttWhbZZ
1Gyx/gvfVgUipTjiKmjTMrlfge3uBqJSyEmXTVBFY/a+XSracB/XZ+yBNglTK3X0VOCXe58gZc428Y26Qc+JUI
u+J8wK5IC5S5HvW75AfiLME+cBMmkfW05yamTIcOQ95TzzngNww45xE8rl7ZM5fYOaE8+cZ8i3/CZnzFxkt5Pb
luBq8u5+VC3X1PqybLpj2jBs2FsTvT+AmhzZ6vgFvv5oECmVuZHN0cmVhbQplbmRvYmoKCjI2IDAgb2JqCjw8L
1R5cGUvRm9udC9TdWJ0eXBlL1RydWVUeXBlL0Jhc2VGb250L0NBQUFBQStEZWphVnVTYW5zCi9GaXJzdENoYXI
gMAovTGFzdENoYXIgMjAKL1dpZHRoc1s2MDAgNzUxIDYxNSAyNzcgNjExIDMxNyA5ODggNDExIDYzNCA2MzYgN
jM2IDYzNCA1OTEgNzcwIDI3NyA2MTIKNjg0IDYzMyA2MzQgNjMzIDUyMCBdCi9Gb250RGVzY3JpcHRvciAyNCA
wIFIKL1RvVW5pY29kZSAyNSAwIFIKPj4KZW5kb2JqCgoyNyAwIG9iago8PC9MZW5ndGggMjggMCBSL0ZpbHRlc
i9GbGF0ZURlY29kZS9MZW5ndGgxIDE1NTYwPj4Kc3RyZWFtCnic5XsLeFNVvu/ae+2d97NN2qaldPdFKYS201K
ggJi0TUuktJ02PCoykDZJk/SRkKSUtiKoRxEBGQTxMYwyDmdmDuPR4jgeFGTQqXp1ZLx3xOP16HzHGXFm9FS++
+D4zUXY3P9aeydNK6LfOfd+537f7TbZa6/1X//H7/9Y/x01ERvyIx3aiTBy9Ax4o46bllQjhN5CiMno2ZoQLv2
sJgTjj+BzKBDtHTj3r3d9iBBOAM2e3v6RgLjrybcR4uDRXBL0e30jz4wsQchaAhOLgjBRIw4q4fk2eC4JDiS2/
UXNrIfnnfDs6Y/0eJfP+20pPJ+HZ/uAd1v0NFMPzLLgEQmD3gF/7Wfvt8KzgFBhWTQST9yK7ryG0OKVZD0a80e
bFIlyeA4ixH8P5hi4yJ8OhgryzGKOVyhVao1WpzcYTeaMTAv6/+mP0aA4YtFJ9Bu43oFRCwqjLWgXOgrjT9AYn
X+ZU5ELZt7nCe076CRTDnQsuZhKYMOiM8CnBtY+AfoAPB9F3XT9Mj5Hr8fwOXYYsbgdRu10x1F0Ei/jOHxO+tB
dv0E3oRfRcTLmz6HDQNeB3oOrHrjfgk6j95m70THmXbQd7UNxElUon9Hw74IuYdTNv0uv/4Zuo5LJXJh/V2EBS
WGw8zRwPybNM+VMB96MAxBrY2DBcbwSZu9BYW4zXGX0aqT2STaw7BjIl+1F59jb2HKujDkOcoiMc8D/OLoJ9A2
ApivhwxL98afoKIY4Qzb+ZXSL8haFjlEotyMPrI7hGuYxRT7ajLZjD3Bohbl9qJV5D6QgC3pewXOYZZBdMI2zp
W7fuOO764X/1FW4wD7jUTAphXHUPq4fEU5eu9a+nsvju8b5WeO4VDXOlRb/4esW/7DAvqp9vTD+hqtR5ura3Ah
znethSJ5gGuZdjQuInwLiYS7AH4MKoES5Dh33OlK8zqj4jSyHKifOT34Hmc5Pnp+syjQXmksLzYUBDl2J47wrn
4iHlYa//o+YohxybgtCit3gpzz0PUdF9k61UavEOKLnu21WM2Yi6jsykUVXoslVlKgyDCVZGrOJQyU5KjNbOMt
0abL6QvV58mWi/3wHVS6/eDG7+ovl8GW6WOVQC/nR/J35v8rnmI01tTXm4tpCazFoI91rrIUwVwj3LWfOsM1ns
IJ8n7n6wpkrl8+EedTXd/UKy/X1fYnCYfauq3eEqc3XPubWQgxYQONWx2xWZR1Fqkf5/bqc/egHGXt1m2ep1NZ
clMfkGudRDQGCtwCJC1cmJ03iBdPFjLoqR4aQ78jfkY/fRm8zb7NvW9/O4zcyG5lCqyWrpnrR4hoDU1yEaheim
mqkrICxQsmtvfLnJx8f+UHBIfvnL38qfsJk/fG/M+x5xYH7djxoYpHqvQ92bH9mnKkRrzALxfdOvPjSmRdoEqN
loHQHRLoaORwZilHVFg6NsowdlSmxna/SmC5dqL5yAYCbrL4yWfXsfi0Deqwat3rWv4Q0184CAw1cjiXmxUu6S
iXAADzm2NNXPz1xggmFw2HekcIlDLhoUTaa78hE+3WK/QSOHJxrbVQDGDkAxqVJCQnwzXOCLWJjmY2l1YtqF84
BE8H27KTRVgtig4cefvjQoYcfOfgX8U+M7dNPGZv456bPLlz49NMLFz7rAUtFplr8rSiK56idGoRUNRBHFuawY
wP2aXVa1sewOriZMKs36BmTwaA3m1ijz4IsBp/eAPmkNem0HM+p1KzGxPPcaoOWxSjTqMZ8xKxTdmNkMRkNep1
Wo5LSzpihsJrOv5VdRwLt0vLlk9nV5uy6KsgOpYn/PLtaaVJ9Ln2kJ/7z9JmuomcLsiR4Mwm8VgqvFS4JXsc/6
```

```
VmNljfksgVsgTbXUMqWa0sMN7PLtbWGFm2TYT17m9ZjGND6DHdoE4YsFaNgMVar1GqlVqlXGVRGG0RdFphr5Ww
qm9qizdTl6G2GOWgOM4edg8u5Ev0cQy27EC/SLtQt1ruRm3Gx9biJc2mbdY36lYYNaAOznl2HN3DrtV26DYZNx
k2mTeZNGZsyN1k2ZQWQn/GzfhzkevmAKqDu1fRqAzqf3m8IGsOmsDloGWLjeCs/pBpSb9UMaYd1Cf2I8XbT7eY
Ry9qNaGNhsZopZmjmqZkatTRS2I6Jx8Xjx8SRQ2zhi8xixvMkcytT9xTz9EHuzJf1fBiy7wT34pd13G19fdTPt
0AP8Q7EWQ56yFGkUmfy4JZRM+Kz9urVo+Yj4Ct2ATLZM1RlOruhygbxDfXh/KRpAnx28WK1+Nprpoum18x1Zkj
DJSZkYkyAmIkz8SarKcuUbcqpzHUgB+NgHdjBOfhGqyPLke3IsYJolhALjMAqNk4liY160QaX5EXGXEwz1mpOZ
Yt1BVNTnYXfCfOtq+478O7Vcz8ZZ4KLjo8u6OHKxS/aWk+Pi7nhMPaEXor0GZK59DHkrAUVojbHnCzOZMXIdkh
fcEhpHs3cgkaVm/V72Qy71YTLIJVrZ+Vq5hWR9LpwZflymswXJkmSmS5CeIKdalRsKt5RPF4MZZAmMakvbO3CD
FArm2iIUykI5YerGRcfWfTTsYn/9ddXx3666GkmcPLg3gcOHHhg78Fw+Ce+Tcziy5eZRZt8bHn4y9+EfvXmR7/
/8I9vUN9cBt9cglNWg7Y5ZnE+XsFjn4JfjRUcwytCKiaO1Eozh1it6fzExJUJWnUuTVBfkBziIE04SBOAVkOg1
SIHMNUi7bWzUoKoBSQwlepKDUfg11IaCr+W0FKawtpCM19bCmXdCtHGie8w/fczUJj+6dd3T14eDUs1cQ/0na2
g5yy02lGosyCjHiVYX66fD+mHzRa9LhtpmGyFyqAxMRnafNOlCXKMAZh1dSTzL35xUbxoArUBWW2TpknbpGuyN
Fl5iIliM3E1Z7XAgc7Qw6WWWcFI4DKKaN/7wU8Zg/jbHJ24i40MXbmJefXwzh/ev/8It4XR1C3/4ysfiG9nw9p
J8dgAM3H4y/sfPv4AiYdl4lruJMS8DhWjNQ47g8z5WM8dytWPGotmjwqaQ5a9uZuFu4xHSgrsSJmbXVtosOebU
Jm+qoScQHA+Xpo0QVyYJidp8a1OhcZzO0rHS6EKM5LrJU3LkjGbYbWwxUVzyqRQZo4d3LvvwIF9ew+ehMpPAuS
vf50gAQJHAbsydPaNP374+4/e/GUfhIj45uXL4hubfGIeZC3JDgQuKIPyNO7oJTX4CYvVYnhCb7DCbYXRoFQpt
TpWv0KlUmKWZZ5QKBXoiTykhNsKqJBZVlsum7ciK8u62mpQIk6Vm8VjZjQHH8k06rBq1KzJs1lJpdYqoUjrMjh
1RvYsCLEPJ8xSqYY6feVbFOuZczQSdR6S6Bsg1GbRUJsFl1yvP+KQxmBFeYZSNM/QjNyGdWidssuw3hpAAWXQE
LSOGEas96A9BouGUbNqrNYrjLyNz1Xk5TK5bC62aW26XL3VaLPZcq15c1E5Y2ftuFRbqpurLzPMMc4xzTGX28p
zy/Lq2JvZOnwzNKNuptnQZGzOu5XpYrtwlwIE6tcZ1hjXZq21rc1dmxdmwmwY9yqD2qAuqO81+I2BvG3oduZ29
nZ8u2JEP2xIGBOmhDlh2WbdmjVkG8odyrsX7Vbu1e7V7dbvNuwy7rLca73Xdm/uvXkboGozmcmKTUJAK93k4s1
6/nk/8+g+8WOmZY9o3fXnv9wv9u9jSsWX7mY+2MHuZmr6+sRhcXE4zKwQX4HvQ8wZUsRZ0vPxl+CsVsHZd5PDp
lNGDN0aFFErGasRcYYMvboEmbWFWRC8pL8jeVcnFYvJaqhmVdlt2Tuzn8im1UxAZhMqLGakUis9MZdfhORfLL4
pfviiOAy9XQdzF/OM2C7uCDMtzCK4Vl4+TBo8cav4I/GomEDstWViHfMs/zEUktko5CiabcrWYXV2Vt7+TLVu1
HAk84cFOQuyEFZkLTKW6e26qgI4Wa6Qg+UiyasrkxOmi//4mum35gzIqucrhc1CVMBTp0QBjZ0CuKTYeS6eHc9
h0cbMhYugBlstCkgzgrSZjspqa2YzJB3PNVRWNDirKhsWQBX+9c9X3r1ik3JNx4Z1/JaKxsaKyvp6cbTvyy3HT
hUWfrD51rE7k2fHHqgVGjQX3eGozCvU2pBJx4zaDqr2ZxY+ULxv9v45ezM3g02sjUeFC4r09lwjKtPyuUVZ88q
h1kGpg4oB9SFDqnYXxItf/MkERQ/cYK4j3WuJaV5k3ra5Z+dxo3PPMmfZs3iCm+AnFBPWiayJ7LM5Z21nc/Xki
KxNnSmZ5uSxM6cWTC6UCo2yuFY6Gis/333vfYn4vpPHxGM3He078Z8vvs+wB+8ZvMu0afMv1/7+z0zrH7aORu5
6gDl99Xfh+C1Np4/+7fPukXt83e+Vl39Aa/q1SYipTyCmDOhBRx10a6wPqgrjQwwUFTi2GbVGrTewyKTRqFczK
g30Tiii47pVeo2aVB0GqY281gjdnVQzLtF68bXlAno5JcOgjdCCMCqWwxw0kTlMDpujnqWfZZjLztPPM+iVTBZ
TxmBAQs3ShJH7HybAvMCc/Bnz/IGr//LU1YsPsQIf/vJebrSv7/Jh0vHQM2ox+PEDqJk2NOQoyjYrdHoLe486B
0o+OqLezCF7Nl+G7cxSg91UpqvMJalCjn9y+sNtEkp8Nc2bquRJmUtP01yUK5+mz+3IG89jUxGaSyM0l5BJfUx
hmsvMNDTpCWEuZt9l1j8tPlbzZN+Jn/6E2bRpdOe9R7myKw3h0+HIG++xdX1XH3rl+6d/wXxM7XgK7PgEztpyp
sxRD11HSaEJevPZBWy+YILufLVWJxRyWcwO5OMHy7J2WH25gxmhss3zBK2uMF+VbbSyRqQqtypL8suRkZ1HA/S
C1NtkSHXBdOES7XbEi6n6bvgchvINXGWaD223QyvMr5rfPj86f+d8npid4SGN+IZV4xZa3udRfOahecluIzFXU
64t15XrywvKhfLC8qIl6jrNUu1S3VJ9naFu9tKCpcLSwrqiJcVujVt7i86tX1ngFtyF7qIuTRf0z136LqGrsKs
orO7X9Gv7dX36voJ+ob+wvyhUPKYZ047pxvQjhrGCMWGscKxopHjx0fnj81kouYvN2VL2L5xTyVRA91AitWlSW
2nJKmBmw3HNFRbNORPnT/3q2VNbnn/1H59hDJeCf927xXzil4//7e1vvPCbvxc/Ez8fu7aS8TDHf7xvV39i8TL
nK3935ncL5ouvPL7vzrG+0eWLb3rpifc/JD/aQd2AvprdTOufFbkd2VidMWo5YoIyATUP6l1ZptFursqSXhfJq
z19YTz/FrwyZqe9MmbRKMqCS4qiafUt1RC/TspafWVlw788ffWzcWiGU/Xsy7nQCHf0yX3+FuiBZ6EtjjnKLJS
hGVWhWdx+2xZQyqI6YlDrmQUIjvwytAh0s5qroFFbfqEagn+CtGqkdFVDv0+q8sVqUpWf3T87TdN8qmk+XJKm/
7AZbWbYm2cz5NCTsM6HF/LCVBtfU0ttycZbvExnW6+HKRffm7z61Pj4+Au/aLi3nretat1yYHfflaepBfc8+3y
ODdHfFNmuRxcE9m7bZFz+r6hARX9b+91/1T+U/J2NnDyqGgX5cVE19eMbQsoBMT/957gZP89ZuXMooLiMtnCnU
YDdj5Zx/wXu0Nmq6pAGv4Nu4QtQgM9Gl+GzB8bLuLuRTRFGW9ixa8u4V2Ht9WuTfBlazI+hp2D/LewXBHNUhEb
R7xgF08Q0sUvZp7AK34af4yz8LD7Av6loUjytzFD+SVWhepBqZIWEYZEUQSb0KFJCp6dl/idYQn6Cy2VWpPTel
LIBWjR4YuRdHOqTxxjmI/KYg/GYPOZhvEseKyA+D8pjFTKjx+WxFvz4jDzWqw+it+SxAS3UJGWZkE6T3GtGnOY
JkMhwang6o/mxPGZQjtYrj1mk0o7IYwzzd8pjDsaPymMexifksQJlad+QxypUpP1neaxFS3WMPNZnztEtk8cGF
CxIyjKhnILkXjNSFfyhIRIdiYV6gwlhbk+5UF1VVSN0jwj1oUQ8EfN7B+yCe7CnQnD29wsdhCoudPjj/thWv69
C0+gPe9cOCT1B72CvPy54Y34hNChEh7r7Qz2CLzLgDQ0maTr9sVCgPhLpS39OH6/1x+KhyKBQXVFTK82nkwQig
yA5AfoEE4no0spKH8xvHaqIR4ZiPf5AJNbrrxj0J5ooGdGD2JKyQZgb9/uFbn9/ZLi8QvgWWldoNFObQTOvIHF
```

```
OYaVZcMM/jebfjqowQ3IIVBQSMa/PP+CN9QmRwEwuGk27PzYQilP4gDroj/lBVm/MO5jw++xCIAbGwzYwGGCyC
4mI4B0cEaIAOGyIdCfA4NBgL0jpAaUJZSLolxH39vREBqJATggSQeAOIPkH4wBwEYWkqByY+QRvPB7pCXlBHiD
YMzTgH0x4E0SfQKgfMJ5LONINQmckkBgGzIvKqSYxfzQW8Q31+CkbXwgMC3UPJfxUh2kb7OClnv4hH9FkOJQIR
oYSoMxASBZE6GMSlMB2KA70xBy7MOCnVlP/xoP2NBl2IrMyEhPifvADUIdAVdn8GaKJcsA2SoBOyNBRQcPByMB
XNxA3BIZigyDQTzf6IkI8YhfiQ91hf0+CzEgY90NIEoN6IoO+ELEjvlSj8cCStzuy1U8tkKKIKpAKgsFIAtwQl
2aJV6JTESCtCfGgF4zq9suogRoQ5N5pdkYGIS5iwkAk5r+u2UJiJOoPeEFQhaTU9NUB7wjhPxDxhQIhEmje/gS
EHgyAqdfno5ZL0JH88sZAr6F+b4wK8vnjod5BqkZv/0g0GCebSIR6e4BJnOxI6hOfKUmKOJ8EmLc/jcEMJvK+p
C5THEHFwf4RITQt1MGkmJ/8+zxKSwZxAibxTTJF/BB3fsmA4UjMFxeKUrlYRGQnF4QikrpFFDbwToucM91+yCb
CdQj8QIzYGgmlFPNvS0DWCN5oFFLM293vJwuS/cB5hmOC3oQQ9MaBo39wOi4gbirCfcLQoE9WuGh6XSmSLLyRZ
+ORfpLZ1HXEUV6hn1QQyJckYdTb0+ftBcMgFwcjqfrx7QNrmigoWqCivz9AlFrpEpraWj1CZ1uTZ52zwyW4O4X
2jra17kZXo1Dk7ITnIruwzu1Z2bbGIwBFh7PV0yW0NQnO1i5hlbu10S641rd3uDo7hbYOwb26vcXtgjl3a0PLm
kZ3a7NQD/ta2zxCi3u12wNMPW10q8zK7eokzFa7OhpWwqOz3t3i9nTZhSa3p5XwbAKmTqHd2eFxN6xpcXYI7Ws
62ts6XcCjEdi2ulubOkCKa7ULjABGDW3tXR3u5pUeO2zywKRd8HQ4G12rnR2r7ETDNjC5Q6AkFaAl8BBca8nmz
pXOlhah3u3p9HS4nKsJLUGnubVtNcFoTWuj0+NuaxXqXWCKs77FJekGpjS0ON2r7UKjc7Wz2dU5JYSQyeZMwUE
2NLtaXR3OFrvQ2e5qcJMB4OjucDV4KCVgD0i0UHUb2lo7Xd9dAxNAlxQBDlnpoiLAACf800A1o+a3grmEj6etw
5NSZZ2702UXnB3uTqJCU0cbqEv8CTuIjWsAT+K8Vllf4iMy99XoACqyWzaw0eVsAYadRI2v0EJ0ubb1+KMJEtt
yckvlkZZSqX7aadRKRQBCuHkQEleao0OIZ8gsevJIFW4quciRbJfLLykfEN1wGknl17fVD1UwTkoJ5EeEFJPhU
JxmOhyDAxH53It7+0EY7EpRQb309sO2eErN6QmVPBCjsRBsGY6FElBMBO8QzMZCo/JRHJOPqpkWECkz9Y/541E
4qUJb/f0jFUAbI+cZ1SQ0CO3WgGw6ha8nsTRZQxNCL2XuA8OhKasQNDfs1yqHQ32hyhDUqG0V0WC0Ui6UM3tG1
ADtehSNoBgKoV4URAkkoLmoB5XDvRpVwVUDo26gEFA90CRQHD4x5EdeNIDsMOtGg0BfASMn6odLQB0pXnH65Ie
7H/ZshW8fUGpQI4zCwGEtGgKKHqD1ApdeSinAmPAXgMsgfEeBphv4hoBOgP0RkOulazP5dFIpIRQATSNw9X3t+
tfNr6WjOIwjVHY1aFuDaqfRfx2XAN0j2ZyQ8SEYJMCCpagSLp9MvxXoK4AuAvcYWOWne2PU/grg4Yc9TWnckng
k/fJVP5A1grGf+soPaEXQMNASr/yfwbqC/jvk60mWMPPCKF3nr8aVBi34d1xE+n9ErF4f7SmbQzKKAl33Uh8PU
FT7YC4Cnv0mXYhl7ZTfAOU2FX0S7yBd88t29VIpgzTCfJRPgK76U9IkD0vRZKd6RaiGg3R/VI5wSUIEuCZkD4d
oVEi29MhIJ3kmqBbTY9wLVD00QqIy9yQHQi3pLkWSH+bjcgQXpUVJEfUc2euj9zjVqwf2eGX7pBjsgagcoFwSd
CWJTwBG/XIcz03pOCWBZDbRPwG5IMU5kTiFCZmJwncEpAxRPae08VELEjTWumE1QVeTMr5egl3OpR7QbIhykTA
ZpjEQpDmfkJEZoHPpFiX5x6ZFpaTtEMXQnuYdMh6g/kz6eip/47Db/jV22FN2VtK6I1DOUj5IvEMyqtO9f2Ork
8hJ2kZTEZ2YEXVTFg1TPAa+lYRkNgRozRyULfSnSfTRbyLDTu8EiTBQ9FB+Ek16HPfLVTLpoR4q20c1DsmaLqX
Z6ZF3eYFjhFaGKR+k16IpBL5aCQaBPiFnQ3wabTJXphBLrwHp+wRqs1f2VHeqbidjTUJDquTeG/gzQs8YQfb9A
L1P1Y9v44sEWB6l55ZXtqhiGlI32kswGUnpP0CzL0RzOVnRiO4JuepJM5KmBFNfms/Toy55fhEpEl5DwMVL9yU
t8lFNib8G09DoBTpiTVCei6XVUC+NHil2kzJm4hP/RpvSa5xvWoR5qY+up8GNNZkubyYu19PRLvu9n+4L3aCqx
+QK5Kf6DUzjm5yJpyIzmTczTxG/XO/80zwwTK3y0f1F1zkXi1J2z9xB6JOnblFatEm50zLjnOmmeR9J03VIzoe
kJ7bCaug6iPnRNorzoJzRUbikU8xLK6s/tSPd/5LON86YIK30Ar3HZR39NKK+Pl4k665Xw8nqEKWajvD1UBXSk
Ev34b81Z+O0eibP7KmsS2YU6SD6Uz1ITN4xnWOURnQffPfKHpPOxUGK7cz+4/9Gxfp6q7rlHEnI52IghdRK5KJ
y2lArPBE5bfDkQeugn+yga26YE6Cf64CVtfDUCLON1C9OukLWi2g2roMx4diG1lBeEo8O+Ca8u2CG8BboM3laB
fStwIvsdaH1VIYLuHVSyg7KezXMtsDdJdORHQ0wswaeybgZkW5UktcKuzw0d8g+ooukqQfmp6RO18pNJSY1Ww1
PHcB/pbzqBN5uyo/ob6dIkXFrSs8mWVMnxYhwJjwbQKMW+kRm18C9Heg6KZ5OarOkbSu1oQnWJVtcVAPJE5JGD
XBvB9mEohn08lAtiCSPTGmnFhJ7Gul+InUVnZU0a5O9TMZTXCpkLCU9CP5rU5I7qf0tcAnUfg/MeKhvnMA/yTc
ZO82Uw+pUHK2h9jkpDm1UQj1dIygSPFtSlB1pXmmgeBG/Ec0bqSQnRaTzupYkuU33zvWiIymhmdrnoki1UOpOw
NEF9O7UjBSPbmprg4ytxFOKeykmWtLQbaA2Es9+F6S65JhyUuymWyFlCNF/ygrJA075uyENsynvt8rebUj5uo1
G2VdRWUdz0UWpnNTXnSkUmmj+rpY1X5MWYUk/rpHjsy2l2XR8k3mUpPs2tUPilZQ93YONNJ5aZA07U2h8M1+pd
rngXOuh7zuJVN2efnKnd49TXWl6/2lPq7XpnYBUhZsp7cAMuqlZqT5LZ9bUO096D3e9kyv5liz19FPdb7L7kGq
39G6U3v36aJ8u9YLxVFcinR+RVGcyTFenznTpbXCAUqS/78WpXMmyIXnHTF5Sf+ml3QKRFr8Omjc6oWa+IUbpe
S9JGabjhNyZEPuGZFoyPzrjrTg2463qm3yQtOWb8I9Rf0fld6oQRZj0kxUy3xhKvp9NYUIQkH7dGpjh9anoI9y
Wopl9KMGgN01zn+xx6ZcyIlPz7/h9rZLi3QefSqqjj3Z+FbQLj8Lc9I7ym35nRFj+70DuQLfO/H+wyN9Jdqfjo
9dF/FomfnWihH/Vh191cBMl+Nd6/MrLZfwrPvxyGT67Ef9qOz6txae0+MUXLPyL1fgFCz5Zjf9BxM+L+Jci/oW
InxXxifFm/sRlPN6MnxHx09vx34v4KQP++XEd/3MLPq7Df1eNf+bDP52Nj1XjHz/p438s4id9+EeHDfyPSvHRb
Rr+aCl+YhV+3IR/WIGP3DebPyLiHzxm4n+Qjx8z4UcfMfCPluJHgO4RA37EwT0MGx+24Id3cocN+LCDe6gUH/q
bKv6QiA8+mMkfLMUPHtDzD2biB08yDoeaO/B9DX9Ajw+cZJDDzX1fg79/ltsf2c7vP4UfuFPLP2DGDzi4fTDat
xTv3XOK3yviPfdv5Pecwnt2cvfvLuXv34jvd3C7Qa/dpfi+XWb+vtn4vpPXzjqucbvM+B4QfY8P/00VvjsL33U
Y36nFO30+fqeId/Sb+B02fMd2A39HNd5uwLePGfnbM/GYEY8exiNmvE2Dh7cK/PBlvHVoFr9VwEOzcAI2JWbju
```

```
IhjIt4S1fNbRBzV46iDi2zHgwMr+ME+PLAC9/fp+H4T7t/J9elwn4MLg8jwZRwKnuJDIg72buSDp3BwJ9cbKOV
7N+JeBxcoxX4g8l/GPh/useJuEXtFvHlTBb9ZxJsq8PdEvFHEt63CG7bjW0Xc1YjXi3idiNeewmtE3OnDHRb83
Wrc3mbk27fjNiNudTrqcYsW3+LD7iIV7z6MV1bjZmzimzNxUwZ2sRreZcONDZl8Yx9uqDfxDZm43qnl603Y6VD
zTi12qLGD4NjJ3XwYr+AW8CtW45uWW/ibVuHlyzT8cgte7uCWafDSugx+6UZct8TM12XgJWa8WI8Xibh2oYWvF
fHCmkx+oQXXVGv4mkxc/R01X63B1ZJ/vqPGVZU5fFUjrqyw8pU5uPIsVzFbw1dYccVOboHaxy84jO3zLbx9FZ4
PRsy34PkObh6oPs+Hy+dW8eVOPBcUm1uFy+BWJuI5S3GpPocv3YhLijP4kk5cDNuKM3CxgytS4UIhhy/ciIUCM
y/kYOEsVwDCCsy4YCc3W4NnO7j8YjzLiPNKcK6tis/txDbgaqvCOSLOBqHZIs4yYavFwlv7sCUzk7dYsMXBZWb
iDKDLOIXNAK9ZxCa4meqxEfQ3HsYGWDOIWA8M9DlY7+B0ItbCg9axpA9rgEazHat9WKU08yoLVpqxgq/mFdsxD
/v4aswBM24BBqasBjOdGImYOcn47tnHzP9/9g/9Rytww798hP43K53LZwplbmRzdHJlYW0KZW5kb2JqCgoyOCA
wIG9iago4NTIwCmVuZG9iagoKMjkgMCBvYmoKPDwvVHlwZS9Gb250RGVzY3JpcHRvci9Gb250TmFtZS9FQUFBQ
UErRGVqYVZ1U2VyaWYKL0ZsYWdzIDQKL0ZvbnRCQm94Wy03NjkgLTM0NiAyMTA0IDExMDldL0l0YWxpY0FuZ2x
lIDAKL0FzY2VudCA5MjgKL0Rlc2NlbnQgLTIzNQovQ2FwSGVpZ2h0IDExMDkKL1N0ZW1WIDgwCi9Gb250RmlsZ
TIgMjcgMCBSCj4+CmVuZG9iagoKMzAgMCBvYmoKPDwvTGVuZ3RoIDMxNS9GaWx0ZXIvRmxhdGVEZWNvZGU+Pgp
zdHJlYW0KeJxdkk1ugzAQhfecwst0EWFIgERCSCkpEov+qKQHIPaQWirGMs6C29czpq3UBegbzXvjx5i4bs+tV
i5+s5PowLFBaWlhnu5WALvCTekoSZlUwq0VvcXYmyj23m6ZHYytHqayjOJ335udXdjmJKcrPETxq5Vglb6xzUf
d+bq7G/MFI2jHeFRVTMLg5zz35qUfISbXtpW+rdyy9ZY/wWUxwFKqkxBFTBJm0wuwvb5BVHJesbJpqgi0/NdLe
bBcB/HZWy9NvJTz/aHynBLnGfIucI28D9wgZ8QpR86JswK5IC5S5EPQ75GPxLsE+RSYNI9BT3PqwJThHPKQ9yn
MOSE3gXFOwsO5OXLIX2DmJOTPdshr/iPymj+nhaxfjqvBu/tZORN3a/266YJpz7hhpYH9/gRmMmij5xtsNpn6C
mVuZHN0cmVhbQplbmRvYmoKCjMxIDAgb2JqCjw8L1R5cGUvRm9udC9TdWJ0eXBlL1RydWVUeXBlL0Jhc2VGb25
0L0VBQUFBQStEZWphVnVTZXJpZgovRmlyc3RDaGFyIDAKL0xhc3RDaGFyIDIwCi9XaWR0aHNbNjAwIDg3MiA1O
TEgMzE5IDYwMiAzMTcgMTAyNyA0NzggNjQwIDYzNiA2MzYgNjQwIDU2MyA4MDEgMzEwIDU5Ngo3MjIgNjQ0IDY
4NSAzMTkgMzcwIF0KL0ZvbnREZXNjcmlwdG9yIDI5IDAgUgovVG9Vbmljb2RlIDMwIDAgUgo+PgplbmRvYmoKC
jMyIDAgb2JqCjw8L0xlbmd0aCAzMyAwIFIvRmlsdGVyL0ZsYXRlRGVjb2RlL0xlbmd0aDEgMTI4MiAvTGVuZ3R
oMiA3NzMwOCAvTGVuZ3RoMyA1NDQ+PgpzdHJlYW0KeJycvWOwLV2XNXhs+x7btm3btm3btm3btm3rHts+/bxV3
1vV1dX9p2NH7MwcY66Rc441MzNWRuzYJATySjQCxnaGJqJ2ts40DLT0nPiyFjaGLk5KBrbSNIomZi74/4CsUCQ
kyhbO1ib/i4UiUTVxdLKws+X8d5yQo4mB8z+AsIHzP+HK5i74Ai5m+PiM+AzMnEwsnCzs+Iz09Gz/DrRz5MQ3d
bQzhCIRsrP3cLQwM3fmxP+vXXxyFUU1Cioq6v9GGDg4OPANPf7N4AubOFmY2eKT/g+Bf1BXE2s7exsTW2cufKF
/YGtrCyN8M2sPe3MnfANjYxPjf2moGlibWOGLWlhb2NvbueKTC1H8T5V/EmWg+eeLBYoEX8zE1sTxn5L+Y+C/z
BK1czQz+Y9a6NkYmfDJzZ2d7Tnp6Ez/oUz/RdE6mdLamjjT/aNJImJrLGRn869snKCg/jVY2MLRxOif4j3o/p9
uW9naudl6/S/Y1MLW+F/S+MYu9nQqthYOLiYSwv8n+B8I6r8xMxNnfBZ6RnoOekZ8Ewd8E3cjc7p/nVLZw97kP
0iGf8EGtsY+XvZ29vimBtZOJj4Wpib/bKC8nAxcTfCdHV1MfLz+78T/PIJiYMA3tjByxjc0MbOwhfpv9X9gE9P
/PJYxcHa0cMfXoqf9x0N8+n99/mtP55+JN7aztfb473BZAxsT/P9V9H/RgoJ27vheNAxszPg0jOws+Az0jIz4H
CxM+D7/U+q/TPi3Af+ByhtY/J8E6f9bUsLW1A6f4z/r+MfA/1MLPp3rf7YzPvm/2pkC/3/o/5OhnbOFkck/nfJ
f/ahNz0L/Tyv+s2H4f+3T/8H/V7f+/2vQf2kJ/Uvpvzvzvzvzf2Uo6mJt/R9ukv+nm/j/2OmEL43/L0OtDRz/d7yBj
YW1x//LiP8VqWbyn5fk/5eShLPBPzUI2JpZ/9tofDoLJ1ELdxNjeQtnI/P/bLX/Q6jYGps4WlvYmsjbOVn863a
BT8PAwvD/JJXNLYysbE2cnP6Z0P/gTGyN/+dp6URsjeyMLWzN8JWc/+lpA0fj/wL+RRu5ODr+Y/F/zPI/Y/99b
GrxT44mJu4mRlDOfMHIj0bW5Zp5P5/yzV1k9JQaKttE+48cTgDjXyV//lrHUze1oj+UKUXZDJqUzeGv8DbZsjH
xg8guDRFQBRswHqRCBiwCWM+chX+nOsysXSMrhkpKGfWn+jQi0hVMhISYjH/O78xHqbV9XfIQt9tKxTHz4qD75
on2PHFloBuoaFLNQ/igY8bIgoOspdn2+wHiPwNIqoTg8vhhIWDPbPPFAYEDobv76vA4RFCsnnTc+9AX+LZWH7x
pHWAVhYTL5ABgPf5mz+YW3NVXOcuskMrnzbzwV4QNsBH0nDck48SeBd4MP04nWJgNG/lpnfl0p1wNnnwpF9rzE
TBZY0g8hKYnViWS6WmuhRv7h7zrB77anmrscxB+fM3LPgE2QfNDvJ/hW6CDyvqBT0oAeDs39/kxbI29kCjWw/J
Qctsyes05RAlJH2CkBkdceh6juCkm42AUGhZNlRPUN0XPyjDPwEVUjAjW06mD/Bp5Bak3itygvnmVhmgYfoXVi
x1XiGBXT6B7l8nfrkqrHB8riOMCjr2qndzgxyJKtD0AojWwmZ7mKEX0bU/nK6IhuJqG72kxgQn18VfuoNx6OgF
/BfmXD9aWY+hz/c5azHidHQH1V5kwxufwRAHPp8bWfeo6GfeQFMaXqk6HAPcr6av2txuZrpA2dM6ZHXHRPCTWS
B+BamdGjcBREwDTY51nxC7n7G77aSb0GAxVWrA7eqdwzNtpjp2H2yujoHPcgouRS2+o0MPoF1usDUSUNh2O7i4
f2nU4BYAbtSJuzLkUWaqxLcDNinCCdBNdudcFP+p/sTX2gWnLRRKA8FvN6RAnsFHAtSCFjxJmg/4xQHYbJhWiM
nZMu3PM0X/TyJMu3688CUVcrk8FvUgC4o1kYjA8pQO1pVKh9ad3hxcM76QS6UL7OYwxpJyVxNchiJgtuu1yu5T
OCKC74jlxIonpDLdNsGL2k1Q61k6iStOV4Vdb3Os7WhGdxK8uajqRp/9LQY26Qhya0O0EEmYViQWm/83CEVooO
7qIHAQmbEPLCeFNurhrbV6mEc6tKrbgaeMS0PTMygLAfC9dmYkMKSUTRvlIGmSb77eukEPPKJ0ytRHHfWBI5EP
4EYlFhgG+0ItyQtl9d11cIhaNbMmKx6Ei7gVhhZiCu/Y8XrDaullOked2nI871+SvDwUzP6mJuNWgfLZeNhO4e
i9xBn4w9hj4R2XP5BlNjGO9aXIs4Q8ztMT0StTezbm8KifGOE3I3Cx67AnXVcSZoCilt/cGcPwJRDyYVpBEf1T
```

```
ispP/mVyKGFrpOl+dis0A4gYkkAVNL42yLVUEop6zFjxv/iLTzgohkQLswjMrvJ1U9KtHHU/RNABgUQ8iRcNTb
B8+ea63o0+ABdD3WSZXH4QjIo/dK2UWEaRa9fSPkmqNESyanPnq38U3YsvVYZkktDceOXG8bd1/Fxdt2r66Th
OVnkMwPnNweZbXh3HHO73mD5rIs0Mf8sa1rytgiMjUyDj65U7gIxi43EuGn04J7HOuYs3ERPDy3MzW2DGJ5CRl
9vUbOc2fWtwIgudIUcrNQW3tFNHHsLk2fI3jr+UIlqXPX1kG/lLKkt5R+3X6ymNwfhmtwLa1i/KO8m95NFy9o+
Ao7r2CdbSBlesXFz3C0E5XWuR/AbwWnaAKAbIeAu2YszIsNy5F2LdA4IZQXNg8LREETsQDng1PerjyYCBpt/Dz
yQJzq2pbOrPn/j54PHzP40IYlck6V5eqhrGbRXIeuBQkm9DvQM7/V56CSPsXUlY87HfijTeBU2C9IIkKq57lID
i2IdAWOITDYPQQllE4H10t8FwXtXzt20r424DG4IOA3VIzxDUwbV9Kbw7zpdZb+hvU2T325sPf8ynH0yD5n2B7
Zakjc72fzf5oVLabd1qk0UfZRymLhLV8UgPQWLGVZHeDE+lvMmfV4eytYwYizFMKZNyBdiWbGbQKPXJJfpbDR9
sqMx/XJpz1lThQzrjW6UuAPq9bHAhyaqGQk2oViftB1009A7Qe6AW6uzbm04dD8AjAmPymJ4cyjVL9uZddP+W1
fp7S2Ss8KHDrM3jtf7B1QXG/lVkqGp7JeXSMk8hI2070PujxZ1/IUJsvZ1qBohbuR/qoPdOmdlzvBGTGsqMMdy
CZ3MtSipoKgGGkrrsV2JyzpITcmXvu7uJGRDTqWOIPVbivYMsThuX+N18l3U5r6cYP78vXa16qJa5vjBNI6cuZ
AxPpn5x21CP51haBK2ojamc9r2Q0pqzV6e7TLUoT09FGtocxfOWhj1NdT2dBUlPdx9b+dxeEEkNrYyyCzH97Ed
CN6WJJLjDrfbp79JKEVF8N9idcDlQj+XX6zOU3y3BIPD0vMLlZI3bT0Nkis2rHSicWjW4LIWQo/7jSwk40w9wy
+Hy0bKS2Aa7tA6+h/6oYPo3pUtXvVHYB9opaP371XHRvZXwpoKLql5nQ1vPb6MzcnY/fg2i0To/FXyz8FpfwoA
k6r7O3fdlUg+WIVlyyUceuChRyMN75mauo6EqB+xldTjTbUPMQHTXCbpjbYwAGkiSSAQt2XunQ8c/uYh0f1h93
eMByoZAsEq8d4Hw8y6isI2gd9t9YFqeMHYs4Ni57h00p0jU9xtosoxIjDqK3HSEYV3v7PPDQYGJx8mShgeHshi
d5sTIqTiZ1taMgjuLcgE2Ddna93FFiYUg3AhUxAMsnUuXPE7dy70lCHunWk/tWELSpmcxYoewA4lgYfBduOP9p
F43JWZpfIfSvk+VZMr2U0RVfAdxVbBO9TqenL91ydmLdYZa3LR5qVgh6SlkaqhI7v4o/0fO4gDP2+fhKVRB1Vj
6GaMdZq7RxCz1oD1blSGnZfw7hDWYHVoEotF/b2KWyuuEAkjIYX54r83UzvnK8OE2T/2MnncIgrPTX3qg8nWu2
ibK9uiBtGVVGT6UDIM5eD4CsEKZJlyJ9kod0pza1w36Cock53vDOrVaA9/hK7w4kYhTrvj0La/XVNXPIm7Qbx6
iynn94nt2eGAtE1aSqOD8BDmfrU3HfGcMFTiJuOERqvMkXCYLLXRkPCF7pYDY7l78T4LzfS2N/mWwoh76SPyWg
7xAAO6cJH6qAHkviXuqK2HkLyOBtt0+mGYnKnXcX38WpEhMX3KmWlEZdAlYgutq373nJIcq2qfVmA6JmNXWP+1
dJgK0BLvZ8JUFUYiMEebadzc7N94mNmcjaMQyE6YIH1VrlmKX9RbUUEUs2yisMNJlkFW6n2Go4E8eatcyeKzIw
45O5eY/w3yUHpKq4KqhiACNjjbOmqjUG+YBoPuAXxPh5qq9HL/pT35WyDC5+wtEacvB0m2lZMSaePYmSezbJ3y
islfXZeGsweQIZwZqCFRSGq2UXNxs+9ypmJVHZ92br5l8FdO7FY0phc3SVUdy30TMDrLyqGqDp1OJdozgGGHhm
qBal/3D4iUyjfVwo+PKg+hnv7mmm2i94EW/N07FyEQAOcFwTijcZVSHaVxmgyzIM5aQrqpNSaPr+oI4lGYLJvh
6QROnIgn4S49EdmjDMHtyNfFlEo4oPFz+Sxg15cH7XY2RiZWOsjnq1Z2XYchAIZ16Knl6OtqGOgAPe5mxRPFbI
PzxxXgOZHIKftz9KYlxynICf5DSMs+JwXWM4HD2ekKsBTFXWsNFXglmP2ukziZH0BSUpC6HVUDLVOfDiy9LMZ0
/HvEHc6t2QCHQmHNjA8AKVeeskaSW1kfoF1v2lBRVvg11FcWZTGaAintthklK/+pe5kpq4uRibNDEGAbXN28OO
K+tjzVLUHUWipRmNvkRVl/fZV/WBtTXcauwhaM2iG0cdqz8Pehb33i2QEVOp2dq1itvZRjDDW4T2s+ODrVtoac
0CUIsY31uuEJZfkPNvMvbvYJXXmTkLJa5tYuiCzLnGqFiId2V+Wa8ReuZXkjRPvY3jZU7i418/DpHb3oFfB1Lf
uk1nIb7O2aAtv7Dv7gLUbtRQSSwXL45Dkm5hpQ4+sFOq7jqmNBrHYxGQ+8Sf50YIXWicg3u6WUimGD4ZI8c7MV
3Vhv4tzLifSrRFoQDCS7iBGQ89RY6JIzqJ6wDV2L8AiTMAUdKkLuVX1MH9f53Su4mxC0olfBcwttnSheZw69a1
+HMsHnQlr2HVqI8iqGkBd6sd5+CYMpR6xYDNyWGab+EJ/LNSRgt5vGEos/v6djJ3IoVWcpcjNob0o2XwOBwb5m
MgX8l0n+B+8ywTUQG/h0s9MGwbeA5Rn5L3HNeGl2Jm9YQWFE99AF6T+/3VhxO4EjcIVOZvqUAlPKu8qCuWlV+i
J/xj5LYmyLY5Oj6DV33kTTq/R9cwfE5hzlyfBgyCvrm8BDCwKrPOPiNKTDiDakPdmhhGjIiPC8n/j6IhUuEreu
l6Xv5+DTqvkE1HkizXwhayOfmc8E4mGCEnSN1G6021gMIEfuAvLeb9LSfse/uk2xijFljdADciOFmFRGALGFWh
0n3EnaLwlpFE+jjtN2FuDcw6bMXjBRxBOc1L79Kv1evs+8IKEEz35N0Qs9NONelsMMPEf1cVN2yelTHppJ6ley
OmCpjTd6lsV2KEGNY4d8HCWrDOgE1fGHdAcKofEmwSW31G5VDzYmEVH0JiPdplQuxoJTnRgXzI7sQoEZYZepvk
qMKwA/ZRx/qutr+To2zpPRLZgYzdGj6g5zYwDnx5LSTZgzvaQxBCSZKtUZcyfg2tBOv3Mk+8IFnZdxSbpwyBsU
OkClqQ7b3+iu7OvmNzy4e4zVevq37aqIfsDdNIMYZ+xawv2znrBYYbgQWB5h40ncsSzLQYIdxJHYcULmpT4g2X
hZ4fxsEGJ12ZV8P+qCZjsqOPBeY2WDIpcnkdlH115Ydi39Y+JQzMPFi4XHTAdhpj3poRLwCL54/YnZeKvpMH6T
+XDyISByOfbFLLWwRoYpDGKKtsItxWe0lL3Ybsvdd0wfUgTOjHx/PICiEl0+7Gy1IM4J+nEupktMID+nQTeHZX
oFfC/6Uw9LO3sUwJ4cwP+RaaXrq+5uilTR7NhTrXEODlIgTyinVxPiib5WIoDlP7Scd778AZQeULR1nw4yP2uT
DyBAgP/IeIxt4kOEpsRkP1OjpirWEEQMquWF1Z1N4YSD1KaF20nVZD7EZr8SGIjNzIa1kLNVM4LRavTJkhg3sK
FZWgBWoCncUZByswt/07+BA5+xz1/W3qatlBayR4s02yLqMr/QSooqN2FGqkThaAczStdNN6/1SO+YwhbZu6Ul
TuNRDOUlTzgsOrHP37nqc5hUT/MUtBZiFz1aSd4418MOFi10PFFBMwVgdLbQwxr0Pqi7YijGb+/GmL/Lh00BjR
Vf5hpaVEDejQKpYbjMLFIux10d7ndhwmqgse5srOswm0mSSsy3M+a2xWTOLwcblxdFyaZMHfjuqZjEQlWkx+0M
wF+RqQEGPdJYjeWXTRQZKJlezjXwBwQaWQcBYdZTKwyeEug51MhZRXPO1UEvZkbYppMPXX+1pXSpC32m5NPT7L
P/7Y39AP25ySwAuR79IwFMqBJtahLdFZizS4Vz7tFD+4PEqd7Nvo24hmEInuMo9YCTUScKFY32sEkREPvDMjIq
LOmkMWSdWCtmtcFh2ZTLcJoE9HBpm/yaiN2WZ5YBE6HRcw4vtMFr3hRM5gNArNmTbbVOapz/1pe/bOgE32Hhza
```

```
dd3hVUe2xEVfyswAyOMj7KxsXCV4W5SasnV3iMudr5y2ID4hxOv2Y0gcNNoaGEjC1tg+EGm02ncgWfkhnJNg7o
IJL4/0Bt1X8Z2+mEIZf5xpK0j4RbdnTy9Vph6lSCdzUkVvLFMCp38CnfKy6YEATwqegQa/GkHw7kWdyn1dCIEm
NGUlhmTyO1oQ5H5gOfgeY8C7Q8OMsDsruRv7O+NoTjk39hLJzvBWiszElq8N8orwAUdzAw42qDEGPq+H5mThuh
FdMNoehpZJmfGWbghj/2xYwJBrvdB5v6OMKdP5BiXyAi4r2XLRsafYl32u8lTjoZ+wSeANMwtwVjdOv/ecGj2+
dUeP1dH9baVpVsTBcQ8P3QCtwlFDecoXiqjCkjbyS6maLk5MXeRaNfhsDArB+ccv1QumegJ18GaCEdI+71uSPD
x9zNt+4NFr1o5dj8T1HzvqbD878q2X0LZ77yzog6i5J2k6Te9Ww8F5uxVGfZ0V13++qYj0GEDJs4IJe5ziuvfF
aeZsqgC/MXWcXh+OBmqHAcqH4BO8wpRlwcBSFpgAsbZtldFEloa5OTtRevgxvSk5Fca9toRErKLKb0f1KYQMfT
MSQJ3ZqosnAgUbaSeUabAH9v98p7cSXzahoa/MDqCg9L4LxCKt1NHa1sS2kbp8cd51dlPXF04InRey/e5hs2L7
D2fZGyQqhtZ+2y/SgNTAwct4oNrL+UyIX51DQC5fomqoyxSpa6ZR3IboG1wuuDf9GpzJKDUE86e4NWPuxv9N+U
Qbl2vs/5/KLWCN0osUq44aqSJH6lxG1wnR2E8+15JkavDhElUjbDzEBGpzmrFc2nWfEoFO6Kxm4MxLgJeS4YIk
4mXDrjtVdKF3aeVdyyzpmh6CYGoFPhv3HrrBUCt7mt2cp1uR1GAkok+NSXWeX4Fj2ITPv+uQL2JXHzL4tCPwmf
1uUjXZeNqdyo7WjJGNUS/UV8T+Hxzsa/+UqsGvxSvoNO0Co+W04+BvXuHa6lrjRTfKPMm8G9knYIsHdUF1TquA
d6TE5Mb2wFyzeDIbHfiJXTioN9aj5SXa9DYKnqWQ+Ncd/Ic2hN7j+GWtl/thb6X7ezmUUFhDxi2rzo1CZ6PxRb
6XUcgmtESefRAMGz5Li+H+rOwdw2BOin8FqmsbUZjEYgKiGTeQovNbX/q9a9yY4RPGDB++o8g87jbzVQUw6RW5
b5wuJv4oEy3OneplrEIhNEIjxIdq4XxCOg6n8Jl/hD0oQEGv5p9Q5K/HYmk3gXx7zecUahR8HeREA+j+lXast8
gKjTnzeTpksYeqJbBeIb2KDKWS+RC+pozW+QOONy6o1cPNmNdi4ksxq/KxVXoFJB3Dn/8EQBL10emMBQxMt7Oo
4xFD7luesYzJw9QJZ+iiAf3KWe8AX6NI++bzXH+MezG30p4ycMhCwHOqWKPLlrfCcelP9/MSs1loPfHooDw7mq
StIyjtcnNPAYVRx5qrkNmfRqimWZTBrq72uNyFXBc9El1hHOmuiZpOXVkf0gD33QbrQ0w2RSoDw/KV1MSEqX6A
RaklkeIHpugXGy6KAawgO5oFm1dlkNRyatDHMBbjedFg6ay7j0xVnlcucrN2ACQjeMnQE3ZeL21bRxMj5PSVTN
39b0u6RDBLz8DHzqQ1A5A5iWFTCBnSUvVsP/xgCN2hZip99xqoQ+OQULCY2U1v8MntC2t1PAB2RGHIRkAh6JAr
cHa7b3lAttLtnGZeecf1pjDdyj3ssxjslPTunJGkJYb7l6Chipa+Z5AQfWHIPQxOronQhrYpSG6C3p9t3/kDCU
Gml9/Pjn8GLEm4DE3GXINipTgE6OqOyvs4zF9oL87TRG3TOm8loE9/nF2drmRIXk08gdEmGX/mwj1J3TaDIDdF
w9RLA20o5EwR0Zjm95xw4dWte6JMvXepJatcl61eTLRS4RkHf13ECUZ0BLXuuFu3aBdQr3sV+ctVvP5tvrHO3K
n+7EzKoeKKnZQAOMW5mVIVF7788DL5aHvGznqEc1cfwx3dTeItVTwboHkA53SjXGQd72g8KpZStl1Nw3ghVTOa
+zGk4hzH4FpRVgE2r8c36AA8DeqG3FhWXrjZIAOIiC5kxRcsHnOmiMLquS0yJ+asxiO362pIZmWAD8LdMKVZSV
pATRSAy/kLxRusVDwpxwcoO95LlvN8i1OiR0uZb+nW5nVtHP+wC8K37oMqwEbpOAeQKKWrUZ/kaqw3xyYujfcS
bnB8rsQAQdChUnrBKDA3GjWkw0suTp4wi9YRlYUydDjZqZHnB8YGcSQsuRsV4scXf8I0GWCd0wVMHnHMCkbcI8
ltGculC07ZY3vw7WmuWIFCstYaWQZHf2p5e/RzF/4qTRhHRSlTemX+vgpc0gs8ZwvcTA8Prw/nzFgmQtvsYqjZ
BS107YPRQG16ZdI+zxCOqsDjCg5iZ8QR6t31kcM3SS0jFEj7TjmlK1ysItrHgooUdQIz2I5vpsuo7MaEfaKcBV
wpyd3uKUNuXjTgXE+kJwcwdFdiSYn6HD3r0rGTND30Yo2A33mOqC6IuUYT5gAvjZhKmbs0/ARtUQCrIFyiC47a
buHdPeVVG7gai5/32qMQAsf/rx5rmkhUNmeNrmScMJdOIajvB5SsGryNr7YFH2stHBVE/lT+nN8SntpTdluCbf
+lRbRNyhburqu54IEZRDOgPeUbPCb74oDGEIAPnOdf8QkYcmbu7n8ZHuvhDeO3BtkgXab6n+FuJtVY8ndGiP+6
uSrs8b1psfPJb2Gemu8Ircds1V/NuBhUhD3wLZruSwwVqXPifrVrWaUmPHKrOUYTdEMbLfSpisJmMtRz5cJh3R
4iVcoDcIsevxnxcxIS4OJIhOgGhdjNTF4bTem8QAQrgtkpSbevyTTsOe9Bq57Gy21Xi9EMPgLueIm2RraFKgKu
2Fa4iQiG20yVQUJl7hNexJ0Mrd5X7tupgd+xHBqFvjY3sH9XjM4u/KWT5ug+8QzwnJIhhCeoQFgCvTim8t1n5g
4DvhnbyafCLnE3yOSOlJSJf4y+GafDuDt3oJ2H9iOva58eNoPV0e04eV+leXdJgwH1rxZvdDThkhCTKSdbES33
TpQScMyRbve094S9QK3UlN5m6UMsYPrlwfyim/FxIDkbsktgAwvE5qrk+2EAtJDyGJI2H68IciAJXk2xaSm61j
nU/xqlbpLfGvyg1Kre6R8mGF9rX1r+qpTF/jSB/FFC42Zi/Rzd9VUfqhL8iyqh0XFteoFrRZ1/OGdO7kijhMB+
qrmvLco//y2kqnMbWI+cXYXNuf6ojfcYoXXEP+4qPLEz83ZFcYdCmcdvkNb24k6O7ymVgC/9M2LIB1q+oXKPWb
QTz0KIdcIYaqbUC44jmAbd8z7g24U0wTv1sfn2c4X7QL1iaBL/cmaZeC7rmzEmGGBeIzjmV4Bbr3l4ME4uAYuS
HRmz1s+jFDF8En0fCR8cF8eqr56XBTxHipMhQx0N20DuD/h7T2kLlzg2UDXvmKTiAsWKTFIttuMqPa+eHijvfr
FJuhTflXpBx+A2W5fbKhQfgMbTQG+w4+UWHM2KcSLnvIPigiBNTHlwn6Mt5Q0J1eYCvv74PrB9Qn8yK2aFc55S
JUmkqdEGSVsnt0pYpNn0AtwySS3m57L08eGs3KavQg+rjGylE0SHmuKaIFqA7wS/jJLUSyi1/wnHKJEX/sf15K
N8Z4WBQF54Fc94nHH1WyKq1dr61kWRpFg91EfnjEEwm3dEW7SiPV9N/JadjWQ+FW3bs1S6EInZAejoeQtaA8ZM
eaarc18Yy26hokz2gIhfVLil9eylvyInfCoA7tZgagAFYO/fYfOgXq3t3RVB+Ew8haL1zRfGF1Rj0XrWj+lamF
+3BsM97TkwY6I1FxzfkNQlyiOJDh1jAo+0o4SdHE5VaXf3kd+e3P0F+CvqHrDVEFts4YwlPbZwneIyzvmPsqMy
obPQub8rwSrvNB6Js4SN7suXajjKabd+FXtTtQiJj0f9GCcnVXBssRjSlcpZKPqcitbXAHDXp9y4SrhC2uXO62
dbgCfs4RVx+YXXn99HFoCLYCpc+NeOWgQ8Sot82ICn0BzXmYM1KmfR03D8SYw5V7uq1O7c/+QYE10aUIDWdcdn
BhBt6im8VItUXIVf1IwHU0H94kSK9Ih2kZI5gGaSMs6JIqGIJGSLR6RGxr5BSW6JRHs+7w4y1UouVOl79kYIOG
DkoiY/VL0/8ojDrfm3Hjq3UXvLHMU4gKDekPOzZeO6+CtQ/hcP++GcngFa5FQNXAEPU0FjwqrAf2pPZKP8/ksa
HZUQ+psm1ltUuiGWQF98noyCwECO+LemFddCz84hdUVZNu2NmTTAOnQ4vN4A2fpOkgSJVeMCLvO3H50E8bMVlT
```

```
GOM2E1WbYeS+VUiVjLdHcS0MBF6OK1j6rvhaB7C++86BrS8gJ3lLGUQxl2QKMne61crWv12IKni69fgdbP9mN7
X0Z9BwXYf7DkyPtRQLstL6MqhSutxjnXVpo7+jiyGKHWZbAe4RwDW2Llv7WIEA22BNDz6hhVt3LEtUR1dtYZ1z
w5SSWXIPRw4ghzJ83oeTeAm2UUZTtjMGxOUOyPTXT0BQGwOtLp/pmssPuF60XfHTqjqU7ykv5SHwz0HfB10TrF
Kso5fxCvKhNAuIN+PHCntv6Yt+pt/U5OpNW+k7wnMNB6FOIo/NwBUeBQIeszGUxDLamL6LYG9i6vWg4PMlTtOr
5ehtBcJwUB+Lg4U12xE74eSOXBBMNdZZV+j0etyfDoo5T5uNmgzHRgVRF0z2PukoD483Y9LNzdh66rR9BZWVKX
x6sybsOXvvkLre73HT9foQOHxkMIF0+mQdKV7NpsOddz0ww8mRGaEGVXAejhb0MfO3We4Yr+lDMFUWcEZnjtzM
Bi6p5C2AvtMb5+OoTk7BcbGS+JjM83mRWmDNRDFWpo6HNRwvVrEOf42CNw6JPLMhiM3puRwbobrXbXWg3pGiy3
CF15z8vJ3rhBMGpHWtAy8Rv0KsnekeOpam3WBKbzbeDTTSQnak7VeJKQM9aA9QF/i341D94kz5Magq1FtslJaU
zmQJWNE+GtEtCG8Pk6bYKxTVmbpsjYTrxPsC6Zna5BEPMxZ5YJO6UXXuBKw871sQVtdLjTknoECN77sztxxXwl
gKyiLeuFzG6UGdQDrcssiELglc/f9puzI8tGTsDjeDditM0g/UcDe3TQ7JB7KSr9N5Y+TH8I9F2LFaD4iiiJuA
BZq+0PJN30y64w5/jU/hZAnOpi5kgj8bl8Ct3AiSItqmFMp8efBJs3ckm5ZOiQt/o4G+KqSFw/aOafvG8gDYrr
5DFkHE3wGJ+Blw9abkZI2kJfFv3vaQp2n/y3a+WuVGHuWOHCCYAvnfqmSOfqVKoaTSonryAoaOD5xKYX3HPMl8
2Y2l7nri93fSsP8KBY+eQbUm245TILExmBGqRReA8JN5lwUvYykwLNfNSTAf+k8A6ILPugFJRH7xcAIZuydV7U
CfP9DSJbWh3FiaU8/Ca7XgLg3XOZVgtYXy5dLXeFIbeTwzxCZJeIxtQVtPjZt+8RFtEWJihY/cQbA8Vnkw3FQA
cxEsQE5iKQ1DO10FgOvzds9GBAUYkn+oWkexvWVn55km78W6lLVv/dfUd6olGiTiH4iuNAD8dcqyI0jTcwb2Am
JWTwhyR0uoMBV2Uc9XfJrpKfXrl9dtCSPN9YnMcRJ+JLVgH4d58hwdpUoo7paXURs6MXIxc8Xsash0mTJx4mgp
wPqxrWFG6TGjwZpbQLFSTjruU3fH75BfW4RkVPin5kqe/K8D7U2rRllTd9m4cj9EdIOlmlVZxGuts70yqvXsAX
t6d8/GXpUU/w5A0zZzeU86U5YxZI0WbiRBpxohRXWO2Lc5LOAu9ihDz1twJLB/KhWq6jE21GDEC9/d2PMTZiPg
glO7bOM4bN8HBCyfZMyXK2ciS9QjfBKfCNIdNpu7YCov6uInGpe4YI0UQBnTak9Rqlv2GC6/78o4GFeSp87RbQ
QMwSj0lu9SO6d0KJEhYbtLbgiu4Zy3Hdu0KIdoUeZx60bj99/14LWSgyxFbOlx/B8NtPxfdicajL0sdKsdGvji
B4Kr863egIyS+aEHTA/IOz0pseDvTHY6xzbPk7FiRopI04Rczvek9pXbnNeONQTL0FGa9MlFP7KPISABH8VnTD
vgLMMvMmmpo1dGtDla2mkjYNsOa6AdRTOyS+/Phaa7sNu1vC/GZbOJTxp+lddU2SPbFkJnS57p35Lo9V40eX/o
n7HNV+0v7oebHSGGYUibF76nEqoQ5jLCK/phEDAtz3sEfeBVBHyhAnQoQVjPZhJkxxjSjkgf1Mlkaxb/eF4a4i
xwqB+8bTP6k/F8er8IIt7HD6ji+puYnGoMRo7m/gx6CXyUwsS2SbJa3asZ+yo9vsi55w1d1Y/3zAHyuaOZ67D2
H8U8IR4VKRSnAC3Q5cl4qEmSBmDbKPHnSAtQNxJF/vpr6I8yaQmD3XY6tCgIumyE+0z8Rzzod9AIn2PXWOi+9U
rSrwvUF6M9rfXBPHgfxK97z4it4H1+TLa+XAf7aIAeffTT9NspaHlkJtstD1KWC23JOsm/IsxY3OE/k1KfNi/V
SqVT07UNGK6rHzOdg1nmtegGagtU32xvToic48XYHl1Cnkp+upv3tSXgSJd211NgjqSE2lU691Jmj0MxkLxiFD
MYPbsKtPXep+BKotfZ4uBDoSUBuZg2mA+Q3ZkdGmRVPwadvD0YFz+R3sGwXL/Q65emGcEvb6rUkBHdJeF1N0o1
VNcLYHwpEnWUUtNXjrerd7PaG4AzneGvDTXZHSDD2936IiDfiOmq4VD2mkA5wOqOSj/TbT0w1QYMjJi5IBZXvc
ZXWVvyRSmRCf7ojnqObeQguKAbC3c8IrQodOoM9i0/yoE1EymtSN6GnjJLsJKbWukIZBkdUGQDiDLytiVoHJ9Y
FG2sJTz5sE+qgRf3SJJYZQLR/zm7BnDYNOAjVNObzxS2/We/PgMtMkOUYXC49m4cikU6axnK5vQx7b3SuLoioD
HDL/SAxyv6uqWo0TdstZTJR2cRstN6mpxGNt6KZQJqTliRbo+M7OkMIN0ZsT44fGSOk7Nn60zRXoBXnYgIHH+D
8BgC1mUFt/HpEzbHsL/jQ6HTt7+TphQKxvImtNwTFSf1r9de3sNHAWbp1gJbVTKtRDMQbHOr+ADy9TjZY3ouCw
2U20PYTt8mBFl3j6b6k1B2mj+ajH+XacsGeCTX41gAoCRo28GEoL2tomHZWMFsJudSPw3aOvLt7CN/t2nnnkoE
gzIw7gWHGcz4U7AztxbGbyvLkGioeh4DlL86gaS2p8qFHQkCaQtADTkcalcX8bY0A4/iVh1or2NuoHgFN5C6IO
eBEULOeHnwDItKbBFoBEHxji5V4DKyxcUz/5sxN1p/crC93BcszIyPPl6l8EeQghqG2RnsJqDGfQAlRRgi20rc
H4l+Jya2yGVfvSFJrIC3BbwuqMLJO2Pt6HL954WHJMI1rDhuXFdsPrjODydZNiQfcBz9sf6W3ra6nPWgHa5hhm
eKLUvjuzNU/dd4MYqVUNOoyia2Y/a5Qah4q54vOtQAaFyHt6xAtiE5IsUtHAODN7NX7dQmLuS7SP/46b3v6tk6
yZRStwmokQc4JtizRIkUMfvoEQxzmCmujWIrCCdzhxr4gPkzbtzu6R35s3/mHkIlxShV3wFs/ZHrKLJPlfWa3s
jW9UUy2EWAHBuCQ4uPFb7HWx4GmGAbqXfuo6tgz+2t37jMMx3IW9Rz58CFqqi7qOG5TvRhAGrGU4pyUAkzUXnt
FdnizL9gBZElraTUV3FbR6I9OBwbpxeiB/T83A6g9CRIoiJJ/oJ5yqAR6dlgEhVb8jemIpTv4c7y4ES4fnhX3L
M1FqKwbPMTwhpFMokC1+lzcXuSELw6pm/BTIC/wYy5mCBjohNGD8xulEYxGUTx0cDOLi03BsU5Pi4x8mkow6+u
unSj2oILuCqRTOXv69vLiICg5Yg/MHAGp+7DB/RbD8TjAjeFC0Ds/aSpOKjzqF4DSnZaYtLDbUlYAJSKXClnI20
eKmNy+vVdlI6tpA51W7jev3kzjyRi7NqvPIw77N8LXZ26cGlC/CJELdFwFSjizKW4kYtSnTrZK570ntpJfo9J2
3rz5RgaSbV4CLfv770WzBdqjAMpo8i0kNPCVFmhdqgvBrBw/O5NJ0qOfpDm++9a5XX7tI9vRYII/W7z7mEZSz0
r6mCmKiJqkxcQSp4vxLE4UguKvDXZT1ZZWDKiouhH9xsQx/x4LP21XHKL+vbP0hRjwLNcvURqSVk0daAo2PGBt
wrVPtNqGfl6VmHyP8/nmvOFebkMd6aAFeFJryWe7V7taNteES+MJDlfdwpFMDzTwG+W2o/udyzwEiriHPY6CNI
EFpXzzXq+S0DfogU9NFM/bd3xuSqAa1WX3u6Mt0nf72bJ87T0RdbQjJnnhCfblGYGsabmAvhZwoVLlRXhvZBVP
fYc1UHYH0L1uGrw6tCg+EAJ02VZl1Pkzz49/dgp0mp0dZKmU8M3p20q8e+6J6y17NICpwaQ2OHBq7doD62hDp2
7vHl81s1/qYSYFBPaO5GMP7/Ox46wSAgJ9afkPgYuqEX7lAjUdBRuOue72Jvmuh4wbKy+owjHs64rAnNzMqjsX
5JLTQFJU+qjGglszQxwAMKHGnIcgU24Wv4PYKjXweqOMCCSwI0IbLF45sno+bQI7jP7Vv7E3i7MjI3XWn4O38a
```

```
a1je9nzYoYiCL5oOQK3gmbOdQWncYoXIR1scE+EJUbFEDSH5UsnMmZtHXXYE5xC9LFzu/PFiS2lPKnRNX0Ex4x
zKVSoOrn+Bc4HTopUoNgB6LkaN05qTEYA1tpVrgOt+aSZeBmALOG8pJ9r3Wsk9fkknIXlk4CySfXlmPet5Qt8l
rDcHXcC+wc8vftgQ+V0K6npxL2tV3BeCVw6TwAmIQ9D8szQhDGuN7yP7xQZIbDw7PRw5yVbP/SpACCz+hMYvbr
A4VV20bTqQoS0iS1DUXezOJBGk4I6FPz37wEdbxK4Y7NaRJAiHpjPz2NgVO6B5B8k3IL3hUGXLbO7xQq/GQViQ
Ui1N7KeKL6qu9JWnvl6KetoDpz1n2G8ZSPdbXTkfCAR/34LF+xX6RxVKhj3wxcgrS4j4gZoCuBS77YM+HMSH8e
UHOYGTBPedkrFmgPv03P0zFzOmW+YA+NA59OMFyPQgrhSYyGLfSLBH/m97a9ZaTrZ7pJS+3P2EC67FAfDi3Uwo
FFeg3+1206rP83ydm/uv/y6es08YQrnx49bLl7EcrasgJBu8iFJUOcgzlzOilzwGhjMcps81F6rRU8z0rwyuw2
Adyr3WDrLZXpMcegYk5G78PHaCrgxHV4wqw5wE4nZ505NC2NqS9ubXxXDyEa8xoXg8foBbFWG281zSnlPeJ9wf
Q2GQKugjopDeh/o8XNsdgiFEs92q7i30qNXTlp9qMSSwdRNAEagzRqnUaOYs6uO02lwJpuuuR6mLSpe2vQSD4A
8vwMrpQwK4KhWJDoi4P3kjSzRrRfBG9H8uYhEqqfKwt0iK/3+IQYybV7PK5jUGMtI9IUow19d5FFv23vzMzUbx
JrGaG4o36ZdyfiDGFhJHsg9HQaLCJwxFVtzBQ8ESilDDKV9yF8QkS7RR9iTRcKdgcS5F5RfFdwApvM8tISwWfd
zWwlDFweaC157VWP+nhfErn6M2qn8CBZottDDgz5xQ0+ALECw6jp17Td5C7F3q/OB/DFVswADzje29QapKdsIO
uNpnbTTsEEB83jedLPVVwG2bELz/Y1PapSkNYFdJjmp/XprBoGoULq79sdBcwI5msBderQThDmoyXV5+KsGDY3
0GeAknaXjJ871GRSQ1qjT/wMNwSX9BJrQ56gHIoP1I1NQ9znGY6oHDyov4EE7Z8E50O7kZ+YU7X2OaddCIT23b
VZ3JzWl5K1hHCZ7NqQm1y2pGgN8vYJ8JAuW3ywCSeSS3nl/ef66sHFKY3+vX9FbO3TrZXBPsaQA1yn7GnZHiUK
U8qtC7NnOieG4BWrYEkIk/BssnEpnc3RiF3gOSIA66CIvqL/b3N1jjoOux8EWiAHdG7GTMlNeihIqcfm8p1cmF
cHQzD2kRkPqIEEjNeW8l6vlIr5l2z2vkCcvM1h/g6ZNl0KsfeoPtpsmzW79wTCCB3upTWGEJ5Z/u5RhwnPepTm
hsA82xtcNQfCxONEB+mzbwUdX/Gkqg0RevVaTU0XnFqyr2i70CL11U1zY1bwxZ33t3xRpK+uXnIgOyao/nz+L4
wJPVHjbh8/I7ckLf7g2Bt5uQRsTbNUrJqf0oC+RfQKDqilwbf5At/kRxLWznbawNuZ1L9PtcrFZBr+pWmVIvyx
KqNuUirOlBsaOXVNYxGi865GM2YLK9/IkISk/IoL5BKxN82ZKM8OYmyhPvKCtqzIqwb56BdQC3ZkBCu/ftQLKW
/brsJpFd6TUh//j3WmB4pneROhc/yZuHydGYnlJNdJhxBxNinJk5vvZOdLct0JBsQx+/0wF/5wDcAV4YEDpJaI
Q885oa9szdu7GkZIpT7aAc8DsmYtrFN1YjEdOkcm5TflBOZIu0YUZKACTfSl3g7WdlR5+W0km6wthmMQiRMGy+
RSvvsv2OarC5rjQADHFeCEm3Lmk3sYN3RQW0119HNewz4P8tHeRD52RFjKu2CFb0bMIr26swRZGcmlEe2WP0Kw
2rYlDjHR3EQcZn5mY1fDv4M1u4UeNHZhjLl4rV0ySN/ZQW8EAChD6WQ4cyLSTG8iHOsysNMejR0jd0Qfak8FmL
11YB9Tm2vXm0tevJ6aGQxFkPLYrvFBS+rYPshuIlyYqBU6aO5mn3Xbwg1czBrmwognUxL2OFSeqiJxTdrSN9TY
hzuAI8vebRtPinpTaYJpO72S0ocSdUwbJTvyBcm5XfRlF2UAGOEdZHUEVdx+hFOZ1FeeQ/p8rALXnkM3UORwW+
GaIJflXNYdlSz/53IDFCTkc+eQZycyUdE/yEPLLAE+athQWNVufVZQ5OPBAl+Nx7fbpHOJ89WEotm5qSkbt+8F
XohONt+J8zsPGvnCnkXMG2ejBhVVfJu0gio/wNyuUwh16TbJ9Dgw+vJcCgbK/Lr2cpyES2TteSMvRuzvIW/hwm
kkUJgy9IBL4W4lgndDX84Orxm1sxu61vSA0g+u+1HDzBTf9dT34aIL4MGKuuSjhYCh8b6f7jwP4pQhwUFyFQVM
ajMNKWpzDWU0apYhOS/sgX8w214MoSZ7jo5ulck+sbMQMCr2RDxV/kwZdsPhKvlM2pMcJJxJtLQFGFxzNcNNU4
9wjxGjGor6ux4aAIrbDLuXN6+g4jC0qTw4UZZd2qgTNLm3A4U4dcceYku/wgYdcJIjtGNLmiU0PkX0kF5AfL1j
7w01g2d2NJef/XGi3ODYiGZl4A5Mfj235e/WBbkMWJvdNa4U7CkLK4jx5A5xPoRIR2fOkbMyyvJdX0hGzB+cRQ
kHaZLFTTdunUGLNyu6x81ir/mmXO3qbTEypisefjKgE9g5JhXmJAtWX/YZtNw0ZeUwZxhCMlbfkkeqXQkIi20E
8165sMQX1HTjATI2LIvpweyJHCSyMseHqba2Qd8r+mS0vhznc9q63ZJAFjllkmOnA9oYCBLhPC/1ow12euoa94
gNJq83bHFx5w+x2Cd7nGyL9zhTHtFEKghuvRtsr3SvfQeq4lXqIYWcnc11fI1cf4SM5KGQ9epeuXYOKwnm4vMG
FXD4PtyjZbJXLTnLH0g8MtHJB9fZVV6ZcG7kv+1sBaxDf2gHt6JDFIu3/Vdk3WqA3NCWpimWlHQpjWA6xWL6Df
I571ZM8YV5TB9BFndIJMguCwn8agCuRDT2wsRLCjD8LH4fYANBsEmMO9U2idy6tlwIqWz9Z36UQKkLC0NL5l0P
zQ11RzpLNLHqJBQAoiuGgv8payNsRsWKks7jHMomrt/ruu+uz7rp+vXCf7UKmUeDmzDwK4dRLSuIVy0siGwJbI
ygr/ZxPK3vi4J6PKaYnq9/miotTW3Kqny9FpFT+txMp0/2mfXFXlKxGpKaG/dXac/qkeHy62eNuKYzhlrDboYe
ofNbnYTudFHTGMU6ZR0c7Rz/JgUL8Igaij4zHQIPF0oxCREwRmeTsOzPkRC4RVo9XMbohX2Ssw4VrGEtAIfAJX
gEYOiYpphGS4/bjvp6yhvuhzXV1o1PUu3w/keU+yBomHTIdcj/3dO2Ac/u5oFfLLF8BYLst6w24oqa/p6zT4Ap
Muq07y+Bh99tgW8mf1BOvL+BanBWVvq8BIkF1kckbIJ2UVkDy2DObLWcxs9AYs/uuoqOQi1qgCGDgJSEUbVbMG
ql3nVjwSbCDzsH4ndETSldNVPy915Jm2AwNQ456NH88BzELqDYeITEkcGhqnwfhXMUIh1tVCz2GU4uR2U22N2H
y94iYCDoQ11seGtKhapzZ7UyK1gJsHZeDN+2RaA5TD6D91PmU1oL+dYwE/9Uy0euvVFHcWukfszuP4abBT//Tf
a4ALsZSVj0KTqhhNmmRrxD6DRF2154qJHr7pAn1c/2x3vxuGN9Sg5mUU3YUROvOouyQYD1/AtxAXLUWR5wu9KM
qqc5w+8YXOedogBrxHEBnjY+Ez5EQJOu7+1dkf9TaSjOaUoTALdUAWK8yqz2AXe1m/g/qJstJLu+mdGWbc8zvY
nqCuk0BmpYGtTeL+A0h1D6gZ1XP1HLwWIdGY14y21G7cz94NYxHj7OAdyorL8QoQLSuQSgltPcn+FKuLM+H9S+
xQO71UoNCysksvJUihIHlXbAgBb4L9r/N90sW1+h80bLXn9yWZwO0OSmVG36fBTMQgvDWTN0+WFRg6A1xz4YCM
hA0hj3OymvDBfHQ8h0D6zJS8yPHEjxWm3PxGx/Xy9ME/HoawNe4sv0LNuhf93aCZNcUHeVzFJtZUA3knyp0gaV
zQsXDf20RpxDGqRgTp11o3HcJW2HsbF5mMk6GE01SYXMxPh6tzmEP1mzRd7gXQ0W39l7eX7YLu/8EW3HN6jCYE
6xZ5B3Omg2ad1YhCQC3l3TlrKJZnaGjiHzEviGbH9TwKS4jt5RTE5Gzx+9jDdUjsGyoDyrLxykI0H3TV1NNRrI
```

```
CrZ6p1Oakw3IYpNDR/TpUzAQ83dlztL6vHdAldIEN57iC3kkpC2vIuizjD9BT2E23eH/XCKNswbX1aNa+T57Bm
IhdkqGnYq4okG2SBm5sfGtYLeAQvG+TLXjEnZSwRIBzx0OMNhe/oLjxDp+ognY3MqaXF3lVEZkzCG+C3JB/aQe
4XaPJwEgSIKj8qjG1as7CudpeE0RjCWHBH0xGpr1knFffwzB24/eFXU+3MxTDBUdd/6rhK2SCWYWhNY/HyXnPD
Y7XbyNkucp7PA3dbuB7n1hl45w93c2KsQavR/6YO/8MvpakAIhODJdP2w6nTIdfctwQPLG5PW0a7FvEK0etx1j
guJzPn7BLwU7PbkhrTMMoReWJiZ08NKMndexgTvY8YCkkPupYWrZGAuDvfUyiD3iR/uGBOTRF/p2usdgO8csaH
q3wXiBJorPPqP3VazxWpcYhhu3dXNX9zImS2eXB0zIQYLlaJ4d1yY5KYEbElfXbx3QcI3/gdmR4golflJ90NZ+
NXuV7JXbSoTCOcCb8PKEQSuy69OrXPJD3TM9Nh24hMmYnKDon7AAZ6ZVti3pgNwtzh4JExloFjyVl3Smjnd3zJ
Hk1qdQ3Ht/JT5Nx9yPCoj2z2WLOio4WDQpFOhp+1qFlbUfKKvxz6FjvD6vP1fQ+3CG6em9W4ov6hd7FPr/HwxB
PkGu5s/T5LTZYRCvp9HOXx4QobUsoQX8jupyZql1B1GUyFlDcC60kkPnRluF8ge+ysT7o/93RzPDY/dpRURCKR
H3NLjbvsCOmSit9Sjkdm8fuXehJ709CAxS+27op+EcNjcLCtI/4tdWdVE++TlznbPts9aluH4IqZ7Lv6WomSwk
joJvoJ12swssWzOPVaCIhpdTYCpn1xWQ9VGLV6a/7LOc5yBtv2UTgCQYW3eox91jFuCAVMLGBpX2wIx84aOen+
4OxodXluTOyhEslRcIiuMpO+k3p4A+Fa5kDw1i+FABwrEQpbpoT1wdceUDFDwv4B3CoFJ+VKsuVEx2/Z9kGewF
lL9FF1RPrrgLAyT7PxNHudnvp7ZmP9rxrOR5x6geVvzG8JcfQiMsflR4Yi1CsJgHl+gOdTvwAtrkXZLHnIumgW
PyYYMxn7G7z0U4tO3OzpgtadTfqKGyHpunWq655vXYSDafSOZAW4qJisPZgnhVaRbptogVHUqRJmk6wuw9GdJt
1qYZkrEOUTBm4VXPP+MjqgpvJkxILjzuS885NmL0lU/17rDhYJrjFoorjmU8/a9TEaQoHqQTIZuRjbPN32hKEk
ONF4vuEwE5PPmaHs6QnihZf7HXJIVoe9fts4JrtwzenII/n+7Md0ZiCWFcDyhYteJCBfHg420G5Z/TKp29jxkz
Zh5jCQfUusDSwbU1loU3qT7qCrZ1eXBBo5vopvTjNQ/b3S99enDgQmDwB2JqKfindHooiPrm62nRS23oWCXOiq
fRv/TtXX7rRtQJguh80l3NUu6UcFdU2ZJFqhc/ERNVSrAKJWpALklgT4kQ5X/+sCniPyah9m93iQMLzK/qsBnh
Uu4YTRL9GbDqBKqw6Lhs6Uty7g5L1LwMrzgdZdcgUfGaDcjuLhwGIaNPKeLaIiHJuRsk7w0zS9E5MCT1SbKMNl
K/LxKGnzgqvYwfmn8s890IlnukvYqrEanT9/WMLV+zTqUb+9FkhUWX6wkq8ffRKplFkBU3FCVLL4ZRgfluW0eA
Nf1GnRLlBGOQg4Vs+YuRq3eR4UqRtI8J5G5BbjcZMgr3knU4hth+7WzSROkZgKvQExnBPNcxWBvynG/uijkhSe
dF7PqMgHe8hAuYN+Lj21BVUriUN3fIl1ORmR2jK0zxNvexTLw/9KKOCom04Z+d8vzF9kn7r5Wx7VBPUBwJlDCm
lRodV9Xlg8sY9YzjxhymBgTUCXx/ErXVKemffZ8Co/A4UHiRnJvQAZFjOO9w6Rjmg/LJCH+UX/RWibih5iAMSG
ewsNJUz7hq2qpHA6CSx29DIocnU9B9daWi0Gk27y25TGXYxuJAZ6GZP4yUGq9ehimZFDjZp/jXSmQMytDyVPi5
YHDWnTr6DWDT7bWtA7iMk1Pzd6q7UMOY94bYPU70GxW7GEOU/uqshZp27nhhPKkHKtNf2QiVLrltErhgGtmPex
EEQuVlhD/UYlE7uZi2BqjPFSWcY68WGFL9XOc1j8pdYwr4UcNqZ5RY28LCa++ct+8mpchQkISKwPnIrmEw1iSn
VNcB0IN8bBzSC0yEKia/24rWOHoxMsfdfm39pWAnPHl+VyyLiyutP0zsCG7mOxRG3eLUeMt+p1NLwG/68ryymX
V7vxRtKnQIZ3/u/85M/cyq87PsxnWZZ+m9RcR0FsEnokWz1bj9VL8tEVfYhGKiOUyUoo3dHEwOABWfnReSeWX0
ZvuqXzyr9HPo+wtQXJ2694EW0GW7UUf0mtkhd7CTkM6iVXW5NwnXFe7o8iujX45pIr3CkwhkTAG/k5ThmHXsqy
VbTNy0kEGjiS6DYnHY0RWktKtacucuci05Z1Zciy817g9tMMfhdDlb6BaTlCmGg5H4yJOVrCXfMNbGQAe7csyf
VNauAvGMp59uoTcpv85Evb/7dCEuxCa/ItwrI8UkZf3ia5p7zwDYZ1fgqXjwYfZPGJQJs3B7ACYsj2/7qcosz7
LDMxZQEnBn5kgxtDIf3gF4I4HZzzj8SDqP/Sq7T4JYBNUu5Ir88aNZEwN8XfySWm5iskO6VTDZreuz8s5fKMgM
3KL9Zr/1VHMQvPijxPkGmUOkfgguUzeGuOxqf6uC+JbWjZbxnaRcNIsLC2FMwWFPEMzlIfv4TP/9SmcahCoECg
MLH1FK4EVbFtomAs9Rod7bdO9eM3blDCCxIF32ce4RILnx+R/0NZ3u14XtqOJI1v5kdbwxeRZ4JvZ9l5cdLrXN
WKLsxJl8PHc6m6k7uog8EdJMjyfC81ttb2luUe6YOMdU1W99Yrr6YquUJl4ICJfQWzfbtNDzfb7D4XJYxA4j9e
zwHL0SbjFLMB/WXn2aEUx9XdURubbffhwrsrr7tweNxLoSfqFwcaiwik1g+FaUucKvlIJZZWYHvmP1H4MrmgEO
LP1hky6OfRNv/4UJGfGoM3caoieILjTSwYMHSwswihvobeN4N8sPImkjpomksq/J2USgUK/Vrq1pi8cS6S6rdW
0HYcMhmch7wotyVfIc8NTer79r327zIJjn4EH320V89B8RSEGyyJZHn5JGc4uPMZ+AdyOnr3JPzuZcSwog/f0Z
QvEEzrUnqzsT9vMFtv8Jq2K3DT3huyJdDlLeZIZA60LQdd8seLUHulvdkzyKuWv4J2UXP6vXJ8OUoypLi7TyTf
VGcWnRWApHsGxwKn4ZEr947QRmxrhT4ln5S67kMqiP/i4DNBnluXIie6MZQxkp9eVnIFGdRU0fVUdu5IK/t1L9
SfyH7asWXFRWQSLACcvgvtOMIN96ZBfqp/LNC5QSz0XRUQWJ/Z9NdGqavT4aODqyeL7ybqg68rhyGX+A2aj0wR
Qnk7cJ6OcgLfRXfMdYEhSq0AIoiXq2PdhHxn+ItaOPxR5uppvSixnk3dslwKu3G3IQEFZQ9GcRhsaIc6Rt97GA
8FpvHs8oPcv8wPutunLL0D7f7zGjzLJKhPpW8lFY8gs2mVZtr/T8ZSsC4mjIzyn3lhx2SwwQ7iN2AaVm9ytvTE
Mh/UuYUL256pmbQiB+7Vx2clnz4OxxZyGG1K6hedxMcOLP4LqhiDjT8+5YlIu0bMsmSND2irrbo7VL5Dtof3cT
i6esykpjxvZ9sylBUcwyadgk6kqr+JiGHcTfUyMwLwQfHFiOZwCfMbYowuZ90d4EUopWUDOTQ1zqMb471kgML4
Oi3Ok0ngISnDBU6I1ytZOEH2zopOfvwjCo0L9aC6UAt/oDPlFf86pQQFdj6q6j2EJKbLoDlZHm00yhg0dqKvGh
/Oh4gUnaUuLjWIvNDnHnei7T0LQONrOHUsJObEUwe80mwtabjDgFuQWpmOE3wN0RYHAyFXhRKzGNVJJvju0Sji
SSCufJkYnu6xDNR0lxMy+1U06JNcU2hU2sN7HzfPH3p+gMWlCzw6I/HVCcaYC9XSKBRqxRz88V41uTNsmADfMN
KMRT82id+BgGyRmBAtLo9l1h25tk5sXx/qRFwxz2X/vLH6HQjwSn7vJBJ3nXPfasTMu7NVcdI7u5jtFqYTh4Rp
1m6T3h1mJdfRt7n6YdOWY15F7ikRSokmB6q0UC/19XS+CvOKWOz7laooIbSOguozGFTtVB/1mIRr7d4IZrChtl
V9U+YZ3bHcgeqUZnjTif56yBK9mhI1DqxiwEBXrnTopF1N4pLLC05awYrk7XtCoMz02YUkRN0bXrusLvUpX/in
```

```
fz4l+9NUqPNGZf4+ZxBFZXp+XSrhlEcPoJ4sODJ2zoXxbFizDCxZ9Xv7lwE5sH21u58ZobUvR1MUnwo3B1oSIT
KwcC6MyGps8fZuuhbTqPbW2PDnORHhzkZewg77sD9/cvLO60OkqoWdQn19RvsgIA+p+OlZMnxB9lOw7wIMJI2z
rLcq015xx9CdTauOaKB3+2zyCECPdVykWilZYcjxe7C6LUsi6Sba3Bpo7HLW8Q42khez7dbMLrui8sMBzTJACL
jAe63nKj8hgzbCQk4Vm8vG4jYe4HpO2BxRQdkLV+RMj/ZWTpQlgIqWUucNzxU3ts1VUqbUIOg6apetmVWprhDd
jXx3+8pFBl2dlkZ6FDOtVJVsY+kCc5OCF0Fg8pOvE6Nu9IUiLg59sGMBeQmjFO/EqBjI+iVDpw4Lq8qot/mr0a
PJeXyiRI4FfG0x4lOKb3pcBIDF3fh/SPyqtYrUOcsgEPFgK0U8jHrGqZsd3kIqm0Ruyz1obnRs9KaGyuEkaOVk
eJ6ZTynyHjM/kpnxEd9bBwWRR61lvqMYvTE+3fTjOfR6soqZMVlzFoYOxLYi/M+SRFYrZewm6noGzaNPgwc1zX
Sfg2qI+PsCsdYQef8bUeXY/cxAQXQ8JSI0SXZP1yxvorEXG1CFU5A3GNeI7jVwfrWTpdTyLrain76ffHMOmkb8
cdFedbXIjeBV0klkGtT835F66PCXe7v91F/qWFpEul9yAl3naVrjkgU2qlQTa9Hadrk8PX1IFOoXeYhzc2nI2E
oyVEhy1xlZ2R/geqSMe5kXaUKJeNh7XRmxRhyfwbaUZljekGfJJSEatEYVLaBusXzdoakZPJ8dpZn5Q58HSQFb
/VQ1fLzeZWzujRFB8l6xm28H3NHc9X08Lt4uMbxP2zCgDokFlEJazrSR3eCgHsmkI++2K9JJZK/8/hh8TgrbcZ
5pW6SQ1P9IYEmhKpcOyQzMgTAXh5WsdCwv1jNEDyQQtrPpEb5RJmDrK8QEhH+m19ke19fFAmJ+Tl50//Jx+U4P
FEF8LLvjjfIlCpsDCPWMkoLi2UQsLSXQ8mvZ1j0RGBDWaH/uSefdRSaCDkF/1A47tA5oYtE3iMsk4pNJGZPIxl
HdEu9O94xpISOGIwjbQqt8Zh8zwQUV2N7QR3LhIuh9544W8uQvsw6cfZis/Co5TzacJSbQG9pLYZjFD33SYA5N
c/OOxEXZHZVGEqFB41NQUkSB6Tj+z1QE+pXmxJuekjNwYRbhvieH3UGw7Pchct4RB8NjviRi6d+Fa94vpx99o1
Pi/t4H4v7z/KEA49Od2v9INiU27mH0AcSic/qqCgBSZAL3cYVN/BC6O4lEBNVSr/PXdQTe6VXe95zZvrcDIzuD
7NWNpV97Hwn5IqJH8BVudjqQrY/03eZERgZWYuJP8xIs9RO0Z/8dUYBwhlZiwpCgDf23Mh3S5r57VaWR6SBtCQ
lZ+w/yU3h8zugH+5f6Nv+KrDt4ZJw87sJ6ZpEdnIl1ilfuST2dTNPWDNBvh0LSOFe6E2Td4d2t4jHtE0RZqEIN
B2Bn+52Z/bJGUTfptZYooAoQlfyPOuC78GlVEV/w130NTMUOlPkKGq9I/g2CJbYR2ePGE+y2lXOAZud9wKPgkU
2P90fo/zpeL/sKNEVSX8DOVSGagdFzdRK8WSLQjOCqP78YdXyEgSNsBRZS57NqkpKJT9gC/XS6NygODr6SyLH7
1oYsVtQv5WOxbEBnmSBVgMN48tirkN5LDVnKKdTNdJalnZrMeBRp3cCPCCqOk/dv5SlzEM1Zok89GLi65jGTu4
IZXk64l7JAwaV2JASDB6azFw5q3Rq3e/NxVvzFum21Qn7h7BkTXlXOQBLUSvJuzcjqhMcbJ54uD5ApNx+n/OPM
78FPjNnYHXDAyYNugnnng0MNI34aI1GJjBlSrmHphsYHaqmrPxQ/dAVJjxnywMPsGKQxiPqoPnuV5I7vkY7OkT
LEfgBDM6U0g/D5IrdG8HvBCY+OnnJld7vcRFCujO4dM4CaliHEfdG1DrQSgFz7+I3WNydTfzl9k6cZaezal2wO
o5/wZkKdzT+qGeCqAi46OgX0n7zN4rVJKZbMAQV9DQCDaUtz1o1d6k1YdRHWS/bWQriNE2X0yWyokvobZnLLf1
CXAzfeRErXTfQfapPeYdh75160IoD/4hKBk5QcQwrbW7XVceQBOpilRXnshUl6y5d1Jc77VCuC+QB+R94S1wZa
woAKtFprRK6B28zTgt190qsDg8q8u2s0wPqbnHaaOr7hJk4pGBFXqhrpagR7zi5Db0TxpjnfY0rKVS5al7lUMI
iD1oRhZ03AoqZtv7U9CGuc6amN3PZKn40txNPgbjjagU39Wo+B5Texp5FEmsiugin0X4Q/cZ2BbjtYQhWoOqce
wj3R8q3UcWItcKUj/UAjtKzqaFoNm0yUczxcJe/OM01YqesN7KLJTpoPgeQ+7SMkD5cL2YD34S0ciYiBD6qagn
5x5lgi7pv+wR3WlQzw+zfVFM5nVKQnW9mhWi7r/oIPj4y7uY+zG0Mt/fXx7V3/FL3dQl9cR+fbRjR4KZP9xMSp
r4MqspA2hwWy/7dt+dRRut5LnpjcpOPI6FiXr6GQS4xBoTmbTKh3ANTun+nzsT51N6whkTUqHXWSGXQr3vV7E8
GS+XV8mQZRb+A2FZ50w7hE9K2KmM4S1FMrOfGzuPTsbn2bjGxZoW2gx4DDV84Qy1ezonX/twtk3RLwUhGt3NhC
o9aj6R9W85ygnxAQ+/w8U5wPypTssRCxwWuxTOsQMLAtUEFPkozJinnoilyMRakyhHLL6SuABpCiftOA2PWwJX
U4xDiT2AR3wgXL2SlITZ5ktDf6rmTVEtld1O3mMAm3dLnqda237IgZ4+GaAO1yB4MqQHsa7LbIJj3VbIwST2Fr
ATGBOV3YcgpZFZJGte4RIIzi9f5J2guTp4e/3ZXhKM+95uHIp7dcnbDJUnThs3BvNQbuIfpBcXMpWRnSxOPzmz
Zbju2tRc/mqvycj6YJ62jIp+NAbdrRAEP4txx4Nhn1pb9yAIbeYlkSbwPZ3e/p7os578qB6ByWpqZuwlxO3EY4
fcynKeB0szrAehhfHOqRY5pvOj9gc7sHb0/fg/NPpPsTfRebyW+oQ/av34MKZnlSJQexthBMnpQbLBhgiKiLrz
o62lbGaylGGv01WwHvTk6A0AEyRkPH1DEKnFjLqdTEhgOdWWmxvZDaHSnTs0bVauoG1M9F7JNtDjS37Zp2r5Dm
lnPlm5NlwLssD+E9Ax+VZfh0g2vWbheh9ERWg3pScl7S5wJWOkQBAZZrC1Z8qzJvHm56iEZ3YfVWSivZqusMGe
sL8UvtejdBws51zhZ9HBIBn+pCUOEkZ8nZD0Asq8Rcd5qOJc6UjPFZKJeOznTgc6fJHTn6xifsfT1zRRg58E18
t5T13h53gd6dgWJTDRlExvbxN/NfDudvrkRp2IxRNpmT6RT5FWREbIQeVXMii3RBI4Mu4+CXRW6/sMgNlRNQom
xW37C7LmmANd5uDY9BxrOrZUdDh0VWP3ME1Ess9xecBfkkUhp5JajOpfZ3Aolr0IokNj5cL9He0i+ZVwm4pdNz
mGiuvF2Cm0G/iXlvYqHeKXVWN4FkRCVDFUpVWRhAVqYyZSXQbQCkt8SvKaxAdMPAg1vAb4gjd6lHseUp3Jgje6
FFJ40v3cBh0sRQB2dbRjUmuWILhPfOG8Eg9vbPZDt9C4N3wCom3xFMZyfeiV33CBm5Kcu+fhtVtvgCWdfbxGwv
DOyIZI5UNHv2BQFvXNyo2OOM4+MLXogMwqqm7GMTukq14yzh9jGvWhY7mMMUf1F0AXUPBNCjif3Wh/+F4W/us5
2DWknxVtZbeeeyuxJ3CWi0u8awokpwKmBpbs0KZzVmeZn6rDTHIXgiQqqqnXpIXaNCxDa0AnxonOsbahMhZWGz
yxS8GX7DLt6F3IEyG5pY1XifhmlwtawVFyqUyxuO6po23c77I8RFsSCIfe70iccEK4wk6Wtd6GWiJJGGnuoP/T
rJzheazeEaSwH5q51HUdzj1rW3v4xgE7dHYbQ+r6td60D30IO7dKZ2JWc3/IZeWwlJoiyJun5SONp7q3m/MOel
TWuxw6d9JGs8AmOFjT8aK4Kl3oS1ckrrbSD50+c0CTJbih4Tt6lLJR0+ikq9GpwMkd1fM/bhyavWPLTD6bLPOB
QydvMnrYJcF58o9+6HunNywBU7dUA7Zem8dmIuye1rT80H35aOvMBbDTvdk28GPK9vAW1uwqpDygLlTnSH0KFK
yq8FNT+a1Qq7SF7R201MzpQFs5oeBo3RbvbPXgff1+CH/UnMvfzWXp1wFstyp9G/6V633kfg6a7VrlKMpzdvb5
```

```
4xArnmdExrqKmickFAMDz7BZbE7JN+G6VHgdzwFRwNVU3sCmQjdpB/5GRUD5154nS9eglfurn8WTUuGrENVsfN
BbvWBIR2tLWCQT0DX1mbjuVvRKndjHbVTFf5bPRj1YsHInxWo2B8pOeswhtXwhAAy+OA/RDaCi1V/EL1M5DqhV
sMN9P+NT9ZfbIcZnXt821XGiVpR+oXAzzq/vM3IVORDdYhGwSpVPIbvwjKyc+l7kc1SJV3Wo/rdLPj4BZ7Xbtf
krYYzPH1dBxAVHXX3IOHHmoCKjKFhgJdeUBmBNIsy7dBpt50rs8d2BZA0wpjdXUiWf7FvthI+DACxkPF3NX7K5
9g0oCg01m1+nELLyT9m5FCvxd3x8y0MNtlV5qm0826L4vgVRZJLWkZgvt9tbHcT3s4wp7Riz/kbgqxqs7ENXgx
F5VjHiS8RQVe1B/HzP0aDX+9rV1RUggPGgomYk8g5a6w1CSw1f9X0pkUNrUZw0k5Vo5kWYwSJOaGSC4O26YtQ0
+TSU4wGFp43OCIC+/9AsenHN2zsbh0Ma1JKXMtwOqfoG2l6AiAWlyNILZv4SJ1zFXV8HiOWbf2HVpKK/Sd2Oki
7ftP6xcI6dua8AL9Db49drC3xWFhtXshhNd7xuDADIhbvPSJOx6LJBCc1DTxu5+jFByDaP08G51l4LZzxlkLD1
FQ8HtmRrzCiszj5eSIECC3fr+BpGpK418n53xdg7Jho9CNbqzUj/4EtCTtx/xyeRtJXGHRpsyETI065B5VtWH8
8Vm5zeSe+Jxi6fEgdfwKWmvuMUgdBIWaQ5GhO2Ha6SX2+C+BQWuhZsdyaFu/HcsuLpDhUVg6BxEuxTjahMDODS
4PrFZtSXmJ6qYjheIaPxhYYZczRlGqfiA9a9ap1R+mB8aSbzNFK4RCMbWJ6hKFxuflJgAgJMZgK91p2s0aV1Jb
iPdh39v5ZoaWH9sMBAqdh09rEZmPQhB6r+9H6y4pQbmvSOlzzJdfNk9XzHq8k1/AClKDyupUIPiYa8JtO+OYnS
rJnxGXmeBHLke0IkpEjhq2rc3WmecujjkcthRs4PdHuoIUc0tjAua3iN+DgwtPfXnj04p49NpuyidpbkaOzyUr
fXX4BWpOIL9PpWSOVTBmU1+Eh6OntrYjvzO9mBEP2OvwqZeT3oZDRfqOgM5tPfeIUivSsoLuH5es1pk3bHSr49
V10l5ahmrFQx0Jl29mk9O3blXoQXVW+/6iIiUV9l7zrpuCE/vkL7ox4KnnTPvXUOVqQdXHs8orbwl9tpGL8+w6
B2gAhO9l2dEtBsBhvURONpNwcAd3/6KzV1vu8t01UdhPQE8tRSdkZb30SnOjghtedUmIeEVeq5ozSvvqy+d4/B
fQy4x+sg1aJDB8g+Ka/zl85P+fDupb39YEC8bQ1A6TKDBLAPYhvznMpJPujAl/ja9/TkQELoJSq7eSx8wwAZfw
/aHRaS2S/Lf/mn/5uu17F/wWEeCz0Oak1cFL+Yt8FcEO18HtU9ThZ4UqHqtSPi9SanjVdS2gFeAtIwNPMpGs0s
1T+J42Fzqnyl19zGfFwaGHyuCeZTWBsk/GHAMT60MvC9hOho2kryQZ7XANAA1yla4pZSUlkmp+yi5OHscqSdqO
7fgAOt+NRqCi049ilS9o650ofPiHQGfsJAEoy5oZWAF2wokhTdrzSTJlDn5KiE7OzUzfiiP0vpwi8WCgr057e+
30XxOEf/kJJ8StgR5clxXAU+z69EDH0kBImr2sUm6T2I9PzYtKhwZRf5WFx+svlQ44qcUwOSEasvKBmce/oUaG
SrQpXMSznW2cwJywBreuvAQ1U6wwLutgOJZQHicYpeQhUycI0KIS2Yw1GMdnQReXyAWW9JgnRX6wV7t3rnD/Ga
kn3KZpUYo6ZdIrByFLEDgPpj5MLHIC+MbWUp5O5crB7eSg3Kr7IUFuCTvyKKVb2wLgzq1iihPYYDgLwpauJBGs
+2kCyTX587ADPIWK6Y6nue0G4L/wHWW6qVvhl5MqNNAunmY8qHWRDyBiAqri5IGtqK92Jl1JprzYgd0gz90rR4
5Yny/hbTKvSw5kHzckAXQ0u1mtYPQJZ9RluW8aXf1fGMk3twKZubmKqUNCIYsfbVexTz0BvaIFlNwuIKhQiwVx
5t1GvP6JtOzdLoeSz9juJzqYblHQOnRR7CRrciUDKvq9nWCeCxqpwuyBo0JDLG7sC62HBDXMZ66J0uSmUsvtx/
wZozrayCSmEYBDGhr8ixjRPKCHAaela9TnEB2F+0HbKx21yKk1TnobRlCqlVTuopherz6qUsabFLdyZ9w4iEF1
2qRH8uGjr4DI5Mtu5Zz3fy0mgzlY9xvds06fcpaVbTKwsHWKrqJ7aThWuKpAuLlOw/1sBFmcJrUr0rdROIAuL+
YA1A9YVCK3oR/yPfXfpzGgKpBFH4GEWVYvvZdA/TrcxcgKMTe0uX6W4Bpj8ygzaFQNB60JPnF7PKsu+AAIbL1C
Iaw1ZgL3ZNot0NEEkzRq/giGkci0vkdBGqrZSiyr/PZ/1A52ciCAqqEKbL+U94ds567t4pA9CZRCbmJ+44TdAm
xzZTfV+rzDtyaAygJDRoDaiCAPWxeAE6zvE9BTDjgZ+1yQXg4UMhRzPfXMZ+RKxo09iVM7RRivfmpT0FrO9fbG
Jvn2+IMc7zzES3iF/8BLeT50agkqU69ww97CKSYnb69egbAg+/Q/ntXEHUiFeF6JlaO3riq3+0c6Yii31Kp/CQ
gR1BypOKU13LFFtJlO/CJ6MCubKM2G9vAGe5zervFlf9oCxvDGudGx6Wk/H4szgPDfq3N2otAYxpeAX4m2q6vy
dAS0sbIlTIL5UHf/AlbiQ+AIn+mRwTgIwg2e21uybbZ+7jg3LlgtVaffRlzinTt/h14l9BoyjDfCGEj5kvD3gl
zEn4o8S+MvpzXTdxhDmRuw41Ty2RAEyNbAoHp5tGLupoZJw/JyJODxnQ7mMIow+pLdqEa4fIxFzEy6+WkJ+jO6
1H9mt1DwelMi2OjmrnZJiw7vdxprxki6Q94nGLNQKD29S2rHR0b1vIaXevIFB6yjlSR3dvh5c7VcFiNHnQhDNM
vJIWnCk9HZxZN5JDOAlRYCF8zcG8YhKMACem55UomEPqhfEySdXuUGydep8xCh/qvIPuU3BmxJB79RrA3q98Ks
Anu3qsoyV9RjgnqzNkskre/7djaR72hG3O9VREBEVM3rnJ7gsmH4r3vLwL+hvuhAdL5O4TNgv0vjpruDPxIPKR
zXDlM5g6UZeeWCam3upgf3LVGMAh2tynWFnQGKqD5nR6AW8TulwxmK85VA5YMgzN36j+GsjsNwN0tmzgtsFXDn
O2wJc0uv/kUu7YH3jF2OH6V+7wl3C1LEm7WZjRL1PBasBkpfUGqT/tQUIh74VNWj+ZKDU0Ub/LpoBpSXgEh9F1
pp7fFXdU/8XJ+r8gmVAucsSU10UTWdJ4J+anJBjCHYv7ZJBINZGryaWtYaj4Iyd74yp62jiiHe8UWAh2458WP1
0kewJTUhX+/yiQBPomMJdAXn3brQEyv3ikYxlSZUWIMdCofzGDavDOEhCLO29EWTHeifrVwiNe0q1R17mpfb3o
CRBPLHPebfof9SjmW/6jFGByyQUsTbkGSXoQutBVl/KrSyo1P8cIlJt986uZajVrUcyeif9TT+8rhFTDBD1xwi
/RPO+4xGQZNCb1I4Q/C45jOJpvpVYGvntykDPaLSb7vRbiWii5SNUAD74LYxFEj6o+aIMKWbZPU5UsuUzTNWgI
8ODs+eCV+BEgxcpXYcxyYb5wdAZ7LrxjllIERkBtDcBxOsuiFnR52uh+lTByGqzfmMnFuoqR2Sm9dBNqi9c684
PMhdCFZA+lSTgKmIf1z5/KJNFF4rYcuTarKcF5y4ZlJFbl0GSM+nIi67mxeHdxqfORM3f2hRxOiMgXFDt66Nda
bQAG3ZuWwkXYBG/OkitlglCvGcoFDrOv3IePzhsxlpTDLVupeCcJbwcZrcfxKh4lP8eCiKQmqUVvamcfLH/xgy
Fju7B8hJS1xRsOpRupANDeD9UcESAQgr8bmNic9VYssu0Yo9igVery85ux2/o3NwBF5NRL4iojbagpDvfOsX0y
RBcYfTDOwIcNsZip0Ge4vuNaYOiKZz+1f/kPLDgrC8OUzQkfguMnvPkWqjqwnnkUmeNoSvxk1/hVSst1p01Hom
GQZDIQpnMV8b3MNJGMHD3AxNzZaKe/OqfAHH5krBJmLucNPeqHOIH7gPKBHEYeGOHnBZbal/3r7rGS8h7+nHBj
kpyfbCcB35ElFLMzJqB0G7mfERevDnGRErgUj6qHlWUAIF44+BJEshR4iC1VhCK+SazG8WKpcTemmxv2AweLj5
```

```
8v6sUE+C6aZ4takhWdjAkB2qBqWjo/0HaO87cz82mRn/8vA4CqpXuLqCgPvRotnb96sNS+IR2z8iz7bZkzjYpn
E5YXP0QsoBsk/J9DIo22YNh9iAQpeieKjsyr9bzZdhDUNFrQgo9JnYhTAqxPK+SG9Yxot6LeMK6ySaGXabeKA3
9SE50E9scgxZZiRmjwkONqaufOnr90EdC/LpMEw5zbDpmUO+MtjGjFk5BTh2WNia0wjyVVUcbsyMXWdQY2u4fC
1hh+g61xTrMA8qN7dWGSZmbr8DIZOGRc02fqAhsc7Qu9mUy67NPsjXwmVRxPKoTFh5aFtDX6V7k/efZxKxC0q1
x/y7/fz4aEniiuy/l1HSoVT+GX2HA4rrkcZ5jk3lOm2iuros49qLd3xx283l4AiaqZYx5DFaw6sfKVjEIj3hmo
pwNnzBcEYM+k6vnc+vNunezsveNxzuQhWgKxF7irapQL9nvzltvk5Cf5uTt0/jamkzCvhM7gxFo0bnozJmgPje
zQpwrT1Up1yr74pW/tAXouNQhjrH+ONVBJ+akaYjeeuL5A7Xc0/M9VuY88bmipASz1DgvaDH6goa2WHX2i6rMA
mZGkDnxjTxv69dEJEd7m7BEZgPWXM6CVyyWbcSiRsQBxINGA37C+ecwfu3ZN0K5sCBgrtx8eumtmk5hV844LpJ
RpDmQmX83zUrmEaSpAPwTZk3DfsZV048Xdht8xWikyLbqCRD/TupfpaCBiVq2EtGc6h9WUMZvAO7z/QNksxTbs
/JUJrV8jlmAsyggUMThP2kzsInRPqg9Fulq+5o628k8hUFR+UVmBZA7zg9WVyTxikmGlBJQlxehVLeT8HjmhxL
7zOrruW8kQXXd9+2FFFa4uo7hUWJR/m+Cgezik1XkStJoxiSQ1ppv4udCba3phFH76XabyF3Yf2CbgkbCAZpZk
SG7knIIxTkNXKv7fQUSScTEzVJ+IhnGXT2jAreX8ma6FpZop2uxM4/KHxO/dU0nWDl283Bsacb6OD5QTX292OO
PXShKa44Wg1ZJYkaQeeR8YcHxTsXQNVAjLMZOaZVzbY93F0Z8f0WUTQPcq5nf5U6V/p0tdFplJ7RMJw8nB/kEl
GBaj7I+cwy+DLoGmZKABUL3RKPDP2/zxIGxtE2fgK2GwUGkaCq9VMXCtWnzEnELmjPEOpEkkC3F1po6HZq65L7
L0sM/CiJ+/16XP9M+Wv8juAjPIkFec0M0xGpwQbFKere3A4kYjDn1lwhszuRZMrjfMeXWTkKTnb/OGe6giRClY
7p3H+V1v04ZtbL0+5m7G7lxUgETXzCXI6rMVMWD5hgPaZhUHlwjrmB3VAIK/utO06l+HK1yMX5I0lMhbrwBumJ
AlkZcqMLdAM/ZYPeq/f5infbcrG8WfxAwqvgZ2ZW6ZRV72kQLCp+AGrywN7egQTpBM1QkOq1Am5D2XiPWcdZBM
26S3becz2gf6nQOisSeHEZZFYvJQqPO27CPO7AKZcE/ir10uzACePxpjiRAXAFKuHnvSf9mayfuKcmRMewR7J5
S7VvSIIlDX+0ZX/9axuBSRCTbT/4XtI8U62NOQFWIP7RloYyyzeNVPkL0dwDvEZ3THTn8QrCcGBHn+MvcpRtEU
xGmN3mUCBG5t6rJImMLG+c9au4jefOrBlXIhopV/mHwVLF1KW0gzJh9n5otvdNKh88kAftfbF93ZO9nrk9NJ4p
URCtUyIMHZsCh4ozcmSKi8EtMyVZPAvMzGh6uEGJ1bdr8fxxesITrTT1mNpD2LvdrrV5GRx9A3dlCIXKiR92Ki
CnDqcobc4IwBUAptrhVeAMNDCikvbu5aDg5goZReENfAjvoMKXk7lGVoSaQezFjIunzUPMPd7uU60mNAUI59Lh
X6WehMBDZS3uztgvb0F1J5JHo7RhvW0tuzgHryfLWBOYbSrI1O6kKgFdmZH2Fw7CzHjkLEzYEqp7dmjViYRpmI
MAy4v3fn7t1VdNdRVylaXXYPHDfVPU/jAKgt/5RKe4poi7UmyOjpGKu460CJarhAQxIiFgJ1CFOygKjFnu1JXq
LoacahUOpkYACZvfe39O1Xp4pZ+f4ySzHkzLpNjfkTa1McW0ulfGx0rvIuqTuDaIpYgMSrQCHCzGRnFboXvGqm
WMhag1Dd3x2KbIoR7b1z7WkiJOOuSwPi4AHC6Caev5y1Io9ddUdal4LppaaEiXlM29LC1ZqyJMOTge+5NaCMNU
VwQdXUIXMbdDnU4IdwEIr+Rq6xMU6Xhh+5MBOvYi4r3n54eKu4vTFsMdcm01CUG4Feg6HqkMeG1atn6xMgf0ur
4yW5tht7+5CNssesvpzULv2VdPrSoWyx9YiSYERPnjYu8wtIIamPK9jlN0EvA2M1IlhZVQVNwFnMLMN18QrO/K
9nULvZMMlAZm2LPJjNzGICGwVm5GT2ms1vGgop5GGe5lsnV7/4bNbOA3Z/PfORLNdCGRpKwmRrJsQ2xYZS/YS1
tYuxBcS9/2Ym60QIVn30GAX1pMNgdlepDWkbbAkaOLz/D6qdRRQ4PfdtdnKNqSQ6S7c075dvoE2YtdZZ2ASy+T
SbSM9TvYBSjsQzngo6td+cS/Pv+PniUVGZEBxCdOls3WRBiCfnXoK/xS82MJCte68aGOd7UvSBv3fVIfQP56m3
5yjj7KArYdJMHeMapgVDG4fAbzRx0BjatzO6x++/Wc5pLj4r8rujbUKeTg7RlCFd89YNzZT2XD/0Un3mBk5jv9
/lm1FaxrHT7PM8AJ4IXy8ORZJjHfUzw6Z0ZXuS+u0X+8r8m4JNdYzSD5P/8zLO5wmIRzPSKBAUSPkJEwGYXZw9
sB/vIUsxxHJMvfBndOBxNPSF3uu5dg6xYzdd8VoLOQmv2HeNkACBXj1HzbBrwKdPeqpmZPM6Jap0xpeHZwPrxQ
fsSAK2bFQ161JEfOBzW1Jq/BdFLa/QxsLxzCE2WurO1fBtilAuV5A0WxDMokVRpJTo4LZ4Z7q76VCi6OfbEe+g
3lrhUvTJ+Crx56qOlTaAQvRQ7dxtV3QQ/h2wGwKtJsTIMk/XoxOG3PWEsNy2YeZ7aufFMA8IV3pE2JiPeez88c
g6kqS9VSSW6nygOPsIYh9pV8vAZlKY0THORwb0FNLl+mrhM03mXjMVAW0OaMUt3Z/qcsFMJ04xNwztnrZTLUv/
wlKqKYT7zMccXRGLEL2dyTygogeXUGstwsM4IJUBmWOiu2i8/LfgKRKOBlv3JoW/QUiMQX+Ci05yevTvclA6mn
PPNvhPH7oC+K95h0LHzmS8YSs56R0bmXlOR2K2IdYMwZwC/0B8tDFf5iVAzbYOtptRRtFJ2/rlMVgsfaDT3C9W
ifMuGVGQ4A6HPqhBB1bo2tKDOXl9s6Yz92fVGFeaN6+7Fqep/CxZY25H5OIl+bGP/w4qaH/tGv6Ip/+rroBpyF
CHL3OZpENzZnbrrWtTppU9hEid6S+T09lhpHX10oNX4ykSE07zRD+4o4DHHaSjlOZVOWPwEBKPcdEb/arOsjRf
e21/9ca0Q7ktCu6gfsjGrgKkhhgyjVARDhffVa8IeQTN7yfcmCBcKH+90ZNx/tfbjC0oUhpbIzTLjvEMpq5FoP
wpEUJ1+uDYqKF/+MP9N0Me8zfGvxXYCdIYX1Cl6Q/N47IeKSQm7VnZLjnS1c6rpQRGm/aiBTClMU0gQOiTZ2RD
FTFQFuVCPSHtTQC293aOvJ23t4/cMp39/VYq1w29tnjIEBogZKris6alEs1FHPMcc9OAzPghrSEmprbbSF8a61
c5e4xmsOiVRuuOw3P3oSEWsVRkL82JD8mjbjaaHOMw47Duvl7PXEsRd2YC6wkPnMLDZXzTAtg/QaAA1blcB2DL
on+7Y0fmlfOYjfBcVJqsRx1xNLfUL5UUTnLsShTU+myxBcVixgQIYZzzcvN0uL+8XMqE0398hAeci84xp051cZ
JvqtFgI4V5x+CM2cPrcP49bu3495e1uxuWYd2atPV0n9NUNV5Lr9ErsX66oteXSF0voccgEjgZQDtnLZ3fyq/E
YtYsg0VfDO3tJSKaR97vwe/f2+bQWidQS1s23h8F+/XopWZHIYGJ/3aXD9Nss8ss1WZRbmFevU6mpuE+hgX34N
0HtF1+Ch690ST8/lKFzueFlvqU7jilk0cMTi+eRCAoE2GB0s4jMvjOhe+Taa6No0dVLWsdthZcKojoPMuNvrRO
9kcadBHRPFZagP9nBy6tmBdRosP39lN8/wB4irFNv8SIegObqGhPCuLNVf6Pyrschlqrg0PYpi6kyXf9gsTpfp
jpI+LBgV3YXlOC5QpAVXWEMhSIrtLbW5igo3VqvTxnvuMltwkoTwn/7K020JMgNmZe8x/VXnRZF83tzm5K5mX+
```

```
mfBFlxTuMCKI/cj+Vr8ZlfZY5xXuvM5Nv+X/uBzY4VR1D6AspVo+5mU5Hp/gNtloh+Mr6UmuFzeVn7095cfXZG
FnDSn4ahN2dxUA7pD5AX7M7YLMMRoMACsNQP5GSGl3gd3bJyYw3inm2IbU6y0WRKbgDtnloCQ/vqLgwxGLKL4S
zSKiFDDoJiemYup9r6LOJF15fP2QGw2DrPSqo/M52s2Ri0VFwIKk10G7hEwHdHgpWbo3njka97o8TzxC5DGwcm
ECRx+s3DDFGbsn+PlVqUesfDl8t5dcsxRWc6Y+4m88wZ8Rx4oN9eeSVgHgzk46e89VVFIPFusTkrE+vV43UNgt
It/mQ9+6apnb4+Ea0fnnxAuTgncTVI/uF3fZq3wha9mR66mKLlxTxuy2NdqFN60/5HllWwSkD5+xF/PtHhgAxt
iWr0zw15hzcV6m8MSQmIBVyU9f7EAzGgOsNATe0UNz5U2n1yaRumORoKg1O0J25KpH3V6brkg5+85DReNMtMb8
b0TsNg9YuQrEo2EilkArugSV6w0AaEVd9BAuGtTMjOyTdGZTVq1V+Bx8DCuYTt4K6vX84NkDOC+/rWXYHFWrPO
IBYecLgsPHbKg8ijs48ycHoXeVV/T907nYJXxfHU/km7Sg+t95JUPkF5kzGA+DrJhoQbuQ8sTakS0jZd+s3P/0
wvG9K8Joa2sO1z4Qj3mrVfIpUOm58fZ+jyeFmcvkG/+zYwaJOrzxGoljxbPl0GE5oaXLdBpeyvo1dk9U4IcPQ9
Wp6XsLDQOEstcZEaotFc8STcgLWOnnOUmsvV5aekd1uYVs7XPm1cqg0V/uONKlQsmwzKvHjj5ZQ5FptRUHoWRn
2TgyI6UhJNCuDyUDZALFcy1QbYyzK+kyeGwjOwsfC+cnPS64Ml4TWHIVEx8b44s2LXJK/KhRxEH+0Db/tHE91T
m2sXMpv11gQh9juSUKLqWDHcjdKQ/Mi7GFJatrCrw19k7QtUbv77FtAtI6ozBm38sEhDeLVz6pIbp9OqG1+GAH
YQeW2zRQFYhWGlZLCdgSCP4wbfZ7gdUUiEm7DzppuleM0M1qZABQ6De4yvx5Gab5fHCEvVmnxuaW1HqJDPDvyA9
yqoq2UFbvE29P0FlguXAR2lJJFd4n2mM7k0X6QYTWZtaeV/mMo4DcdRpjFNXyIXHL0ECUUeY36pI+i9nl1VMYg
U2BjX5hnqr+K9V0UrOrpmB9KPm6kA31cC2ze236nAS9I4TuOrakI3CEae7iTeVX/Us37lax0poMsvLYCu5QmHN
cNe4CfvAC4mYy8IT/nNl09ByN9aNRwFi6jya3iyaa+saGgDXcFILTR2O+Nfss7q9/44inmp5xvEgfO4DeN/J7z
Hr06jkBpz38WHWRkYzvqSTNOB2P3VkDaFacjw78jP5fQ6MLYCvK4KesgKqI8R2nEntGX1mrxUv+yHfdyaAtAaL
1+GZGh3msp5Et2lMLcPmGVmTXQ4kxRcQOChOXhsC8KUqsus4PBSC4ijWiIiNiVLubd65MDZvhn92pGSzecojhd
cwrvZYCahwWzIacHbvOfyEBNS58qSvQHFQ/UAPXJiybqmIlQO+0i1N4cfTMbljWrkxqfNqcDeQQ0MAraB+O7Qm
fh+Lc8amcMWlwh+U6BpAFhfxdpzuoAXYJIAFaz6lUpP5AeYFgjQmjZpfjN+39O1qqb/dLaUePYraxf0Bk+6BqT
ULwBK3rkT5QCQA0o39rTTAC7tvBqTc6Hm/EmYI+Dz1ne8vJUc4N4SmtaAAJd7oThclJPK840e7chiQ2UUj0ykL
FYQ+eqkFTRpCGZSWOmc6CersZhuGYRMlaTzbl4oTSuVQQJjqOm6ecto8FOTIhA89p5HIb4S7Eh6T3y/5Clvmh1
h4Xq1OkU2DbPtHpXkGbb7MkAWBxhymdSaOj3Lg4A6gdy9kSZ9I7r/0bUpQ/ET/VpSdopvUBE+RIyzk/M9A3YuE
a07vyjw9Shs20g56QnMWvsTEEIP7grDFry81EsNVMe50wTusIk28jr9cBKmHHUHRPClaBv50KgMITdgmL8MFNE
oxtXnAkAV15khGtNhrjG1rgIJdY91mODAP51kSXacue0qzGTJDYSpYkxTHReFjUYcXThy0izJO/XTTLRodV9+4
U6/MmdWGtlBUCsY9OgtYIT0oxLKhrJCbWkzfoIigDSOZKQjV3pYo9OzsLEL8vTwHmclDzqjn+zeF+TSYIAduY0
oH8fj4ByJcdwkDjSb2LKqkJn0VnRj8Ry+OjsTUyUAOSPWDZahDQcHSiHUYdpBsLry0rYyzGU6y15moQOsrPHcW
ZK6LfdJ45gvAtqvwnd33yKrWnH54xZfVin9+lDC3K7UJh1xtbl4u85bRHZBAtImH6cVRmsj6N0px3FzJNz5ZQi
bOa9GztW0jf5XboD6u5nBY9rwANSzUeUYALBsj9Zo4IHBid81nTmo/ZLz9SoX5j2zTpLfkGMXanGwMrE1lVxSa
++nOV/k0rBmJWEGAsPSLeSxrz2jBERYiocwqFGi3kUsFeBFQEWzYlHTAv4kSBXih+m3oKCliw8sof+LngKvFpj
uGUvRPUqEpAS+Zm0RcWCd+0xFMraIc1GXXm2YwWQDimW9tRt1WNx6JDSoRHljYclV3HINVB1qjthRfoD5l6k+s
4XZ9bt+CeR9E7HU9fmVTlQttPSLsfqUcLqLCMfi4YqIv2W2TZPHhgLjwnS07nm7VnFhg8ulxmpAKsI0kGOXn+R
38pphjsinASsH06Uf0Ow4lDj2HqQPP/+PoF4y5AjWBob4psqFjaWqK73ZszhwsOt4KywcqB5HbmP6nqBZFN5+Z
+ft8MPM3gg9lorUGMFLSIufrLb9tKzLzQhrgJnuzzXsBpnUeBJTyTiNXEzHadjjX68EkMkEgAfVxqLKks+aKhI
pFp9yEmw9DdMZVQnWAf13+R/YxFpEW0bjdeQIHc5ci2Ci5lcFoORhVULJG6xc8zsSXPPab+W6U4fhtkrIjt1em
MyQAuOrivW31Qo0RAzMfrsmqlTxqdZAJX/ttfv0l6Wp71DdJrNEXZdV4NPQJkmKwzhNpteuu3XUUQhWAmDaJ9d
t1q5fxAVHcDKKtMpNc7w7jATbwXoglQ4dPuCy41I2MLpsA3DmUIHcVt9aIixGiWO25w0CDFShKwf0i/YoWpa+P
/fu6CH1kNggRMqSsfPGwDbiI4jIEBIHmWj1oU5XM+5AIbwmtudNi4vKc/2GZ5Ep0X1EmAJGfggpQk5VbyYoP4V
aMZRq9KnhWVjq+YMzGgvd7Kf5PDv9/Zjo08mCRjaMuh730TcX9AaEesTYYqvTWkdPThxhDdSFVniKKGQgYohwv
/A3ni5SAZm+pXGcLXbljtPjG4v9C23UoDlY3CM4K/1oop6WT8XxJcNWhsEvSm9d5/P0Wn1hOvZhIyYLtgvf1aQ
efWZ1HvBsfv3E1h7VJmUlzYHXo/Fb4O4ktRgTtz1RlCAXLuD7M0/sgb3s09kFk8CYVZv5x7cw8GjavnqNlfJ2i
fyk4I0aFmRCPerdSgsip4IdmjZSQYcZB8d8C6g1c76qyUm5UQjflNcZhmzlLujQ0qPTuJBDpAfPyzscEjYYsHk
VuMlv6lsBMJRWZIOfGaaqS+ASfsSRC7gRejICvMHt6yyBcMkI2R321yGX1I6QztX5ELvy/cDjOPxqAugodocLN
yod9mSW3rpun0dp01+Q2mePFMiU/pCcHZwXJHJugvXGo8dqkWR93DU3YQm+3YGy4S6tZX+9CUMRkQh+THiA/5C
33E5xVE4IcqHIbadALYgeZz4lx+kNq578DcI6EG22YG2XdKWlsVXGn2znEQHSZMXQ1BkTUxxGCbWIFRjUs27EM
9JcjdSBzIvulZ0UI9p9ycEipDDNqPPd2a6OQMdvPcqUvTKAeNxVD72Y+qUpqTlE4Qxq0wx/JAIik0L5iZdEvcQ
AAFb8/LvnD0ZI3fuI+gix32m8TzRejUMFV/zMTu0y+qzsnSxzSkubpSHpqrQZHHNN7x7Vh8QZmshqU+rK9tHzx
+Swyc6PFs7qbzooZgbHDOoEbIu1plWCJIKWsyeO76E1Ud+rsZ7qaSsONIhFw+3LoIcX+StilvtccUFynKVNHME
hGweRQZ3dRjFHFBNzjmK8M/KCw1ecQwdUvapHX0U5QlJRlOpZOKIttDQy9lUorpalBIs8auY26Xx+GsIbcyhzB
6ZFcMzrdllgUurGkn3ebFDHdW/wl+qdx6a6YBgclomsBZS2s2DiKOX0bp6nyrV6XhOpQore+gteSxbZGbFiWbS
nl8OIt8hpg4dR7jZfZelwKhs33/4TzRINv305wIFKMMcG8fG1Z8zU7y0KHed4Czpsg0A9HoWzJHOTJ0p7BQnBm
```

```
S/dMV87eDMK+viPYUJJPe4p/LMug0OBEmxax7jdro8fIDA7YWQsyRLcDflPwMzVbXAX5mPwN2Zf3fzqNrs1arL
jkEaOynyrHomQ50lA7U/2pFYna1DcEAGKmUFYfrjTUkJ80GNcmxbVVAojI7cMPWWK+hkLkguc0omXQECqeRwdz
KxNpuwKPF0kvIxWoJ1flbk/oFeU+mmrnSUxyZ2dsWAOq8FtYSJIiDjdqnh/1yQS787fvfNETU4tQyME5ihYq+e
VWzfeNAL1UlzHi2JSlg2UbTvVddjVREn8wRSkRvNRW2Jjf44Ke61UzAecwwbXhI43dKcM31TporCeky/l+XTBh
b88gVaB/NVPM0ysEdh8rNbgXcXxV1q8NWUyOa5oEYbOwROfYGsBLFlTwdn8rc0qirkvm8TjqhvYYlPOmnyvGLP
upMCwuoUxYGV3udB89126kgkDcC/PG3dWwRZJlr4E87SGBX0BkuPrer7XLmLX4WldQ+gqEVPvq3/EKoHRVrhlO
JNvRU6TJtKzQc7+3svDu05WRXhCtyaaOn68b5T3ek5t5AAUrueo5jqCxqbKMaS7tBCl2QBDhB8WjtqoB6pwH5M
M/AbOLOuD88W/W1Q6ejCLV9NTqLEogMnFQcet4G4nl4aso4hK5Mu/coqubV8brlMCzFjCFRBfpckRpCcKAruiE
FZcwyuAqLhTqyX/+08gDoSiWXtY97lECO5Ae8YgV7YVRmVT2CU5fbbvHCRWAjOMHynmNPNvjXIjkw5+/j6X6lL
Kb8mSITO1cCSXycOc3vPvPOTnaHRUsjj/VbNUyCyxpnBsBUJZ9EaAfSfXQq0Z5f4Na/r8NMUBpZfUtY6u+6ZmE
d7/Qe2E8f1ZJxL2BPw4e34Nfia+3UIQVyALKfwBIa+G/25A9EtTQLYxyWXy2DNve5kY58Z4RQvDu/t5PPFDd6B
Itk2qHYakvoQJhM2JsoO4JHv+4zdMHz7RhTDUr/EYw39kSIdPKSWkFACjkNDatTUo1EYoZpTSaCMczKuhnOcDa
Ylom0UN92Aq94Ow5oSNyqDuVxran3uSVoYFeiDqDxam9jsF+OhhvLE+BmZBePXyWM3bY8WzfGXtGtZlT1J97L0
3qOJM1zg9WGbvDCeMgLWZOEJcaNMN4rP0TQ7cJXon9+TBJNzWi3p9kjXzcuCaWJN2J7HTLOD6G7+HFjELI7m+l
X1cCahFB+lWB2v7aE3wFc24VXv1527RX3CIk2zA6XrRVSmWzh9qkkZUqukxUV5V7Ce898CKmTE49exTGa5/Uw8
G8fojv0t2dZinLYuLv0vieu2x1cCi93FKQzik149O4MXmoT9Sb53uC1Aja2CsXJVc36rQ9ymIPy35NQP8phgSr
rs8gvcK9WMTut0/VEkjicgK6VWTUgrrZ/It8AZ/PEyB/wlVQt2gHA3yU17M7VIZ0cWCib2JMnJwrAljAAlOxGa
ecYAJwAb9GYab+OYbknLc3E00rDqieU+ZZL1fgCithioKy+G9XJoETuCA6/Tt6BLPZmvYBI0rJsT0E41Xe11g7
TE56Xz/Bt2pSTFD0dvK6pT5x+HBAP1GSArpwQEwBjMYSaZwJspJpe9lNDjpzJj078w7sUqPjwTum6eFQayJNVM
IiPpCvoD6o2p/gP41KB2saXy1ki1fSKWgL3eezYhju80erX8SGqR2ycEf520x75JsW9qzxpJoIDeCgODdX/TzU
U5MMngtTcMRUgFVpydwBsXRxICIFawxaI0cMxRm7WFoXqg+oR1/UWQg5nElhrhHbTQf6rT+gZlKEtCZmWmSGsx
8UsF2pOiF1Fzgrrwlbp/skjmByy63TqCraeHkHqInXmJUs4m7wJFJvZPfZhxBwzGUkM2FVCx2vVwNrvzIT4nQP
kYUCn/iXQ8N0dP6QGCZVmCVb4ay7xKoTnkY8rc+KSRZgE8kbyKRk+DTQAgyeeW83Yd8ijJDYE2HEUAWZkgqA1z
2RnGZSlJmWBCXaLh9yu9moh/CjiWwAfnYQbnQiLK7UFu1q7voiPnuS2hMTMBLs4QjvZYeS/6ktD9yOC+NzTRVE
/ZUPHD10fCJS2jnjJ7gHkk7BnhA0KrrB3LJBpZJOGJVgdNQkLLaZP5fR87a0yX63vTjX7mCK5nkSJ1n0yMVK4W
pTKwv6DnXD43fqMkWnkoK5mOxy54maTRP1NHJdhdnwMlm54uD/xcAHoDhf8hGotnd8h6QgmB6fdr5yVEyv6vBR
DsCtH9B/xK/FhvR/dTWiTlzbU4DEN91i8eW2xVcCWs4W+dcpOoWPF5bEOp3c3fPzDCAfJoJ7yzr9dEWCpnwmcu
/Ssf+iY0s9Xba9XisBJLI4TImzS9VPMMgV8V6TgoabRyKJ+SlEgNbHmvJa1fEq0VVbN1pK8edE8kdd5vTBsftG
WURDIEyWelrYtJVRm2bAUlvaikwHTU5Yv0DVIEdGdb8errwElJpnJgNKURUMyla2GRR947nQMj13qH5bbN/48F
t/CSRlViFu1wkATmLBlYJCpoZTYID0MeaDq1iJl36L/dBck7y5n86J8guXpBkocm7Z8lhSkHuLQsSfSsdu5Erx
BWWyt6PDLVM4LPdllxx7DO05UTqXGCWNHwxNkdZY5JyPhNc8RFBYNofHg1PS/MrOEvvlzpqa9h/pKwf/2LFuy/
rlhF9iJRKnYL2cXxxKZO7UDPvbrbOu2tym8Rf/Y3scC8xkXpKxMUeaanXjF7hF6habzitk8SyAccEXa4QRkI+S
h5NVxh9YobYjLn7joyTaiJ1DfkFqodiE1rwIFylTLw3s1MOLf5keTUx9jDvdh0L5NPN6YeXkCDUibWy3ZRIvmb
px1mtfJZeax8RWvs1wpDwbfJ/wG+g0eOhMVGF19aBvDMfZOrijdyOe8taPszig7kmvsuvlXjMpra4N94jXmPkq
1dFIOJwpF2VA4XhP0xw2SRytWH0vuXhSdxuhmtL5TnLAQDWv/VpFTqj2BwAU83ZIq6OWvt66dOdfeG3K6NWk7a
Rkk5aLiGe62fan4KLiy0BBigyErUXGpO/1nrX5k2JP+NV04XxRmeNxcBph9VmVNS87wBRTsAij+etKLRs7+sj7
/6w3+2j9aWFDYN444cBERroUBbR0tE6HoHMaS0bajKbwmObuebuCmppv+8EhS6uCr0BrLGqEM5WBzRlGezkn7A
BqT908poGvme9mOB7VV5n7mEgVigjrg0fngMEpc5EP7Lw51XPyzdASY4Ogt5jX3MIJiyosECgHncUVgmnS6hcD
tLQMzEIq0+i/thh/Ye+xfCMhmYOe2k4xZElOT5rBLn56oQWCqQFtbRJkZ7bPC6kNtaGlYprFFMNTVENx+maxpk
lvFKPND6Y8/7KoIkvck6VOecVcDYPHwuiQ9HMfaRZx9pBoPfQJGB41pVvARvW3ddGFvK9uhLPvcdZ/Iq8WNVBY
1meepnWglW1HPBs5d8zg/Fbli+1kTEZR0kP3JY6GFjMKkbW71CWAXJVRZlw3XwvCM+DNCWYMeUV5sEsKDYy+c/
x82h2PlJkzQXX8xKQVk5aifO+sc+fUEz7MwCZzzs0DWc29yQBKCxQKDzd0cZO3Qg/2gqzeGAI4Psin+hMkRPim
KM6TZxWNClQU9bvHBcT8OrnwYhpHX65Q3cFQPR5JgS8aQyvZzzjY2XuPxBkGHBaEgvJDH9n1A6KlRzkzkqijBj
Qhw8i+/XGRKL/STfsOqAr9SRW8lctnlKRYirLbk+8cYkoSxjLh00Ik4MgvJKiLDEuEuArFFLUDnFqjwA3lwujQ
Www7uSmkPCiKz27wi1H0614XdsW1SGGaFfMm7bj1zypjwKrCNDgNPaNwomTWewkYHy6aboNq5z4lsijgtoAxKC
sAqfONi91xc3Bh5l4dslqh3ICaBMyZMIB19tm6Xg84pNw/ogN7oYLD9A637lLqGSKzrZk3m3pTnu2AvzvZdYJF
EPRIEZOGXT8xPXLLibd8G5ctnDtYEVAHJ3ldiX9wZ7H3OcWczchN9jODAXtj/m4hOAwjBDjb/BWe2eqN7tc6nN
r7+OR1ooRB4DrdiQ4mwy3DAH4h6FbgeI18eNNxjOelrzjs15eJ5E8O8Y5IiKjBhs47+XcnIjvrjPwfu12i7k7b
PE/NllI4EAG6GbO+xB8UmK03Xohk8RjviZK9EHtjMaZX9dvh3/aUEWTsjK7LZJZ29V61W+206Z7MioCRsPqr0s
GHltzXG5Ih/n3oj7hkOoyMTuSsZ9O9/ywWSLFG5HNFPW9BWQB3LTtDV9eJttg2u/3bk8lxPB8wpr+t/vKBumA9
9ZDBr+PZ5CI8UwczK/Rzf5IdFmpKHrJfKazpplX/rjcBimsHievlN2eSzqeL7gaXR31DVR/DgAXcrVy03n0coZ
```

```
qEn+TrMMKcuyCq/hY7SsWorC9ExMzn6aBw3oSUbHYb8cK+m7OlzPnntBtIi9KSD1EaLeEcA5v9amzflSmz22FN
XzMKTFxdatxynHiGXhWcvXNmSprO94iPsbIviGTswUAJUPWaA32ABo9DtahWWZ+8LerNIAdWN2ppE49BwTcj1y
F9FQXysSN5d6B1q3nGedHR4KDRSrg6psghh5M7hIlJF3z4AHQJA0jlKeyne4FJ+HxOWmkM6pbelrqv0n1oocEy
xCnh6XcLy5+OweVplrT9osaI3palUCHdo2nR6dVc+Yws3oKdw5w2oUsJCVQxS+ZkpNIhgcVot1IV4EbPacegyF
38iSrvyxLHlcr0ieIHaYre7EmieaKsHJoaAkqCsvdDS/gABapI9TXym7vNR11YEBl4QKq9ACupO3mRO18Gp5of
r3TuhAr5ev1kNOaxdCrDYiT8jF8vV51EPq8cvb3xCpmVHeRDwcqsNPUUqSJSdWfChpxlJar6jApz5rnlvrPrUP
y9GPNRSUeAr2U59Hhr94CMFq+wQp7Fi1HuPg4ZQldcVXUrUMG0WBDlCbKsCL/oaWeEIm+/DAefepMk+Osy04vg
C81IKY+LR96uCOHrSSvpwdBPcx1RV82Tqoaqk7DVyqJxZlvWBmSa4mYRlhrkW3PhzT4jUOgNBe+lcnZK3MjkeU
WfD3spCdYDW2wdSBr7IZswn9cvhqapzJGQwuUo3nSRteVKEXYyo5+NT77Id6tKpFkOKjK/ETwzvFI0jxGgvG04
2VcM7i/XBycnZkWBx4Ewfbxj9mMIAofG4h/kgn21AfEsmiO2NvdMnd4n9phLIQi3lEix2ML+hfYqZ+TBK6GX9y
Q9TUEB2uOTsJsE//mczfGVLIOxdXosv++LjiyTSnMstKs6g4Sj2VNPBZUmBw7vIPZsPaArTUPw8TQuYFqXF2XB
eVhptFf+E8D04SWrlTL+cabmuHFuPwkWK/hLRBdzsfFOBd0a2ARK/DE6vf9jIqxB3qsyanVQ9zIcqMt0wlqtIO
N/mZXcgpm4OPtCp9q01SRo/ZAa2KYeZdl45bvDXojd4zshVWGd6d8AI80rvhmHuGzn7T0xSxkHIm1Z7T9+iCqX
HJ1zow4G7b1AdwTkpLxpTgHgn15sfK0+NPXWG1Lia229J9kEtk9cEVV2ysiCqFnDHyFT9i0Gq78pDxFSxJ9dNo
vUOm1+7AYaTx5fSSDw40JIQta1/sg3YEF28W38jAfeRgiwwURd6/SjW1HZhalufJVOc6B7wK1smNOzbRbcYLxF
Is3yN78r2kwxMRdLSfdsFg2VPiU7LbU7opbjLaJoSUx3iet3OrOz7abh/T2sE0p9hvYC1vKrThT1zsd34g1F31
RqDN1oZiiJsLWHRHp9Ua2jbInupE9Nu5wqSudOEbmr1ISw1ZROUGuelPozW8+YPLLQ+PjxuUaMOzldkELNOcUC
u0UX+MmRWhvqE/UPlwM9O68soqWpgmjxUdFTbzupGmz+ApIsYW+2A2DI2zsMeGkrb+q2zDu63O5zpIEvjp85ca
hmLQZWGYdjILbZNp/CwDNW9djmMs0a+25PPhozEkUblK9/b37TinlTLB2EZUwBmOmR6l/gIVYQSq4WH439ecPG
Y3UKaQozIacmq04MKdVAfvGzNtu+2r294/50AnFts1mV8KyOeiVugLWfi25s9B8m5AbsZyeKG52C7OHFuLhh6R
rmVIOXVyR7cvQnm9pH8hzlPUSU4dFnH+qc8nRbx0hpalFWnwnbVbJ9Mn1nGOuoNYIttK4hMO4l3SAG4wyL3nF6
ViIhg2XE44Wv62UEUoUOLsBULZKec0uBZqTc3ZB25pDcJI6Ops537adg8N8FzQ3y80pAs3CvLO6FoVYe2hKQrr
jlUdFXjc/HOvJEk6KjJlS7PAa9B/MdaQORVZKOtylZedTaJpgBQk+mt9AzMuM5nJ8xExTEW06gaLtuv5SZe/3v
Vp+l/u+aJIUtSyVzXPNos5QnQMosanlfMcLvIErfAUMkvba5y9gh/H3BhYty3e4NMAjjHFg1tSkt4BBHXlI/Bm
/Cya12DjPA4R2o0lxGHIMEHw7hEIGM/eVzpqD+gbZC2VdgZiGj84LtR8EX3gcLuJ/oKmqmlhg+G5WxbzrzuuuK
DlfUeVWVJUEsv6adwLRbn3cfocsi49Wu1jqsXbpvfmOg8VCNJhHp/AgliKI1AUkyLjWsL/9aEUWtGmPkzUx3Wk
afcigiJOzzxUOC/uX1BEbG8mjuem6r2vZ3KY7Bco+iwhYzEXZ1r1TI52XbhAcq7Kv+W/whAXm0qwTU1ss040p3
c7FEjdhmP+njCSjaykegWMHq5RjjTt0TVNt6quJtwRqrCWdmxgHhM45oUaXr3f8D10iYalPAZlnL9UDSH3XGq6
9OItSTYtiuwbtmWh607r8hvz6WSGZfv4HUHD9TZ0qvvw5TW0Ev8P7ME2yuxtjtpC2P18RrPG98ZY0RGK/15gyl
0pTa8eanSWEenj6A7NpfATZcEJKYUpqppN9oQWalrSbOmeosHnGrWK8s82g2ZI6wk9TLcGjYijQo1zfMNkxi3l
stRASTgjTMfzDPWdIh+Mq6hP9b/HiK/7dUcDZYGXFxNs8aRqa8OvH9ZP0mYCaykEol3RSSipFMH7HOn8AnmLt8
n7eb2MYzp4e9nZLB9wcbJ3Haf7pQzTXgaXShQ3hm+PuJpA0Se3pvBTyI8fMzzxzQuApyiNaBRX1eExnq3ziNjb
LyLjdvEPbdpzy92vjoWCauD1ITHHo4YZ6u8CAl3t8w/gP08AexNQ9ePkTcewRCJ1cqS9OMA59XayiJffgUOf1Z
sLwdrYytLQWXpPfKB7rFdxnZZfJv7Scc4VAlEVyKy0/EjD66NEoOPWXDTzU5Gq8Qn68ydtu125nalDeC+oQEMC
+uOzVgYwiZquoq0tL0vFYva5YoDmbjmip7jlAmetnV26KdYtuuYWXn6Rw++vLQJpL7inzD7PlEHijVYg+WWXbC
KXwDBAdPLT5jc3OVRZUnS/Phka6839bn3A4ckdWsHnURI8vF1FIbw32AyNRYcCwtwoRFcUXHbo6vyVIXnJNEnW
LXab9/jBmP/+BuBYVZHbpZk1cNRSFiyKSTEaeqo+kxGQriaCzl3Mbx8Yoiuc4dkeUN20dVjiPWCc/c1KxGUeMS
lKTelpIrdcKy61r2pvQcUXpMc2Bwzp3vYO2tDlI9WiZGF1D53736NNTQPHO/d9Qe6rwnkU/H6fTgAVDlUkvNJD
KUf7w7mmdj2qu2y13qbmwBp6E7n4Tzi0TlOchD0K4630+AyWhkGND2U0ZieTbPYHdwJBWi4/z/HSyYmkk2E10g
0EG7RIdbSNxHohXAozp07FGR/aKxqh6H8PeVz+ueTcdVVzc3XBecKPpX+k1ehMtga7+S6wBZ2H0W92p7eNisBt
MCrrvqClgypkcr5Nb66fmL7hJskPe/9iSe3cGXhFXHwDjClRiIVA+1XiVGpfoFC4tvv4WnwbFPQlvDF1zcULjK
/Z5i3VGI//SMqVCfjMGG/qtzlpMhDE0XiMF33h0Ygf1ZtMYHvtHQUEGashAIcKCFbLs2JOG9K9EudNj4ISYuVP
oTY8MGh+mZSfkVcVeh586bHih970fvtmGMVq35v5Rhw9sknRgo9alOOa3f1I60XSVwKy+DSirz1v1yCor8fuSQ
t8AkPExmAJGJpPDkAJEt4/DsiEZ1OxTs+bTJCrx7I0ry7n14NgzDi7wAhW5dDF9T7FZlSRZj3jUIrg/SofGqCC
/sWhxv/+uO05zhGxmEokS6T3E71gVXodAkJWcV7y5+nOg+mquG8UHeWO7fndVWGoSdqwdgbzBEeJ40kcFT16XF
kcVyqBh/lkUqtvR4wttxPYHHrJJB4BWdfqbTKCF8/Rx/K/hY+lkw0ucbk4r0d0brPAcESNMcQHyLsoHwfNSS7q
1yZffknujolAxkwlLX2DDUBxEYPlZ0VAisRrz9iQiJFdIBYZAMOPBCLfo+HuMg70C4mX3haXrO74bEb7Ecn4VR
qPQrE0g5BbgfkKlMlmrfyYC8ULRT+9IYEXOKescrMEMmBr48iYTdEAOWWEIVgiOiMW45krWiUY+9ljS8CskVaL
Aua+KkByIBjvIz0ieTkqMIkgTYBdo7Efx5KIxTWVTMzK5Y4OQZ5nBubTAIabR2c3TV878tJPmkDba3SpTdphdI
yNUEDXUoxVZfyIynr98IXqgD7UFiNagR4KHBPUubgdci8NBoxlmQnkdnyNtaxIyZASrJedwptpUGC7ALb/a1xt
xLBOS6b2eCwIqNPhwfxP6VIy9EUuVErYLjisb9mpyTX9mQp64bMNkANKiZXlgHm7VtuOuQlql5CFHRaCCJYT4x
```

```
kYZzx+xAz5rFCrE4hklbwFdSWpXcqsUQK8cankHw0umNeccOGo0wlnncB+CfTmm6Lp0tMOq9GwlYUal96EK2nF
Zw2B6CiwHqRFjksydUJ5Dz4dnOcJPunFE5b1EEpplwG+E7J0rc8BJfhV75BsMMdZchez+gvrQ44OP08hSsXT/k
SiX6XRfZHIGLRelfrlBFJn5lWMkIgLYueoDPTSCQeJVfd8xam9E8K80bPI6yzSrnf41zrjU9Ul7a7Gn+04F+Yn
iHyZMlgklHBSzzToNhCLp+4N2R9MAUX5dSSmSWYDjIgM8ckGLgPdNl4V1Db/lnfISPGEJU96WsjUVgoAP/DW6+
8zKW/PenQmsQbTfckHWthP6WgkxiLuM5K9kmd24cLha3dgnmUTWA7+bj6Hy4Qo+ki+L3isTDBxEM2sDIHw1G5f
9P+ixCNvuVBCh+7YAi01DUJDS1Y8ts0yA/uHmqrjwQRG7T47oBSwWnH94E6GNsGM6gS4MmFaOBD7RguE7duUj2
XnbB4i7pJlwdMcZ1DSox21822rF5zr5cH6mRzaxB/Pi8jhaPsLTlYx5SKKTxB3UPK04LorJcH2PPnqkV1F7ff9
oQ0z9IvgVNF/D14jHKnGviiKHhFBuhI83cf11VOZ5VDn2HAbNHQCjv926+WvLO56M4opOpJoa1WfU5IR9d6DcF
oMt0oC1v+VfALZ1d6GxQzEVT3LcBuWCntGvMOG0Ak9kes6WuqqrQ/aKMgv2zLBIinG4Iy+Kfw/6APiMzUQrooI
616dH94pICGoNQ5JGbtmj26TDWJ3agwVmUqpSTtXG8omoDEMqfayzlVHr8e1gEWd9umn8oXJzcymmLqLLR6zaY
CtLWVEBz/+CK7tLxf+Jw9ZmZaNSOtTWhg+mE1mvmPemBcaDjSvczu+FPNlHQlTHpA6RYlhpuK7M/DJBKhl8T0s
T3Kc9Sckl5CZJ6pR+4o9AZDUjuAXDkOZ/mpw7R8C+z5pnp7qfVgN9PiPDChvwGuxrgueUUAQbY/hiEscWkE3ug
IHZ8gCJeD3W0H3RPbAhyUAthHD0SbhgZ5jY+vejUgu0JCz4gZ/2DAAA9NZaOKVq56fzNEl1wpC8Sg456+2KTgC
wrx0s3vkRya/iz7LCt2YIKi66Z+L0ER2xqmFRiFFKevGE4uEMLSuNOifJSugGExHV2zKFuCnEoJ1wJXpVoXf5B
KbBwMmpt7q6up2uhSllylsMSkjk6v4V6VgR5G/O1++kNDwy4t6bTeNk96euWf8A5RQtO9t3llLwLwmFCcqfw3kJ
f5xBVUXaby75DPqM9ZQJj/7J+/1ifU0VS4j31ZOG1Qmv+44hnV1ALOJJq/EocnHcWk7s/dp6HDp62KWH2XnZM1
s56vzFEYVqb2U5p7tsyAzGPIGZfnUlEBpeTyMSyMc+sdnTcJPDA3ieHBqfUFeEaeqrrsPxv++sA3xxcd1CbMKR
g4EhSDZFu9KBZYnek9BFiPjobkEaqR/3WfEJkv/8D9AxCM6TRshrjN0tFFo4uYm/QGyL8O88GPxjje7pRVYIgB
mxvzAXcNRjZPOQcZaBBlRoz8Y7uFyuAiTk4Z7qLJpKtDD7OBQ9afyHDnUG21QNAWrAy6zZvDl+Rv6nkyZKzAIb
kTcjP/+YJPh10Z0SPhxSD26+Cxstj7F/UoQ8+mvnVWFGOSp9F22GsC1RfGTMTwZCFC2b1eQKJtVzln+5xTjJ16
V+DydMRaQ/q3tVW3jSNnZsFapqzNFJ6YshPjzlfPZRfRTuk8eY3XOAsqqeTlA+AR0q6KlPKdv9RdnS84PzSaTm
dMVcAZplqlb2mk7A6+ZnpHmuCuWitILz6KW0sHF5J5WJT6VP1B6Gk6EfOxQGmG/ZkLMZLg5XKhx2S9bTo7ZRe7
JZJCduHsfjVQflMvxOzbiJ87BwHSsqB7qE/x2nklRUzyTshi3vPa92lSEvFXnZ3E5Ib4tTDdW8reYuqAZvCgfW
2k51QmBHGGqbY5lx/FV30L86MPssiJmIJ68fJURvNSz18nMYU73i4mF5GCUc32SzhjFv98bTbwvWkRgggFMgJe
vdZDjsa0CxakEEb7AjJt4eIX2c6hKmetlyO/l1sgkWxtq/JmSnjoK22Qm9jODpzfz/NwmYpA89a4msfIfM3qDB
fsib4A8cRGHWSRDr3lnsidXcAnRJYwM9d6YJD+QX1dH3SDlIY3TMTWscDcedYR1OIcnrLlrk+XQXe6LWGHW+q1
xqfWwuQGI+ZSoRiZhpoUSiRbrcRFkN6XgNnjQXbYKZ2JUbpiyZ71rDdAZ96ZBXM7fSnT1cEAmA6+BDDVkFJyh9
OmCT/ddc5SHfWzbUmXjXt59X5x7XUgnqGcrfrGJntcrgyyLptK26GYPlFep/zKxuG0D6T2zJZpemA6Sry/i1rj
rxq59aTbUlL0WvH3s3d+GX5cPVdnRK/r6BR9WZV3EbjzzndgprqRnB8qB23cGNyxHk3W7p8IFpP6TdtbtHxsuy
ptHOg6UTVa0bqZ5ZkRAEpjYTy7q2cs1gvgKsuFmXtyrQZUlei7HG+4XmywrgV8eE+q+U3CqZa51GK8LtL2lczL
kmaZSeJl9KKielzYfsApgg13TCcFqKUITHg9196x/pMFnMP6qsE3FbTGSmlGtLpGBlq6U6kuhHrOl9VmWD25fG
BoDpL/rUxjX+hjTfv6eymo2NytrFV5+eo2J7llLfN3TsWDnh5nHT6h/uDKkZZ2ryF4r7xOByx8VmQXt+v9ZY5+
BMTNP3CDGEoyoK7sgoIHu9CkXeuRAQ035ubIF8TMQ/D8B1pW/6zNGQXnmbfaMPeWY6gd+anpr1I89usFBGB3oj
PIuP+kQIdhH8f7gfdaktUYPwTjm0zqfxkyaooJulrb4OmeiuRDY4NVA0di1WxWhk6UCyrOk/BiyJ11JnUaXSwB
Bpi1LmcvxHonY5+J0HyYTA2SKeAJoXQxORGYMv04c66u7esKYjnBl+1nW/LMzxk1P9LRlEnY9Qx5mer2qyb2nU
GGj81OzOcLNM6D7RFsdM11r3sW05vxhaFPgpIL16EpE07vIk/Kv5I6ezPdeqsPuX6fa6nDMO87ZqM/lHVcNW2X
8d5Y1ZjIMjXJQBCt9bvAP4zBK1QQSdmB7ZJGJMeYjysEDaEQDHjQYjyVaCFNdFJbbLvrOTrmvFXWVUWbbZ6pvF
AZI2BH/A8VMAV+305P34Bb0X4Zxq26qSDkzq/FYoKSIsWxa9C68Z5SHCBUuWv81aiDFh9IHkRoYBFFhva8vleO
NGd0u0rWHWN6g8Hd84zUnHOLzuN3hgTEgWWYRXNKAr91jHf/Po2qRiQMQtD58zxE1cGSEp7RSrIqdcUujxHKfU
sHqYNhduEETU11lIz4au0S0AJFmcehkIdD7x011WixJ4RK2s+N2GC/KXa5EKUDUadLg/bD7pWumIK5i6qW+z/w
aA6jOmdlWNBPIfCibiLxYJjO50Qha2uzi2+9TfGFckWrgssijWKjGFZh6ugYM1DhVyCuf74/l33Fmi8v+IrRTq
PueqjrB0dkXnwgC6rk5d9gFLy5e5jY8rcGiZCrujX7ISVUEY3RGJ2ivR0bJEwjM5Wh6yjv3xrMxjyp4Eg+2DXz
apIP1RkUApAfb3hk+5tdlvCSZqxVUdasDisFocBjHvedpJxBJUsJFq5NxgQUlcFGD+oKkQVf91D1dLWHR39gJT
lT+dLh/3sVpI9p5LIZOtqgt3dUP/5MSgA4NU+5mEyPeMTBcf/0Rc6PW/rKKf6kVIyPuQ7qYd0o/6D5m6Httzci
OIiZNLb6wQpyDP0Ry0ratr2WtxXLloJz6mN+HiiyVuw4TPUJidOR5v80RWIK4rRGRYSyW/P/6BHlXgqPsmp29q
wsX4LIVGv0RzLpLpp+8ZO7oEkiRYvPSXYTm4Tsw5lIbmy3BopVswVzj1KnAG80F3GQmp/OW4E0OVtG3NVBbDEW
m0dFjTB2wPJfpBW2AZzUGu9E6SnUqoYFucZq/PIXyjHBSYZ5xrW/eVw+DV9YJ8AN7bwBTZhRB33LaVWjSPUYZm
0iYYcXrzC9w1HmVqbsqQfVYsWnpg7a0AqjBWeBdFCqAhGxpCBjEc07GiazO3+YUgMY41oVGJFysCKtL8ANVsRg
NUqGy0ZKQULEuqXNHwRpaFwVQRedYPNCAScGJeIpNnk6P0HYTw5nCHCbkIYYYJAvlClJeVHNR1rk8d7DN7sy4X
Z4TXXqi4rZhNY0/w9w8Mo7BvZgdsHpehquLQpdvEthbqnT91ds8OrJweiwzQjDN6VY3czwrjY7lC+4edvxs6j4
jeyMpcAhdRLW3vziS2rZnLSLhVQZKr4A2RV28DbHkeVlaJyxCxP8T+V9ydDF+v3aNcTWaD1Rtq19/Zsnlwmmru
```

```
S2VjC0CHB3K82xxSRNCI/oZo2P5hXdbf76ZiruYe3EdXUg0kSdwUy4jLaL9/IuK+2kNVtd3aYaJ0XzUCPSuXX7
e57POgxfktFDHakuGhD7bOt+35y0NxSC4qMMFBEaEwcWbqH+rBS7lY2l03M+BYGZ+24j3GIOIFr3Q9mBQFlrA3
Cjbq2cGn52zNFtfEWJDZmFoF9yHlK6qPxBa2vGyR0GVHPx8hWEfMmWSXS0FjLdGr1dMV/XZ62XDWHi20QLnuyc
5oHUBStcy3GNQ+/7ORJtnveISB2tSEUBwh5ZBAUeh74KBx8phbvfJ3vOM0zPun53yAoTkp04VUw+B/ft3hCNkV
LO80drmBUBO+9f6e9d0dLqjItrnPKoZWvflk3KNELHBbgt2LvvG+inko0lfKyeL9Kvi1RhcNC+tzMkq9Vt0Fza
AheInZQdEaJ6AYDY8bEwwzGSf2WAio0zPRGrGlCj1mWHxFA6SHSVttKPhZPaiuxeSuXZR/sFElCB/yxKo8M0Fw
jbJTn4qGdt2N1+lWtRsjiRyL9uE4aEhOii83yy2FQzVslme0V374r2Pwygds95mnOsyjlNAMa9BIHDFnuXemCG
j/KBMJanjC2mQSxZsE/yHdOUXrWC4DBGLC7onxty+cfNTd/zmCIhBv3UYjNZNc3JX/cex3Nk7CNOMMCGtpjGgw
Ehh+FXqWrhJUObrg5pWKZ5wVeaRVsRwS4v4q2ASCekUm65gtxKoM8IctK/FS7ugDb24zxS9dWEGty96QVgo0ym
/z1VMVleV16qpOY8Ldul8xf9QTIBoJeXgEB1dv8ne/MjNZU8WAQ0KRLjpDMobyMJb4MuvqzK80lhZFrTcObX0N
9ED+TvzUMS6wX+HP2K9eJmf2kH0US7an98uWh6vozqKxdFpKj3NgxZxK2KiIIt/KWrJ2tvOe5ts3kUfUCxOC0U
ClIW2Lu6BlP2+3MUN7//mDR6HjGT96qXjtHFWFQKAc0nhfw/tbzDQnB+zAW6dE+kRjCKo6tnFgnVYNFlhtuQxB
Z3qh23hNiQB8pRww7oEK/R9aIOaDyj3Gn9iVa54Y3HgTOwakNvasHks/wr41TRTy4bD95lJsPikC0QrsrDd1Ln
zBPYVbLL8nsvRjFI8TE15JaZyBgIxXLLe1YdPMnSJe1oN97CwcsuQjR49nF9czch9Fh7eqyXcyA61dCJBor7RP
7PryDYrzn+J1nFO7j2/MsRpHGmL7QVE15NuEB8sTKAv52UC2Oyw0Di7bpkJcx8vyulRJL2nkjKPhGaPx4yln4a
8L1cVe37Gf3ipEfkaOXAri2YWi40GS0Ep75YHYTEs8t0UMISKXl9pGfDzw68+nRmYq6TTASJl48i6atjNshKDv
/1sTmDSpi9pvUOTHgKnuELFotYCldLSBBHTRjYa2hU4RC24kOp/+kl0L+t08GE/IZCbMMnWH9hTqRxZteIRH5E
/18Jp7vxGYwnE1TGn0OFaqTS3ICrLWK5yVRa2ME4fxzD1cJRb4OiTcDlGskcTSOcfJPKhFnx6ysCeDYqmF9v0Z
wfrG4YtpksbVqBDkS0jMMyd7SCU9QtmXXdlNMfxNS0iX0zypKyq7VLXQ1RPnG74mnI9DyItG+uxc5V688qqm0Q
eybZsCTlIchnAxQ15SMDvh+UuPjsc19t4r0IdyMCsPXjgbXYSGVC6VA53DaL+QsrXJ6fnmGH07fc5hlH830YNc
Aqo/peJaMgtmXIo0hygeQ6AGccZfRj46UJrqGYB0o077h9KjmGMYTq/S4XAT+g3fachXHq2gjttGGHXvO9p1jM
PSTlZCtSkCGdX4iqXDe+L9GdDlYs2pD1lsvAJlE8wHDzgXbpHOIbbbafBTzskTIQ7R9uLyCmEUU1kU0/D573lG
vR9TaKfW31znDY7ciYZLOADNm/Ec3IHRbbkCyiPD8D2t8N84ySIcHMxMcA96IDakMT4tyHswqUez0A3T3bRbtg
dhwTev7fcZ/ZQLGmm5i+qOcmRjkcEa3fCvaKFiW10+WIr8iKxlLyGh+yr6Od0/HJy7u49J836hYwSisod/VQVD
gs+E5+mYtjNJSDbitp/RgKQmPNFaLTiCD1gU+XMl/OdocrW7bOlvPTDhY8AqgtR6np9kzIp7KgSJvJxhVNEaRy
3YbewVfBGFXWH8uajYhJWjucKHKgDBTkbTu7a8YFBxGRJttxOK9hbZ2Wa9dsPDFF2T1BrU2tSv+JWXcRi2K8/H
GC8Bs6qBvoGQs+Egl6VjQFj8DSBLYrHRg7BfnOlOeCSkJxTH3C+ideijq1rqx6ExQys8yi8ZJvDytx8GfHKqUl
Q3R541PNph/typhfwK0v/yGw2kYwsCHBPJPJRvFnSlYSBMhXMfvH+fWvZi5cNJ4mlzaIT6MScNaj8CvrEDOo6o
6kOrO81mrJKK5gQZkv7deXVJLLUGQU8u8mkkaYfp5a3kwYe9wZzLtTkwquJZ2iNr11GOvnFWC3HdKfYzezFbks
2LBf/wcPognd01xEJRTu9VFxh7KdqysqGWnrd8X2FbUuzIwZRuU6UYcRKcamKtW2jlJbr77zClRdtVmJzdb38s
Jx+g26iOLtODO//XgbUqiaOlB5y5vXJ/3zzmAyEppCI4cbbLNfSX70bGzDVuWixalM9VfluI5NI0AH+uBld0uk
yaGXnWLCYfF8rQQWw6zyZCI3QjO6XsJHHXjjcWqpiuks+sQVjMIBdNt7BQlm4ESNKNWWZ/NxCmMTjHI7LITYpr
wk8rN1MzQlmYopl+2FfwXtZoTMAvFeskIrmZTkp3nJsMKAFz973ml04HOgYERrb0sEujqbQYUpSg6SASC5++Z2
KqBslYzWalGsCveR+BDTTo2XML3093Yiv2m+ASFUMM/cxR8BzAJZNTiZAVlVo3GPO4WQ/wRQNZ5DWG6xp/bLKR
FrJinBkoyyS4uHMI8SkoYhEIOMNOSzkeiyu6G5ThxPXyJCpffB3JMl0h7ELuMwxeBzcl0JWpAP4p6N+KqDWSuW
EMTZJnL1gj9JXGXbIfy2IiU9mZE7MQGSJWdvZIt6ekpReu6lZ1dBeXOsgFiuEKhBBob6fKx1akcwI9y+iEO70W
27ij8b3XbkwCP0oZbwFpswHr1XPR1oHIBJjo79HCR+EakgBm970e+N5tHzcBLuFhap4cyhWJEBgnNYzocz/v6d
X8YeD9TDN4b+vvQJEdSA6Ue4f7gLEihuHwZxn/tOf7SZ9emXPruzUzRVxh2G3KaMxPsTAEJYPwSd8L4c/2T/f0
djBvCG0btgeKX5+/nGZVm7tuUxJM6yb4patp2knNU7hruizozjuzMj+QPjRXLJnigPmjmNAeTh6nXXecX3v2Gv
m2Se9bYo9ivpDlzVpSjHwKvwEpBo7leoD79nrV6jUSU6kXpTYd6O+SZfclby8mdniOhANxBUHIHuUBmEOt3JRy
o/hxejRlzpJycnpaJ3CYhu/D2mBZrhk7FTMwkzoPUvLz6KBzvHyq497FCwteOSc6jmNS1a5FvYE6+0O8HBwBuB
nAXTWDKDCRNpaLlINsxkTKlpgQbXuxO4lsNpjhHHUcmhwhO1AvBDMDMz4gPFL6YdtJ1gPmWg+HJyHHVeZQP6X4
IwdreiS3ns/jMafl3RM8lkgrQPEtWLrsYHD+CC8j+mrAT42JhyaMwMxKhT7+BInRacDQQFIsWXhwZvu75Bsg7O
7bS7iXir5IAmqtUzfehtMKo/Kk7N9fiKLe98jM6y2siRpKq/z1I2/kdmRfnZZSjDs50AAY6/pTXtjMlsv8BISO
1I9BRaUONEMzlkCNbTm/HYyFKCHjd8wWqCQ+vg45m+DV18xclb7XLUvQRUE2EbqflodPDAyphMhIwrmwnMGM95
tcgby7rpI9nLXCprZgZQ6qUxCVXUY9Q36wmmBiNr+0SzS3kYNWrAdVoWTVMAFGr04M82vwh552Mz5/aJpAR01t
4PZYO9xsNNLHBi8Y7k4pkvQ95/ebVTuMkbONf+yt5PtJ4SgIStmXpbkQAlgpPoGViZjYwh51fkYRjBfyhyJs3B
SV/I8e8lIhlQNqV3uBiVqvHTjMSTiwFs74rX/4lBPLsz85B+nI4QvyTS0KghZyKzWXJYdyE8TrMAvzcvjMZ0WP
AOBxIQTbhzbnV8nI+M22re9y9cktxRSrqucyySj6ndDQDJqB8bphuphev0+lKZJctlA5Fswgp3e71EiQ+UbEOV
cJJ4XkPTbwYnvNADkaw4Tn7R7YGeizweEr5EJ4D+LSOlp29w0bWXZmiP0v+ABecQdCdtF/HkGRvCUzkLusdppO
ISzK2Lig6jonlqMCMOQyTHhJ3eHDLgWn63K1ZuzdYAc1qnno076KN6E0ZzpawRdgv+pAxuiCdYEZF5XZrFS3HJ
```

```
3N1wqY6TaecL/H303D+rebkJ7A/JdiHsdZ9xwupZtNLS+alkvSZJy7g/YAMM3LOAYamTqyOQ4LXZNtX7AQVHNY
VWiLl4Qu3boQEpIEPX4QSpbSu9KDTDpOX0wN/MNNxPpXjQ0HjF/c1dUet2fWHvZXtnBGAVdn7tCgdQf4dKXfP/
MCL5Al5rYbxwOqbsZPsl5poP4gfmy/GKGsXIeLJ2OvnrNnptLQd2ap4ciC+p8odTdmokfETnvqjUkpASqJHrh2
KPhzxWmZaDESNXLTAKPhWL+RA17NYH7k9ZQzxuFxXCB55rzR1/W9lSFxTEctSy8W26D1WT67+tblYBvhbctLTr
BEf2PlnGx0zAAFwoRyxDe9ROAkMEF7sMhgWdeagTkGDdwohWOWwFtaFH0mhxTBp3zR1Davm/f5EqJRcE0Ny7JF
ua1Uwv/UiSsAA4w80aSJcvRoDVWJnzs9+1AJpOizby3EEZCjxCADMOVeV64sJ3SikMLgfE1lHV0fRIuQ4U+rb8
OGNqVuy73++fZLq45n8vQOyBBLGB8Ovt1Z6ymBivJO0aKcD4WcPWN4kjvsPQwPHGL1xTw6iJD3pAiCAR7VH37Y
+IZJ+kBGhdXs1E9QO6drx0vIgtuJbU/qXnTOTqURodICre+pkhRG58iYiktvHa0zWx3GiDv2zVNLNFCco121z0
QCvWBtImOTrj39FlGY9x7yEl/Vy6glPNY5+847C8ajXjuZROSgMWmSGFXShJaDrTzUlqHaFU7m7JaDRor04Rs0
Mk2AW6v+R/vnsFO8w+HF50gnGaNE6NwMy6L27A+DPtJb4WsHju+egaMQjDPyJp2LsEHHTh7RaKXhhWpqjkdYK9
4SwB7x36XaKj46KAcOlHbP9F68RB33Lt5ErRrdJY5OwLO9JV2pKRwFnv+App+PJKWRNYhQ4gTXtAjReueAyURC
57OYL2qtkGk/LOTbbjH+sRWry3lIHTGamgEWDtVfbl4leq9dOaBMuLo9zKbTdxGWao56h42XdV2jEaRSFucOPZ
YfFNLYOl3CcZGoSgcPwoM3Bu8iqScaXLOxj6OdqBL6m4JObBc3QOsDDSnfTw3VcrtReEwgiWJY+RUKFRovd3ML
9vYtptf+sz9ty8gXvI2Nb1Lz9KdVQBLJxvYwfbODZT7qxUVpMzufB89//8XhWqTY6QNDm8qtVQg/OP5TCWplWD
39CVA6AvMvQJQ/pvUMHrV23dCCd6lPVEGQuek+UocSex3R/QFHHl1a6d1J67ZjPcal36w4ahi96mTPQvb/Idkl
5yyDYj4v85J6dZ71vUxhj/JYZGLkCn9UiX5feKEdRKQHHF7vsRRUEPEb8sIrpDh5NZNT0zEodsWVV5O9VMLpqD
Z1SNJ9Ma8Oqbb+zThfrewGVH4eeexXGSzl0lXh+/PSsxTFNbPXurjt28gHdi7SIQsR+SSuIFIrdEBZqc2EONoY
EbjB5aun+ClyMcsGGH7Rum6jsrJqhaMZt5Re/UwSzwdRwCKjUORHb1qzDX+JZOnHigSNtr9Kq6z/92efaS2dhG
z3mL7ZkbOjgwjxKLxbqIlW7NX+Ia3L+gZ4y8KoBOuupL1vTxXeeA54V1IUC77z6sB2QOimOzjdn9/yfHDVzBKD
/GwKpIXhZ9QMGh0qrmGcMP3rnrPlDnDEDI50ankSLQsd+/i4pIhd1Wgvh2m4GS10mLquHMxrwBy07kXSCT3ggq
WCMWOjX/WQJtv2xmc+uZgRgsib3bxzxHTI2D92U9QzLbVwhLi8tHSHrsLGcUk8fv89U4L5yaE3XcSmMF1bp8U7
xmLRWv/9ixGMeNmkZ8a/9NSuYsVuzbVt1zmlRQ9hQmhA0drkdrJsugqeq3bDhlU4jT3gdHRYoQSg90gzUbC+Tz
oGKIY7wGxLsRo4oPzr/VcGCluBEiF9MuTI6z3zI5vxmG49tPSNaDJ1xHqGCiSrZqEV3Ol0vK8t84N+aB11oXk+
3+Nptt0QWhM3xxqwTTWMbJ0+6I7nEqYdLOlw51qf3aNL2nJ/cRe6BeLhQTEOAvnrbKMfFlNdZ/ZBRBhmSIQNJf
ChAOFQ/N+i2A9f9kNljVxziAllNSDrwpHcZoNJKPptSfhD5pK7+EKDNEFAR3xKEYzjbqgHUFXsYefoaidRs5St
9Wclek+OLSJAo0bwo9n4fubVwTJBGj75oFhC/zF2CJ8VlNHyErZgOu5IZXxGtu6cvwPCMAlpEJi0uQeD8/wqyQ
x1KZKZzDLpUT9nj3JOWZfVCbyhMiyZkvrc9NU+w74zK6R3KLJewx7dv+OCFKXhK7ZRHNXo/gfZfNb7u9fgkp/X
qmdAtVurJ5/JDCawwb7vR5qV7sHGF8vTDZV+/V/EqlE/Ea0HXxNhyzl3MbGDqmYnFucU4S0sVjpfzYWnK8K7be
rQ3ujaunGWYsivqvO+UNMN5IODVKHvuEQNSxZRcnw7fUMq8XLm3T8KUg6ESwFOOtjZEUNFR86kb5qhv1H0kIVj
f7oYtGlsSZyFEXOTQ9JmmT0aVf5n+dpHQNK6spRNMoiiBN0CEg9kXmEkaQSy+4UaBmfxpxuLNJLwSqJPIRB35U
6fF64uKA/Tofa43OarHRufpFFbHziNid0aegrZJHgI16V9fRtSQU+ir0JbswrbtmT/cwX3l2j+tK//InF+zgya
E49L16JMDIQ88vzR4yWIJPehGO+/oskMeqqMn19nM7cpVsYJzJyENMrIx0LmjuxhtY1gSlH0S6YAbB0WJi57L3
FLvfsfP/TMZkJ44U1yqp45t6l8qY54d48S9yyx9TC1Ia2ZdDYjR51Z6OPbowE4tzUAmnzNH55HvQH3BbFU7q7Q
JAut5DcnY6eqS03WY/RSmGd+OZ5U+xtp71bgyj/aXe09U9HPYg4wRp/SScaR1CggdHiuz9u69NIN2AsTaFhj4a
XpaWEkwi66ZeT2KRI8s6qhmUigfDeiweW3eWrNPWWhJMP0UCfrM4VNN1XgdWKu/GUdhy8mG2db2LvHU0pp3p4+
Yah7TPRlFFkyuuZB5feMPYIEG/tc9jvoTi2ZhVVUELybyt/D93ISZw+dqt25OcHVbCW0+4ajU8UJlPVRKfKeaj
Zj5LGEVN4ho70PEC9+peTPggwg+WvACGYmTW5MD2TXALzs7c3S9pCIxwFLSrQqOK3wW/PtRLRiJyRj7TETEqe0
cPQCGxw7XgGKCfddYA6wnX/TDZQxp0LUz0nDtvESj90BtBL+KAXIW3/KLH2u8cJKDkDg3TVvCNNGK0gXT+bhVU
nrC0SW2/2/Jgvg7EJxL6pgd+OzJUDY0Cx3dbFbjQtHBIxnVUdK7MTKgWSv56Ttfru2UDaSabQ2+mkEqZKcHKSe
SJVioJYVGJ2Q+Vx7OPLUPE9/NOvTo9ResVqYz+HGxwefOOutovGeQMW16IL/N1/Dhj4ITlN1F3I/8oAd8tX8IF
8Kvz7NNbi1n/i2IaE8cdmGT95MYHxyd725wzwg+4l33V6LWx3nYU8N01lPG3/mgljm3jzdbw87vp2XmPuEKzYR
OARkJXnH9wOGJ/dA5PDGt2KBNUICsp6GsBZLcinIJEEMkLT7bidO5o3KfmBpv1ECtiK7yXS61FMtwu4Fh/sR6o
YJ71goMUnIIVDb+IrGf2Af4iqIZ3wjzQwcxEjBZy0c1PdnL1WUZb1vyrpfvTfCeyMe5YjjHuXxxweGatush+iF
pnUZezeWZd56iqwNJXixiAvpqNHG4DMRAMoZVBHWbZsEozO6GGH2zNKlfUlaRMzTdfYcKbqOSptTUPRqDwAodW
xL7tw/wZFL6rqADVzzauEgE9Tv3BvDQCmqAL++2zSB5ld+NahpJ78TuAuzqhW+egWJsryD0Q4MeB8dfgUPyctr
kPt1ptIBQ1BHQC379KLVNBSyHI5vDZ6p4p8PkTu7LIJf6itUNzH/kh46p3ktPTqY8vUN26LJ4uJvUliWSSpigb
Y6vpbO4i7jX33QKZxB7xiZrAcpd+1QbZ5jXw9DxkNl2VbyjIB67uFutAaP8VyAvjuYadY8JBXiGdKdGbRqzobK
2p24e6s7NheOeFSi0gDaNVW8a69pjiDje+1qzWBixA9KUDqmO/LG9JcyZucYZ7DMdeM2mecye1ehSASN5SbGrZ
LGiEBpeDzcrY/LC5a0eqeLZ5IsYGb85zdnpSIXpiddF2LUtshrbC5dQY0ZmXAlY767cNh5tIjpitVkfUDJB21J
0wsfxsYvWvHfO0krbbBHyeXQlkXV1jLRF+o8L2fedIkv9L35HtQPdB7FiVoOT+RewJQrsca5DY35AHXMNLPRUd
7wGxUD+/5BkJ+tr7x7lBzjejMrymNvVRYej94ep6ENxec22f+PcS0+lOOh7ZxFrbIEBstLsXoVFqKhRe0hW23i
```

```
RmBlcRP371AmPc8ZTsjgJZY037qQwdNZX8Zh5ouTW/lLW9oHufPpnLZLbphSWzBJ/FCONWUoC+TO36+FkWYNDk
1p8rNxOTcCnbQm0N3tQkGfWF5bCOX57ccN+z132nGdxnQ4pJ1qtnA4GcTMfyaP9+OXBO/P+LgP/WMy5ynD2W2i
D0Me0ZIL0YVmbX7Uw6hlfLmqV0lKl0Kb9HtSm3JFp4m98DXO97lYcVPHER8qiao05h/KBK1UfCVBF9HB75dgU0
StnyThJug4Is69AN9X3KAoRjqylvotZGsTXt3fAnBr0jjDiHfv0J6izX2YpNaTbNHMgRCpVZiYibUSVCGRpRN1
+DOE2CTFcWgCFCj1Awa6AnQ2rOH4XzrhBOo5zXf2s+h+xRBQMvytwdeOZVRCcr6Yb9tHNwMvi6BMRGMUgMyOV9
bNkA15DoJbwGRUWpZV4fESWDuiosLnijpVVlhCx4nmQGVdQpuBi2FzU/eBa/cZCgpfK+pnPQo33DBzu3vOPEhq
Dox0OzInTf5IEm1BA1xVR54Yn78Qeqfvf1KRIZCeeTxEMnWdnYPQvKJLsTdCpoEgQFZGVKQ/EcVN+sRpqTnqZz
bTLY3HDjvJsbWJFsZ3BgFxpZAz6Ws1aZg9rej2cHsfkYADHNCbIg51agHany0opQamRcxL8XCvFRfLkMzr/DVj
HDJ5NFdqwTPxlsfEHlIyJr9yM0RiX2OepSW4z2W0lPWmuV/1LpaSdHwnv4dd4oZCML7nXlmtrd0TjppKVAVZVI
ObjdCsRX02rw+dxxv/4Xln63QBjV1GHKXfxVaZZ2MIS3nNfhRSPQnf56BBFpugwCNcG7nwhgNwDHE51Gx/BGx7
RQQCg2GzHkVWAZ7EHetuzXgDR6Ape6BnhZCDGC1QLdEV8sBU8S/9K9mbaZj0tH5FwqDhxd98d+tL108DYfCam2
IhRSUEiDR8VMUlq0rNUcBJWiY3x4LSDXMgS9yNLdEO2Ja/1ByL+isqEjXJKYg4HPrgEioCa3I6dAIPY67/icIM
wO3CTS8z/qwgkd1cRj6PZ0eDnFD+kivvDbecJbHs2DCC9xUTFr5u0cPpj+1WnOfWYvr1DxCbtiLCcVT6qy+jMu
SO3io430P+cclXuyLFFU1zsM09Et+NemzZGfg5MK3QifqA9RcTo9+T4+oIi84TX+DgjpWeqrLltJf7CIH23zQe
8VXlCZdgs4E9XJ+HkobDH144Vehf2yZtkS9zebc8H2EEhvfEbQz/+CarzsrAJwIILfB32/2IPYULFkZBHjEMqF
RbsIVk5kl8cruW70WrXQ+zvKUiyjuvxj+5ykYg7lZprwRZAGw1V0YYjQ108Pf+xrRsn+XtLkfBGEX997Pq/nyK
6PYXGk0jIQr76KLKHRcQrCOrX9fwSWqaQSTRQLJcZLs1rwWMXkOht4HlzFvJOAnmadYUA+YIWz3SuINO55HyXl
tLZZTVf2i8fIy5q3o0fqDAjgWugiIZ0lzOMs0KPcb1bmmP+M/SojLshUq2yhs755AE+sSrlNOlqTdREQed5cHA
q0oxO5Q5Qel+UdQdBVMbbdgg6Ig/CdNNu5Jf6YZtory/4AjQAJK0rtzBhn8ni83/dS0F0hF+iRmYNWHeKmoU+S
KDDRVkPqYEMSKDh26kKGyhvASbDtjvV3mUwirQq8sFmdKdK5bCwMpa/YewlCQlxkFT6K6xQJ1Zkz1JUlQMDCpK
ShCJarpkn3Htn3Q9La+ivWFngfiBVXXcopztTf3SRrsZuStHmtjZ8JuI5n2Xxbdvs6RDfodRxhTMWAXs0Fq/aB
28ZTfN5yaqJa2NdA6yklV8OKhfqRDWDvNOshqcNcVIaUPEgasRWSxv1VjkowehM9YjNhQ4TwluEhgSQwL0Evlf
/8eG5aQuJA8jHGrAF2TL43W9N0YkqoepXs+8Ect6F7bbnl6pwANhVUynq2OfF1uyWlEHNDrnwoXugPYFBRxcsf
kxxDeoCz7RoG8UxaATjxCj6wWkYINtIITeZCGX/kOCx393BpoSTVWaFbQKCTY60GnEInvvsyq/f4mnzhIb4pqk
juzWFGyyqZ/5MFHqrthX+Zeh+rjouPr/japA7IZwczq3b6q1fu7nYevCes2dGT8wVY4Mc91sLf6bh/b+nQ4aQl
PItHR8mUGyBa8Ztlx5v8cM9KBihOQtnvE/tPVnbrIq1n3zxvNUT4WHQdcSJAHxusXQT+7iFiCaLi6lkrF6o1Hu
SL5CM5iP/VOQ0hZN+5+m/SCo9c4vhHBw8pN1btguSCXEAsiRZb/fUZB2f/C6iVQ8w4pihSm/f9BfWSWM1FVW/7
URPdMkbbfmgvY2p+UmgDP+DaFxTGQXA2uaeNFt+c1aVKzO03oo1NKkuRXIq7tc/AXqgMscNM0ZEtibzILWvZmR
HZtcLf5/YG2mkntx3sooGuW57kAiUsRsv7XoJjGChLByvTLBnUcan7RSX4bvwVRKiT9cHOSFSU8BnHYMF38F/D
PG8Rsujp914j8d3/SuEilV+Yc1Zbifw3qq9M7sKfm2qLJmA1J0MTNPf+IHWaR5VNn3qRopQRXqErr1/ExuCp6g
2IXcDjTFBteGvigiu7+fF77U4GJPQvUrtxEZpQL+l0DYDBvmbMRRtooQ9d2+lLRBzst7IThdjM8jHICEcAGNHT
JS86ur5yAtVHNCvHZKVHfy1Pl98APc31oXMctxV2SdcOUgv7HRqtkjjfhbxM72O5JL2bbkCMMyX7WEZ+3zKDLK
53KRQ1GEI8urwcrYzsav8GpASE9TjpLWxjY2bQrnUVXJuMvVG5sUfDi1TzZ1vdgU/CsnPdNY2E25AExOALia6F
URf1lST91BJsmlD6uYGnys2BoYO8BRLg3/pjLyaID/KMxnYoR0UYLjufkptDnwe3OS4chLHewtp3Mz/AZHMrQo
VpXb7RwMY68zJTnnusUonkBggCn8l6QGJclN0+knGiX4mkqv6fDLPS0cI9cibaeF7kjy0xK9GUPrebt7aM/DMF
MWWKpe1NziyjYrdiWjvIq0LCyOz9SeND2gx+5CsNnDZrfDmaD3p6iDRvEt0ymWfNWKoxGnLQFv4W3N1/MNpKVm
PzpjvrpghltYTbgBZJK/OwyJpakcodRWt5AERC+Q2S6I7rfTYzWOtv/9h1lQosfOqxKOrBbH1Lwn89jEptSQmM
8QWpBvDOUAAgoCvF6SP9IFE9pXcA2yUs5rFlv98/Uw0Hmsi9/TCVHMC90kQbUQkfRmgDI7Lll3H+1aCiMIgEgk
y75mlIMitFQHibIH2A0IF3Sc5qclzOk8Ym3j9QUYRgTDxi7nQSKSpJPv4N0J1YSsxQ7S0A57RIHgOq8zitam5h
zavH8UawU5PIlLpm3H878LK1lSCxOGU9IB3ApJSlZxZZ0j1fS0dnfZDJ+ssdZEtTTH8YyENDyRqfcdLk32WAPb
W+l+86KduzdG7aJvL98/oqMiUDowKWgcbVDjv23M56ajfZgRPmRFpk3DfdvfHz41DcMmfeQ8kZNkCL3npsBxpO
DqafdRB8+KpxerPbN+K0BZoi8j1RsJuU/G276FF6YMbrFCvscO2oqXerMcxXJrb3Uo1vOBA+IUAkIDcEj0wmsY
xAZp3tnz9bnnJWlMAoikDS9HyEf2lO+nzQZFasLbnzQhf+UCklKFoIEbm4EHKwG1rPOKKQ1IcOdIYwF2XNeRN6
v72/b8WOz9a8dfmfN6fSWeYBMQyneUvXX5GJZ/+DpHA76z/+C37UqrExwlTyfJmkNi9lH4MQmHHilHU5UQPcWg
+RGDjxFmu6SOQUBQ4uJP+QdLkc/vDsVKHw2KksAOtDndZM2Fi/0fUj+GCM4testCEU0BVf8JlPi9z7Y9tdEGw5
RfCDSMhDC7IMjcSe37xHNS6hP4aJN5aBmPlqgHP1voYIyz+XO0DixnNSxTWCqXmxLkWFfMbDovzUZtAJ2OlFIp
wgD4qY+ppEza8rBsi3LqgEAb0A7rTB8d8XzzE22A6ohyZaqCgPEaxBpGwOsa/KDDJCBEgAyEIQe1oAQMeXMT1d
Dd3FAZjZlEaq8mWyIkZd4NeWauju5gAzVBM18djiJ1Fzpc2V13FyQcStN8I9ODp6GpspsAWLay37ZnikxjRwpi
Co3cpkDpoynrv145b9I5U0zBJtCw1v7dMpBt+eyhJnGYd0ROXs3UX9rVXRUekeymUQZp1flmuHGf4jzK1ImN5S
eLNEwHEdmiQBGVtg4COQv5Rr6LsrzK0wXsPWglDUAeKOUv+9nNRSqRVaH8rfZepkh4kkGfKMNEHFo92FrDaBU/
z9dPmC1gIeTaekLwdMjbHQqfo1AR6dwz/JM8h+Dhnca7k1BkXfE1s6r7JCvAok5bYlu2Nm0yfiktVe1GyIr+9q
```

```
oZ5MNknyocVj4gzhsY+XSK63vvehB7n4C1ROXtE9Z8djJtnxWb/HECCOHC8BDkZ3kzfy+ZNReW1e1mSnRefkFw
/1CTR2fhD00q6S99eA9e6jTLvEx1v8p1vsRnRhF68IzXre4W2/Bnb6ppTNAunCigdmUJrnwn1tlU6WdvFhMbjw
xoDhRiBqNrqafjEvhUH5zCnbQVOntxp/A99B+2WpEHmBbg8oyVKodV2CQmjE4e2aii2oybU4wx2SsdSZY5zwly
LXsNrigrizYYlp0UUr68SalGRw5bKSwzzKEsAbNQO11224eDThcwE3jicJcrJ7EYLttODphVQfFZUvCR7drepG
daLnLxP3txMbGUZTl5sU4+RX01nyI8jeQDt6jOe4Tsjjnh0mySp/OtUcOFJ2ME4UsVsrcqfWPnCLucy6X0pyzW
XsEedsnz+kt3znYtzPTzTaHXqPaJVFx2ewt9L38sznS7mXkfWgWV9K9dUKCCkD6TfrJuhpOpjesYG+FlZhrPEp
BD2tConchM9XCyhXWQJALOJP1ASLVSRHkYZnwjlhnxI4PnKZ/ZyVpyCsCl9JL6P3ekZLOLvA1PzaVSkBNGXT3M
jbOia9FOD/cConJEsWX3WHscD4pRJSr1Ynppzm1WPdn9kzxDlOKz6XX6T8Oqg0UAQERqSAMbGKpBPGLxdggTlS
AGH+/6iQ69wkLvz55sRG3evvJNuOiKWoJNAKTVwTGg+JwA/N/swcQkRmRUtzjh3H2rlYsl7rhgSpRoOF6ZwO/9
JcrIJ+DGRBnpXb8+3L1kQpcM8PiwXfq2IpDoGRuw0XKwAqgyV479b3e/8TzU6xQCQ0YryzKTlZ5Gg6hpPHkei3
OYlDQLp8uZce/3KUXUsaNSVwAk85y3Sje58G/QKfm1fIJOryKpF9tNaIyWmASdM9SAkaeADS4JXaHTAfy1mT4j
Err31TXCIvIN/ucyQJVy02pUQQQh2SsGIYAYtGOwulx9qMHURd+nBWez+nlQnU4nxsQ8MfzqZjp/pEgmUo8IWY
KJk5CaZkYr/HLICHe1WYLMDK5b6rVObCauRz/XgubW9/qDL3bKsC96OyPDqIxw9CrGbxjeUGZcklzQqCMMz/YH
bHCKK9rTPSZmAbss1aR2d/MH7LVEuMySSEajr6rlOKqsmkhZxSm90JBIsouwiNhJmqWgBpVYVwYX85g4IQU5X2
aBB7oijefEvJbuZIJ67c6LBZZand2edfGjs2mhzvignDDjLgajrRJzigslP9chqrz5xbnHbCRz+kn+M8eCfJ9m
SehnYmvQ0E3Xy69ifAAivYNSU31RdWChiTb3JA5UkIPyGY9yNfWJu3JmoR75n/+9T2NjyV+BVKSkNndUNCiGc1
wL/rsDDBQrSpIzZfVUvetGxYAcTyEMQGewuDNZauqXVj5igwo5NdGrTVns3RVG5VLiFCVgAKqETW80B6JlIMIn
x+FGRvdIwH7FktFRMQR7wzkQEmwXWM7Zk+ty8BsRbw/pNr9Eg43q9rcSy/qsbpReYEkvczgaExwWG7ODK1Yw0x
DavfaXMwP0CbOZ2qjk5zsz4Gijck5h+IAq+2nFj0qMgqWt5+OSPjVLZ7ECaOcUZuL/Q0Bhkdc4fCmUa5ekclI9
fkjH96lj6AicNAK9K6JhfJlmVjC4Q/oFcV2xx3vafIT//1R14jqh/hMcCUke106C2D0m8ss0pRIS3oO5i0eMOq
VZen+xamVXOjgC63EA75wn7UutB4bRRZ9ATZ8kP7KVMCCnxUdyPFrgsOMxEJchQ5oPjn1557DYSA+W+otMvMBy
t4BnoTEt4T09NLEBmnuI97isofkz/1QiP7aJxzTdszSiVAMGcw5YoNqxRPf1m8Y3PUEvSPogysmMSIag//NKOo
LwivhmacKbfH6uhvL720F8rnO8edI5pWlDRkfNjoCSwC//pYg/6CObIznrjs/ihtMzXNOggbGP1HQwahzauuWs
LvhwMJ668HOyUnJalyaMfY/mInaoDGIFzjYn3CF1C+qgE+YMl/rlBOGxpWNrQFJoPW0uvK4l1t/fdrNu5wYcqQ
adSZNNPlVmUDyuDcF+x25L9DouBBrZbEOkaGh0sY6A/6pRF+kuAbFIPmSV3ZcJjKZu+enFdGiZ/kolz+1xU84b
pyhupz/MiQiMeyHeVmjCvrS0VvBdRMv9gvEAUNclxRQ1WM4kwus2U0KqX9L4yFvD0kpy0p28iWVxxip4offlQr
UQWgPE/B1gNX7pD0zwTh/upAo/jUbUA6fPQygeo9O8m+LeddoCfcEt04jCTIP+nq/ZvuVd3zf93BYVar2J5qR1
uNyTPfA6dX56YHRT5I7Ts03jJG8+riLRxFvVoJIfkI5LtgpzEy7e9Z6h6Gvigzcb4ebqJppht+39fj6Rl3VWzC
3wFS/Hmha6x8J91onuWV9LIWXDu8YX+cY8s+ZwRDriYGMvW7D154AkeXCD2EgNyacZ2xyLaaumUbCOI2JBMaqb
Amcoj/n7U26A4azt6DMax2BLzv09MZCCRnoNtSEhevGXzo7PlCyWR+w90+EN/B6tIYMcVxd3ATJqnYuOafqMYw
pTygjsEt7eGvuTPmNln5ro9itEU2aUbLVwD6LHbmqJpsZDhoNV5nACk704Ij4VR3EjO6Z2hGpo+R8EKF0uXuyy
Zf9jWdItfthBdGFqlEXaWHsKWjEBhN8fm+Yl6it+L/sVTS+YckdvZfyHNbPrpwY2Rt2bME+oQIzo9zuJ2LCwEj
BFyBxpxLLMjDGLfOyNbtT6LeuCUfMQfyC01+9HxdQ2TVKjr+2jAfLIEWHpef/6F+TwYDXHc5hSLQZJ4Zk7yZCc
ARzCOnsvqVokU7zq/X3VagpUITSZz/1OAVpZP3BaRkjCjrweBJhuYTrYcoOwLyJ3qFAitwfZDyRG2Ma0fCcZ+0
1SRv8KY78j1zdIgMGgKSUudh70S19ICuPeWe7cLTf30IHW391q4sjbrxNoZkCmqPee9be1D7LyT/ZPlQxeJJDP
Qxfqvcdmm+vduNkryjqk9Ngzt2Wv7GbvXQ7NbJKfbPB086JMcI64DMD1EUQr+CBKopZNVYTTCnQxfBqUIwdgoI
j6g5F9xV2JzVaWNRIsavxSnnMfz2C65Ln2W5HRppkh6sqy1xL4LguyQxqSTI1TRzI57FuZde43EvJ4Zrllhwuq
XGWzZ1vee/+84QGINPU5LFWBDQyPyQkB+UdkBh+5kHRLUertHrNDYRV6zCmsU1M30379BcRPAl1okLuzLMIcVl
7EGkjDw6+aUCzVkQASkOGWoQEM1u1i8NIi5Nak8z7FwRMQRsTQsopcDFARHwC3uwP70PvQZ803rP6JX6HVE2Nf
Cc7oge9P/G4dlGOEbGEctA+zcDf7T3Tjoxmbwuh+u25rGkCH+F2qZTXLKxYYbv9/atYv8NiGHw+JHs7pAG3wQZ
+pHbulkSGAiVcA7UgSvvi3lftnLQzvCZt45Y4SWHJH1EgSVVX1siJaHW8+O5Zu0fl9XTTRRH+7AJItwYgZ1iOl
/2SaSRTqh8cdC6a77Wy+he7+7i3h5NKD3CXHrV+awPqRW7tkMbA/p4YHLlRsaO9pUAHdRsVacvAleZSox9q06i
zO9PH95XT1bQdRBw3RhrsP/sVWQA7s/iXG13WXHQPiEFk4+4As3YMypnqM4fNWCHngPwKCfGg/w8Tx5KtXuv6s
j3RAeVbhzR657iKlJbG4qgAJETIgNGqXDQ7auyaBP4Q1XUWzyod9D5Mz+xMq9o5Tgp4zpSiYWG13G/mC/qwVlM
sp3vOERh44tRhjkR9oJXa6NHn4GITQ9P3MF6gpEFXomILXDlMNRb44qg5UGJM26tq44peBWELCxGAD1j5mlXdI
B4sb15x5RPiHHbrRkwwy6VfSybQR47CJgduZiMUqIfWUKOAiiT536FnjDwvgU2rzEoDiFnrLuX88trV5D4Na9O
S7cN75baGwSgmvIYd252q+R2BFfBewz74NnVMbPaadwB7zh6/qw7+qmN+Mzb/oayEsxuz/ZGExLO+whavX/v47
kj7qFypcCdO9TGZBCwP7d0KqrtHwp601y6pY/XAo0y7xUrxcRbm3EuVmASuO/mEJadtK9W2g5McbA+f3mjsRHu
XbkgWKnApCcxXhvu/3OZoNneM3+e9aXqqzxV3AIzZJf8EHAKsepcVHFc1gHLgvVTfOOaTOXHIk9S9FkyaBt6xO
yIsoHCHdyrh+KpvdT6/0pSpV8PGh7GUu/L5O3T8e//msyMVKFYvpz4phEmS8CRUdQCZrizyktNLhP1A/octiRl
6RoJH9UKMC71aCLu8BwR3uov6IaXKzJrlhd1K11WEl/FDZiSedOtTw8V6CLnFeKqiv7cGWd5ctkvkwV5K47K1w
```

```
3JtwGVYjRuplqpzWgtWH8oWHvnuj0LeVSblgh6vCJU39f9jlD74KiLIBCaYI2SWYamKHmi1eAVLRJssTq2Yhd+
0yB321WJlZN52nKSn2HfZWO+HAgXsOFnGouo9FRvRqkrKcGreUrrKKBjDB1aER0bggXJE4EF3uL5QTN6fjF7yL
enadKq5Lew+z0ubVraWY0yhjkbHR6sUjusHCb1DoWIWR3jITwlumr/lAG4ltf7euPy1shbQhDdtHkdRmAsSsD6
o0ORMMh4I9QhSd6PH/T0m6XqpRMzbw5UXm6ltEmbG8hrZPv6HP+/NKORv77tBAELGP33f4H3dvmLYM0mDMPqGS
rvYa5EuKV1Z/JGQ6InmPHo9D6p4xRX2xykxiy6PwriMsg0u1pm4iUlIC+Bt9HkvJE4o4nkAcnim1RpDBTGSLGr
lmOVLtp+8Cz9PyAVK3AlE9t44AQGneLFpzjBF0hzxkImIqYRgpU7aeB4RWp/cxyybTWscltGE3G2HzPhCcDmEa
rrQmoyC3wElFyKd3KYy3bxYKEAehvJU2VQ/TnHvJG9pNh8uNm4Y68vcSbR2Ga+n0NjFGVRC44wMzPtzyJO9+dR
9aDRm4H7XMoTScqBGfCmlpvZlc+TuVYKNBkhHKnzFSsLzkoFzfseJE/ajqiUoewp7WoO7vR1kLiN6YVvMwW5Ig
NtukrlItx33VIc+x8g8krQ1KXLfi8ng6WRp9w666Adlo7W09h64pqIIcxzWQsjAxIyME8ECV3bmtdRPxjpdYkE
cvJpaO3xW3l9sVM3M4zRafrI5u6oFnVw5kZ951SKy+ut1t42+fJ79LLjvCARwccUArcom04JCwTvG7d3R8zNr8
PJhYIH9Z1a1jFnGYhb4YSc9be371yfYGrYVfzJ5vd2QU1Ul52iD/deXwdCIIfuJX37KUS43WWpd4Q+IuYNlMJZ
u11S/iAQrv3KUTUOtbKYuDt6P1nK3u5DL+U3ACdQ/c+s+9WA5oKDfyK2vRVIVb60afrXCPWDV9MwCiTo80T2cT
DUS9R+mWnHXF016zC9rqYJGFdEFh/FpHOgYARhDaVqEAOgDWDBO996JDN96Moh8ujNqkaiwmpQ5znrPlDnBID2
T4oglkigFJcRXb2vDa9fmRwYyoPH1y9uqdpDZv/w9KoMb2QveZoRJbksBnWV+JKnQVjEJxZlJV6tDe8fUG+G9M
FnB0EIMy8VpgIJTpIzQpLOEJV5bV/OSR4wYLQ04ULhlzba8Q+oAlmCBs0r/HUiD8r0a0kGX40C+3MmjbDle+T6
J1z2p5mAQdcykrauMzsrDOzaeuXOhV9Z1tbdJfEkVpLWYamtTPBt6RID/+JTn1PRfwWR5ab8Qht2PqzNoHmykP
lX9BbOx8LrqJ95j2wDXA+3j/IyeSrrGDi+dFGtpHhGKZXfcLvuuVaPXI9tg0oRtByYNNNhJpoeXkzrxTcNNcrx
mcK+OsQfqMg35IUuWfoSkJyUFoAJIOqdK2mAC+iPCjVdoxKQRikXUpVGdGZZRo/qd/PMWrFKaQEdNA+h8000In
GqRo5889+87sWy0R3aRTxKR21rIpL79ILEqrTbMoP0vv/NzVkOrEVKqqETjVTwgXnlfqUb3spzye3YqLCBH2Vn
nQTiH7uKW3428A2DSp0pCYFYCp4/3fCYB4BMOoVNZDQYcJcygjRjzoo4hTDNidzaZrnHDLknPhnSK8vuLPOUhz
wnEFhx3vwJQCiA2rzA880SIAZHWwuBChicR7Y/tE4nq0za50LQqmji0p7EUE/+oAEIeGv2YIR2bf0t8scVl5u2
Q6I/Me0xcbCIH0oxgEg86Hchohx88HiBTfsj5M2rK+AYSTVUQe5PYPZqBsqUKUQzx1i3/gjMGkjqv4k5JmHqQe
Z0Pp7iD7/KNQV1tP9tWs/mxXZefqqzKpbkfgvQ+k172ygBX51eJnQH+yrWho4PNi1eTxJGqvoqzePZzoj+vhad
tO8OWfGEG5g4ai97pRyoqbujQNAsS/mkRWGrZ0Pf9NSbs1/TG7DQfCfNytF6vxk/lLZaakYNYOs+fwJpHYkhfg
xdq2eSWLIxPoa1VHAtwn7fg8EJ8y70vjKkGy3TixIJz77sQqARXK9BdrrhZfJ7iSBm6yh5o+sLd+4FB1EqRhm4
x11T5e5qMbtU+0WRetpRTDIIhUoTP9IlsUSAmhXfVuX7cnu2E2HHReTJa4Jqz1A9Z5G88ZMpVy0OFOB8I+QzBf
fGGKbYkYPV2xXcEOm3vDgjIaea72l0bXxOKxRZea0GM5OxJKuHGH4brWZ/kE1zTJLiT0BUTb8MIkpCHxNu3V4P
BkXZ+xyw8COQWNn+npFX2t3VifEmxNWn5pS3jlSPlPOtbaiNSSxVHyQQhKAI2QxCwQcj0WzsIGjPy2dXtj6koU
hrMXBKfT3ffzVCfxXW+INIQImevuVZoq55n2X6sPH/z8UIb6UiJEWnnwayaKFslrpsGdWyaG8lajyQ602L+E7c
oMIr4MyPfmAjHMPKl3i3aw74YlEV+m9cQINVM8dRSpKBNqsTn6VYH/HbHkG3iIBnihT320wlHrurWHfeLj1e6H
jR9MKvSd7QULzstAhWTSekxc/4zjXgFeSmO8UcfSDucwP3vNIeYjELkrORl7A7k0k5md+e5zPX1kZMuwrWx0jk
RqfW6CqSi5i7ftaDHHd1aX1PdVtS4F5ntP4Pz8SqhFeczTIZXnBLsgyBP8MXOxnomb0B5HWFZeEAI6PiMq9uCd
giYXPx1HsUWgIFZY8jZ2Ic8j5n3eJCT1gSDzu7VNs+9I4aTw3gBbgML+j6s30767JWowxPBfJMeV5YzzvAlrPo
hgu+AsyxlAtynl8tKuzeTUO/ACTXATOus09fy6jEcf3r9mOL0SXvAiCxTT76CaUHk0nkGxx8Xw3IfwCmsXn9Fv
psS6SlBGrfXA550ory+x57+wNqYi/AStwsfwYh21yVsVWUgP2ZyufzUtTiPCjG3oU70pKmoVP3c6MQgtEK3nJm
9TSTP9pyjQ8IeJEOm95jtg5yZflLbzNhuJN221rC9k2pnI+MzMquRlwV6J0T+6gtX+WtT/KWGM8Z2Wv+80Fqd0
XKs6aGG7GnevGsau36CQTmo9rDqYDGbIaeOLcTol+V+mjxx1JEQBHDf9V9aIIloeJ74qobSVkbZGBvhFSOtnF0
txGXckCcwKnxWkzddDbQUUR58O95erHaQpeF1MbUmwxjg8t/7o8JVSG47+vXD8Cp7+0yGOuZO300zd+AOeuWeX
+SF6k/2M8ru/yyaecAMjOxDUTEcvyiM4RQerL9C3OflcZB/3DpjOoVAlF9NG4LJR7lK75s4fqmG8PywTz7PZB5
VQ5ExekaOvPNcsovjLvBGbOKeG5FURoE3ssI+1m+8RCxfG/lLZX53WSJjRYXLJkcrswbXNkbnBCOtW9IquqRhl
I7BsVhrsldwgbGHxvAQ/o1gzhWSgku7BTFxKlG2gi8TIru1iA5HyG1B8naBmkYkkBWkKTWx3G9AGDtZjK8jPi1
Y+uEoIF9DP1gbF9yJ9W3TkdtptG/ghPoy/4DexlrkHfGuaYx/hyoJBOP2FTKlZc4Dafl1Tmd4B3MJkZJKIeXAV
KAbfE3+StfO5ftKCZfnmItW1Z8gXJ4bP3B0HOicBolB2Cgj7+4J2yeIkTK4kHsj4jNCftKRmUDbfioHcfaEuXJ
GpRz0tOya+YKXWq5W2jnpykvS+7XpTxyRATbfDlb/O/y0gGRwBcYDQqnR3TAK4ycCLEJ9zijuxcOW+9itC9Wai
JZoUt4WFCp0f/qkwyrMoAFRtmCx8haJiQHuxLQOZIP+t/oJpDlUTxxd3UfYxjln3HWWXGKwEwqkeZEXNepMKL/
y4/kLWYBWwHFfJOwR6+4qgYpw+IsKNwpwC9wlrtfck6slMVYpDssUOhw0hLQGSWXjwu2WwDdx6hOm/KvKRIPT3
l8reEgZWzG850WYLtfD6RFCAec+UxI9O78TZyLXbEmmOMxlKTOPHRXWsYfQlN88PA2i4AJ3ECMbnIQpN9fUFQ5
On//ewj/Y/BumnVO2wpieFZHDP0Pd+w8TiTBjXvXYcXXooVlg+aimqxDdz/bJHdeDF9Y/SM88LjOMC+eTLaHor
Mh3CsE8G61Fm55NANfifc6SXYb1x1mTfp55dm74nSYAZDs/FZl3RQkiuWRHn20gjQJwDS1EMAlQGWidI20PkjZ
btEN2TmGX+4NNTJnxZ43pQshpxatHvRib1Mn/9ecso0D8YTgz16mVwu7WBx40ORtK0XuIJIMLrGsm4YdkGQjtv
4Sw5lw0hC1aSVrIvdT69l5GMzb7dzSuhtyEU9Si1/W9Gw9Kf8g5NGzFiN8vidJXc8fnw1oJrpMVVbRx0pDbE3P
```

```
aMo6VDuHKKXi1JuWVAqlokCqk1PAy1QITdsNh3B++agXc6DFDAtN6BthXnN9IjI5y5pmDXmWpPAlPV3mnG1FKn
BXY7maWyYr+cxRrvqXSIFIfZK86SpVwP7KaKU0iD44g8B2tISIjvoZPoKpKFON/8lECWqwJOKFnrsI9sGQ7sOG
Atqe5dUsD+ja8vEa8EaX1PunulWAHg+1IFUtCIZYIeXd4aDYbQMk1YyCBDQrZDLwonzC4DA7vYHsatOq5tHSEf
8YpPA6s7PcX4GQNrXy4ZHVhpYbSTAzo8aG1HTHxCwDm5QYTH8KxQZmJkNVZCpXcrkpcmbIDpv9x73C1h4LQrM2
HB86Z6WhrE59Qxa21A74CM9EV41oeDzUjaVhFi2u5+imOjkdT7L5KGhsrBTH3OdhE8njt+8Bw32/Infq0ITKsF
wFgP0Rh/OSqrea+GGDioqqQhLsd5mCbOU7pIBlbIVU7Pufs7IOQpWP5VyF+k0cX42rGgSDOoPW1k+lDU3mdhl9
VYajtcW0oCPU0LoxytPOHL6K/m1X5NZBqXW7iK46+XebiEYp5IeRXxrdT+dPVEcsNJt+44KW+YYypifQHIQ/zE
hDi3Z+2ty9Yd16BEkYW3cDCmMLMLKMXkygL62mkbdSVHb5bFrgz7QKRUQRAjjtaayJcSfld164V1KAxteNPD/v
z3cIk1LMmLrGBHha+e0eX5GdBeqjp1EFOEpZ6XPVWjwDPaUWZ8tjG3m1Ocij5sGxbOKnX4w21MjtGLypbKWvMM
av9OxAi11pbVaWKGf0WnCfpnoNrGW3JMG1YpmYkuw/5cDsjyFoNRa2lZ2ksz2jrk1aBcyLZFwZjsjEU5sIorbL
aI2HuJQ1Kng1DrDG6HPDVkekHJdYn0RRGunu0usYGrh1q8o14D0rpXM3ee+goUeJxRv0TEwohircp4SVA3txZl
M8RiYYklMt6YddcZhL3VNipguc1VJWz51KFyyYw1bms5jnXfccZ7LZN0E5+4hJpLHPk+Uq4tdOfX8j6duuNn4s
BFzZPIdQUoN5klKnw4nozeCmcrg3qkv2RjbSnSPPSBpnm8SJWJ11RCacaP9Zzs3WFfTZ7963DP6Hwjkoq3pkvj
f/zv8fU7edoSD8sNXZnq9XFho+w7gu4asinukTzMLaEuuJo4kR6o6cwlyVvZR9mJjOCVV5S6g0OlnOUY9osANh
vHdAqEnfFtc/I//Y1ZwZ4P2wEmREl2G3nVxf6GaWuGibmhtnsixCUbWTP+n4kL9pQoMmxP5RoBuiZ//HwJsbZx
A3BXcOIzIpzhcsD2BvNVLx3ZOmAXV/8iS2tOISmjmJXWQogU3KNrj8rcmyFmf+oqT7SezcuPSeSfgYprsIe8rv
DQ3M2kEodSSb0LQvcPbaufSu6QzGIHRBIcQeo2ZwVZXYM1IwnUZ5G8AzB/GvO7+WYEJUzQrVKsOQilL6yknl3f
+5EWs81qEQcXWXE0rTmnyQ/uHMNoS+4N55XwrBaRirOcvaCPQklG9o9phkrPn2kOEg9U5009sKAJAeWSC9y9am
h4Wb4OJ/22BQDkVA5nDxF/Sgoi0Q8e8SzC+Q+5zeM5VKm6qfqFOKbKZOUfDqdA32Hqr2x8gJoUfh4n4o1462NE
BQ5itLJ0i6+5gIBhHiz1dNY76c3Et6O9EP8i9Euu2i66ELbw3oUlniAuRx6DHr0JLIjWUQMW7+nomArQje8/X/
nTplD9+k3NJwz02EYnoY6jqwskb3RgN/3obeu0finPHS9/StBPcC3EV3vNpfR3O6TQ3AniEooIGes/dtrkLfAu
7NiWLEz2UmfnxZGkZ7TCrx2Mx7X0wMx3uRB4EbjQKZhpcJx9nCz9ltFjQkEJ56fWD8/1g4JWZHBw1AjROT6VKi
woLH2sRk76iSRI4RrOm0KXOyDSwCEpvmszRNgeaTwPqMGTLhfu1kY36z4dJF/AhCo7B6tgadeotOp/8CcbVe6+
JH9/ClwiJDfGmrf/xUhYFFfG4hyGjDIXlngol9aABiqCgcPS5FStqpT2kr1qrVJouPXEmXE9I9gkmqqEcjbkRo
hZLnmQCZPanu1YDubjeBqModqNcFx8egbhS70yyntqAw31bnHElFItBr/sAS7QznF2yhmd7PVOpd8r3nIToDC1
+fmczjxZtGc+ATQ5bq7njhG9p6PL9gIMJulw+EwgvoUQzxIgtxCLjJodEYteQMEHXbCYZy7bgVxW+6ZmDt/4mm
DxsDwWD+KZiOyNaaW4+1ro0AQuq3o35cY436QifJpVu3SN4miyHVftdGrXFMpxmDiAW6uLG+0Pi0JaIYLCKL+b
Uas5ARCGb3rg8ftTG2ddcJWE7mfsTml/AZerz5KY1lqXKB5v5ARINIkiXAMiw/G9Cxf3rie1FwJQjHuLpFr991
zNwIpP4NCPKVW4jCZ5TcRr6AO0AMA5lxrLboLlq6W/xpo0lq8bGEPOzNt5bkDZDy1fT4CGkr+TpWEb1x6M2JDJ
EjvSLbaIaduLZDeRw+/lTOs9lHRCDwh2a0a/7G/wqL+7JYiTKF822JtdUYVI9kz/2ya4cPMDoCat1A6eMlzo3T
wF2wfMIMr5F3RsIj+CrEZPyzAmq9EsiE3uAMk1gXTMa0LidhIF/Sb6qYagpKudYjgeOhX+8Qhp47d1g5unUxOB
/un2F+nUKRwpiY4CiZ/hZD+IYule8SzvY/OwNRSqHQxOKYVXQpFs/wXObR33jfSMMa9O+XDIEef3WPejWqL+pP
gE4dzbS9FmHE230DUBzHTU2+iJSEyqNUxrDGBMfl7UYoIY4ZcM+NqK7a9Q/kMoRUFlfQ8Ej79+7XJdhhJixkUf
4mExbSprGJ/OuYzdnDs5aMShSc/rvKJXP2FXrRJzSVePTw/XKbLS5l947ZoSqSyY6Fg+EbPQpHGKclsUX0dDcF
SR9fnbOmF1v8ozeeIqxhqSa3zHUWQER3t8JCI3HlQHY5Ae09bT/AL0yhBcVaVAh6O3uKRpyR7TEiD9FtrKYSsN
c6spqwI5t9KmlmGVl+185vG0bB50fYQNEz5TOzt6A3vfcAzPtmn5ZkTaeLHI2sdINm5y1arec2a5R8vfQPSFSE
itjrjrEjyDLKacepCh8ukaPuhEI9+Rtmilin4LGkb0PYwDMore1AcrIzWSAakdg2j0uHNoihD04EXUXIAgGUdI
3DxW0vNGfWN1Z+pFc4b4GDrwJ+lHkgIJodDqquaJFH8MkyY/K4WTmUztc+vkMWWybn4tHiAnW457iD/CX5LU1D
VQI/nn4MKlhMeHFiBgR+IfwOqD1b0t77s7t0AX/AOJy3qCpiWKly9+D4GvW41JMPaH38ha7ZrKrWtz0Ym5TA1s
69OmzkD/LlXvUgGKtdhFA1BRqxLKs3Uz3XR6KfcuQjtRSBDf4/xUVH2+q2lz2aAjbdOpgfl+rsvDaT7lDR/Ez9
OikYkPYq0seC0Rr4cZtDxyeslCjvjY/M3g7dMOTd9utdIIF4b8yWBpy6hG46w5JaZhYNl+yJ7cXiKYb55JdPI6
IFAJeR+h14jnEtEBlYAkJHQ2iUymA1lTsQzjafWbB09Mb4TzANElKkMIcCN7D2/I2Y14OG5YxeBrPV5z3h0we4
/wzl9JoURPAf2XEQdpay3xiVKz3mOsMemHWnWkpOxC1svLXvbmoa2FYOFlGt8YnCQip8cjnprE0R/Z1ML0RczR
2/ZtXUb25cGLrnkg4YZueJNOKGyW5md8YVL8fYRruN/TctxScdr7kbT6MA1Sg4ZZCAEN0ERv8iilQwfHPE7OkR
UwteT7HNNXsxhmuin+Y41Lp7h57mVpIPabrQUfMUusoifZGOOIhNUQbYzjWsowApOnJ1hAvji2ent2+6Z91wGY
rZPGlMpukO6KNibSa+q92nwaJaqtibOgBcTyDevbYhIkZM9Jh/aCtsaxjGctbYJsJKWlqd9dhpjcxdcGZ1ZLQp
kASvuZCcXpNswp2vWaZpLVlIx69hiyH99mFW4BQAH56tKLGizhrP2objyUWEg7LzuBhU78pEHufFerpC06WePc
7yWJmgF/BWmMNrTsaqXBAdqgHdA5/uq2ZC86yy0SOgfPXOsf3q13vSS5AyUrAxuAtQeZLQMs9od8OXIXWXa6HC
SlEBTYry45Nuc1A8MZve6A0RBVkIJAlRASfOGPyiTXs6BXiGIEHMyKflB1pThGDKyDCQ3Bw5rqWejwA0SqhJeU
zqSK89uUI9FSwp1ShDAEQ2Wrunk50NPuWVERXzW6bmui/iHUVcJs6KMmll/EXfyr3elej6+jRzEAoHDPcJaenF
cZEDFb4x5DNtesi1UI4twUuLfeYOIeUWlpoMT4nAfpVGXeWd5GXgMXGGtqH0tGNWG83obHRa6ju+ySlkxg8YNd
```

```
N6GTBDXsDWwRxf0VVY+N1E8+bxFj1UUjEXXeaFudQ0jNnt+7F0tEMTSmJSsVkS/IBamieaNUfZMyuxZdnyk307
MtQLEns5m9VixNnLHbcXRrsaKuJYRHO3R5nbF/gFNnMTxl1HD+ywh5jaoBGoJsXeGGOU2+0citosig8kVtCr6U
kfeDb8zuk4/lirb5s8he635JloKSMqM2qIlQIdib27iqeMybCsh+5UuJG8tE51KOJU2IGM0NAZ7UPYeenrNlBY
OnvnTPgbxWcuLCNxAhaSFFJ/BTSuGcovkL1ow15J4qtfTG3rkNF/TPFhYHxN3m2qJcBB10yMV9el15OtH5Cz4S
2w3837bPjUfPw7YpYsmhUIHqDDVCtyT1zvJyrzoYvvK2Fl0W5JxZitqDCweANHIm6k1wrWBLXg0SYde+SqcMN4
lQ4er3Jd8FXH+5kYqYTvX2wRUNeUGmaLbOLoNW/rRWxvLrDbKP2aFB1LLRSRSLHcQX7Ccmvm5zMlJPgZiDqbtC
ylhwmDEH6MZ3CEag6C+aXiDdgo4OTYFRQ4R5h2pBBTjxYz0qY/JPYAkfDiJhf9UNNbExbPZkhDVli6xrfdDTaN
4dpx9tnirlW92vds9PiB4ciPfUQMH8GMr0LvVWzr20qYQdtWQSO57iTMfguTQYx59iL8EIDYed1TVxHvXsl9Ov
CQuKmP30kEMbdLShjoTB7ivk645K3GbNSKRGKhZ4SIc7JNjM/hTFsBe29VXyZQchQnms9J3+UG1TN6HROEr1/Q
82forLgePuhJXL9oriMkeqC4J/wBojxx8sYXA3YFkwqylndpv9GcfwR/L8vWqV8geMrJ4f+DtfgP3DNnSkPcmq
9r4eYEMftNb2/q9P6gbpdLPUYqAUqUklPgayCHF4ZMQfD5wLfMhBp/hMxTMtPpWWY/nQ88oIniIZMjrS8bTmsh
g2h7a3RIHTwLccIKVV/lk5Ejd1r2bCzE31dO2QPcZt40W3dAWDJRagcZ6C2GOyoADpZ/zYdUFEQQ5WpDSNvE2Z
0SdHifWV4Hp2Y/Osy2G9457kHVxuK4k38zbZN4kdDeD/3QeGIMAh/5xfXNNFMYh4elj+QtDiJw9buo/Z/LINhX
VaSWT47Tl/VQrjDsWjImHcsSfhgRqSt06GG7ltcO01BeM4+Ti8Addhn6LxQ0S2u63BShRxgDwQ58J5ImPVY3gX
UHHS9irSMmenbIPvV9jeAfHfzEYeCF+zv5mMd0XEHyAtNporMk7FkDcg9i1iFEf96cvsdLkMcK+Nmv9B9XXH2Y
XPeho2/h94R4RDVsHmaocsOCl/GG/kGMuUsysiYqKIvFwa1Zq9GdI6AtgW9iHb7N7LLwgK26fM24Mvt4Xsw3DO
X2CrzV2cU524OiNtKwgIGW+7tpUIc/V7G2ihifPGvpQRY216Dy3z6gF+cMnmloppvU5KDyn2d4a3Hg5bTj7BBf
GQU7nixR7EkTzeI4O6LpaiVt/zMSKjZGgbnyiuCSnAAuOzGQ/0smAIFlkm1JGbuL7N0sdqiq7mfUCbGQ1O1z2o
ghTF+Suf/KZjXTs+7TlQfqlgnVzJTY4KhyYDzAStFqE/SfaRo6WsWCuDUVyPDwf/u4VQOVYWq4c2eM8s9KaZL+
tDYE3H4x7rQ4SgvcVEtugp168a6nxZNynSKycz+rxOrhILt8NJvYn1v5cZwML4b+66f4hNKDgFpEn4dUkcZ+PL
7wTdWJhVV3iuzXeirPI5KhysGdeRkYGFCB8gmXhd30pO9fosaeoYJhhMujm34oCpLsoPLk2WhtbRr7HBPClsZc
aYzWFxbMiBsPDVI06V8deM4xnUVIL1GjinInOZJ377hTfb5tFEHFqr6HIubxazAVfqIOqZuoyNAAFTNuyw23fo
O3+DfAWhD1jnB1DERU0AvPSQX8ERclLztHDw+rqFx8EbJ284G+veCbGPSkyIMutSvbcHTZP0flBrGPjMeX+i8g
gn8LtrbHQiVr9nQKrjWk8LQDvVTicdlgDNpuK1Rj/WNfDXy2a7bfa7/Ht6m1YZN8cOSI4nWE8iVdWExNYKn5MV
DUtlNggoMludmsmOJICSuU0dALGDn1Otm9VYUSJJ2QFcnUD1/bI4PCWRuFpPDlXlTOIBeiW+Bvgscw4DBa96GB
fLEdlL9p2+LszZPkamP2I/Q3exc9NcPMndbRcphtTLeZlIz1NAwXeClWuMWvaFrKbf6rgoJ9w7mCllQ/N9Kt45
Khh0CUzCJDXYq25jks9FQE1tAH0DT8nX4bHhOFAW1xgDXc8Fd9PwZKjGT1/twW1s7SixrV502noL7AL2/jWbbO
WRqZqI3wiDcnF99OPVI0as7nvh/wd+xUqr1FWRbpR38XRuqc5rzUtXxFtzJQenQ2qfK0Rs585AY1linpE4w505
djnYTVxkHLmwPOFFRKMMBN/WCY4br2vPxJhtYj7iykl2A+CC8gH2CzzFM+x1As6gymgjswhSIzPbilCJWKFTYo
3It85YpH6npmMVGEr5gAxQ+14bAJTHQFu9TLrxfVSd+1YFGAs+lD7KH3qQxf7dfeqZUSFv6u093O/ZT0f40Bog
2ms3LiZGiu82MqJWW9TRzlqKARzW54T+9zeyj84vb5o66LTt6yACfUeltZKHSSX/fnQLQGSFEgpFgPIzOwU35C
a1UCSVZ6CRPm8iSSzBNqlt/p6DXrHdSkt3JgSUN4uO98UIjEgwjlwNUY9MgPJZIen0f1zmvpaHuVGwDGEKriPs
H8EVOfAj84RigbGrqtkqT0QZejl4B2CWjSb1A2t4aLljwkJyZBsKH/qvpQExOqMhdZbmia6xHMLNI8/B55cZY9
6jwtFtLsTyDSrTGg3qC+cAkcL7KfBdFTrJUiajohC0u1zkjdnEtzqfOYngCNBmMCQTSXG7aIaN5rbqnjgsSC03
TGUjgHT+nq2s2AmNZxD7JtNM510Bvhrlw/ALmB7I2W2yKxr+z6WotbarDy2OaSWLrTn93nungo4LadjOrk+s9a
XMdOojgkZ94g548OB2YrSAIkgwXNEhG3mV/gRktT1T4BQLjsicwL/z1EKEU7eTOgBCc0NN9CgDDEpeZpxusgUX
mh3YNR4vppgMJ9NmPz0tlIWkPpi7YoqMLOGEQPXzHUPWrShafE6JJL5gfki5d+updFYg14qxpY5v6v78anLf6g
tRQCLCvqtCsPvQ6m+6Nyubx0RxSLPHxjgi5+qH91tfWqu09S+zJI5FItlizMSG9tCcKrqOkjUDDiTf0vEv1o2v
EC+7C0v5ScKLmMuAMBUrDI72izoxmS8msrWEYZrmWWtxzSCeWI4gnaY/TtsjfCVj6ecMRiK6iDYKhbhcWfsMvu
+M1QiLkVLjA4uvJkYZ8BMFljaHCukeQTGFMREIbEm20USO26VJ2BrODwvsxlp2RTDDgJboummRa1Xj8w/c7O9K
MhUvV5KlrWlqsGRpl/QBLqXbdy+CmQS5z+wnPPWEFvudUJn7uZRQO+upGk3/gXBzcBZufmMOJl5qazaKfpZAli
mddRd9zj7B5CpS+VXWHUI+MWVaOYujI4D51Fa9wbLzHlol+6cmaGzo3U23guLn5lDiiiJr4m2B6jWdtCr19aQd
4Ud8DRgWyYZQLgvTgtUZVggR7gqjzUeii3FpjAh1dUJ3HTsz9Eli0bSgxFZopGZJImXhio3uTGnHhCvrprERke
Gz/QecNLx5SvNppXQ61JhwQlu0hbqDk+l40Dki6Sb6wBvXluPxi8nKjH6NFru9c2Btl1QQKfNpAhoC/ONNfyUL
ODJfFJNYndliXGeRNHJtPTPrfHE6R6uWtVETACURbtkpE9fVYpUTOCvk7+ybe2fgkI0yWwA/dBDtm2+EBczLm6
DLGLJYBcB5loOoKENhgLKOgKZ3/oBmaaUSb22jWiC6ov60iUHlgQbSe4+GKDHWtSk7luHJLPEpAyvrqYfZe4gG
IXRqv1ao32+cc4NMXmd91kZ919CsMCgstjK6E7QNnoHEugMJbRO4M+zKCUuO42TtFyL2CzJR9wne7sXj8/nNJ/
2PmeeFQ8QLBACdfgCQq/ItJPJuLh4YIGf6ASxLwBwZGhF4fNd/GhjQEOhtuNvJlD8HETIfUsuXZHBZ0zSn74Bb
kPjVQapugJ78eL3dTBxKx0/7PoqhnlZxB9ymv2fC+Cvnk6hhnR4EVkPN1+t5HPSaQmv0D4+JKX5XLHZfGDQM1z
/h7dqBy/AvHhPcxFmJ93fVP96LZRWVnQYDnHKTLR+uHz9DCsdbvAP4mm/g9l+derjho+HGUseR3wlPRhOlyjzh
346FNs+YHBNkp8GIckpu+8XKZrReXCh0ZkaknG1dA3n9GpdPvL+gT9SuHppyo4mwCdO5aDIPL3I73EfM4pl19p
```

```
uaC7JTfk1a3EGIVSPxboBpoH1Mng8fWlx3y0rCunvVarYqGTMBMZLtNXKI4xLWILGdxKWAo9kWOaPX/c7gQNzA
sZ8NYTqJ+Dp/rGHqVTKqKAkjSK1HCVsMl17y27stjwtqFWQgFn0IALHWrGMj6qSpFRtbtnVu+TKbPlumCf40at
fJPWGXU4K6UqJHpLkHe2ooN6K8T1ndWLxNCjfmc+ApI1rGbxPwBoZG8R7rJ0rcgcxCEqNAMhs4BHval3WDoJ0p
qUrNCu76WwAs1rMhebXBvmKEZ/9QCaVy5lwRFXv9z2QCMmV6Wf295lDUKJOSwB4ItvpHfaJrHvqeVbXaGZr07Q
P5S2mIZ1sXAubUvTTygL5v3rmViyHzssIgpMFRzyYsU6dnNM/g3axWTb2X6BUOqFMHMCQ1xmxziSLubBsAVOgQ
eUKzOS9vgufTqdeNXJyvvUziHlSsvHYBiN+Dz9r/7WGBKTesnX8zLdKsI7yxC6p3nzFBZQUtmZGTSq68po1EyV
GL1wDLN9yMN4Ye6n5zT8h4wQQCXWzHPvsRnDHLLDGfkGQ9Ki8VnqZKSMCMpOFCrOivxGU96nNeCs+OTHlGfFdB
lMJre/yH31di+uMrI6nH0D8EKjSJlxpoyRzDEaIt0AtVZFmV6sRZoGAU8OyvlBtzfRCSwdBqbBav6miQISaFTs
msJhXJwbDlNVPlpPq/4NyguY85FKbwAW8k0H35CCn93Rl0kNV6muVxDWATXn/WwjUqxHHJ2C20sP++5ZxEdvAu
wyv0BR2vGQC8VEqmR/MdAEvKNcdU3U5fUZaLpRJqBpm/gHlhio0Gp1MTWE4GwM9NEJz1hySlNiNTqaLBuVC+YD
2hhutzHc1oZ3fWvP+4V2d+V2xpWdmsY04N9o/N6KvGbHh9+1RfdR8mu7LxAmvaLLlPSS+xCJUWE8420RMcKry0
jD1zdosPg2RkZzG5RfUwCO+rs53XndphnLsXJTR8lz0gfpAprHL8WOYMUKKfihi18VMp0p3TdKrMNgFUJPTKgY
zxbxbv3UD2qWrdjbyfwBVW5BxnLuV55pYP1t2zdy1IUj48xQ6ElHb+z5R8aPVAeLKwuGBuOJh0KcqpFn5mWzAH
Qa4eW3inmkzKq8xeygAVzwBxXwHUJXBHTMTy7qgcyTgo1Uf47lj2BbUL+LK7LIvAAu5M1sYyWm/eTHcL/XzKTa
HsvNqaZfWCsLsoO3DgxSeroiRtepMWfvFYRmtxyahd/XCIz/81ckbiz/UFMd+mLIopJ3ZB3lcs+yX4prjpsrj1
/cUt9BDlcEjTnURiBu/QUXR5QUuOykLLJRvN2wVJM6yl0QhZnqf/bLqp/TmbpXtc9a9Xf8ALFHg41mow198cZE
kAmvYUyNUy31zBcNSGtNuHdEmrDAcUXLhIRO9fORjN5f2fL6KU/bCivJ4+PLQRmArgkUP7RKWYt4sS1TrnYXwA
XcgCGxwTclg19iiIQDvY+t9M1CVURc9hwj1VBhvDmdmoevcWDr6FbH+GZ3/UYPnX55dDGeGndKE1ZBSD5w7hNZ
coqRgxHtux3TzNNuBcT+SowPCVqMf02YjDwKugYxk2hQM1RkZI6PMn8171PTgEgPEx47vysbSiSWrVmU8kqAPF
0vM106ONaq4aCYH0tkNz4odduLQn4DYDvVuIKZrVtD1v/Vgb/DMGEO3l3rJLHUEXPbxdpVzH8XBpKy6VAx2QZQ
SwF1uT17R4qgC93a2aCc9gQMT35fSfuMQHhXji5mqI9lJBtIRCCi2qdGWTAGRqvwCbFmFvTzDTOu3OtpuLhI9y
R0Qugmu093D49UynjRWOchUJ5mY2c3DFF7fU9nix6C2adqDjy6cNuX44Lichgx5y5PpMJnQM5L8wTMTkoEz4TJ
r7fZNcumJatOMlaGXw1us428MiH4nBHsC6nojh7EMEyIjv1h8v2XHBtxCS04cLZzPbGT2CNM3jtRrn9Pgb1QXu
HgFPEh2J59fPbbaY4leGfijmVcZ6bxSIPUxncv/Z8CfxQGeoDnKRyj+EGvqmzF/cTleLo7t0NIvnj4yKI489Bv
JwXQI5vqNjmxnZb7ZzTdiUMgKxj7Njs2Nix7aRj27Zt27bdsZOObdu23dH867zCuZs19QB1WVXfTSm+SrjFsD3
s7W9bHbp+4DvTdJh+Ohtvo7uy7V77dqydcWRICcOLLz1pbvquYECya2TJbNawpzsh1TmQiLIFphjpl5dSg3j+M
aadIUabUkLWYitp1AU9q+CCFaEo5tWOVnTjjk5JN8nje2DMnzFMk5EM28yR+/wmiDVLS+r5VZWWtJURZXkMa/l
uhavnQgbmmDTbBZt52zbaW+yN8TDOvQNYBH7MkA7ZPw8IHpSRMZhHZohlLLt57K47JmcGd7e1Q0ZK7XK0SUP4t
LdKF/shXc/gK3HsKEdZQ2lgH6J/9EmnkfOUztsAZGOrVLlhn34/UuEyVSfOT67xoXreNWRU+B/1IYSYbwOIfLS
lsOp3YAcyxlznyACUCn26470wvjuvQV/AnXnVvD/JukjbExhvHtjwMIgoAlCoZUvlI5SWwUtKnYlnk5nrFOZmB
YXtwgRxRqhQTv8oi44cjxiu0tJNkQreumFVgtCA2WU61blamQ+dLf9m4EahKLY+Lw9yIBOzuKBG4UsktAjDxjn
F6tcrbmFpziqL0tfCgAqPQSh8e6KqTb78SC2+MbjEC61wBWXtaUsFl+Hxv6Ip/G8XZG89QkcLRt51bGeFuJV+l
R6/vpkQvDg3JHOakcO+MwOecUwSzlWEEDEb6UeNGxrPbyhRqnPua/zY0716emwynGocgoopzska04YhjBWTMud
gerP7oavOEKBtQbytBWuXLGacBmLaust63cgvrnRidkXGDZiWke4++AwMU8EGw88F32YIhsOMW/vqb64arKLFP
BvJUVWoL4W2Y5OZK4FNLvVLG2sqbqIONK2dOBF6qsQD9Hc4bNjoPZP8aanrb3VR/Dxf3cEg45VNA3TRC5s7EZw
pKRyTgnAR+/7CvRURz99PvaYfe+1Gx7jy0hTjdgIwOP8Rb17UfGy7i5JAsPX8Xn4F6HHP7XO/b9z1Qcy+JGgzn
jpo9JiJh0HzK3CX73zEJZz7oSZWtzn8ZADRahylM7gCMXL80enRJtcQbgUho1ywAbcnkzu8yTehOfBHKCKgD9W
qLVk4BqMUxfa9ZbtGzYAZdRkWwFjNQ1J9UP8Azbtx5Lz6B0MPlpo86FRNRZIH2m8O3y/4rq/ZcMA9f/LcIeYc8
v3oLNWS1U0AtdbyqbkzuWCZNMrATrZqEZ1pbxI7nshUnotMwzJDV+qi6UTAacnz1sCduIZzBNo8ISVPHqfhShi
QmOzdpy1qPQqcweHRbLZMm8MBazeVPSAgA1FWp0QVSojJK3/jlwrrgNAidwk/TQuF9B962mjTapFY2XBjk9os5
T3VtkOgEr+xbgN6AcihkhFQE7bgMF00BCODcM5V3wYGcOGlx45x+VJnYZW+657aNi72lgU8Ofkyp05ujf4t8OV
TGgR7Gyleb60iAzEwa6oDeh1TH7ICTZnmjcr+GFAbqSKU/WawgfrT7I4wiM38vt4POZXLr1B/3dCrXR8g9Wgzl
SvebUGUW6dcSQc9q/H62WkCPjjmppQASGubyFIIVfKhv1Ts16cmTDKISSvaY3Ea3QZ58Na9ZyhG8tfM3WkYOGU
l/KhXg75De1aebbv9RfP17SmLXcZNOv+tsplJYwpfMOziF+Krd1TZxIuUftGcuMDiBLsjM8mR15Yedh70ChhXq
1pJ4K3ynaJ7wEblrghovUM3LOySCd8p42S/ls5HMUW/Q8qwWhtzIWkNFajtk1HGi+iEyu2BeeMyrH0ErHUQr1K
SIcMcYv24Fa9mRCrt7GujI3fkWBPX/PIE0blY8uhylp+D0AZV9mnQYbKPzDiTUaItOJQnPiXHiqP0fbu2w6Jxo
JETCNPGHvI26kLQxK6GBT/+EpU2CarkCieMTsyswjNbeTw89ffdKeTIkC8voWGKvO6UfZeR3MFnnlfakcSorhW
jfP5VGHYSxu42/du6CI7vv9pG5roscn7gE3XG0vnnOM4u+m99e0SNPlXSkYhpNISxoTJVrIU1ZFTPQtJYX8aF1
7fEaWug1kXBy+K1R79Y0yAtWoHrtxdvsPDHfga14fz8jb8Aa+RP6u/cA5lcfR4uQQ20D+bAymN0Zk7RroYA75e
yoDkUEO1A2Ninhr2pwCrLJJ8k7+lqZSZGulaJtP0ODrm+D1dPSFuW97qeEsoikbwPiYfoGhN24vrrNI2yvHOlK
zEoNaVm+GVzaMSH/czzvMWCeSQ71aPrZ34f0jCbAZxBm4uRLsebE5nyn8w65XTSPTGfogw39GA7MlW9+prtHkR
```

```
NvTbQhjpDrQUIEMI/UMdHh+8nK5oE8lshlap9lOuy/XKmPbxWbeBw5Ht4H1hoU4cfyeT82BM0N1xwQ+0ymqkDJ
OFYyu3xTj0A/AxQ9nac3kKktdhAj7HtIHq94xBmhb2k4V9SP5kfOgZRzkcou6Zf1cmgilYYxlRIbEoX5sk4prS
rUX0qnYfMLKlfkcZjtkJCoSNdn+IeKvBkr4wgNz+F2DsRkCtmEvIjknAF72GkPdFE+ziynUiQDOvq20mW9c94z
HeQMjojaIlCTa+NDmwLTZnOM4WtVGoLVWGIeMB74XKEvELGrg0LC16F4LtF5nxW2KOW5T3oJjNrzSoSANG2Sks
CjbRinh1Wr6c6Xi0nFLy4EPTf4FQqQAXNASEXoC8FnAmwH4jDux4k1IwXxXk9p/3svY3NETp0ABVaVh2GZroBw
ZFbX8iJdaSBw7r7nExxf4WbTl6wTuk61T9y0Hz9VNV7RruIOPcj1COYZQ6Wn5vnTvQhsjemDcRe/36we6QkUX/
vjZ9FWtUFOFlCZ4cKlUKH9V9LEDrfKaDt9obpsLmHBbctceaOKnXi3PCsKGRO2Ue9kAbhRmIyB4huaDzHNN6RG
MxhTs6lHGngmcwnwnl83oUv3b4eQZRLb85tsiZPsxuqmJVsZ2zqGvFF8lVEx0IfMiParg7JGLs8EuHIk2B5k2u
ycR8CvDXChwNOKnY+XNYpo5eMVsfrofFteee/rjIPX3k8ss8n0jHXx3j+AVxoMw415+J3vvDpE4mAA59B5ELlw
y8B6LRS7Cke3uavfqLerXfuKchv6DgHYXG1n8k8bPtR8x7YoP8eFYpKB5trg25fw3CFYT9jeMxOzPkTlLhCmYz
xuQ1GLtm4yB4dlEzeVai0pHkdzed5o5T/hjeLGM7yte6Z3KlZC9QGwCy+IuO93FZg+8arayIrnHwiMC4y5NODO
m323M8jH7juj1LwvgV8cPFBwwK+bkzwMd96p8HSXQfbPTUU29rFfaJnIIPQnu7wbX93pKDkcrj9guJnvxZSVnm
F6dkTv3z5iSRghha20CuDk+gSB/n7923/lImTWw+2XBhVG/zE4ed7estYFMDS/BbRdUE9yKN0MDmszJmTUL2GO
/YgF+Oidro+lUCbc9x2W7c3eAqDZi7gOKhYj9lcYdgOuH8zbGzufzunuOjcEdnfZVgDH8m4bC5+l/Wo+HzKeae
2sWg/Y7oRy5ftFgQVoiV/Iu8fTV2gIl3rZN7G6L+F2hcltlIPMVDckMb096YNe8yKUAs14eGs2bYtUhtPPUrTI
eCnypGSFrWxYaldeoivL/kRp4TnaUgewJW5KJ4l6mWZGr1NPIsLjjnw1b2ZqpEj1DKros4AVUvKw4lFzlqWr0S
HcLgM91M5ZkTVBUEfEgulaEWWhIkDPsVTHUviVQXp4JxsilxUb1+Lip72Led6czBvP1qUMDMY5QbaSSrKjfwgI
F9jVBXySop1D4mU6MbPqcC9phk1mJv6jeuhwFIRiMtzHVUbXjKzNYcP5o4MJ2LU5ma0mt4R6dp/tEZkHHq3hph
F943wubIepD6FN/rDJ9PZ8QsuUPI0BxKyv6rOzAG1PEimTkBu2bDzHdeDwW3hUA0GyRMEibzyOxKS6ijkFbB7u
QB5a9X/OPB4JCka+Kj8NLbwhEcys1ZvTgshj69wyho7hPJ6PjOW5uKLfztkZJ1T1lchhMt84+SznaLZYYwMddV
W2kxzFG4SKwdtnEzyJab7+dQDr7CSsNyRDPqwHX9MbrH3nuUkUnjcl3YQqYNTgVal5+WWR/FdFcOcK9ku3XP6h
0h5/qFBgXJj0F2Nlu2R0M54sd/o3R8paFjbARuNiZCjFsh9l86qu3TauaHdsbapy8Yi071HhXP7aOh0qYqCJ1o
EXG5StxfClzJBVe8B34wyH1b3Rjph/NTOE9GVnDeq/sAn/BMwYrr707RlUKMpLzdxNwDwiupzTOrX09+mjrHlj
Q58v9R683eEJwtJUJp+/3s076leQ+TrA3Vqym0hwggd5yZDpTSMg787WPcvJHIMTEWb7QusG0DZ5vkmlsQXjq2
Dh1bSsPlUisaru99aaWwIMeq8oFRJWRV7XlXWtukxjsbkAysnJH81cOZMKP9FbtI+P+NCMOsdZ1qsSjUEQ+3NT
e1YPBWLNt0q/SqylC6FTG6LasbmoCCgkp6UOmrd6PTGBT2zrAriMQOxtWXqsUCtx09uoF0ve6PiV5dOww5ueax
BQsvqcOron7I0dWSg9WsgEBHqk+gZPo+WjQ3ma/Pxmu2vnRmjkK/yZjdtnlSijwy6XZkUkinScAnctXXDdfLZ+
uXB8Zu3CBRGHbYzcTdppQVI6RrqN0JeMNsHIQmRfBB07nMByI4eUcfljwzOX21j28UnkMFPs/f/IP8E3T974E4
BmYT1ApG+crVXV/ZwyED0OMmZxAnN/BTkoNxLc4E4aNrZ+IraqYJBnSSNIMwWNQCiza0FRCY5FW7gDFimyZvmi
NJkUMEkYv4j0mQX/PUR80jkwlYyEw44W08euMOPMNJBmWJoVeFxfEmwdW4xIjHr1hNSbn6FC8i81AT5+0geORN
YEK3ABwovX4LegneUY4FgMEH4wZX26p/kgDm0LeMkJzfiSoglxU36Fy1uxRKzN+f2pYyyqNpV4sEhC1/oZoZww
2vWgUnhcOztzWqdDJGQWOOu36ZmeS3d7m7JGFL+K/D+uuhAdt6gye9qR57SmxlGtvgFWEq0F3htZCXFZHdQ/n7
1Wtj/B8RMONOWK/vXEtmTQwYOG2z4sdGzzbLW7GrBq65YcquO3ZnSrTJ0g1O740qA5fL9bFnc5LKqJz77yBFk6
xyP8ceKiPv0CLfy6NpW4Hz+JA134hIxl0/m5a+zFNiQVA0dMRDYmk8jMIYxE+QiCpbi9e3p1cYMcfXdL41ZMKH
OgJAtNHTCLnj9YQiXxKXFlWxrVSmo4LJuksKFS3N/6v3ciJJIf7yB7jEGtjJFSV8sXAEs2DBI+5zl5QdXPrT3z
bm6kFCmxVx9sMT6q7e84baGtXzumtUU7d8KbBdPRLjwHD64NjiRM6nIoCtDAzozJmV4V+EzAcxafOUfoa9q+ai
pdfTxs7G5hlzQTA3SSrCGQhP+auRhxi/q0iUQ0eCnPwcLI+KDf3PUvfB7Magfcj864IbvlyIdwhJyF59ihLCEJ
zn7qxWjiPkQ6RNR0HHNeU5SMVgBK6Jkc8Vra2N2dnSSIfmFdlROwTElOlaQtUIiat7uKsuu5PkSVYjoqv9g3Ak
ikz9rT2/KQovrQpan83LhDN0oGC5KodpDycaApvzzQRDbPx9b7R03ZV2Cm0LLsF/sPjultJmtsCL7xaX2VOoSf
wO6etu0RrThFU8Avne91oltHS26zHsSCFQaIqRZSfswxTC2UlGdXK3Wu4VU/G/J97cqlTi4+Vw8U5JYPha8Zi6
b+k4wGH2kOuBT9vlxCWWxSUQUsmeSMH1P7rDG19zAcP2+FzQJxjGhpC8CaYYoRf5D8/XHFN1RmjqBCpqfF9dB9
ucDTozCYnz0qfcxMTB5MMb9BpWACInuxek7x1WztAjmBCKjJz9aGIYe6ZZvyN3QDCzscQuwkjrfv9y1qarYtmd
/Xqv1EMrYslru3bCl+icKv3klsnn2Mk3Z7uBkxSVfP6+uvkgyJ9RfZD83Ki3S5haHiibMrUvBD6iIycLHn9FfS
SlKb0GGW/9gzVrskJxqc618zF6NPI2B91QtzuCz5o+1wBPyM6d5y82DLI50d3zDweR0ig7oey6bJolwLIRl1DT
XZ0sm0/gJODH+VZIcsz6b0Wx0d/EW1n9evL7UNYUgTJHFwaE27zy/CoLIUxWAsLsasWtIosAzfCsUy5vLw19h8
3R/KrTH1d/dBg7aZKVDxoopfGLDpdgJs/27Fs0myCbWRq6spUxJf+mbLTAs2F/hG+dq1/Q885DVupB128ppXJR
53lPSNxji45v0rwwRp9K1t5kQntLOH9Yyatz6B1Z/pQqYulu/6o2ORc7zRrU3Auuq5FqOk/yOqQf3dJzVI0nUv
DSS7Y5L/RBOIWVQEUsDwLLQZHOAR8a1EZzCThfDDjCCQBuYzpHDBESJfymYOhwtVYdaQ5FlL2c0DhKyde+fgIm
jZ7hKX1DsLyC8ryo1C5aPHzrHxzYy0ZVsLPdUttRae4vTKcQ2z2J5TALIm2ZMKjap06U9JwfNli6a7KNPd/Wwe
JULs7eZby+EHCXV1GlMCLvA4LH0ZCtRZC5IMD2J/C1MoKgE4uOXY4mT3MrIRGQGvkzwPwpUcY/HMOUawX98cLI
4km8cNBU7EQI5CsczpNE7AkD9TSujoje/kowtFiCTX93JbN9cHjWJnm56TJu27NMFShb4RiIZxm6On4prfbr0V
```

```
NI/9wULyp5wLFYIiupVMXpjtv6OxM/Ro1I1MxcdlVzFV49L2UhxW1+bulk9cDw+Tu5CWys7kNfXWHewpTjoJO+
/rxl2rXJ6/dszlNKT5jyFjqQJGSZKoU13JrzeuyWB/zW9wJTPvdfgl+sdJhOzKHjXNlXbBXSCOWzq/Dfnyt5ef
6jkj7tLVZQ6ExDGYbhCUFBEypgP8v2lgKocbTBrqHz9dFOQYA76qsH+pWwbLkKCQcfWFA3J28UvbL3aVulBCUH
dSRmTHFm8eRRn+F5ozp1i3ztEiHj3pOdpUq59wd8J5msgF9bEf116ivoTWmMVOQI5geZtXuzPspBlHDd9XYJDY
v6HqKJyz8OJtV7Dqn6Z6xuXa6YxS+QWtrdibeoecoxX74hdA1H73Kn8GIP2ipkZCxkaNjegNWI4XijST0/fbgt
LWXmMJy5Nf4jbTYT7U8Ii1GftDjhhj4mAWbuRlKasoD1ED8fv5UHEfpO+CREqL/qjhB1PMk13NoLdhsg7MIxBE
8f+74g/9hAO6zG5Nnlx65lDr06c5lwFQDHx/QVgUt56vt9DEjtSlU3/HZtc5gK4mG1HWtwlE2fc08sJCtLFU9u
X8ZIYcvpaxp2DXJPOq05ow5erw+4fE9ifaEV43r/6A58M3x+E7JbJVHW0e4xI3Hk9mPQwlASgVRQLvneQiFGwt
PyhKm50CF0a0t40nA/WJX/N9OX0tZ7SkqgCYTY196o9QvnpjgXA95Pq0S5OlgJEB0J2WAy9pIPkaqI7uzRJEaU
/iKuJ7IMc1NNJEFfg4iReDl1PxcM2RCMsIDmCDT9jmlg/UEkeWPBHtk2Nyxo6kaVTlY0ifZSkQUm1FleTkD2pm
szaEYZXeVXatcu90PEkT6HVbxuVX5iqQ1fK8ErKZa7FsKNES7pP52NGb3K0ZsKDdNbflv2jCM58Ye6P0M+YJGe
DJ4VGI1MxzFt2RUY65YvEFNjVhD6aWWJtSScFKM3IrxuzwcI961tde4QU0DXmi2hc2u2k+ea0iB5zxpDtO8voa
WVtLoUadG7FebRHma+vjGCQakVPw/RN71bso5swcF4WUEHxj5WxV4G2tovt5RTwyhOY0vwmB6+1YokAsXA/wxe
IAYb+BXsIJtTLEt23cVyiBTwAGGmSW2GVMAiq0sBGHk7ru9pfkea1TC/r1hOK5od4udS2+hshiE6CvFGH7D+Oa
XJShBWO2VbB67MdvX0mmsAS3FezWnYDd1YZwMKc6ai+ln4L9b3LQ3UV0q3EDuuyXDfIezF63ZR/xaEYhptrPDB
tmysCFiLTbPQaP9s089pclRxaecmvS9d9KGu2UbQ1lYn1xPJmnYO3H1SQmNNI77i9oHk1eeQRlA7m41LWIQo9R
+uBj9RFGft3OVX4022POPZ8ree1/c3bcW7CSsz8QIVtUQSlWXOOhQbGnPx+WnRKruvxwG5YiHY88H3nlwi83Gs
qep86I1VBNhnPYvMC0mo9DcPgcZI/k6ehIsTG3msKczlsqk+jM9lpmcDJNNhC+XQ/VIFEqAN2tt404Y8/2+IxC
LeOXg9YHG3zoam9oTyrmyTU7UxybplcdfkCvGY6CmdIpdM8M12t34sRCDXwUhM2q7dEaimPHoB99WXjb/V4y3J
CnBllUv2JIY8jytD/WFs0KDLLE6QELll84gp2ae1BsH5f4hDXnKlvIVE+ordukeyfrm8N4hlomKmTmCVI81Zvu
FEF/1mpKyZ+XLFlUpm47iFIBLqOaup01i6ZEGd3qGsb2JSqHP7hp6ohLS9wSR5DLkl6UhpOMHRfXc70h9g4HeU
NcB0NliNQZSGslKKC+mQElJGZnYdv0o/OQVYaV3lufk6triP3k3WnroTknO4w+gyPSiZuiFigR+iP4zUJtkoHF
6aYYDZNe+47vL42kqwJ7cdLS84rjqSpg49NNbyfdsPxAV6E1brZ2BWNV+vcnhOn4DGOKVeZ+XHl+ifAZJPkFmz
GJ/EzvWJyd05eVTrrNFUBEEEGU5p89XqsR0bov/CvVXV6r92KvggqIN/U7hJR+9iXUvr7jGIucHMDkkUhmDZKX
eytxbT/ZGAdvZJpfkKBxWuIn58zxbBGMVHAjnkqhmxG87NpQjExdIGsRrvddfbEwavEzTahk6PyN+HLCOHn3Te
HP4mYtfomeawq/n7WX3c5AEf5edNxtl5CelD3UqnOqAcxE0S2Bx0I+qbAhvo62AOYlqWAcI7ayuyZBY72QPZLE
l6ano9+nBP40lYyiE1dPTdhUDTJo5MKWB/XELH+nYBoNTRCasoWYPFCeYloRRj4pNhLKUSGkIfuQi9H0QYHqhn
FAuiIx26QjKeDZ/TxmsVQLq4unTtmQL7+vSWSgKBj6Z4DC+K03RN/CjTNu2viSeossJPEwpRraXkO8k/2OWNuQ
cZIwKMi2vqc3T5fMgN/exfG2Pa1/TlCcJy9ZA5k0BAatdc4xPyewGsIFJOZ9cjqqsbd7P8gqMDOrtyFnxXm8Pf
vjszKU9DcY4Q5QazCwaZXZRwgPPtTCXuF2+/hzVOdFVLfACoMQkzcgIer/TxN5O2K2uQY4IcCb0M6KzHf5etmW
oYVGS51/K7zbVC9MKE+a9c9WrtZh23yq0OUGcTZAmoyUvrQ4sEW1ptmfdDPnYDGGvU8Tzw7ssUdLjSighHu656
lcfnh2RAv151m7L7na7gbyxEitJSVm9MR6h+jOAnc38UnWXuiyvCjv2EAM5eAuW+DYgbmXZUAeHnFbSExQfFkW
aUIOT+dL/em47sLCm7M65s5dxkbpzcIf2eXcfyJ/gDGibT7Sxotp4PcQQxvJVKchrjctRoL4UWfetLUioCykzF
zgsMIwUUXts9s3ysu160Tg3lFcy2DPZ8cuydpPrfVQdBOLW08u1ExLarVVNJhoux/E9CWa5Fe2LvgEIzRC9sE2
LghcCzxF9xjq/dSgsnr8xnz5T8kUT11/WH4gw/ptrFSc1aos/HHuwNQt8wL7SvbdBON5w/pwC7oyN9gTAmlSv7
bJQ84o/ZqL7vV/pGRoo1V0wuQv09tROF0GCV9WT4g33mwI+j9nT2UyarM+/AxSNaIz6KzSv6j6rKDOBpQV/V0W
plpo8fbMDW+toBWH0iwl6kYE+5jKzn++GCXa1Ah4CTtHXdPCLLeGdiNpJm03NPqHD+PhG4pW7TiayHJ8Y8Lt8A
fVqldNqVwHqKnBa4d++WdBEWf2FG+r4gAbmJpfi8RhaEDd2n+PFp3mf0xL8VdHq+XFUyZq1daX9Amzev1rrdX3
vnv3MDcYMMnoyesLWC4Vv2csD8KcO9qGKzQVGAyL5H29V12/pc1EERYbJemq0R4euBTYGzwPDTngEQX4VXAlty
4hc7fOaIPr1JbK0Nn2WdUtADM6cqx6+1g7ldJxdZ+eOSKG3JGK+3W0q23Ct/IAvW/lbcPXRYD7XbnApPPRZMGK
50nqjTc6+8KZBQsl/HNlxCe/cc7nlSuXYWhJM+/uDWcBgubXKWQ5O0BGleORBWoGVxHX8kjfsJSAlQlwQzGlkE
TOGKo/+1T0zJcaVYDSCkccX7l73eYGiKaMKNW/Z1QGjpT/38SJvA3FMGQ8UO51E9IKj2Sgxa0PRRQiKG+NnNdI
ymH9gOG9UGwsDymtqjjXWeq1Eq2p4KizaO7vOmm1NK2nhwzouoclcs/K7zkuKfP7j2c2U5iAkxQxN59anAWH8Z
bJdyLVzweuQpyfa1zpPFG1RkbJyGjYdQDDv9wmXdcVA1AOlt5VRVvLXF9qJv+wK7ogruXPDGiSC7mXT7ThCt3/
9ZPO4Ru8qUvQsY9rMJW3b38Ch/F7O2EED1K/HIbSYu4RPnn8YQwwVcv0eCSSb9CjV0utQWKJpMEzxAZbFQ77v7
E0PosZwDRuxu81ZDg42aWOJw3BfUmTIb2J3itvbaEoeCfFrbUTWaZ7D8mVaz2M0rEXqOkYp3t8T+4uFqkn2mLL
WGk5/N3VnlO1NGgwBdoBIQAiqLeAtSX4cG9FRHHwpeBkzQRZgQ1wVPAZ4ILNDsP3TcVMr8er2dfNVBOpc1xXwc
hEJd7wn9bUD0CLpAkGJmAIphXsNoboGyjxFC6Ndn2Cn/0piNnt/+W5sYolgWsWTyxIhfE/wjrT+QhQiDtRBkW+
ziFLR7Ca0XnRrKApoSmMjy932WRmACm4Ct4ocGfSZrWyBDaDIsANpnKTLiw1AmQwGg9xE+8aVFBE2Rpr71IRuE
rbLyP381vCfScvOb0laP/7INX+449nCL0IEvelrtJ8M+RftDGuMis95yL3Kj1+tk3WwiaxJZMiQpMk6DCq3Hu6
JCA5+nGo8YJv6cKPNUkgEc/CJV8ZU9nNzSuLqexLDPMe2y5roufDskz3hmpRTKrHZUAzAfhvTfgkljHz7z4ksp
```

```
yrN8ExjyFCFS2KoCeSLfYi7zO2Sh8Mq0udVzF/UoM6xbjSukHU5RUVWvh7O8tM1kQf6cMmdyFlNCclhN3WX5v9
S6bhapjL3dG5gsLqpRiDXY+OF/GXsENxfkPO/GPfcWsnp1hH5P0ZXDmBGZH/4ZMwrxPA57dDdZbmYquVLK8N5h
vItQCk4cXkh1mmnRMqY8nmfS0cCyQEt3csvpvPuP28yXqSzUQlR4ekJtzq5M2+7I5NJxGSorw6QgK0ltBa8UbR
zda+RzliQT60dPHIuVXm32s1oWWOglnymkikHnd3KnMI/K5jSoLQX5BLGsgVqNgcnmnIxwxgfLJuFvYPbiXKFA
YkWozpbDSSqmx8e8lGsvjweiCR8nrFpbxv1SKNqda/ZiJCi3tZFa/nLU4+ZHvDwDuxgjhFPB2G7kKHVmZVj4Jf
nLxkykOI1LXKq4O2kZlY8vtNYFqUp+fVWYsXq0SYOofAdWIPqgEYhAVTxop16UQoaF06va64tgzkc9PxPipLpU
kPtZGXctdZbQFWTapaiN78x9pVEa4H8vldTq2nvJqsLD/6PT4yFoVrHYUJjqxM6VSTdU81LvneIcxrf1jHR6sR
m4DtqCa7Mi544bUUwi3yrBOfVR50Na2I92OrvGC4S6uFmb9S5/QF+gMOfpoT7Qs580c2zUVS7YtH2HpjMT9NiC
F+6bLfLc4Ub5QUmrZFBtkmuOyrufTcKS5qwYDjZ8t981O6rDLnLRclmj1o0fxe23o4I2OjpOfS+GpTorqRSeOG
k634xpVqwzoQtylORCpWfYCY6UBDtl0yeIvT0p4OOYOb2UpCbJNoW/OAIdCm7zzrmqakM8Xxwow1R2rxZqa9OS
MB+l/TL7qqBZp7zBJMGa9p2wGM/zN0FDnC4KG0z/NrLK1tJjCSTz2J3ZrqkRL9YWOmlkaCmvfDDNp0zpXZdGrw
1/ciNY7+t44v2Z0c6VdjCpkAkTQxhiRtx0y8S6Iz/Q7owc10UxGaHK4YnAjssCKnYbU7So3Z8DPGlKXKOrMOGK
hg0Ixvbsq4M2nUghxXiDa9CmQqqv5fvtYvbM4NRwA/AeXWQdEFKd4215nYmaOnyUM4CqGPlGUzM0rB4kyrKIJM
PwhTJ61+x/wBESAuj0FNQlWLwYFOnGGxsf1E13M5bHrjiAvQZAq2ClyoXcxvMIYwd56IM+j8WxOI9IfgwUKVH6
DU0y+VvKb43FPMxZxVz0x0B1Jr54TJQqL8KSQEIMG3OvzEneFabnVDqgNrNEqkDvGpuNBxChWww98c5YOrk3DX
lAjdytZ9GNgHOEV5KmAWpTRoyE6yRZ/jqy+eH8k785r14slw31UmfAQfMUGUFGgtNPPcueA/rWK3QXXnbutydv
Yis0mWi+7wNT3HWYFoyhMZyK0pkwb4Rqo8G9fjl4Lql9EgcspH2t7IDL1kJ9j7H9lPksxIUzSX8x+dzjEBGCyL
tn3ObKekA7r1fXy6bFS+5kYecZNcG+heJzjdOZzwKvch01oGMgB2Sj8l9z9KknP7yxZIxtqsfyoq0AEx0+/PfN
bHN87ZScN7r6JZDgVQZGtYFQNHTIQShphK1bqoy2i/kM+Pl+DjC37kwCtWZIld3JMMjwUG+V1/eB+Zo7g3XIU+
rYx2g4o3/BWF+r1Fj/jkCzVkjxLwte6xW/9M/LWu1aHq6TTvndtDeUQ5uiNECpkbfRECMFNzBgdWQljAEudkgT
1n2T6Z1tVmCPxUmVkWJ2zdfB+WebRs+fNL1UJ3mAf6fXv9BwvHfCM7cLuX6nH3TOfPVZ5Wnnrp60Kpn+p7fLcc
vrkVsuAgJ1k8x8zFc2Ixp685qWkTsou0ydhWt5IiXo+UX8ASHKPI5X8eCQyWV+9GSuh5gJeqZlcXbevmCQ68SM
TZibWKbSGpdT0wzzgIkh0ZJTAe6++OrWwlI5c+CBc8jsacomMa1auRWN7HBvrtn/n2v7+p13ifItCRZ6Sb2XlR
xwttDxBjn3sCMdiezc4y4EkaVeeUsPh2rfZTmR5o36XM6OQ8uQWgsZtK+XrjGqXWH3KshNKkR16S9PdKr7u3Fg
jrzl6NMh8WFw5omEyOtgl+afIZAlDbsB/mUEYxCA98p4OogosQJjgJTyPmyZNRV1OhFkNQfV4WtteJ+I1PWwmk
Qsu3XzqFxrVMh2P8UhzNW+o9H2/6By4UKSWyiEtfGcexcmgGbOYMFpq31TPOyHlrTXYLCCP1YpSBT9DUhBavCe
APtpPCrT55dcS8QUm8suxozvppusA+5+Rq2bT49Kb+5Pzx5/Pz7oOcXN9RrSAQWtbIMCFJffxJV56ajbX2Nb5x
+iaM2jhcjkbXYL3IeR3g9UJMwaILItsld2EBVfBtoLtGPfAP4xAVQ6GjlalOtyqI4SkK6Tq6asqtAoT5OUnzc1
54ZA5ZZnCNiikrfvWrokAdcr7QitMoQLpSids7S11DaRFKl9EePuQqpn4FcL1bzdxiVfUckzRyuyPoyTHQtcOs
MDVuAwtSchWYtksdPiHWwP2jhqb1irHHW++xaKaFApyuavp3kp6GRTl63kbr7ZaXndKFtQ0ZZIiDtWMR38v8ua
lbT+btQRKpz8JzCe5+PcTwq3qoetSsBen5CK/zz9TePTti54qVDegVo0DoT0EzlL75Ga4VVx0L0TuQLwvkn/FE
0feVrTpV4O/uUVnbJuaVLWkA6kPDG8xJ7C4zLgLrJl+cJgBOiT/nq8LWlvIPqnrvBURUtoisyKdmxaueYjMA1A
qPlE6OzZjU/YFxkqOLly28S3TU40/BmxffNBy9RG9z15NffCw5CWz/bF2iPFCIauQWsRoB2em+PhG04X4jTzXO
/epaP7YCiRedP9VDXwKurfaamqGpcnY618UOEYbf9ue3GYhHHdRPEr59Km6crazOn5nY3z1cUrxJxUKbFuL3HO
0QEFLycLkHEoWyyd5qPkdGIVO4FK8OWFpkkcFZGxQL7NPDmlMtgJS1Qw1zwwGU4SUihEgQLSDksWsAEflE/YgS
KD9Tx85L0jkJLvEsBY8oL3cgziQNYzGsZLnT2viNptMqtmqMNFKCQd7hImKt3bZThyqhK6D6SKdLzdXH7iklHS
AEj7jHiPDwCR2JQ0YKfSIZhfvgDYyia4Ba6g/0SKEv3H7mOdzRxRFXaOeri7wxk/7+F1qm77qRmMkXaU50EOR0
osjMJJ0Y+nyd7jCmMmBDz/Ge4m7IPhL397a5fWcuk8aE6a5JtnX0GTtnIsHuILZsmAVK729Nc9YMNMUnEgX+0f
uUEHYN7Y/FjVv+y1I5/3FKoyVslVd/ehwW6fLJYnRCc+fxtNPyKcVURcg9zOLJntnDw32D4RynfIweH8aItPGO
0Pg1ah91BYd2OBq+Gtm+L0f4nKV+mo7wQeDLLPjjIyyfso+5ViD4Pj7RFzdiazJ6PPMsz1PC8lImVYO/4absBG
n/GH5qqgXvq96zzpVzyF8v0vDGrc9a/baWl32HecvWYG9teums88AUw8PSlddmeUJoICV8ApTqDsEQVKzRfdIt
ijhE5GBNWILTE5XrzFVwbIW9ArsXRwSP58XIb05FK2doCD8BiP5be6+WlcjQYXmE3yx3WG4CKfRuILD8vqFN88
XrxJsq4/qZl603S6F+vj7Mho9RB1SPElH7rb618AeeH/bO3R8G/ROr/2leChokfR6NqhGpgcZPMKH8WmTomhwq
TsXw8f84vKWL3srTEnM971Vt8D5t+p/geo07XOWtTbgyKtSG3sC4p7i3OqQ0hUcTmsbL0/9ux0w+rk1noXW+TC
QfX9OJNCRWoZ9oX7YYWGVb9MJ3DzQn0gLxg85OXBrbnOTflv3d7GhVjAh2kC3PlMkIOqYDzot4c9m/ZNvVJF8A
paK1V4T0Z49idBPAQXrlnyhEkCiOKAJeRqAKIPDuHeTduFTGjRdE8vaX+U70yBsYKGF/27SbFMsCiaumVS7KsL
dI1rR1rAkjPWLqrTFG01BeRVazU8rkA5adqpo8xdF9zPmcfHSPuLKk+456CVNb68sLMj0hsa8zzGBrd6tbLcx4
gnYnrUkUunnCuNK/5O+7y6nfZwCa5OCFanAVOQKUorr8v66dSfN7Ur1UWohyxZO5NBnkp8Rt4aip8wKnDWFjbA
zSjhgXJBgMLv68tHEnCRV/e73H5PbcHqsjKFFJU7kd5hgNFj4RaE0zXAARYSNOkzdD0c5cSDcBIWvFqfXl0rVg
QeiDA0yqBtETtbVows3Sg3WW+951b9qw4WAGvKNtWW++XKkK/uQ4AudwvYjNWYgx1qYqsXcNVLYLAss9IaIevY
zLZXXXvAHVgY3Kcj9WXdjOpQQ93oduNKcVd9EvzFXEptDfP/VkzzYNWHC8/VFFKSK3n5xVz9p+KiA5Ywv//0Br
```

```
uptRnXTOqOubgAPvUFswQKsxl2V/cjMhR7FBulTWIw5rnk9mvzKvf4mT+Zp4C8ia9JSXyZN4SglwwJ61UQjqIN
EXUbjr7g/NzDUn8SEiVQcSZPZQ8YEvOMBosvRBXkiKMdoM0hMkuPsw6psqCJyYpl/0XfRvPU9jPVPGNtXEzwBs
zVgTiYQbXbycaY7Sp+OnkKaUEchIn8xARSyr/VztOfI5hn8Cx9K4k56305+KzKyUbsac3RVXSxsCHJbOnFQiK5
7Zcv1+CY7CFMQAgsW+rEaiy2vDT9Y1atbOyqqtG1Z2+W/qWZ6+ehGfoVoHCauEPqPAoP4GCv8ur//35unyduRC
i8ZjVg2+xUWSOg4ncEFnHgoIhJDWGJgCjW2ZRL97EawzTxENvoQontx76rF4wHwO8qKddLw4i/0852cdVQtdFO
BHI9jdIL9wR30p5hMzC7pnkYMg5iNirvRMqw4AhVPpzG3XLPQw5ziR7zy/aJN1oj0RTf4jdmyRlwW7Iro9bfr9
9nYNHDo/nVkDCBFP8QX+34ojvF226hgaSK69HBFUYOs0RYkxL1T4x9RMRH3V0qK19Q3rVQ3KJrLaA22KCp3GMK
VZNNwtJfn12oy2y/ioHC+y9b889Yq75ykYapr9B+gkxCaW8b2LOVQCGEDoxojGurYVXsdNpTjClxkmx+xY2ux1
Eom+d7Jka04HQpqEK4BhUztZYoHu5Mavpk0c4KqVoDcZuYznK6abCYBK/kclCBo+hiommWuJe8/KYZ1+n8EEDs
Te41cJA8ro3+zT7hZv4/MK+ItY3BgHK+Rupmj1ml7hJjg0lg7Eqj4uRi+vCCxeg3kzpANoMk8cHNMyPx4V/XRA
kaw9iEpkh6JUpqSFOnzFbrmuepkJH7qqRYNXBcwazHGvq60RiUeNaIioWJWKUxWe50IykXoEnOkh5B6k+y6T3y
AOHotFFMuz3qRSzNE1Nv0V0UGPOj4dyiLzpD4iQwAxv9P/Z/B/w4DYxtTQycXe1tDJ2uAt5Ops4u9k6mvpRngf
wA0vbZhCmVuZHN0cmVhbQplbmRvYmoKCjMzIDAgb2JqCjc4MjEzCmVuZG9iagoKMzQgMCBvYmoKPDwvVHlwZS9
Gb250RGVzY3JpcHRvci9Gb250TmFtZS9OaW1idXNTYW5MLVJlZ3UKL0ZsYWdzIDQKL0ZvbnRCQm94Wy0xNzQgL
TI4NSAxMDIxIDk1M10vSXRhbGljQW5nbGUgMAovQXNjZW50IDk1MwovRGVzY2VudCAtMjg1Ci9DYXBIZWlnaHQ
gOTUzCi9TdGVtViA4MAovRm9udEZpbGUgMzIgMCBSCj4+CmVuZG9iagoKMzUgMCBvYmoKPDwvTGVuZ3RoIDg3M
i9GaWx0ZXIvRmxhdGVEZWNvZGU+PgpzdHJlYW0KeJxd1s1u2zoQBeC9n0LLdlFYEmc4MWAEIGUJyOLeFk37AI7
NpAYa2VCcRd6+OnPU30XiY1oafhzJlNfd3e5uPF3Xn6bz4b5cq8fTeJzKy/l1OpTqoTydxlXTVsfT4bq88/+H5
/1ltZ7PvX97uZbnu/HxvN2u1p/nz16u01v1Lh3PD+X9av1xOpbpND5V77529/P7+9fL5Xt5LuO1qle3t9WxPM5
1/ttf/t8/l7Wf9eHuOH98ur59mE/5fcCXt0upWn/fkHI4H8vLZX8o0358KqttXd9W22G4XZXx+M9nTV3znIfHw
7f9tNq2OLau55c5N8wNcsvcIgfmgCzMgqzMihyZI7LNua2bDfINx2+QN8w+npgTcmbOyB1zh7xj3iH3zD3ywDy
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/AnOhOcic4EZ6IzwZnEr4vf54nO+QU72rJzyebvjSz5ApqNF
+ICEhaQuIDkhZYNyMcTLzygiQtIWEDqfNy/5GnHjEUmLGDItaMHZsyVG9YBOreefTPNgRnHZzY9Y97MRmc0OnO
j9E0237AO5sobZjQrL040MdOZ4cx0+oabF6fPy5sj4+bolhsa4x0bndGHzhvddji3c1u7Qx+6pdF+buQ4Lnxnz
OhJt/TTxxPHYe4Wm4/vOA5z1zPD3A2cF33YLV88eHo63dMvNwFq9j5X4xtiz2vnnt7nbXxz6Zd5cfzAOj3qDL6
Wxh8Gw1IHxw+sM6DOsNTBtRtYZzf8eZPhAYon/M8Hc3V4nab5oew/A/xpjOfwaSzVr58Kl/MFp/nfD5Io2CIKZ
W5kc3RyZWFtCmVuZG9iagoKMzYgMCBvYmoKPDwvVHlwZS9Gb250L1N1YnR5cGUvVHlwZTEvQmFzZUZvbnQvTml
tYnVzU2FuTC1SZWd1Ci9Ub1VuaWNvZGUgMzUgMCBSCi9GaXJzdENoYXIgMCAvTGFzdENoYXIgMjU1Ci9XaWR0a
HNbMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMAowIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCA
wCjI3OCAyNzggMzU1IDU1NiA1NTYgODg5IDY2NyAyMjEgMzMzIDMzMyAzODkgNTg0IDI3OCAzMzMgMjc4IDI3O
Ao1NTYgNTU2IDU1NiA1NTYgNTU2IDU1NiA1NTYgNTU2IDU1NiA1NTYgMjc4IDI3OCA1ODQgNTg0IDU4NCA1NTY
KMTAxNSA2NjcgNjY3IDcyMiA3MjIgNjY3IDYxMSA3NzggNzIyIDI3OCA1MDAgNjY3IDU1NiA4MzMgNzIyIDc3O
Ao2NjcgNzc4IDcyMiA2NjcgNjExIDcyMiA2NjcgOTQ0IDY2NyA2NjcgNjExIDI3OCAyNzggMjc4IDQ2OSA1NTY
KMjIyIDU1NiA1NTYgNTAwIDU1NiA1NTYgMjc4IDU1NiA1NTYgMjIyIDIyMiA1MDAgMjIyIDgzMyA1NTYgNTU2C
jU1NiA1NTYgMzMzIDUwMCAyNzggNTU2IDUwMCA3MjIgNTAwIDUwMCA1MDAgMzM0IDI2MCAzMzQgNTg0IDAKMCA
wIDAgMCAwIDAgMCAwIDAgMCAwIDAgMAowIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwCjAgM
zMzIDU1NiA1NTYgMTY3IDU1NiA1NTYgNTU2IDU1NiAxOTEgMzMzIDU1NiAzMzMgMzMzIDUwMCA1MDAKMCA1NTY
gNTU2IDU1NiAyNzggMCA1MzcgMzUwIDIyMiAzMzMgMzMzIDU1NiAxMDAwIDEwMDAgMCA2MTEKMCAzMzMgMzMzI
DMzMyAzMzMgMzMzIDMzMyAzMzMgMzMzIDAgMzMzIDMzMyAwIDMzMyAzMzMgMzMzCjEwMDAgMCAwIDAgMCAwIDA
gMCAwIDAgMCAwIDAKMCAxMDAwIDAgMzcwIDAgMCAwIDAKXQovRm9udERlc2NyaXB0b3IgMzQ
gMCBSPj4KZW5kb2JqCgozNyAwIG9iago8PC9GMSAzNiAwIFIvRjIgMjYgMCBSL0YzIDIxIDAgUi9GNCAzMSAwI
FIvRjUgMTEgMCBSL0Y2IDE2IDAgUgo+PgplbmRvYmoKCjM4IDAgb2JqCjw8L0YzbnQgMzcgMCBSCi9Qcm9jU2V
0Wy9QREEYvVGV4dF0KPj4KZW5kb2JqCgoxIDAgb2JqCjw8L1R5cGUvUGFnZS9QYXJlbnQgNiAwIFIvUmVzb3VyY
2VzIDM4IDAgUi9NZWRpYUJveFswIDAgNTk1IDg0Ml0vR3JvdXA8PC9TL1RyYW5zcGFyZW5jeS9DUy9EZXZpY2V
SR0IvSSB0cnVlPj4vQ29udGVudHMgMiAwIFI+PgplbmRvYmoKCjYgMCBvYmoKPDwvTGVuZ3RoIDI0MCAwIFIvR
3VyY2VzIDM4IDAgUgovTWVkaWFCb3hbIDAgMCA1OTUgODQyIF0KL0tpZHNbIDEgMCBSIF0KL0NvdW50IDE+Pgp
```

```
lbmRvYmoKCjM5IDAgb2JqCjw8L1R5cGUvQ2F0YWxvZy9QYWdlcyA2IDAgUgovUGFnZU1vZGUvVXNlT3V0bGluZ
XMKL09wZW5BY3Rpb25bMSAwIFIgL1hZWiBudWxsIG51bGwgMF0KL0xhbmcoaHUtSFUpCj4+CmVuZG9iagoKNDA
gMCBvYmoKPDwvQXV0aG9yPEZFRkYwMDQ2MDA2MTAwNzQwMEU5MDA3MjAwMjAwMDVBMDA3MzAwNkYwMDZDMDA3N
D4KL0NyZWF0b3I8RkVGRjAwNTcwMDcyMDA2OTAwNzQwMDY1MDA3Mj4KL1Byb2R1Y2VyPEZFRkYwMDRDMDA2OTA
wNjIwMDcyMDA2NTAwNEYwMDY2MDA2NjAwNjkwMDYzMDA2NTAwMjAwMDM1MDAyRTAwMzI+Ci9DcmVhdGlvbkRhd
GUoRDoyMDE3MDUyNTE0NTcxMyswMicwMCcpPj4KZW5kb2JqCgp4cmVmCjAgNDEKMDAwMDAwMDAwMCA2NTUzNSB
mIAowMDAwMTM1MjE1IDAwMDAwIG4gCjAwMDAwMDAwMTkgMDAwMDAgbiAKMDAwMDAwMDU0OSAwMDAwMCBuIAowM
DAwMDAwNTY5IDAwMDAwIG4gCjAwMDAwMTAzMTEgMDAwMDAgbiAKMDAwMDAwMDEzNTM1OCAwMDAwMCBuIAowMDAwMDE
wMzMyIDAwMDAwIG4gCjAwMDAwMTgxODggMDAwMDAgbiAKMDAwMDAxODIwOSAwMDAwMCBuIAowMDAwMDE4NDA0I
DAwMDAwIG4gCjAwMDAwMTg3ODkgMDAwMDAgbiAKMDAwMDAxOTAzMSAwMDAwMCBuIAowMDAwMDI4MDQyIDAwMDA
wIG4gCjAwMDAwMjgwNjQgMDAwMDAgbiAKMDAwMDAyODI1NSAwMDAwMCBuIAowMDAwMDI4NjIxIDAwMDAwIG4gC
jAwMDAwMjg4NDIgMDAwMDAgbiAKMDAwMDAzNDQyMCAwMDAwMCBuIAowMDAwMDM0NDQyIDAwMDAwIG4gCjAwMDA
wMzQ2MzUgMDAwMDAgbiAKMDAwMDAzNTAwNyAwMDAwMCBuIAowMDAwMDM1MjMxIDAwMDAwIG4gCjAwMDAwNDQzN
DEgMDAwMDAgbiAKMDAwMDA0NDM2MyAwMDAwMCBuIAowMDAwMDQ0NTU4IDAwMDAwIG4gCjAwMDAwNDQ5NDQgMDA
wMDAgbiAKMDAwMDA0NTE4MiAwMDAwMCBuIAowMDAwMDUzNzg5IDAwMDAwIG4gCjAwMDAwNTM4MTEgMDAwMDAgb
iAKMDAwMDA1NDAwNiAwMDAwMCBuIAowMDAwMDU0MzkxIDAwMDAwIG4gCjAwMDAwNTQ2MzEgMDAwMDAgbiAKMDA
wMDEzMjk1OCAwMDAwMCBuIAowMDAwMTMyOTgxIDAwMDAwIG4gCjAwMDAxMzMxNzAgMDAwMDAgbiAKMDAwMDEzN
DExMiAwMDAwMCBuIAowMDAwMTM1MDc3IDAwMDAwIG4gCjAwMDAxMzUxNjAgMDAwMDAgbiAKMDAwMDEzNTQ1NyA
wMDAwMCBuIAowMDAwMTM1NTc2IDAwMDAwIG4gCnRyYWlsZXIKPDwvU2l6ZSA0MS9Sb290IDM5IDAgUgovSW5mb
yA0MCAwIFIKL0lEIFsgPEM1NDcyMzRBQ0IyMTJCN0NENjVCQTk2MUY5RTlBQ0FFPgo8QzU0NzIzNEFDQjIxMkI
3Q0Q2NUJBOTYxRjlFOUFDQUU+IF0KL0RvY0NoZWNrc3VtIC80NkY1M0U2RTZFNTBEREJFNDg5QjBGQkJBN0FGQ
TQ3NAovQWRkaXRpb25hbFN0cmVhbXMgWy9hcHBsaWNhdGlvbiMyRnZuZCMyRW9hc2lzIzJFb3BlbmRvY3VtZW5
0IzJFdGV4dCA0IDAgUgpdCj4+CnN0YXJ0eHJlZgoxMzU4MDkKJSVFT0YK",
    "digestAlgorithm" : null,
    "name" : "sample.pdf"
  }
}
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2020 15:38:59 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" :
"JVBERi0xLjQKJcOkw7zDtsOfCjIgMCBvYmoKPDwvTGVuZ3RoIDMgMCBSL0ZpbHRlci9GbGF0ZURlY29kZT4+C
nN0cmVhbQp4nGWTPWvEMAyG9/wKz4Wkki3ZDhhD7y43dCscdCjd+rEV2qV/v5LsHLmUgHAkRXr0WoEJ3e/w7cC
BnHjmybtMOGX38z4837kviaHEvGT8fA6Hy+DzxC6DF9flzd2f0aGcPl4K5colch2xxGMdvVpxnNXhoY5UOFXfH
cnXvKZRzwgloB6Dtwwwmy1Ai5aaLf0UW9w6pWBxDtE+jcvGScf6enkclsvwpBNOJDPgdoqU0xTbFH6dArBiAes
```

```
AQXCbEUQuwBArs76FAkmNBXMdkwZn9TyoOWgNGS6r+6SeVnDRdJ3fF4SaYkEERq/ZZxUIA9INs1BHU39Lnbxcj
1GHDTVJE+EZDXk0ZmzI1M+pB3ILzDUqrnJ1XnOfJM/LswgSKqZg4a2QvokpaFssnteVoA0Wdyz+p2RK69uNlPN
eyrFpGZqWejlXHYPyRQMUB+s9iJpeU7Qmas2dogYfd3tAvKLzBl21srvZ7IE0QeGONeseSBe944au2PAABzgaq
F+vu9WwTKHFDmsxElrbHSEdsXdLFttj0/+fMIWwrm/cra8gUWmGuqRtD5QDeyTb8Sq27YHKbH+lzUDKGfSzU/s
6tDU7yyBXuif3B+aa6fYKZW5kc3RyZWFtCmVuZG9iagoKMyAwIG9iago0NTkKZW5kb2JqCgo0IDAgb2JqCjw8L
0xlbmd0aCA1IDAgUj4+CnN0cmVhbQpQSwMEFAAACAAAJme5Sl7GMgwnAAAAJwAAAAgAAABtaW1ldHlwZWFwcGx
pY2F0aW9uL3ZuZC5vYXNpcy5vcGVuZG9jdW1lbnQudGV4dFBLAwQUAAAIAAAmZ7lKScopFokGAACJBgAAGAAAA
FRodW1ibmFpbHMvdGh1bWJuYWlsLnBuZ4lQTkcNChoKAAAADUlIRFIAAAC1AAABAAgDAAAAwv3H6QAAAblQTFR
FeEseflMoWrx6ZcGDbsWLdMeQd8iSesmUgFYshV00imM9jmlElHFOl3RSmHZUm3pYn4BgoYJjpolsqIxvqo5xr
JB1rpR5sJZ9sph//5F1/5N5s5qBuKGJvKaQv6mVs4H/u4//vJH/pLrRqL7Thc2dis+hjdCjktOnlNSqmdWtnde
wo9m1ptu4qty7msz/sMPXtMbZtsjau8zcsd/Aoc//o9D/rtb/sNf/s9j/vN3/tOHDueLGveTKw7CdxrShyLakz
Lur/5uD/5+I/6GK/6qV/62Z/7Ge/7Si/7yrw5z/xqD/yab/za3/z7D/0bT/1rv/2L7/1ci6/8Ky/8e5/8q9wM/
fwNDf2s7C3dLH3tPIxNLhydbjzNjlwN7/2sP/3sn/0Nvn093owOXMxujSyOjS0OvZ2O/fxOH/yuT/1uDq2+Ts0
Ob+1On+2uz/3fHk/83B/9XL/9jO/93V4c7/5NP/6dv/6uPc/+Pc4Ofv7+rl5Ovx4O//7eH/6O3z6/bv4/D+7PH
06vT+8e3o/+fh/+nk/+3p8Ob/8+z/9PHu9PTz8/X69Pr29fn9+/b0+PT///j3/v7+AAAA////LxzPJgAABItJR
EFUeNrt2I1T2mYcB3D30rPacU5bWZ3arU5Bypg1Oz2x1L4ggpuXZJReXWBxTDdCTBQlVJcLBVo1F5Lq70/uA1y
3219A3H3z5HnJ7wl3nzx5wpPLAF3HbQBqqKGGGuprpj6uEFVYpmzuKEt0dkYVOnqfOe/0nWSPzlhV6Z2ZybDi6
t/jPqrfJ5PHExOHL9KUSaeepyvJDMOlzyuZVPrZ+XEqk0snN55lcrlUljLZykQ6lU1Rrv/qyuHJxvphLk3H68k
X6eNcNpOlw6NnuVQm+e7k3deHbGNXsp5LUXoj+zzDuohdgU/n9dm5r+e1zRLZdrfV7lVsNy4to9dvVD/Gup2yL
9SXoi3yF6IgGiTLoshXt3ib52nHqskCb9cMEm1BrlYtWaZNqtFWzRIMoSbU+qzeFO/zhsjLIlk/b4nVTUOW2zz
VLFvg2/LmDokk3BdE8dva1ualyMb6J/4bEnijzzPkgmqyZVkyG72qcSkbtnUp1lhTFrdE26qxO2AbVcHgSb4QL
doRa4JMlmz5dpVp/9O6tH32NHqJvTfubsylg71woulS4s0fdBBLRDyXdYZ2ySWPVmN/uuSQ6x+1OzgbGlydC0f
c0GwkEg5F5mZdirqRUCQco8Hd0KfRz79/FH1wIzQfmXf9o15NRGM/hsNRGjwIReajj0JMHXsQm4vMx4ilSGI+s
Rr7LDS7t3rD8f3b01/+fnvKs0TlYrdZdn5nVZlIpbJTzPdOKBRMoiaL1YusNAtqJ9gsqn1VLxQkiSsU6lKR8py
0KEkPOVOSSDXb5TxXeClRgaQCJ6mLXkHK56mcr3P5hWJ+xuurWiJpgVnLp8xXmJFMTpWKKlOfqpxaZNdDUlPiH
krcjCMVpEU24tJ3HPvNy/7OEJNMp1k3PbPzN+jRadOsU3evM3S7zrpVaktc22y3qWmSp1K9rXaCfVUzaTezRaV
Teh9jvWXmP/OgF/b88DSurLT0Ukn3WooS11paSY9TvBVv6A2dVkhbbumNhlZSPF1r0UpDa2j6D/5Q7y8rbKNSv
BRXtE5LIVZprPYUpbS0oi3FWa10okvLLOQH9b7e0jRWXDXqrf0WG0udSg3lN03TidHZ6OsNRVdaeue4oWkNXfO
DeuSxQ6Ov3zrB7bEr9+3a07VutM5ykJwrch7T9pXvVpnXAzenp28FpmhoeHQoMH5n6NdXt16NTAbpl7Gbd4bvj
g7fDgw9CTh+U9+dDgaDX2zT9NTUl5NPJ8fo70++GhqfpnsjwbHx8cDoZPD2k8C9a/AVZ+1afntyrqUa3/mghhp
qqKGGGmqooYYaaqihhhpqqKGGGmqooYYaaqihhhpqqKGGGmqooYYaaqihhhpq
```

```
HLnLy5o4swXxc7pzLW9wGJCxtWADGNWWEhYDkho+q8F+/sJA3RPsxBxPSCi/2ZlISYcENOWpwV/YK7jdUH5nVp
rnvq/bNx+A1BLBwi3Yk2llAQAABoZAABQSwMEFAAICAgAJme5SgAAAAAAAAAAAAAAAAoAAABzdHlsZXMueG1s7
Vpfj9s2En+/T2Go6L3Rsry7zdoXpw8pihTY5IA2vdeCliiLDSUKJGWv8+k7JEWJkiVbWW+BO5zzsIE4fzjz48x
wSPrtj885m+2JkJQXmyCaL4IZKWKe0GK3CX7//DN6DH5894+3PE1pTNYJj6ucFApJdWREzkC4kGtL3ASVKNYcS
yrXBc6JXKt4zUtSOKG1z702U9kRo2yquGH2pRV5VlOFNW9HFm+nz2yYfelE4MNUYc0LmPriKZ8q/CwZSjmKeV5
iRXtWPDNafNkEmVLlOgwPh8P8cDfnYhdGq9UqNNTG4LjhKyvBDFcSh4QRPZkMo3kUOt6cKDzVPs3rm1RU+ZaIy
dBghU9WVe53kyNivxuBJs6wmBwbhrm7vHfJ9OW9S3zZHKtsZE0ew49ANH8+PrWxIPKpc2neDlSxoOVkNy23L88
5b0zVAjZBjbnLxeI+tN8e9+Es+0FQRYTHHp9ljzGLG8R5PgQa8EUhcCCy12HquIV2elTzQyhIyYVqDEmnFyhAZ
9mkV6ZyNp5emupYdyJJBlnBnLsQUg0CHe0pOXwXdCrn+QVY9RbAlKFLIobJr1NnBaJFqHmatIElaYuq2DVlP+V
VAU7AVlEDSJ5LIqgmYWbE1h0NfpQx/gKV9V7haegUUEqYy9DGpUE1nKNcIlpAYPJy7Ul3qoWUd2po+T7/Gmoa0
rsH1Md6Hm/TXAbv3A6ZctgdUxwTlJCYyXdvbWVrhmf2Wxu5CZ54RtXsJ7LHBd5hQaNgBsXMMeeUHTfBP3HJ5b/
6nHY0CM/r/4n8if9TzX7DhRzV7PHUSmeeypKqGArZHqY0ETVxQoA+vTijZrp+yp8FISMOeqQXq/9E820lDTyzp
1GXOlzX+xRnPOYMNtWBCX3aiyd4orA/m6y7sFZ9xgHfrADakQIYoNYLnuPitYw7F7k9vgmmyQOVV0VDPw2/NV/
PGneUiuTXWPcbrwQMfcAWkNn7T7Nfya5iWIzaOS7yahaHY5WxHrdHCedZQlJcsfqA4TTXJu8ELjMaB463/kYlF
HUiFIUDiXZTKsG/EGgJGYce9Lu7+x8e8H0w0zscFH3GGsqb5SqNIfZTvj6AKsRLZQp6wZH+rkVkhhN+QGCtJAo
9b4LF/C7OB4nHHlFBR4mgASdIljiG9h9lXNCvXG9KmnX5eJZ5r52KT1mhY5mq9YR1QGcNMgM/DlRlyB6oUsykF
3ElFtjg3UHbkDQ/wpXieg4IQ5oQblkxK7OmThkztoJgOKzAEtFYOYruGLVtOU9AnAmktp2gokVCdJumD56+M85
IZyM0PRA3vJQ66sbNbti13SfeVJIADIVeVTN5HSxKVKRj1FghhWAydEm/Aj1alsqMMVzsKryDoawyAzE0PkpAx
Hz4vUGIKGhQ0BciCuPd4JwI+ldcXEj2VkSb4USixfyhbFB3Fjnq18xRatMc4f2nUyP0eYuR58GK2Ju8YbVY9CZ
vqBntT9+QfvkUtCvaKRBTqkYTAsHZWIYl4QLCtZCbYGlrAoUgaL6yY5mRwiw1YjhJYKGMoaZcMJrTxreJ+VBWR
awqq1CXG4AAQIGFv5wwLtBRQqFsFHoSaEIfojabuylVAtJtKt/i/n8n7v3II67e9cNRkBzTAukLDBeTpzFbVjL
rsVyRVPZ86RVeRvyQsldlWy50mugYhP0GAorhUuoAv3ZiJPihNzmM9LL5CyElUnxHVKbvonRCXprYn7BuqSC9E
iySYLSouMVjWEKt0LnVZtqpvg8EJ16Kj6qDgeaaFQ2bUug09hk+w8Afy8UfW54ch8y6VP5yLKD8AGSlbg7ul6Y
5aMe3XCl9PwN9Q7TUJHsejyHg4f8Ks/623CyC6SoK01VgdsBHeakWjRQacxhwdebvPwv0S9p9m8YvrUq1EbXQ1
S24UzTWiDt669JYhRz0bVqxq01qmF94AGoVjHrTcIz70xZd7VGb7hfSfCh1YG8tGT56yTXzydek7ouzcjQj799
MzUjTVWSE7jKlt6bF99NReoJu4yXunylMTKucUAw6wbr8pmCNXh6tk5F5j83R8RVjA6AQ+BtLdl2XrynZ5oHMP
t+Yo5bsUkzwuNedxXgZH2kG7ZiWgK4YTqtwlp+8yBazmjog/LcFwbTGzrOxoddWTo6iX+CI+/yKMUSNvrMx9Er
r/d+XqcZ8Xil7X3AC9b8tJegxMrInrGa3AOgBcKRpuqoc6fc3DMWzQVYXsVq0D61P4pKaThzWyuQdcRceeAtkC
JVdod8rhtT2WGrdZjCFtpofSIK2R1tOoR/24Bi0Dahjbk8EZHkDpAvI3Q2QLiD3N0C6gDzcAOkC8sMNkC4gb26
AdAF5vAHSBWR1A6TXmC3+jxDpknyYCq6IhFa5SOmuqu/BGgKqjwMp50p/DyEW1Q2/fYXcY1YR3fDbQScoG6TqJ
wVfxp4S9JuD1ud+3qP9nW4hKZIxA+mwgU69RqS1YGia0TOOfbs1V56rVfuCMoROraRFgZFU1TRaxML8jFLv9N6
zt9HWvnbra2jQSWPkCO50toPVxkdY3c5B5WOZR8EAU++0bSgHmuhfHS6j+WIR1TcBhuCue5ar+ZtRD+spAECFu
KD6l1P1UnOhBKYq6N84RMO3Db1hjdHJoKjvn8ZerHqPwDb+UI6fG1/0XVj7c4iaQZLSqbNYLOaL6LGdxL2XoS0
Bzw2/5okW0QAPTvWr1BBLm3+bQHJGm4M3Tv6spLJRYGPDjgtIYrc6D9+3Dz72dwgL8y/wX62H1tl5mxGsn1XMR
+hD4A2eKmoj8jQEa0KOZaOjma0e1JrOvoz4Nnuh62VCT304/Jvyd38BUEsHCLzDyaM6CAAAky4AAFBLAwQUAAg
ICAAmZ7lKAAAAAAAAAAAAAAAACAAAAG1ldGEueG1sjZPNjpswEMfvfQpE9wrmM4BFWKmHPW3VSk2lqpfIsWeJW
2Ij2yzpI/U5+mI1XynZ5NCj//Obmf8MQ/l4PjXOKyjNpdi6oR+4DggqGRf11v26e/Jy97F6V8qXF04BM0m7Ewj
jncAQx6YKjafQ1u2UwJJorrEgJ9DYUCxbEEsKXtN4bDQp54aLn1v3aEyLEer73u9jX6oahUVRoDG6oIxeuLZTz
UgxiqCBoYNGoR+ihR0c/q+pgV1bklJeGg34ZHpsFwVBgqb3QteKsebeAJaNkXVIDPFeOfTvXWcef7XwyK2W7Q4
2qnI0wwU3nDQeVUCMVNUTMX9+K+e7lo0p0V1kShxftrRnu0IVBWHmBakXpbswwUmBg9zfRHEQREkSz3WuM0pG8
Z3UNMNh7GdFkWabIi9KtGADf9/kKjBZA2YNi9pjnRobVp934cc4/TL7uAlfZ9FftAFdRW/oWZ7Yy3lqY0tow6k
z6oYcGvCo7ITZuvYrT/s7kfpGlIcfQM1btV2R4UVTpFakPS6BzRzopWKLFhWzSI8WpwbUpUy2FBJ29f2RG9Ato
bbNDZnkLpoHrEGAGlf6zA8KPo13g1I/8jM/enjmojvvv+Wb/SZxVsC+VXIYyx7kw4eON8xbtvivXomujhDd++G
rv1BLBwh4qlPx5wEAAC4EAABQSwMEFAAICAgAJme5SgAAAAAAAAAAAAAAAAwAAABzZXR0aW5ncy54bWy9Wlt32
jgQft9fkcN7SkJC2nASegwpbRoSOECbs30T9gBaZI2PJAf49zuSTZoSvE1ttE8hvsxIM983N/nq4zoWR0+gNEd
5XTt9d1I7AhlixOX8uvZt0jv+UPvY/usKZzMeQivCMI1BmmMNxtAj+ohel7qV3b6upUq2kGmuW5LFoFsmbGECc
vta6+XTLacsu7IWXC6vawtjkla9vlqt3q3O3qGa108vLy/r7u720RDljM/fqip7+qUqRHxWZF/IFuOUNU5Ozuv
Z/7WjfJEvTNOotbd22G6/fZUryP4ccwOxtc1Rftku7bpGKltPHFbPVqvte+/Xd77T84ECNsGktr1jNgndESjnt
fbJVf21iLeL7cPM+JD7yCOz2Cu40Tg5b1YT/gX4fLF/2afn56en5aSPF7gaQUQYg+6CyTnoHQ1TRAFM1tpGpVB
Ox63sKFxpuMcIiqTPmNBvFn8cs+SYywjWEL021n6AuXeIGmrzNpPfRjtL1UZxa2yL5UZ5VxZi7/1Z87K82CKmn
DUbJZHxnWs+FXB4rjixB2e2kzoqpIgl4Fkl2R00BuMi/p2U9N0PxHhConbBtkBlyHsVcMY2mJouijSWu5zOpZc
```

```
GxlZ6B3F5MFK/tkuPhQZVwdqb70sGIz0GAaGBqKfoQoml77n4MrgU3c7j1f4HKDe+PZtmF1LFDOXmP0mrQ4pgZ
qhQJ2QAD05z8m3AGDIviSRbP8nusHA5V5jK3SB9KCVdJGei8LWHQEo0zn176JMjvGQkdPI/K5YseOhr+S7M+vR
xj60fXpMzz8D1NwMlAdVTGI/BpLvZ5mBwn8DaDAULYYEigt1odQg1QZKIzZApNrdevWdq2UMVMzPBhzSegrOJp
82NwHYAUMiyCvInbDog4bSdOZce5H8i00Q9onEhRisIH6fKRR/r/EfyyjhmQnjQ8zPOWf9be90otnLR/vDKuoI
nCURDHppUFVO7goaeQGYMo2LqAWPneB9avskptUtLp2ZLEB96HMLGG003fOEsJ8lM4MpTpJ1YM/kK42OyvICvO
PXCQcu5Lkv0EFRItRelo4sLH7RAIUgLfIoTs+mCEJaKhzfYhDCbGywr2jsWxIF25Y796WFr98wsOkyD7f4Dwef
Szo4Ov7UgymrtwWxGxaoPLNzqPpuCuMnHX77SIahKVckgNdbSfUqq4m8OItI+E/jtXKKCHlfa9EnrLbUf0txKn
yq/JREzYEuuCcSJoN+eAkuHCq6lx6ArYISrOwAfVaNLGxFEN4wyIZGvEqQyJ9OKdSAjsopcaioOrb+7TISpcD2
GD09rGIgoJ8U90HqLm40qGJaU/UwHFVXW44SFfmA7Zk/wWeCUPQcQMuDSE7T0CKxXnmCCGSUPT5GR5rsVO/mp1
j790GxeND5U6b3z2cFo0vcVY12a/c/esmorRdhVlnyeYtMXTuyWtjPwsYHIdQIuROWECMzYMOUBRj2booLUYBb
WD68gCyI+E9Iecx1+G90FqQgpDnYxTqhxsidmxSPdkmMdG6O+Z6dxA9kVqH0MXe1OjE+P5wnDBj57WmkjrRskF
xy1LPh8cUwWpXrY5rGyteEdKBlozuQwldTbesyJdjADajD9h2LkEDW3mrwhm+BgeMiE59p6nE6NxXdPbKjICAn
FEAXuP3+Ezbk6Qcdc23R5aFFJ16c1oVEy0QcW+YkNqVKE8m2tZ/+OMVVh+YovEAJXjkfUV3eZDKFw8FSFp2gIw
hbOPoli61R/ZZ0z0u/mZZWyV7aL7RAw8dRUyTXpCeEHKCS8/gaq1YrTwJ3eZHUpKX0eOutb2efaR0GTf3uQA47
C5R1sdtUQcS7OO1wytXnb8B+TF8gaL5St5p8bYQ8kR0kV908gDCRlAAr/PsxFikaIu57Iyvv3jbPTZtnvQMxGg
H7AG5ixVPjwtO2psrqi+Myr5OLd7IrgOzaYbNPu4d18gw9ovqba8NnGAkk/crO4ZzJlwteAzqkcAWV2S8XAGGX
be6qY7ezDRwH4a7qigjZmMtpTzDrAlSxl+8iiEUUxlOIV1Q9lsy5L7CGGjf7ERT2QtqP83wxW7dwS1N4Dyz+Mg
vY0J9DPvcm2cntU9Kii6s1VjWVCVOHnC/VXnwfWiz6cbP8LUEsHCCh8lF80BgAAeikAAFBLAwQUAAAIAAAmZ7l
KAAAAAAAAAAAAAAAAHAAAAENvbmZpZ3VyYXRpb25zMi9hY2NlbGVyYXRvci9QSwMEFAAACAAAJme5SgAAAAAA
AAAAAAAABoAAABDb25maWd1cmF0aW9uczIvdG9vbHBhbmVsL1BLAwQUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAGAA
AAENvbmZpZ3VyYXRpb25zMi9mbG9hdGVyL1BLAwQUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAGAAAAENvbmZpZ3VyY
XRpb25zMi9tZW51YmFyL1BLAwQUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAHwAAAENvbmZpZ3VyYXRpb25zMi9pbWF
nZXMvQml0bWFwcy9QSwMEFAAACAAAJme5SgAAAAAAAAAAAAAAABoAAABDb25maWd1cmF0aW9uczIvcG9wdXBtZ
W51L1BLAwQUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAHAAAAENvbmZpZ3VyYXRpb25zMi9wcm9ncmVzc2Jhci9QSwM
EFAAACAAAJme5SgAAAAAAAAAAAAAAABgAAABDb25maWd1cmF0aW9uczIvdG9vbGJhci9QSwMEFAAACAAAJme5S
gAAAAAAAAAAAAAAAABoAAABDb25maWd1cmF0aW9uczIvc3RhdHVzYmFyL1BLAwQUAAgICAAmZ7lKAAAAAAAAAAA
AAAAADAAAAG1hbmlmZXN0LnJkZs2TzW6DMBCE7zyFZc7YQC8FBXIoyrlqn8A1hlgFL/KaEt6+jpNWUaSq6p/U4
65GM9+OtJvtYRzIi7KowVQ0YyklykhotekrOrsuuaXbOtrYtisfmh3xaoOlnyq6d24qOV+WhS03DGzPs6IoeJr
zPE+8IsHVOHFIDMa0jggJHo1CafXkfBo5zuIJZldRdOugkHn3ID2L3TqpoLIKYbZSvYe2IJGBQI0JTMqEdIMcu
k5LxTOW81E5waHt4sdgvdODojxg8CuOz9jeiAym5V7gvbDuXIPffJVoeu5jenXTxfHfI5RgnDLuT+q7O3n/5/4
uz/8Z4q+0dkRsQM6jZ/qQ57TyH1VHr1BLBwi092jSBQEAAIMDAABQSwMEFAAICAgAJme5SgAAAAAAAAAAAAAA
BUAAABNRVRBLUlORi9tYW5pZmVzdC54bWytk0tqwzAQhvc5hdG2WGqzKsJOFoGeID2AKo8dgTQy1ijEt69skti
lGGLwbp7/fBpGxfHmbHaFLhiPJfvg7ywD1L4y2JTs+/yVf7LjYVc4haaGQPJhZKkPw9MtWexQehVMkKgcBEla+
haw8jo6QJJ/6+U46enNAPbssMumebWxkKf+rp+q62ht3iq6lEwsiUxhB5VROfUtlEy1rTVaUSoTV6z4CMznnJz
gRkysYThfovtBZWwQ9DB5i80Cg3GqATHkV03RHmngS3tcEB7IxZBepRuotxA2l3VAantWIEqHuT3tyWNtmtiNl
xH24sXzCREHFB4N13OFlau6x3hX1S8MTlVv9wcW4t+/PPwCUEsHCJu2OBwMAQAA0gMAAFBLAQIUABQAAAgAACZ
nuUpexjIMJwAAACcAAAIAAAAAAAAAAAAAAAAAAAAAAABtaW1ldHlwZVBLAQIUABQAAAgAACZnuUpJyikWiQYAA
IkGAAAYAAAAAAAAAAAAAAAAAE0AAABUaHVtYm5haWxzL3RodW1ibmFpbC5wbmdQSwECFAAUAAgICAAmZ7lKt2J
NpZQEAAAaGQAACwAAAAAAAAAAAAAAAMBwAAY29udGVudC54bWxQSwECFAAUAAgICAAmZ7lKvMPJozoIAACTL
gAACgAAAAAAAAAAAAAAADZCwAAc3R5bGVzLnhtbFBLAQIUABQACAgIACZnuUp4qlPx5wEAAC4EAAAIAAAAAAA
AAAAAAAAAEsUAABtZXRhLnhtbFBLAQIUABQACAgIACZnuUoofJRfNAYAAHopAAAMAAAAAAAAAAAAAAAAAGgWA
ABzZXR0aW5ncy54bWxQSwECFAAUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAHAAAAAAAAAAAAAAAAADWHAAAQ29uZml
ndXJhdGlvbnMyL2FjY2VsZXJhdG9yL1BLAQIUABQAAAgAACZnuUoAAAAAAAAAAAAAAAAaAAAAAAAAAAAAAAAAA
BAdAABDb25maWd1cmF0aW9uczIvdG9vbHBhbmVsL1BLAQIUABQAAAgAACZnuUoAAAAAAAAAAAAAAAYAAAAAA
AAAAAAAAEgdAABDb25maWd1cmF0aW9uczIvZmxvYXRlci9QSwECFAAUAAAIAAAmZ7lKAAAAAAAAAAAAAAAAG
AAAAAAAAAAAAAAAAB+HQAAQ29uZmlndXJhdGlvbnMyL21lbnViYXIvUEsBAhQAFAAACAAAJme5SgAAAAAAAA
AAAAAAB8AAAAAAAAAAAAAAAAAtB0AAENvbmZpZ3VyYXRpb25zMi9pbWFnZXMvQml0bWFwcy9QSwECFAAUAAAIA
AAmZ7lKAAAAAAAAAAAAAAAGgAAAAAAAAAAAAAAADxHQAAQ29uZmlndXJhdGlvbnMyL3BvcHVwbWVudS9QSwE
CFAAUAAAIAAAmZ7lKAAAAAAAAAAAAAAAHAAAAAAAAAAAAAAAApHgAAQ29uZmlndXJhdGlvbnMyL3Byb2dyZ
XNzYmFyL1BLAQIUABQAAAgAACZnuUoAAAAAAAAAAAAAAYAAAAAAAAAAAAAAAGMeAABDb25maWd1cmF0aW9
uczIvdG9vbGJhci9QSwECFAAUAAAIAAAmZ7lKAAAAAAAAAAAAAAAGgAAAAAAAAAAAAAACZHgAAQ29uZmlnd
```

```
XJhdGlvbnMyL3N0YXR1c2Jhci9QSwECFAAUAAgICAAmZ7lKtPdo0gUBAACDAwAADAAAAAAAAAAAAAAAADRHgA
AbWFuaWZlc3QucmRmUEsBAhQAFAAICAgAJme5Spu2OBwMAQAA0gMAABUAAAAAAAAAAAAAAAAECAAAE1FVEEtS
U5GL21hbmlmZXN0LnhtbFBLBQYAAAAAEQARAGUEAABfIQAAAAAKZW5kc3RyZWFtCmVuZG9iagoKNSAwIG9iago
5NjkwCmVuZG9iagoKNyAwIG9iago8PC9MZW5ndGggOCAwIFIvRmlsdGVyL0ZsYXRlRGVjb2RlL0xlbmd0aDEgM
TE3NzY+PgpzdHJlYW0KeJzlem1UXNeR4K37Xn833a+/Hh9N06/1aL66oRENEkh8PBA0LSFLLQQKICMaQSPQByB
AKLbjCNv6sJC1ZmLFliNNonidbGJr143kZOU4G5NznMzJcZw4ux5PMrEdkvFMJicmVhwnm7UttHVfN1jy5OOcn
T1nf2yr37t161bVvbeqblXdRlMTRxLETGYIR5SBQ/3jSl11BSHk+4SAfWB6SrrZpEgILxJCxaHxfYeKwj/9DSH
cHwnRafYdvGuoZM/9DYSYkCUzfzjRP9j44blqQtbkI2LdMCJ2L9+lw/6d2M8fPjT1ya9pn4lifwb7HQfHBvq/X
rutHPuvYj94qP+T41/RzvGEyNgl0mj/ocS2xBPnsI9r0G8dH5ucGiT5NwkJHGTj4xOJ8d/3PKbH/kO4vinEAf5
jHzOCWtanHK/R6vQGo8mcYbEKNrvDSf5/+mjOEheJauqIlYyr79s+3GWSzdqbb9/+Xt568/3/m6vQp5rz5MvkW
XKW/IT0pgciJEZGyBHE3Pr5NvkRYtknRnrIU2T2z4i9TK7heIouTh4mj/8Zuhh5jFwlf3fbLDFyiNyDa/ka+Qm
sJd9DVxkj74Ke3Ee+g1LfRdwdf0oUteBrSAWHbsH+lFygZ8gW+hZ2HmcjNEQF8iK5CHtQ8hTu8+zqjmv/jdBT5
F587yTDZBph9aOp+/AfieHm73BX95It5H7SSA7ewvFN+AJnRPt1kC+gTr+t4kIrg7oot59+ndIbj2Dnb8g+fPo
B907Pco2kWWODZwlRWrq7Ojt2tu+Ibd92x9a2LZujrZGW5k1NjUpDfV3txg011evXVa0tD5WVBosKC/z58hqfN
8tpE6yWDJPRoNdpNTxHgQRb5EhcShbEk3yBHI2Wsr7cj4j+WxDxpISoyO00SSmukkm3UypIOfQxSiVFqaxSgiD
VktrSoNQiS8mXm2XpGvTs6EL4bLPcLSWXVPgOFeYL1E4Gdnw+5JBasoabpSTEpZZkZHp4tiXejPLmTcZN8qaEs
TRI5o0mBE0IJYvk8XkoqgcVoEUtG+Yp0WewaZOcv6V/MBnb0dXS7Pb5ukuDm5MWuVkdIptUkUntpqROFSmNsKW
TM9J8cGH2oWsC2RsPmAflwf47u5JcP/LOci2zs6eStkCyWG5OFt/9VhbuPJEMys0tyQCT2ta+Ok/bR1NCUuMXZ
Gn29wS3Iy+9fTumP43R+oXfEwZGUL2zsxFZiszGZ/uv3ZzZK0uCPDtvNs+Ot6CGSawLua7d/MYZdzLyUHdSiA/
DhvRmI+1tSceO3V1J6o9Iw/2IwW+D7Kt2+2zdKzSxPzdMUBGoDtSpz8c2fuaaQvZiJzmzoyvVl8he9xWihALdS
RpnIwsrI65ONjKzMrLKHpfRmm07u2aTvH/zoNyCOj7Tn5zZi/60n5lCFpKWP7h98qzdJtWEulVaCVe1eXBESmo
KUC3IdSsDegpjmRXUjuUPqWbJjRMU2OxSjYximJwWuSWe/k4PZ6EAqTSYjAZSpu/oSirNCCj9aRu1zJeHkKM/j
iYaaVbNlwzJ40mn3LRqT7aslpGdXSpLmi3p3JQk8YE0VzLU0sxmllpm482pJTBZ8o6u50j45uJ8peS+GiaVpLu
ZEYub0K8KWma7BoeS3rh7EE/akNTl9iWVbjRwt9yV6GaOhhoqXsTpfOqMSbqpo6ttp9y2o6erOr2Q1AATx/tbP
iZG7nKnxKDLJfV+vdRF3Vw3EgqIkCIIyE21+E7q/Hp8BFS4imWu2lQrdYGbrFDjMpLFUkuiOU3H+rcJ1TB32hR
dkaZlXZSzKer2dftSn9IgxWEpPTFy6JlSoytDnB8jAeIoilFRTJdZzOelLjkhd8vDUlKJdbG9MfWoWk4rQ9V52
lYdt/VuURaqifhweKXDlJmMBNy3KjfZqvZXu9GPDW9eGZZm9XLbzlkmXE4LJLjyzUnCXFiptrnV08/Osxzpx0O
MJ1o9z7PzisLO8jA7trPy5sFZeWdXrUqNEeRe991sLjtpg7aOptIgBrOmeRke3DGvwIM7e7qeE7CkerCj6woFu
ine1D2fj2Ndz0mYK1QsZViGZB2JdZikduzoVXr3cwohM+ooryLU/sA1ICpOv4IDMnCNpnDCCo4ijk/hFBXHPmi
lrGHUMcbvFmmQ2edT3cOz8W7m40REjeAXkiDXo3bk+nmgWnPSKCeakia5ieEbGL4hhdcyvA49A0QoDd49K7TIv
88qVVM3acbXoKYTK2AdKZsHEqq9ouP1SxXzWs3rtVc4iiCZ5xhaw9BXdFrDh7VXgOHDNp/N77P5mqm0nA/nl4c
1ne8/3cy/TFgluhGz+BmsudxkTinWu0DvBH0G6M2gN4o9u21QaWu2UZun2TPoedHD5XsqEeQ8124uXN1QF2WtU
lZYEl30AI15xj2XPK94eEN2T4wuoimy4gYDFUx6ri9Dw2WLNtrnJHbSULFUUVHREGZve01oCUK9SxW9fb2BwOE
J4fVe7Kwt39OLn3DVelthlc+lw/Wn2kyXD3E+bDdevkyzL9Ojl2nW5cs3/vXyjdOXq9z853OqqnJu/JZaWfvBX
ndVlZtW3XjJXZXSYezm21yE+w7xkhIypZSedsJ5B5gcZxxUdBe4qSErO6s46/EsXl8Q9ZpM3iAJQv1M8FLwepA
Lsh1v2hJlrZJZUhb1Q/RBEUQS8/u1Uixb0O6wiaRhqWFpdUu4nz29wg8qQsJSekO94HLm0XBFPV3vsnDymjJaV
bkuXJFHPQBrLNTlqyyjwLsaJnrzmprqczIbt3WVHvniYPAHL7Q9sLdm+bHqHVXZ8BlbIAo/sW8+ua9Oozdqq61
uMUP59Dfu+sO7RXs+P90OF0O77tm69Z5daqEFpBg3LmB9bSCnlCJNa4CAicCG3eQA1pUXCO8mu8nz5CXCs97T6
F2mF01gYpttiERZq+RWb4zOmYASk2CKmS6ZkqYFk3YOgesmTiUsLU8RmjOEqDZmQH+KcS5URUXDEgQCvYf7JgI
BZtjDE3t6D6Mq/CkrokVh7IvL2ZcuQSSSXVqaraErvr4N7XQF7WRCSz2k1B43njNSjRHO6C/oqVEPZ/gLPDXwc
Jyeo1RL0VmjxEd8ko8KvnJfzLfo41lP8XEbfWxhYuOW6Bd8MO4DxRf3zfgu+fi4D9Qhi78sKka1tphBcMe4lAE
htISrnpgI4HInhDdXjcfs56TymgLVaKIOjSUzq6VNyl157Rev/vjHr7/2j8/m1A1u3hKvFsXq+JbNg3U58NN3b
pLl3/7mw//5u/7HR9avH3m8f+/nDtTUHPhcykb05s91p/EMOukPlSs6K2Ra4J4M6MqAQAYMayCLggZAz2mg3WQ
29Wg1Tq1WkwHtPMffac5wms0ZFNotVksPR50c1rvQLliFOwngHRJOmkFnLjSvNx8184ZPWPdZ6TprxEp11kxro
ZUzmjmrlTNzWiJ+T4TnRfiqCBdEOC1CgbhLnBa518R/Eemz4osi/bKKvkeEIRE6RIiIYBCzxWKR+4XIFHnwJ29
Gz4lPivSkCO0i5IuVYrPIiSJQEd4V4S0RXhPhSZGJ4u4ST4sUBRTj+RELcJqT4tdFjUGEp34l/i+RisqXvhJ9S
fwpgz73t1GccoTNt0ukBeI6pOTYhFfnHomqE/tPnYk6RdCK8AcRXhWBSfuuyHWL0MawIrJwEyrHgbEUR2hoJPq
cCPeLAOMiDDKut0R6RrwgPi1y42x3gyJVRHCKbBfqbEpwUyR6RgRFjImURyyteY/p6yXWc4rnRM4pTqmbe1XUq
AxVzsyoWwyIbSKn5RS5KBriGrjtHOfmIIMDTjFnRYuxzciwWAQT0fbZ9ZzVbGacdiSWzOVmxcw5zUDZV/VUS1a
00txspmYe3cFBGsL46Q30HT58OGALp6JpRSgcQhxDYQTCcDoxwVyXIfAzMdGHTh1IOzPSqAQB/LeHddM07BNYd
fg0XviuLbOmLhQKsQF8TmUJgVOBF/9ks7Yc+lb5DSDjF9TobYCwIQU5l7dULT+5fKFqufkIJd+BBhgphTuh/Ef
```

```
wTf437/+Ou/jhoCaDhfAPt3NPfLiHm0d4JZadx7iQRQ5c5Y1Ar938sRIyWKMOL4zBMeAADK3EIlgky4LlFcuiR
au3eHP6cqiSA7scQw7q4LIo06VgMEcxOwnWmN1qsMTM6WjVEA4FhO+Fe+HwBG50qSLUi5kogEFbLsRzvpJ9cAe
Z9VyYng9syFWUjeIXl5uOHgW7ITPW25vPfWd5VJ9hN95oYhGNk7JLjzjWBvMwpm3BmPZLjMM5uItPKVVDRdNF9
LweDPrTenqRh7M8mHnQ24ncmhkgAWjFRwnMBBYCnBSIqwAfUMNxoCzqjm7XgCYzluNyxERSGDMKMlZFO9T4FRa
+x4JYg5qDAmrQFV5H51iNYn4LyJKtsh6DlphZxtUxk7CtgRrH1q23lYG8Rkv/R/CB7uVj4f1PjIUnqygGlc9D8
9TyH5e9/ub4xtr9/pLR8IljEXk9/PzI8w+0mE2mwNpy63tZpe8/l10KL4/MdRdmCvSXesNraDe8yXMnNFFiJPV
KiWCOmWnMPG5Omq+beTKqCBogGkUT01zSJDUavUY3Y9QSzSSXpVol52VAYxzOWeo9jKlDgybw2zRV/jCdBPuNE
DiWfwunKmMs0/OR/qo31Xjag/M9rNlK1pBjSqc/DzTZn82merM9115r32rnT1uhhAenkxrGctasyckn+Uo+VfL
j+ZfyF/P5Sl+zj/rKscSj1JDpe9gOxC7Yqd1unjyVCZla6puG1NJ6l8Kh3l6bvQZVHcajksr1tprQ2nKCuYIdl
GKoqtdUVRawRGEF2VYPqHWX08LrfNzDH76U+OqpxCbf9Mm8jetCdrlp6yO7Xn8jEJs6Nz9Irzyy57P3Tc+c673
/QYPVYXwSqD37v/6n9ofuu/fkY7vRn3zLW7kkngUfKcfKLTESuitEtR44bjtno1obHDedM1HOBHotHoo1UUuFU
gGkYqaC1iAQqxivmKt4peJ6hSYFcNsrICDyua3EJ2DmfMXH632+3FieuyzmEEsKd/AGgcQ4K0uPYfWkLFWAekQ
qKnpVF1vJk2kfc6QPCZY2uFmWMwvzOA+oBwjUvKnV2fJQF+vOHwA9ddU0bSnoemhvuHL4b0fCh8PsJD+5rBylg
2sa92wsP1RQMhQ+/kluKLt0vd3jMtff87XpyeceiJhMZq8v17CcFQplcduG5naX2IQbNr3hp8wH7OgDv+QuEzd
8S1kz7XzUSTXuE266P+funM/lcJocyJT0GVGRvY44wH7t5uJVBG3YKhkIGDNmM6gBq2BDekTPRsoR4BkLx15uB
9FaNLocndNBMiwas8vsRsihRdjS4oATDnBcu/mvykMFxdEtGpjVAKfJQtgF7cjTZnZh5nZhTkeWVovGabFoXFt
yICfHiXIzULCGJ05CjcTjUfoHo7zH6aGp8nvQc9zzJFbjr3re8hgYPh+RDPUsIt/yvOcx1jBspWfKc07F6irx9
SoO8Grh7vGlCvdKa2Y0hoW74FE81LHXAhaLjpgFs2Tm9GaHK4ezxLQZbidvzBSsRKPjzDHOiKEyvJSJ1bs9swZ
b5gXhcOjwSv3OSqaA8P09vRV4LGrYg4PhU7fmBwj09R7GwCScWlhIPfoFffqNuSMQUFMUFo747e0lvaly0ZRyn
nQCcaca4O4aXd5170+W71v+L4egavn6GDx179deuQ/aDy7/sclVWpoJdyzPYyvAefgbjMju5XdBwNa1/FT67sM
PY90lkEXFqNG5dNTA6e3oCzNKeXwwqrWL9gJ7xL7LPmTXvmeHf7HDq3YYsk/bv2z/uv27dk2HHdg4fc0OKlNNY
0uUDdNme4edVtgBJDs4WQCBGqe90j5oP2d/1f6W/T27DlF03A6KPWYft3N2ZhZ/UVRtczxqq+Q6MqO0J2ZexJw
PGX1Yp/UZY3rQ67m40c4ifahXSJ1EDPSpQltN930s5acTNxrjB2vLMR8DXqIwuDvCXDqH0aM/BP29n8+rb7R/8
M94marV6HM+GLZoIXP5WOomlarFA/iyYyx1EA+5qOwkW0zGC8anjdyvjR8Y6XEjGLNbTc6Ak7Y5dzsvOD9w8qy
30fm083nnr51awanU1EWdXt7r9GLN5IU5L14UvZe8Se+Cl59DgHrTdwi1zXKrrSLgXUKz08rnxDxWZ3YscyVDs
xvFyk6F19UK/carveq1AtRAw65ULLXlUe6ja8Y9trwiUSzMs9nyCkWxKM9mZPeOExDgf3YrFqk+2LGSt9neCbu
HYM4W8L74sCKYZFS+NctKLVyeI4/msfJjg94UJVGdlClRQQoqQSDBmSCtEYJzQaoE49iZCyaDC8HFoE5SuwtBP
sfU+rMSKGH7dODdpcQS84s5RqNmh0ewxbBoV+MrS+MQCrCrcTrAYrzFxIL1yJ5eggZNXz/ScVTN4qgB3UqGUXV
RSJf9Lf0bs9avq7CXHAzPfurGmdMQAoyvpfdtW3i58uCXDpcPxHsK4PrQmV1+3mDW38jU63/Ml2WVLicda6uqs
uTAr94++sLJqMmebVX9IYo6Ocx9GzWyjjym+A4UgDszkEktYr1I7ZLJGvXYS+3UbIcMGwAPHAubHoMtinPqc43
rWrXVM9XQVw1KNSCwttVZyPTgNVqihYXbneAsKFgTiOXmknXhHUariPdJ15oYGkCtaTD3YHZlN2yWcbFYW2L3t
NXyhsUL9QUs77CapnDlll3PN0BVKvFgGq6qB4fOwrmcrN6BHymjsdIjy8sOazjat6G5tzorb93mzr7ysxZfdUn
5Xv+a6sYzrz2wcVd17sPNAxXct7M2DLTdOJFdusdaJGeVtO2rrd9dXyjqgX+kpKUiN8d15GWLazmPp46yWH3Sm
4VxxnnzbVrK30eyySeUGlqtt0V5LTzjhgU3NLi3u6nR0srFnHEndTp1hBM4ieP0HG+OGRSDJWrQmawu2w6iFnc
N4R8EllaiLh5/TL8Ta8t7Axq8ldrkqgYIu8Iu2eYU2c5cFg62xfvuuTfR8A//sLHcv9lrXbuxyTmxjz5SWvj3f
99x41hjk1HbaHRajal7aAhz5s+xppDJ158jejRciUGI8hhyrKxitgqtJtOvTVQ0+Z/0A+93+l/0v+Xna570v+e
nfmbFrEB5NOAHpx8W/HjM/TDun/HP+Tk2eNW7JqoSlTrEKMlpncEAQIlX8EoIvOJd9Gr1XikmW3MEkzdm8biyC
XG2a5jhwyvXG3Yg2E8IFWoIYDd0lm+E11khElAdANI1R7pUT9VdtxTu0Jx7R0dH/rqeRv/E8oF7d3Tm1tdW2Y8
t7z98GMxcXMjHc920v+3GoywK0K6dfVq9ib/h+CgmADmN/v8exsON5J+VJ5zefG+llzM5IAAbgeYAGCtzKunxE
JwsA1dZVRltLAE+y5lF9cUOMNrBZLKAWWsUjdSa582jQl6euWdtHamD6sW663WU1L1SR8vrFGy4oh4XGsVV7lJ
cMRevdw0VQUcRPBCE6SB0BAeD1B8EMQinBdgi3C1QsxAs4nN0fesA1jnkvhziBa83h8d0raaJMFammLJX73WqC
4XUA4SR9E1MJnt6WdmqlnDpKg7UUi6cmcex3zfw+Gir0tcGm1zIbgnqjx/a1WBzplF2NO+8M7h1fHN+3cCnT3x
6oK526qsHB660NcrBmdiWA5E1dQPHThwbqKuZ/M9H6j+5v8cHI1/LCvgcxdG9G6J7GkvLqncd6912rLs8x7b8q
yelEml9W6BxV10wtHH3A/HeRw/UmJ05GSlfbcBg9JTmCeKDZ5QMgzZbW6zl9Bihb8jMyfb88YPoGRkq5WZ5UOa
Oy6/Kb8nvyfy4DE5EdSCSZ68p+Vl1QGuS3TL9/nUZXlRJOZWXjXNPrvCm6BmoUacwJq9GVbaLatd8/kL0ggxT8
nGZqoi1p89Gn5aBsR2XObcMvAzvyfC8DEyOigrIFJEHGME5mVO55hLD0bYV2qfl52V6ToaAvJtROmXKMC/JHIP
ZNqZkzYYPZHgW10gvyZAvsw1PqeK0ggyUyCDJ5XJMnpHn5KS8KF+X9YIsYXdB5rMyMnJbuVTFP8Mq/lxfzOsiO
TEu22qPGfpYPWiAVQdi+VeNPBV4M2ZJqU/9rWDlt4PUaVz5ySCg/oig/oy2QqJi2Cl1yFXrVxzJteJIHmBxSz2
oP3/iicCOI5tLI7lrS4WCXDmYY3z//ZeW+TNc19rCpv1fPFRt0r98j9HkbRyMXOz48A++0lJfyifwsf16yy/Wn
++z1v6eeFN/z//vP/6P+z/6k/TyVt1p9Br2x36aRiGfzrfcQj6xSgQf+8u3i75NmjV/RzZyZ0mM1pBinpBtjF1
3llD6lNrfgs+z+PTg48PHzv8T2Yi0AYQJ8kURdiIc4ifJaeRB/yVB8gTkwjT8Bn7D3c8t8nv4R/H+e177lK5W9
```

```
w19s/5NQ4XhReODps+oK3KR2vSaKSbEELvl8iZtDeFUbC7sWl13fHUPgJTxNEyJjoynYY64ydE0zCPNZ9Kwhlj
IE2lYS6wkmYZ15G7yQhrWEyfUpGEDscAdadiEa9i9+j9bymBFfgYZgy+lYQupp06cHXgD9hZoexoGInH2NEyJh
atIwxxZxylpmEea6TSsIbnco2lYS/K4K2lYR97jXknDelLEv5iGDSSXfzsNm0i1Rp+GzeROzYr8DPKm5mIatpB
Pae/eNDZ+18TIvuEpqWigWKooL18vtScGpWj/VFDaPDpQJjUePCipBJPSRGIyMTGdGCyTtm5uamlv7Ni8fZs0M
in1S1MT/YOJQ/0TB6Sxodv5t47sTUz0T42MjUo7ExMjQ+2JfUcO9k80Tg4kRgcTE1Kp9HGKj/d3JSYmWWdtWfn
6ssqPRj9O/FcWgqvfNzI5lZhA5Mio1Fm2s0yK9U8lRqek/tFBqWOVcfvQ0MhAQkUOJCam+pF4bGoYl7r/yMTI5
ODIAJttsmx1B5vGJsbH0kuaSkwnpDv6p6YSk2Ojw1NT4xtCoaNHj5b1p4kHkLZsYOxQ6C+NTd01nhhMTI7sG8W
dlw1PHTq4FRc0OokLP6LOiKu5VWuRsVE0zsEUTVCaTCQkJn4S5Q8lBnFp4xNj+xMDU2VjE/tCR0cOjIRS8kZG9
4U+EsOkpOf593GTTWQMz+BdZIKMkH1kmEwRiRSRAVKMbQUpx3/rEWonCTKIbZT0I0UQoc1kFKnKEGL/w+Ygth9
JmFR7CWwT2E6rvIxyK3I1kRaU1kg6EN5OtiF2RKXvx2cKqfuRNkEOYTtBDiBujAz9xfm3Iv9edR42MoL0ozi6U
8WMIC/j3EeO4AqZxEacawAxo+osE0hZqq7rL8v4a+O7VGhydWQtrovprYxU/knevyb536eRlO73qVKmVNkpyhF
VdidS7FSpYion08WUOtuoStXxJ2bcjjMOIT/T3EeUA6rsKeynJI8hPJzW6n7U+IS6gkGVb2Vvkzjzv7UB88EJ9
MKxj2mJrW5anfMOFT+l+hQbG1Z742QDZp0Q5g32rwxpbpc8kJZbpkKHkPL/lG8KT8i4qseEaud9SJuyeZkq8xD
619a0hkZVv2caOnLLHlO6+XO+FlHb1Mk5eJscZlnWMt6V1U+m1z+kzpPS2ji+x1DvCVXbZSp2n7rHEbThCEK3r
o9ZbF8a9/HVrKzl9v38v5ybSxURN30445/4zBvi3wIdq8DV9wvAK92weAN+eAOkG3DsA4h9ADPvzr1Lf3u92Pv
M9Reu0+3v9L3zzDtc+TtgfQf0ZElYii3Fl8aXLi1pjda3wUx+DbZ/Wqz2/iz8Rueb4dc7yRtQG3tj5o3kGxwri
Xve0JsibwDX+ToneoUFaaF8YXxhZuGVhcWF6wv6mW/NfYv+t2+GvNZver9JvVe3Xz12lYt/Baxf8X6Fxi7EL9C
5i2C96L0Yush97vEy7+Oted7HHi30Lj56/VH1j0FVj2bYIn2fhWOfefgzdPzkzMm5k9zMibkT9JnpF6bpZKzYO
zYa8I62lnizw1mdujDXqeVuqj9SNe/1F0XifYq3D4l295R7e1qLvY6wvVODi+WR0Mp51b/6jXEPcy9wOn17LM+
7A5/F2PUYtW73bg9tV38f6W/zoaAt41tmtnCbI8XeaGu119rqbQ21/rD1Z63vtGr7WuEL+I08E3khwimR4lBEi
eT5IrlRd6cYdnUKYWsnBdIJYdIZst60Uqu1z3rMyn6wJ3RGBA1cg7n5jp2BQNs13c32tqQ+tjsJDyb9O9lb2dG
T1D6YJJ09u7vmAf5D94mzZ0mTpy1ZsbMrGfd0tyUHEVAYMIOA4JkXSVP35OQU+6k2AIEAgkfwTQJHELVnMoUkg
ZVhEpiEyUkyOQkBNqaCiCGTAYZmGMYDyLlnkrAXGw2oVAyanMza878BOUzSmgplbmRzdHJlYW0KZW5kb2JqCgo
4IDAgb2JqCjc3NzEKZW5kb2JqCgo5IDAgb2JqCjw8L1R5cGUvRm9udERlc2NyaXB0b3IvRm9udE5hbWUvRkFBQ
UFBK0xpYmVyYXRpb25TZXJpZgovRmxhZ3MgNAovRm9udEJCb3hbLTE3NiAtMzAzIDEwMDUgOTgxXS9JdGFsaWN
BbmdsZSAwCi9Bc2NlbnQgODkxCi9EZXNjZW50IC0yMTYKL0NhcEhlaWdodCA5ODEKL1N0ZW1WIDgwCi9Gb250R
mlsZTIgNyAwIFIKPj4KZW5kb2JqCgoxMCAwIG9iago8PC9MZW5ndGggMzE1L0ZpbHRlci9GbGF0ZURlY29kZT4
+CnN0cmVhbQp4nF2STW6DMBCF95zCy3QRYZMQEgkhpSRILPqj0h4A7CG1VIxlyILb1zOmrdQF6BvNm/HjmbisL
7XRc/zqRtnAzHptlINpvDsJrIObNpFImNJyXit6y6G1Uexnm2WaYahNP+Z5FL/53jS7hW3OauzgIYpfnAKnzY1
tPsrG183d2i8YwMyMR0XBFPR+z1Nrn9sBYpra1sq39bxs/cif4H2xwBKqRbAiRwWTbSW41twgyjkvWF5VRQRG/
eslPIx0vfxsnZcKL+V8fyw8J8SHFHkXuETeB66QU+KEIx+I0ww5I84S5GPQ75FPxDuBfA5Mmsegpz1lYPJwCX7
o3GvYc0KuAuOs4IFxpwj+MzxLrP6vyMF/ukNe/R8okPXLMRq8u5/Imbw75+OmC6acMWFtgP3+BHa0OEbPN3Qjm
gkKZW5kc3RyZWFtCmVuZG9iagoKMTEgMCBvYmoKPDwvVHlwZS9Gb250L1N1YnR5cGUvVHJ1ZVR5cGUvQmFzZUZ
vbnQvRkFBQUFBK0xpYmVyYXRpb25TZXJpZgovRmlyc3RDaGFyIDAKL0xhc3RDaGFyIDIwCi9XaWR0aHNbMzY1I
DcyMiA0NDMgMjc3IDUwMCAyNTAgOTQzIDMzMyA1MDAgNTAwIDUwMCA1MDAgNTAwIDYxMCAyNzcgNTAwCjQ0MyA
yNzcgNTAwIDU1NiAzMzMgXQovRm9udERlc2NyaXB0b3IgOSAwIFIKL1RvVW5pY29kZSAxMCAwIFIKPj4KZW5kb
2JqCgoxMiAwIG9iago8PC9MZW5ndGggMTMgMCBSL0ZpbHRlci9GbGF0ZURlY29kZS9MZW5ndGgxIDEzOTkyPj4
Kc3RyZWFtCnic5Xt5lGRXed/d71uqXq3vveq1qqum603P0lt1Vc2MRt3SjGa0gxCSLCAgekYtzcAsYhaEIITNY
kfIZjGWg0jYsTFgGwIBTCDYgBEkQMD2AXwi4pjNHLE6BiR15/fdV9IBzDn2Sc7JP+nuqXrLfffd5ft+3+/33Tv
nzpzfYD57PpOsfPTk+m0Lc7Mtxti/ZYxXbz1xxy0/ufxFn8bx+xibv//YxvrNb/j3r1libPFulBkcw4XG1vcFz
j+D823HTp571o/1dBXn38P5JSdOH10/cIo9ibGlRbp/cv1Zt+0TV+Jw6Tp8tE6tn9w496wv0/1zjOnKbafPnlt
mH9pC1S+l+/jH8Us/BRwaOhdSaWM9PwgLxahUZv/f/RQZ29oSr5M+22BPY6fYWfZM9mzM30vYK9nd7A3sXvZmD
FPELf6MnefZPF/lQzoX09xYMy+yeTEY4hruTfN0uLIq6GxV0hn9Sfq0q7yHq4N5MexNSxu5i3XLB2bYsYN6RjX
HwyytdbM4TdJ53k+7y1nSTYedum1yKoHHO3gvyqeDYQ1XxKsqs4VCUgxUOp+OV3hR737i7sAKwaWpT0+MRd3Zg
meFEqZorA2ltr6SVtpGVRYL2qtuqxbjehxpXy/vuG7RFoyUWkq5dHguDIU1QiZjodi83QhlPCHeL7goezIU6kK
jOW8o7uHvISn0E4SIjJae0GLMk3jOoAkDabzDz71EcK9oC/sOXTqG13hZb99gviiFF+rCvsc8cVd9e92reBrly
/Wo6I8tNiZ2715u6aIxodFRKVJ+1b9fm9nLZmvddqvoTyxcsafQmmkV5VjRePxyvP1mJdQNwsjwRult3i2M91a
ptdWJlj6Xn+fCaCXVKSXQcTmBDgqYvFAsxrw/xH/Kv8BugEcdYcfZbewO9jz2YvYK9tvsd9mb2FvY77M/Yh9gH
2IfZZ9lX2BfYWy2HgnMXUZm4D6TFBNNc91PMO2DZLhMV/gwn/pI2kWL7yQ3idgkzorqmGy9yjMzHKzQ9A5XMOm
DZbIXVGnN0qLt9QftzNAdm5uVpaJZN0vqTU5VdvE06lp1J1Rlz5mmK7qynNk0kzAeOkX7cD1yDZrnw6xtOvxzp
WQiaKxdce2uoOpXC0r61khP+SYs1isqtFyXa6ViGCbheFaWXHBdMKau4sU9B+cExnH52sNrrWiqpmFsRQy3Vpv
```

```
/WmNUz3MuxKtQnifaN9rTAsMNKxHK8lhyDROyVERJzvUMZkmFQpCByga+xnGRozBXnGPyxLzVZTyBsjxsTifSV
IvlnUfo/eY+qubd5cm0ZquzDVXxil5hfGqmJpQWwnR6O7LURFEEw/KLkVTSRAGHeepQl8aqc4f3LycKXSrVdvR
rfqp1WAw9rWGKUk0KoU0bZaX5NIylwIc6ULDGQ9IYsQbT4eIo+oeOXQUrLxpYoBCzaCKs7APoiuDzirqEXvJDc
Ce4Q8zFPmE2PyGcPXIbj01Gflk0Lvwunpb6Fk6YvBf2uMmb4n5mgc5/zf6G/S37DvsB+wn7OdvEuPi8wqd5h2c
ck8jX+CF+BX8sv54/hR/ht/AT/Bn83/AX8ZfzV/PX8zfyt/P38D/mH+Yf45/k9/Gv8K/xb/Bv8x/wTWFEKKoiE
ZOiLbaLedEXe4BvK4Nhr25iCStNarCrJM0Rj/4k2Z6tmymepINev7u0mCVDlI/zm10YWcTj5YTwLbZdmxh7AWF
kD/cHKNwxFl4xydNON3NVE4r1BkM8nybLae4UhJz0DF2EubZtAk8zXbrlrJc+eyjU6zv8paL9HIf7+SnMvJcmd
Wo3NUQmabc+7K4AQocdONcaBi9d6fYtYWruaMlgja8MM/gOILWXLQ96QFyTrfLlZNjuZg5rcaVOTwD6exiFrI5
qUSJzz6fdFTggudZ+XJHJ0mKT2taJaQRTaoVNuk0U7NboMiqqp9koUpglNMJI8uQUV9CybGXQ7xAiwEHTbpvwH
mOT0fA20Xg0dDkbJkMzdK+0PQSL3rDfsTE93+52suGKG9562pO2s7KGwe2ihTQ8ie1Q9EhM3M2/LXVimNTtIj0
SpzVc6tAcxO1Mpl24GpWrubLDQb8HJMK/Tt+2484AVZuMj8Hi+evIzO+CB3AlCnA8EVUCLbUXcG44XB9eYsl0l
aeUJu83cGYJ9wEoFxBFlEIxeIjwDD3gWQln4kYqbgSeAlgAXJw34bEQHwI1aWEKCEkKJ9IvKFuiwIPoAt/3atJ
XJSsCpUJdb9ccbknl160Uyh8bTxEBne8SMH0ffupzAjeLs20EOwlhEzm0kJK7MMYjRf3CDwqOSxAlraucLq3ik
zoe0Big9wodk9RnuDSXysNXSaMOvBBDFaMUHYm6Ky5odLgk+kUhyXIrNL1SKKWm0Bo+ASzUVNLS41wg1qIo6pS
G02u0UUbHVJLAFnja1ErTMX4sITaacSNGU9JoU4+55e7VUlnqXAFwjzaiTdSwQ+4OlTLX4UPKgjRykQbCLtN8G
WA7P0d134WCH8Tgo+zDn1ZW++KwDrQf2lBHU1FyYaM4NdkgzoE5l1bRRKKnRZ8jkINbWMuF73uex6XVpoR21CJ
btP4EYgVmnSYSRmE9rmAKgG2ECAxERVgvREMVjEYExApgMdOtSYXKEQ3QeAQf2wz92Kja0kovvuat18DFE2GUq
s3VPFgE7AdlMGNAcC0ffhW6CxCl2Z/HtNxJwyj4+zEnZzG1Wj4GQ9jKh6RqDsISZc+4GUTXv40hmEaF5zEfXF1
qiFFgDv+Kghum8Vp6CV80FDsOS/Fuah1/CoU5zMYtOPMxOJvfgtmpEG8xO7gcTQOqq9EcP9VNH5Bd8AM05u8mb
+KrznA02ayLnw3f45e5OCWvkvo3cXNI3QG/ULmBbX6U7LZGvskjFPTNMzEvEgNQhq3pFLHbgyeJqntM8BUe5yY
kNt9GVvOjWoCwuvnv4Fqy+HfS8+RQemSUUuxzIYvNbW1t/SP/mhiwC9jF7FL2GHCpJ7J1doydYM9gt7PnOhb92
+x3wKbeCjb1fvaf2WfYf2VfQnz7GruffYt9j/0Y0c1DABnnLd7lO/ki2rGfH2BsOE/EathdFYTXFxB+E+pP8zz
smCSlEEGY5eKUoz8cgD8NbOw7HjVIQYoocBAmZkOQou4CRRpHo9Ke7aRDh6RxTqSS5V7XsWyAP06ArfUOBZgL+
LCbZGlm4h4gesVSEKSokaS2nVHcQ+SQCAJ4YpDYYT8bcTabhyI57FBQGgwtETjEQ7o+HCwjkKRDuwguD9iHHlg
VeG0G4K8jZqxxHExzVGtRuwsZwOwEjTZLKV7xCBvsOomA19WTOHVBNNtbaFQ0jMCv1sXE3O5KFAZyFt43d8mUi
oMgKehKtSxNmfycwFBUbjpUyWanfWn53WRPnRpmvE53OAexgamJaYCFr2HSyvPosjC+VbBDThamVKDIxj7ijNE
ZnxC3OhveTx+JIFsXElVqUxbGoZRIJPk7fMEI66MAqiGk0CrRuqgs8LuvREk4wkUO4gHZJ6nJF5XHKj6f7JSEt
y5cY/AoYIBiC9m3yK2Z0BJER6gazkLqTBX3I3gLYERbYOQwShrNRn15ftZWVhabUtYMosv0xduTfr9X88Kg6Jd
roY3jqgIzFGtPjiS3IIoV/jqCUDNPL5rRAsj19xgWIFgHrX2sgZ4W+s/RKCNv0/x9aM97iBeK/fg44zqvIPvEf
RiuLrzJCOOQ9w9x+zPUAfUPIqBCpmBFpIoUHszzFQH0c50+4xv6adoqkFZ7qxZlwh79JMF9sReYJnZBZO1p8G3
91fH3Y3oVYUWRIglgCN+YWs6fiBeqb+CWuBo3mjSKhG67aKDxd4OmcAn/niYnFw9AGxsWsNeO1PA72B+w90IV/
Sf2Z+y/sb9kX2X/g32bPcD+F3uYMx7wGL68Df7Ygx9fxC+DaoJddub5bL+ekheDa+TMMldIzldyghdDYOeyCp7
dNtY9J0dSi1R2MiA5baYJEiLQInDAR2S2092QzQP4+aDvRNmwl6ROi2n47zCBg2ZwOvgeKBrIT2rJrd1b8GCai
6cOEbw052gRB93rDSCjYhLdI4G/lnO8lJR7zmWJKo20Gzz6kWdtjSRe3TrKO8CFbFW6MiQOM1ekNqB6iQ12qBE
4Bgsl0mrEcYezfPMawX9SnrG6JAt+VNwGKaICq2wAJROoMI4ai2NczeyYqHralsx4twvti7ipZG1mMkWgxUwX0
jLRKl0fj72kMVbQ3PjCVY9g2qiHcvnQtG75OoiK0PBA+AnhBVbLsJFWFfQTImzaffjl0okbEtouOgnobT0lSbU
pKPgg1DWEJlRZVcbYMNQac0tBD/gxkLBis1yABtcyuAOPe+AzwhOHA6Jbnj4qKQcgykYf4i6aglhdEFTBfTSam
9ZAWiD3Dfih8uMqBSQ0ivMmUZiecjSMgEfqAF0Hwfk4op3yvuoBbA5WZsfrE4XsYKf12Dbe5o1PTRXwhnC4d0F
5ZU97YTS9vT/b2B2NHbr8EtBauqi9cqW0c7mXXP/B68uD4Tz4io5q5VJQ3NbtRs3V/Xsa0sRrhw40xHjz8O8/Q
Sbt5gyQC+I1BOPUk5120L3yisu3cfipFs1vobt/hC6aTyiubxDGe6rh6qm4+CDAz4NULvcDxGQQJlUs7qF4j76
ONyd13HxhGE6H8mUgxxgDfZcpmMYrJDfmneT4NCVgWxiNa415AZ4HcBgVbn5WhSHmAmc6VAXlXaKISPlCNP4CI
1V2xIffpIz3KQwlJWzOC/M4kt+CsoJVaNEH+Zf459ll7Cp2LbuRPZkdZbciqt/OnsNeyF7M7mKvARq8kb2FvdN
F6owczhk+PpeHiJ7k3E7mUdZrWsAN4MoUuB9xebL77g7EfMQ/eqq9wEnruDiL4E1+ZEkRkSwiH0UcxBMdOzCZJ
F1Xp5yaC6YQmUPrtJBBmM0ooNIDiLjkX+OUkPIxxkZ7kUc9B8n0qvVK2C5QpLImnh2fu3KpGHtVL6zUkvL4fDY
dQmcUECY9EidK+/V6jT/DmD7lpcx2RYqEBvgEUXhxvRAHOY8JPB3XJtJFyIu5wMMVipeiAPN1IN6DdvFJz/DJU
qjjCeuDEFfmdi025q7ZEcZhrjaiWhHTbayXGD9ZvKiFqqam6yrctefCidrC/HaoJ4EoCY+DILhMqBLFwbokz0S
8Vc8jeSG+pcQ3UNcppzW0IFLLt0OcYAQuwOcamY85r8VrOYLlzzjVhlarn/PAZ9Wtza2fwyzuZxeyQ+wKdg27n
j2dnWbn2bPZb7KXOgu4B/GAsmTvY4wDl4Haj4C7dIwMHyXuMqQc9oESlCnLEVuPaJnskLwlLDVpRgC46vCeKFI
WQR4PB/UFThre1Aith4Nehu/Y3chqOdGjRJftgTIlOCb4pmknQAXXImHb5z+Mu60JX9swICeoL7e1MYFfGkuSY
nGseN3E3v6OoOFXs97eiaTXW6hpgCymqQwZ5Fmz+cmcZyCuCzlF4k7cS6TjlZV9Ky1fjk1PGRCEgpjeOeYVZj7
kWDpc8wISeRc69YqBBxW5k55XbvRzhh5wC8SAQE1IW4AR0M99sJlaUphptyOv4j/2nkNhc3rMBqT5gnqwzStOb
```

```
itve9L6TY/bF4alqBpZf7I5FdRazU6sIvlTTbpOb74OtlajBniY0TV/176LJk1jvC4o66XkzgsP7d/+EejNP6A
im88h4jZGsoXk5gtwna8Tl6jQ9XXgK5fPKZqgh8ByF2kioBSMhbGU2a1NRPwvsgXWZ/vY1Q4p1tnNQIozQIrns
ZeD+9/L3ua4/4fYR9gnwP+/CsSgFETE2xlxfE6kN3LZS/zlgLHiPpPUsegMs79s046xvRRokBE2pHUTgxHHefK
HonAKSu1SKKDEFgHe1F2yKHZIU0PQX+n2OyU+6K6RVrAgzMmyyzm1u/Q4TmoDSrxn4OYEZGmHEmKDOuh2d5CZG
mrLEsropF3K/GTUmjz49/qOUeCk/ZDy66Zcq9oqHJmTpSgRlmumYgPgSbHSqWgvqHqtPXXEAC2/4BKFVeCvl1C
aBPwMAwugqhFAUCqBcxx2gCtl/CoH25iUIwEBjHcdntYOSmBNZYhksFvSw4iv2iCIGhLfcPwiqXHS15CcXASUq
SDCTkKSy4p2BgtT9OlJfmZuWKUrYXdxMFGYAVRBaIaBH7WqyqYLDa+kUTkvhDuOLTavej2eLgvedZkKYuBkRZQ
UEa/L45PmJ9HFCbzx03qVeifU9rxnCXepEe9LuP+yPCuCXj0j4E5lc77541wTO8zmR6jsb5CW1kBfs/kpz3uj1
qkiQvzHmlTtPkJ2dCsogtmEGnFsG7Nsij3Av8B2sxVo1EOIZb/B/hU7AoV6ip1jd7DnI5bdDWZ7j4tl72F/wj4
IXvsp9jkw26+zv2XfZT9kP0WDAl4FrwUfHuZKU1LE6jrGmS8IAd+WKcpJ64yW0G1pkYwU0EYSDiZu2lbmhkbpy
H7m0qouo2gdjI3SknmaftDGm2DydTvsDsi+02wncU9U5TKTvSxZGTFS0pX5QZrnOLsIpRQJKfxl0KdwnzqFTxT
pu4IZ2oVnY5dGRCuku9rklIVMHuG5I79yuhTfVH1+g67bmOjqyHuM/fvS7uXFKsKRoVwcBGEggFW+FZ5RtST1Z
/ozYWDqcV3vnuTloSOQZLVFaWSuF19KJuh+fRFQtCIroFOXW5TcJeUoe+hyIQgrhKviJRKSUsbOWXJl5VKGUoR
aFp2oDCQpHh1oSwi7AwSS6CunNQmX/aMlL+05+cNdVu6AVk6Gcl/JCGYECyyRF8FNVdGK0CNWJTNBAli7zBOIA
AcjdTlKdXESmXpB0dIV3hJNT02GIICFhd586IV4ptk/0K53Ou2arJUm3eJH0XVbFl3qjr/MEVv4HJcbZPd7YeA
mdYr5RtewZbci0qTwfiOunoRba/VCXeC7XVwfx/Wh830llzTht0/1OdcKaKDv5fJJ5HhdGv8TkvjLZA40XUoXo
tFbDIQSQHI1dcjomESnll9041MNkyQOgyRpFPZpErIfRkltxZvkEk2o5Gxia2vzo/w7iAnPA094BXu14wlvBk9
8L/sA+4/sk+w+9gX2ZfbX7L+z/wn/eoBtYkaLgLkx3uYZgoojD5awtZOtwPZi53Lwgm6+jIYT0nqJcaa5SrqKG
GYK9HYW3c/X39qOQqKCHmlGuFTiyImrnLzXCb+uW2XgdQvvcCn9nluMTdqphY8QBV3lPVq2oKWF7vKQvHUFEYZ
cIU1W8ny/k7L5atwwk04doro4Ba/FUzkZGfZlngTqrWQu24M2d/FKuQPeuJInotA5uUwEBj3I2hbRbJhTaAprb
4vqpaJVkF9E+2idrBiFBZpSbQplQ7rHVyD0kPseEBCyQsqJmbqv07FyMewc7JIcAJkBiYG8MQWbZ/2kpqrKIai
DLtanaqV0grLhBQgZiMOx1Pen+ATCxCRlYjSgnqqhZ6T4HMhLYLiYJdCFxhC8GPoQcyFaRLlWDUuWJzSJQyN3k
9iEtRtdIS818BvbFJQ8OSHE9dFssy4KWbdlbAEC9sWOmBphlCf1MyhlKSU82kQ5uxb4jtK9q2uT6IJfbpUlqoU
vSQ/9s/W9q/uTykwFw2SbM+MABBFMTE6G1VrZ7t67NsYL8XihIiFGSyD24NhTIXQpuG8F6rVaK/FQ6V3d+kJta
q45XiqMucyScMGThLPw+DJI2wxASZtX4caT3SIn/YyrPEMLFaVLlBHS3m20SiGWceUKrTwX1hZJ3KIYOHBBf13
oq7mmHL9Sl9PaCD+iUAFolil66fRs3fcL1Uo0CaTafBB4pvXLTWSVF2wewiP2MloxeLwOjYVpXIp2MDa59eDWN
6HdHoJiO44o9yz2XPYC8LB8LwOx9fey/8A+zD7GPs/+kn2NfYP9HbzwB+zH7GfsIfJExnRO2dtkn4B+OWL0iEU
dl7bIHg18FAjJ9WDzFOt6cLR4lNuBHKBkTewyO5R0TV3OkrwDXkG23l/M3LqYJWbVzYxbl8vg1m51fOjckdjWc
tqhZbTcvyztirBmCX6SlXhviJopIZPly4aJW3Uj5yYpCQKZ1YcdtGONUw2IyG4ZsAsaN8wQuJLUZjEhSAexOU7
z1sT83TM7m42SgvjSNK6qGNEikFCTM9OYKQsTDQLPm15YKYWNkBakypWS6Ax2duud3XEb7jLWnDJhozC1NF6q0
soyuF+lHFKkoSXwgHA8Ihtockl+1qGAVDd8yrEjWu12q+CbHxypyRlaEgM/w1zP0wKQFYsUFEO3dEScsKo8E/L
QbcHwSN9JO6dxBjtV8FjRCyWtq1B4k2qXA/JVyrfSYhvCB4cOhreoXZJWxFLZXBzUSQt4Vd8Lg8DCTs10qwFWJ
nEO9+8eXtvTKE1TXDStbTNWx+3txYlrdpDxF9I4CYxf9Sf3N3mlMYb26PLsjl0Ngi0irzcAoTTR3sd7tDUDTk6
MVixZsdvTysX1vZYW36bz5Sx1FbyHVnz47QrcVtsewvoC8d+itU/WRhS9MRsFFEZpX8oOQWIW5cANlNhG/reKv
pyVfMnxyEukfBdKnFL2IHeEWmze4xbFwBfjrQe2HuIPgy9eCab4hBFPPAsl83x2J2LZb0HNEE98F/tDp2c+xv6
csh88T35QACKzrz+yHrFKH5FI0jwb+UjGMxc1tFAbqe4sBZLBMnBrOFpx7dYhLWI428DlEhFEEB/c0i6dkQNlV
E+fUv6Dvst/dCWq6HdscgElSFKQvAV6NYJZYnJDX3aszq089OPBkLjoYK6p24vw292xA5AiLaZ55QLkZkMWfOG
bdNovxY2Z8bQ4tu+iAy3iAS5fglGDOo4BV978gWtuuJEfR/TB0/cBkTDCoSUMpCUqQcpD6ElH3mhPDjEKn1KRX
PRcImYWLiEmgOKwLFoYKAJrjd2uXQAQlOaCsUBwyFjRYgWfGLv82uuu3FcqwM41oqFXjCu1XTs6PohmpcSrSV0
W9x++JOZRrREmBw70ffSN6EshKuHDejWPcvwHL1hM7VdhJT6/Rwe0asGvEflSh7zDmQnJKtEHB5UlB+9cHCCWd
if15Plu0SFwpGrzfqV0V0A7kbW+mvjrnSh3E0zrFlzqwsl8se62tDDWZGzry/y74v7R7rLXw5be7rLpf8I+zj7
L/gv7iuNFlE//EftHMCMJIlgCN5rg2/gc7/E9/GJ+mLItox1pGYRo23Ty9DmtSJnuSEcDo7NIlrjboQHoBeo54
dB5dDtF37qNAI5CdVYGQOERwXp0Q4XL6Siyu0jGTpA7/LaQHtkQatqC7KRZbtfzDsk7Jl6jbI6FfqGo4HZ8NHl
3kLq8Nm02IMGfkqJ2+y6cxbezGuhSXgl0Ssf1BcCdKxxKpbvcT3dlASKlQytv9Mywu0QSHceojNyJpIxzI+gd0
jHdpcUF2tniGjGkpTL+ru6ErPs2KEtnfqJAbIaAopIVRZlW4KUNrJzqTibQJo5bS7J2CASAD+VZdS20oUeP2wC
GYdQ4UNYzpaqG0KjOTNGep3KceIXpydRA81cqYFYQ32HoVWeTdHCOjGcmX0hT39PGgqYo2uWgp5QoaWF7tPqPQ
mW3cg7lIsDYKrn2F1P4LLp1u5CYvZVubcy5CyjWaB1ZVHxpdM76LyTQvw56QjXQBcrGwx5Re0nmq8igcm4hC7e
4nIBnotMEvqGTBC+KVq9+3MzESs1lCDzQSgJvxCbfKyXFYhJ6szvmivHicG12u7AeGirdQjdiR9mzRUgDlIQ8o
LiH+oLYhzIUxfFG4peny1fedVFltlRdWtoZ6bBcL/jZ8nA58yjMVKtV368a7b8FbVtDr66miPHwzyhPAW9Flco
uyMzkronAM5nvuaLBMqT1RL4Jqwe5VqAMm1XdAv9TF1E2v2XtlW5p//v5UuEefD41X/vavIEk010Uduw3SWiV0
Fc39NaCO9PrHkfPPJ6KE1ogGCUerYwa4zVIyNGSWeL2cn1f3ADm9RLEjdey33Mr4O903OtP2afcfsK/Yn/Dvun
```

```
WzX7GtrgPr85zXm7Nt8lHuXK3q8o5e76XFJdWnEHTTkK30ux2pXadViDHJ/I2SsULVzBPVriE+CAZPqLvnZpKE
GNcoKBsalrvEkeb5/GAGFrPpV7JS2sgSqOEPXlXVjdu8Qtf06NdrzVoHdvJo5KTTVl9aTHtjjIJA9fCJF0e0F1
KREiQQCJiy1m+T6zEM0C/1eNTRRHphSt3cF5vTo6XyuNJDdbje8rCUIOxxJ+YHDPCLxSAzFqrSiEMOyFQUlsTk
palTYikT+r1sts2U9CU4lJe0a92psdcYIIv2mi8TB4CGtYqa10ZK0GcfxXMytgLhDjkIg0PwYqIM5AktgXpF6R
jK7QPBRJL0y6fGVhiSHm7Y9bqijZejawikzBIKBaKJwHttcED0y641CQPXW7hHnguLb+hsZfSYleh1D3YIQmnG
pNTgR8W61UNziY9NdHb0bS07GZkFIXcgyVHlkeK6Bxt5oC1G/AoUvxw3KhWU7boeVOene/toN0/cbEwOZtvLgk
1ZFKEUCdpPxa4ww4fMuzrFLJCNPHVhrsYxg+Hvme9q3D9x5TG3nyKlQ3u1oZO0JdQvycJDluk4zyFkH9e2D0kO
1u5O70dcqxNIZSyCQdpKTkQe6j76uHvqHcQps171SCoeoqcxd/6+dbD4jj/IptnffCux41Wm57GbmMMgFk3iGG
inS+YunTxKDUcW0rtmmlFcUYjNAH1h3mBTr69Ot/5Os3jfMcc7DCDHE8yBCSnD1La15HE4smeqhnhzV94qFmZi
KauuvrS5XqxeuRQpV1pHyiVwoKHtvvlRmXu4Pjm7bSdmAPtKNtFu2LnWhUpC43iRBy0HexQrsiIMoxoD+Ws1Ju
NJdJeo/2vXzdiop3Mpp16aOpea1q09t8wfMyzKAPDJ2qGKLqu7V1bW5kR6XL3Mw52t/xareJBnlZ39tey1lRod
/b3PJFEJm9i2mFK7yRbeIyktGkzT+beQJyZKK3b749/9Q9c/2c3lfb/A/Plt+nyF3/3pu89sgV+64HNj7pVfsw
GE6OLnFb+t8wv7JTnv7JzvsLvRhn6nw2vxhntkv+drS137Uc4fge+JVAwYhba0vIf4N79Ww/yF209wL+J+3+Bf
w9u/dzV9JC4RN9r77J3+S8PXhneW3ggemf5WNXUBvGO5OPpbe7NFbGX9sj+mnbQzxQ/8uj13qNlOPrTGx0LtGP
/6FiyBjs4OlYo86TRscHxzaNjy7axk6NjD2e/NTr2cfyG0XGA2t80Og7B7r706P+uaLIfjo6LaFt5dByxJt89O
q4yn1+FlsCycHYLv350zFmVf2R0LFjEPz86lmyRf210rFhVhKNjg+Op0bFll4mF0bHHInH76NjH8Z2j44DdIl4
zOg7ZqvjZ6LjA4C6j4yLbJ8+PjiMM13sPnr7tjjPHbz12rrX96FxreXGp3zpyR+vomfPPPn5qY7518YkTLXf7b
OvMxtmNM8/cuHn+6LHTR0+f2Di3fu3GredPrJ8ZFd7XevTGPlfRo6f/9KB1/GxrvXXuzPrNGyfXzzy9dfqWR18
5+h59/R8079i5c7ed3bewcPvtt88f2Ti2furoxvypjXMLo4f+mdvXbJw5e/rU+onW9Wc3Wt3WwdMnT26cOXo8v
/DPPPto92CCp91/OjjDjrNbITXPsRbbDuibw/cyW2RLAMQWROgd+DyKUrT0SimdDUBli13MTuC39QtPn3VnG/j
ewPcz8XkzSh7FvdP4PI3SGyi1Dmm7gSfO43wd5X655n3uXb/6xL5faNE/vfsvudJC/dQ+OjqHN9Ii4QYcjFrwd
Fw7zW75Nb385fNfPvt/M3p05xzecxZjsIDf293vPOrdwL111HfU1XjK9XPhV970f/f0Na4lZ9HPUyhLrb3eta7
FuvhH/T+J3w03h8d/qYTDuBzbP4DB+DU//xvo2dLfCmVuZHN0cmVhbQplbmRvYmoKCjEzIDAgb2JqCjg5MjQKZ
W5kb2JqCgoxNCAwIG9iago8PC9UeXBlL0ZvbnREZXNjcmlwdG9yL0ZvbnROYW1lL0dBQUFBQStjaG9jb2xldGE
KL0ZsYWdzIDQKL0ZvbnRCQm94Wy0xNDkgLTQ5MiAxMTMzIDc4OV0vSXRhbGljQW5nbGUgMAovQXNjZW50IDc1M
AovRGVzY2VudCAtMjUwCi9DYXBIZWlnaHQgNzg5Ci9TdGVtViA4MAovRm9udEZpbGUyIDEyIDAgUgo+PgplbmR
vYmoKCjE1IDAgb2JqCjw8L0xlbmd0aCAyOTYvRmlsdGVyL0ZsYXRlRGVjb2RlPj4Kc3RyZWFtCnicXZHLbsQgD
EX3fAXL6WIUyDwrRZGmmUbKog817Qck4EyRGoIIs8jfF5tpK3UBOsb3WtjOqubcWBOyVz+pFgIfjNUe5unqFfA
eLsYymXNtVLhFdKuxcyyL3naZA4yNHaaiYNlbzM3BL3x10lMPdyx78Rq8sRe++qjaGLdX575gBBu4YGXJNQyxz
lPnnrsRMnKtGx3TJizraPkTvC8OeE6xTF9Rk4bZdQp8Zy/ACiFKXtR1ycDqfzm5T5Z+UJ+dj1IZpUJsj2XknHi
/Q94krpC3iWvkHXEukPfEuwPygfiQIx+Tfot8T7yRyKfEpHlIeqpTJaY/nJN3g/yYmN7rpMGaUqR3SQ3eOsFWc
Rc/I+Tq6n0cHy2M5oYTMxb471Ld5NBG5xugzJDCCmVuZHN0cmVhbQplbmRvYmoKCjE2IDAgb2JqCjw8L1R5cGU
vRm9udC9TdWJ0eXBlL1RydWVUeXBlL0Jhc2VGb250L0dBQUFBQStjaG9jb2xldGEKL0ZpcnN0Q2hhciAwCi9MY
XN0Q2hhciAxNgovV2lkdGhzWzQwMCA2NjggMzk5IDE2MiA0MDcgNDAwIDc1NCA0MjMgNTE1IDI3MCAyNDMgNDk
2IDQ4MCAzODcgNDg2IDQ1Nwo1MDcgXQovRm9udERlc2NyaXB0b3IgMTQgMCBSCi9Ub1VuaWNvZGUgMTUgMCBSC
j4+CmVuZG9iagoKMTcgMCBvYmoKPDwvTGVuZ3RoIDE4IDAgUi9GaWx0ZXIvRmxhdGVEZWNvZGUvTGVuZ3RoMSA
xMDAzNj4+CnN0cmVhbQp4nOVaeXAb13l/Ab/ihqwOZtvLgk
```

```
VtTP2CpuTJhcIxYzaePdTxqF1Z3HXWPD7XduLL/Ivfa7W3u1O2XOP3SlNHR1+psDAxOjBvs/W5Xy/Qxi9h07Fv
13HJgX5NI7PGf/e/VaR66J/S4Rajrd0JV16BxVIYGCf6FygIgJelDu1rMC2avrKWomAISvydz4KsmKHFWuQag0
AVlo+aWbMZeBB65bR5bCMwN2Txc+/57rl3DW/Cb29/lXt1/EY/e7gJV6NidW/hnnAs5IZ5zqAzt1StrhFtljcn
ovTGh0Rm85ToMvFbh1m4jNtWZvbuNAvIKXiy5A5J1T8ZgQWOwjK1UHiODFyqaqu9wNzgbQgODIQDVc7yrs18Fs
cOtwop/tv84N+v2xT0rDQnfwNPjXQ+dXv1WHbf/JO4a6wysdtf9oORzcBvbXnvm3Pon5vv+q/nEJX/LbAfDaQy
uzwBOPHKjMmxk3jIPIGFJCEgcM4r9NMwUBwDxzAr36u0umKD6/B/gcy10yiFU1lOfG8FnRB21UnChpkx7ch24V
qfkmwCLuWEx2doIaWSiCQO+1Ncd+OakmVIHLgeVzMHGjwy3xocXnhlzJwfnr3U6FsiJK131S22c651nrnz6fNf
W1PU/ne6Yv9o78Z5eR7h38gN+KxhQixD/LJhpQiVUFqhlmA+Fyhpwrqw31IRCISyZA7IFoj3Q/MXTX3kjiOq9R
knvt0i6mzInvKmR+Jtf/Gr0Xz/P+Ngv85xe4m5aZKMRnhlu8gCXzuj3+/GfwR5KN1C/8pEmmgErt6cGe/iQC9P
/uNfjf/Kp+JcS7/lk7I9/J8Fdvv0ZAFHiFulPiQEeYjj2o7KWxkAHCJbNFEZEYbQGJO2erNHc2jVoUCcUoo1Ci
LQAoYFCCKlRT5O0foDlBh7aaG9dmR8ZMVtMdfgFrt/7lv1/40zXr1kgtmrcfgzY2NFxdF2Nuo5C5ILg2SlCtVS
1g6r2BKSaPdkJEXRa5TZAyxbcxUoLwdZdToNcQHEB2Gu8cpsTDNJBS3C4bHbZxg9T22ye/sOZ2+C08XdH+McPj
lyqMf32WGdk+OKjA7+/2t083eFf7mq84Ka5dm5Zj3PnN56dDeBFXLv/k6XQ9NsGXEuhs0+FnIjlcBwub+VehFq
fQuUa6o2eg5bI4Ro9lBknoDZaZibmCdR1ucZIa6wGG6ApWHf1GmQDH/SBSr17+lnbGoR6x/zH48eO1XPcSBi/O
W52NpzXam+PUp3H4XIHdLahLCpbKXhmAI/jVBRdHFXhgjJmRYD25DaAsM0qOwEnC5DaYNlpoTJOK5S6m6KHbPY
yNrcODw9LTltZb6yjlMUu1QKiLjMgqq+jiDoGBmnJaGtp2HXuejfYOo4HBpV2UFcf1P/D9sRJ/Aexa416Hpf2T
+iC3vWr1/GYp+cpE375p3hxcOZdSwPBIXvD5SenhoZODa1c6Gr4FE0NyI2xO52cDXLRCn0sjMpm6lkTOGRuosa
awaEypsmpoRlioxniZu5Zwaf64C7P8lOyWuVWcMkA1U67mNwE5oMfks0m8eCCk6VqpfA7u2yOu1JibH7s0SdoP
iysjX8QMuHcce/Kyebznte471+8hrMXWDbM77/BfXoleO4dp5rfEjr7RH8zjYsRLq9wJsiFTlTW0L6PQ6ycoP/
THGBBlnUagJM3sAR1eLp0Hgcfcg6GcOvz0afeff0Ln7v++OMPQa79+ze+gWsw/vKXUXVPGYa6MaCJQ72SbiO7O
sTxsKPoKCDGSuukQcYsEbDGQK8GCHVNtZVC6OD73DV8a7+IryobytKtJUVX751bXJr7Rzh1PofK3UjpBGXUTVd
DPRCFWl7tto21lNfooKkWYjneAqp7yJ5NDgDRYpUFum3THkx5DojJKaXr/csnX0C0u1kkg1XS3pSJ4U3JfRMGN
/QGrYP2SS09uHi0Fpt9GO1q9QbiZi0OH6Kh3WEkN0LySkaAs0ONYAB3dqk7lRMirBwAIMJtuOGg33eFKAtf+kS
jkDGZnn8fTkxekcZdOIzT7q5Et32gqfvPH2vRzcw9/dyFbi68HByNWa1zHoHv7Xrq7UPHs31nsq36B1zHh1rMN
X3uaLNr7YHxxOkrmpi3Se1zXA3kcgA9AO9vZQvNhrrQrkE5IPUGyxxNYi9g6GVR8rJyHQ9IjXvyKOA1apUCFDJ
2EAowNEetFDzZ3hkM7uqUQ9EEjEep+3hYPhWAGj7WOzRMD0ZeAxyMtPaWfnYwUsvW1uAMDtYr3mt1alccHMWDo
xzU8Wmsbuo6R5CO2FGKYtSv+6bk4DivUe9pOG/G+PJbM17Hgu9dmH1mavi32BwbNg4vh/3O5msPtTimrPi3PoO
LZsv+SycwN24Y1+nm8KynZ629zoYNrXpTw4SJmzUY9r8y3Nr6kMlS21ZPJmuFypkAf5edZWZQ2UlRqwmxTb2sh
cuuTUBTtKE2BiTnntwAuxKnVH2DVTbBYbFGwaWJFpqNbgkcdb+yOTXQnapyIrR5xuYH5uqdyZOO5dEG10pH/Qp
nCq9ahGnSDUXx6HSjYd47vP89pSaOw771TbCrD30JlXtoTQgQO6GHxk7og5qop/FsAV5LPeW1NNN4BgOSbU8mE
LxqMIlV1kPUuoHstkpNlN0MdHNAboLqCCnV8caHv9quVIfFKpluyu3Cm1LnTRjcgG3V4ZXarTc62juhTOB6UCZ
leEir5YbJ3N7BaqVKsUrp7RtsOOhztEqqJ7p6qJP7VUn/2wy9rSdXWowDZsvCVpt1JTj1AY8xrIktObVnaq2Lm
07dlf6Zr1rwx87VmyAB/Q5BcLSGWk9eO9kkutzLuK+2tqcBeC3TDc0P+5o/adAreCL8sZHab95++cb12rGfcm3
8P1P2K9+JfP/g3fhOJ/8svG0gpK+yYB7/rjtDKHaYc+Rjp+8n3N+gRa4eHUN7aIxrgPt/wrzPoFo8B+MAOsb5U
Bx+x4Eew3+HjNxlkP8o6uXa4FkT8H5E91j0IfQSPoGn8Yfg+wPuOLfDfY3v5Z/hv8z/RPhDTVDzuLZTtcAOJ31
BeaGHMQfv53DTPAOWU26Qe6hq53zVZgxP51WaQzq0qtI8akZXVVoAmU2V1oCWx1VaC/xnVFoPMz6i0jXwXvZRl
TYL30BfV2kLate8H+SwwIEuq+YGo3mgTZoXGK1h/JcZrWX87zFax+gfMlrP6J8x2gCrPqvlVRojq/bjKs0hi/Y
5lebRKe2XVFoAmTdUWoPatXdUWousOo9K65FWF1LpGnQJznMKbTZ8UPd+lbagCzY3o43UHluS0TXUfluJ0SbGf
zejLYz+AKOtQBttip0OoO22TzO6jsl8gdH1QLfYXmR0A+O/zuhGRv+I0U3UR9ubjHZRvXYdo1upjN3J6DZGK36
5mUyA0ccZf5zR7XQd+yyjOxk/wuguoOvsIqNPAq21bzO6h8k/wehhRr93Kpvbzic3EkUS7O0N+ug1xK797DrEr
sP02tfLrkFyfmGVTOdFMZ7NFMk6XPLJtVIxmy/4KXc5miksiRulVDQPgiOkwiML2Xw6mhpha1SYl8R8IZnNkN5
TfUF/MDgwVBWHuYliMVcYCQQK0c1oJhNN+DcyJX82vxHI5bOPiOvFQiCuWhFYSSQLYEo6VyqKecIMA0Yumi+Sb
Lxq8DTw/YTMsId0Lilk48WtaF4cIdvZElmPZkhejCULikciSRZJNBMLZPMknY0l49uUUcrEQEUxIRJQlS7Q9em
A6jgvZsR8NEUWS2up5DqZS66LmYJIomAJ5RQSYoysbTNxag5ZVrWT6SysGi0CEj4iJuF5nmyqyIQqCtTVfASsO
REtUoPzJJujk7rBym2SihYP5vmNRoZJBYoDp2IkmWELJrI58CEBS4FXW8lUiqyJpFQQ46WUj4AkuTyzcuHi6gq
ZXIiQy5NLS5MLK5FRkCwmsvBU3BSVdZLpXCoJy4In+WimuE0Nnj+3NHUB5CfPzMzNrESozdMzKwvnlpfJ9MUlM
kkWJ5dWZqZW5yaXyOLq0uLF5XMQmGVR/GVYxlksALKYWIwmUwXwMwKRK4BJqRhJRDdFiOC6mNwEg6KQEbntXyV
AqWxmgzkGkipoNE/iJJMt+kgBzDIS+NCEhHzc2tqqpmJKWaIQMBonC6CxkBPXk6BBfGxdzCkBTcaV5MqLECAQi
WXXS2kRwrKVSK4nKOKFA7U+mnBETK+B/VUmha+UiaYgvsDOZfN0YTXzqjHOFLPMz8ryvkMPY1nQAb7Q7EsWC2I
qDrkOitmEvFgopYpJQKBqGSwFubCe3WQK1ZT9+QD6Ccu1qs8H+hLZLciTPFi3GU0lYwwASNUsS3EApED92EpsH
7GcpFk/+t9ZMKOgrNbpITwZN7oN1hXFjArqgaXgKSukSrmpyaJMpUVAp9MSpd5kQeVGlJYQTItloXtU9cKISmx
```

```
BkVcf+iBJU2JRvFtnoQhrMEfj+Wz6iH7/fVKMJnFKjBcDG7kUPE+nhNPCA8KEMCb0CqNCv3AaLcBrfB6lURSlg
FpD29iMRPQIyqBb8Dt4No+SqAhfEejn+ef5F3iZ/xwvoymQyaFtkEuiDZQACQKvbr3wDSJflQ4dovsP0UOH6OE
q3Qe/AzoI1HmwcxXu06BHhG8ctGaYrnWVovrXUAkoanMB+auyy2B/BjhLQG+ARArGeXXFkUNrVuTIEUxGDtlxt
+QloKmuJLOBgNQpJueHXxANgHf3rq7oTTAsczAegZe0ANyjcNSiEhm4JmCFDaBKcKeWbIBEDu5ZiIsIHhdBPgA
YHMUigFZgZpJ5QFFJw5wSi1geOAeIKRI5hgIdZ+HZvQhPq/J+4BF4HTqYWdFLYETnFtEWW0tkWG0Dr8QsiDJMK
D/GZh6OkchWo2tSqRjYnmVWpuFOpeOwTkWixCQUL4rgocjuIotQoWp/5UnFj/MwyjApGkWCFmGdNaCSYBn9QzW
9iywiIrNCwaQiU2CrxYC3xiyprD5d9X35Lt8JQ0yxNQp8JSdoFovMj4TqweZdORO6x4OjtvnYczrzBFu3gnCe8
XNVTd0qltTaFJO8nz4/MsL3IE/uzor7RSrGnmYOWZhgmkWVE1Xn0+sW3FMM7zX2vMR8iLOq8zFuicldhoxaQRf
QRYjVCozpPzVGGH8S6nSJjVeAM6quWWQ6lbki8+mwPUmW6zRuirVKTPIMjyJDREF4Hp2DtadAr7L+JDoDdswxW
yJVnKfZeAFkl+FLxxdhFrWRZhG1bwUkpsDyOTai3FW4L4LcMsxSKmZZzZNfLy/jh+pCyTJaCUWYS3EuqPGMqDV
XUFFKMRQSrKeIag2uszzcVBGKqj0idwidX7eCUiyXNg5FTFnzaKZV+gnVSPeXIsuLgoqWkT0lzHqlQyr9cYt97
+2KqSNW0K5I8ZhkGqNs3RzzPan6IKLHmHzuSIUmmTUHnYviFVW7VJRhngVuCaIgqtWyxfxaZz6WVN3389ZX7XB
Ud5pVRey+kpXsK7EdIKXWryKdY88qFh/teffWcYbtgQfxvNt638+ZSeUUP5S4VHpfku03dN+Pq329pMa8oiHP5
pWY1Uk1B+7FTLFK6Qs09zYPeXi0y/5fMpDm1UFfuzfO9/OPVsuW2k/yKnabTFdS7eLioa6aPdTFlQwpVONB82H
7F2BOK/jgfPSbxGDmSC4f3U/vn58HslEmRbErsjWPZur9MFVierAj3b27He0sh7VWdoKK9souWolNVvVyQ41Dr
Kotxq4U+3v9VZ5V1thSd/J7Z/rUTppi3VT8pX4WWM8tshquRDTOTmTpX+C//1fsYpVOnGJ7JT3JbbD9TJmfBur
wifSAVjL1fs9eZyd25eSVZTmUZjmt+Bw7MucijNOsQ2ZYt7sCNm6jR/EZGN1fb0o4JvQJs8J5eJuYFYaPaKW5+
nNmqdiJrEPAKVcYhzeRi/AucomtoZyAYyx6m9Ra7IDzbuHIW0iJYVtAm8d+CONJtsuXWDyihiHDAwY//E4Y5g0
B5W+Zyl9yF+Ewfp/P5/Gdd0v4vWhW0i+GZYzfF5Gn6f/hI1kjs1LdEhBPRlolrfdqGEkCOSfx3rNY0iiEpCXnv
og0T9xBnFfmpze9ZyNSnReh/wGKk3dsCmVuZHN0cmVhbQplbmRvYmoKCjE4IDAgb2JqCjU0OTEKZW5kb2JqCgo
xOSAwIG9iago8PC9UeXBlL0ZvbnREZXNjcmlwdG9yL0ZvbnROYW1lL0RBQUFBQStGcmVlU2FucwovRmxhZ3MgN
AovRm9udEJCb3hbLTExNjYgLTYzOCAyMjU5IDEwNTBdL0l0YWxpY0FuZ2xlIDAKL0FzY2VudCA5MDAKL0Rlc2N
lbnQgLTIwMAovQ2FwSGVpZ2h0IDEwNTAKL1N0ZW1WIDgwCi9Gb250RmlsZTIgMTcgMCBSCj4+CmVuZG9iagoKM
jAgMCBvYmoKPDwvTGVuZ3RoIDMwMi9GaWx0ZXIvRmxhdGVEZWNvZGU+PgpzdHJlYW0KeJxdkctqwzAQRff6Ci3
TRbBsJ3EKxpA6MXjRB3X6AY40TgW1LGRl4b+vZpS20IXEGebOZR5J3R5bo33y5ibZgeeDNsrBPN2cBH6BqzYsz
bjS0t8j+uXYW5aE2m6ZPYytGaayZMl7yM3eLXx1UNMFHljy6hQ4ba589VF3Ie5u1n7BCMZzwaqKKxiCz3NvX/o
REqpatyqktV/WoeRPcF4s8IziNLYiJwWz7SW43lyBlUJUvGyaioFR/3JpEUsug/zsXZCmQSrEZl8Fzoh3W+Q8c
o28idwgb4kzgbwj3hbIBXGRIe+jfoP8SJynyIfIpHmKevKpI1MPx9jPDvkU/XPkJnqiTyoin5Bj/0VOw96nwrH
xLj/r5PLmXFglHY92iNvTBvjvge1ksYzeN+4VkxgKZW5kc3RyZWFtCmVuZG9iagoKMjEgMCBvYmoKPDwvVHlwZ
S9Gb250L1N1YnR5cGUvVHJ1ZVR5cGUvQmFzZUZvbnQvREFBQUFBK0ZyZWVTYW5zCi9GaXJzdENoYXIgMAovTGF
zdENoYXIgMTcKL1dpZHRoc1s4MDAgNzIxIDUzMyAyMTQgNTM0IDI1MCA5MzcgMzMyIDU1OSA1NTYgNTU2IDU1O
SA0NzYgNTk5IDY2NyA1NDMKNTM3IDQ5MyBdCi9Gb250RGVzY3JpcHRvciAxOSAwIFIKL1RvVW5pY29kZSAyMCA
wIFIKPj4KZW5kb2JqCgoyMiAwIG9iago8PC9MZW5ndGggMjMgMCBSL0ZpbHRlci9GbGF0ZURlY29kZS9MZW5nd
GgxIDIyMjY4Pj4Kc3RyZWFtCnic7XsLfBTVufiZ+Wb2cTZZdkMWCOQxCQYILkkM8lbZPDYQCElMlkC0tNlkN2Q
hya67mwREDOoFxNKLVQlqrVKLqGgtWqoBqdWWahW47b9i/7W9Xmulf+u/kXJ7aW+LMNzvnJnNbkKgCPi4v3uzZ
OfMme9879d8QDTc6SdJZB0B4mpu94Zc184sIoQcIkRIae6KKrPHtP4I17/FX7UltLy9u7vIRwjgPXlqedvqlsd
mvtNFiLSZkBGmVr/X17t78wxCRt6Hz6e34sZUtcOI94iPXNHaHl01x/DzL+H9cbz3tAWbveqEX2cQklqG98527
6rQO+IiCe9DeK90eNv9nT8MV+L93YRc814oGIneQG47Q8j1UfY8FPaHWo5Lb+M90pNbcU/AD/tJwqWB3YsgyQa
jyUwtScnWETZ7yshU8j/pR5hG+shB/LxCdpGHhJ1414LbN+HOdvE5sp504s6PhYPCJnEK7u0kx8mbCLmRHIRdE
hEWkKm4S8jbskhOCB6yB3HMElKFWUaDRKQqaY9UK/VJH0iHyQwpIh2WGqWIMBUelevlnfg7C34ippDXSRbpE94
lEbIPPoSpsF8qk6zkXTgMu8jvkQraG2lsITvIGuQlVQiSHnGNWIs7r8mHyYP4CeLzw8LDwpvI3T7hDvIWuR8kc
T55WHgL5TpI/kruAI/Yg345VWxB/l9DXIfx/IMkIhH5LYESVbwS95B7pNXEvzNgivwW/xwnPUjZQ3YY+gypxvF
IhWlsp/Bjod9wL9lO3oQvwU3wG2G9NF56QppPtmgagEayBXE/yM4YWoTVKDv7rGHYxW6pUdhFPpQajU2I+ydMI
qS5R6xFiVrIfvztNthQpjnCetiEnLKnGeSwcYFUgOcRg3EtSk1IEKaRFbhaQ54hz5Ep0Eu2ICYur2GG/Fc8+ZD
0Hsq8Rfia+FdyGMpIHmmRjqGuCfp5LyEvGA2yBKJAnIptt5hb4dvtun6p8tOG7CnOIbeKzajsJjW7k1crfWfO1
CyVxskNu+X03ZBr2i3ljn/vXA/fm+JcWLNU2X3aXaZjdTeW4V7dUlyyO9zGfXcZf8aI7pZz8U9F426luVW5y3b
X+Nl32fyzp6DaSIvaK7XIOzAbGclYV5L0MTF8LJjkHlEiBQeO9F9FbEf6j/QXjrRn23Oz7dktEjkVgXGnfq/2G
q1/+3PYkIfx/zrmgjnyW8RCnK5U01bxOxK5jRqMkC7PNAvpREqyneovwj9XkYJrjuKi8LmaZEFYNnIqIp1qH4/
f419/Q/y3N944nfOG/Nbph0TfySvF107PYvzddOZ9qQdtnErGkZDrCuIQzBtMd8qOJwV5b5Lw4pi9KX1Jm9PHO
USTw0QWiikj3OlI7UT/AXvKLGT+6In+o7Zj+DlxDDcKXXlzM0IZ2zN+nnE8Q55L5gpzxbmOueNkp7HAVGB20iA
JCkEx6AiOMy+7SVgmOLIzhalF02c4rMJ4hdhtZGoRMeYL43MMRqnn1HNJh19Y8VpT889XqifU14S8U78TjH3iY
```

```
3c+uNcqfvmGl167+upnJjuFmQIVRgql6jsHtu155mGWMV/Br9WGVNR5zgtkmyiYSLmETKN6+gtdyTbZJdfIjXJ
IPi4buJbGv9KHgfL3/pg+OlEfFjKaLHCNNOxNIXuT+lI2jzGnjLgeUhzuMVx8TXTbsULX+Llpa8gaQ4+xx9Rj7
qE9ljVJPck91p4RPbYe+5qU7WnH0+woaY7BkToKRZ129YSJRUzS8TnsW4zc9/RTW+99+ul7jwsp6rHj/67+SbD
Dux+8/voHf/jpax8+pP5U7Vc/QuFnoYypwkyWAAnWFuNO9IckYY2rQh5rkM3ULI2lZhhLLVQcK4gWCzXYjSajb
Jdkk8ko2kFMQmg7lpESKotgAPJdiynJQs0mLZgsRpJsO3Jo9CzmQv1FRaNnFaK7G23yR0abSf+V41dt2ZDzHLG
in7n+bBANskgJflLoJPkKqtDrxOvkq2khrRQXySXURRvEFeJKeTltpGvEHvEWuUdeR3vFrXKGkZhFExDJIBOZG
ASjZCImo5mYJUqTiHUsOCSHKS3JZlWkbFkxKEbFNN58Bc21KFbFeo04G6ZJU+VC03TzLMvcpEJrOSkXFoguyY0
2LjGUGEtMLpPLXEYXJbmsLutSsd7UkFRjbRGXg1dqkhsNjcZGk8/soz5LN+kS1oiroFuKyqsNq43dppBpVVJPU
o91g7gR7pQ2yevNd1m2WLdJj1i/a71xGUHHMQvsjzDeLIwvOyTMFma9z74Oq5tU9Sfqj1T5rY9TpGPs9+SVsu3
kcW67fZjTF6B/2UmhK9WAerdbYKO1z7zfSA0oenmK7dQBHsnoXgeOHGJBtadm5CMjRfQhHhgODOipjuswbEaNh
gVZFc6HHt+7d/a+9SPz02FPiv3gS6efkxqfaWmWZRLz5xswFmwkncx0pY3dS6ype2XTZmufsA1GS8QkzrOnWNw
Z3KuLGOH+oycO9NsOHCt8vjFzXeb2TEDKLI1MwxjNEXmMjhJivowMwaN9fbOfveXgGXLm4C3Pnn7t8XvueeKJe
+55HJ4Xv/z3/id8XqFMMOGnzKs6Dn7wwUH8RT28hz3WDejDlLhcjhLYLonb5duMZLvZlGVIB5IlWGxHFu4e4Vm
6F4FdMxv6D2gBXHSCpUt0zoYcYc8IGCGJy2Zk2+VpuZjsHNmqsEB9QPC/ISw4tWOXFJnfN//kW7u43rFySAuwM
qaTh10T08aOgzHpdlkidlmWSmzfst+XvD316xJ2EMRGRYGmj7aBAXWycLfDs3D3KM+NC3enem5EXuDMyzMbMG+
//DLPf5yfE/0DwSJ/JOxOt9lHz0LuXEWLpXq53nizdLPcNW5jmhH7izRpLBaa9CjpMnSOjYyLpt9ONqTdPvb2c
benP0GeGGdHt8pFMaZNJzOuEzT1GqcxW0uOVIPRQLCpeeVUZUT42lTvosc3fOXNVTcfWfoHIdV9Y5p6YteuXd3
C12e3b6vo7i0pPXRV0R9+9KXHQhnqH7kf7FPrpR3od8wPrnONi/lBn3Wz8APYn4E+MI97QznzhKIiLcMdjbmCy
6z5wm8zJWFZ7oAHYgYTx+dMEDijWn4TIuiNz645RM6cObTmWXEmesPj7PeJ088Y6C6fV92v/g0/+73CH2POwO0
zB+2DHQexCKWuctmOwkp2kIzsIkuCKIBdxDRlR0hqN1OBXSwUU5zZjgmuhBolATPHDzELaSvRZEhiDkQ9zInQf
Db2Zec2NDB/YgZla82eB+yx1HeuzDf4npn3fipJdKzkoBPotdJVdLG0xLiUttAu4WapyxilX5Nupw9Ij0jbjPf
Qu+lO4Unpu9Jjxm/T7TSdYguPOduC2U12mMda8mCCnGuebFGSMX3ADPlqI8tnhckVUC67zQssruQGUi80iA2wR
K43NBjrTfXmBktNcjB5ldCT/A3hPuNTwg7j7uSfJb+bfCa5AJ3ILLKkxPOT5FNXCrveVvep+94WvqeG3xbyhDy
p8fS7p18R+tT54gJxlHqTsIX5COszdmNMmjA75bscZKv5NmGrzSTaKJHTkotIullK4XnCPgt/mPej6z/XOFLgG
YLX7+zcbH7NE4R7TwjThCz1PfWgWiI8Ijwn9Kqtao3qlQs+7hbGYDJzCqN3qtvUdeqtai/z0TO/Uet53baQEaT
MlW4RjcT6UpJxo/wDsj/pWZvJJhuqkwVTEim32U6hrY7O4j1IP6/EGI/Yf1hsdpe9xt5oD9llXttTDVcK01iuQ
pfFOv/t75dfFajs65MaP978y5cf8j5gmPQhj481mCenYHxQkkv2Y4bIsow2W8mTow17rXZlQ9a+9L3j++ybRye
R0TAm2WyyZIEp1T0BtXHoCIaLnavjwNETp5CVV7Efss+yz2L9UEdhRmFmYVahUphdmDN3oivDlenKcimubFdOT
UZNZk1WjVKTXZNTMzE0cX3GxsyNWRuVjdnrc+6euH3i8YmZsaOxQ7EDjZmNWY1KY3YoM5QVUkLZ6zLXZa1T1mW
PWZbQaVwrzNCT9oRpV0+fmp2YTkaJL7379G3BB/b29c3df+fTB09/LIiPb2t83uN/6Yb/OC5ObVnTFHl7T17l6
dt2tXhfefQHL6f0fDU/f9fEiadYrNrYuyn6iYG84JqI8WmQMTrBwC4S4HPRjvFZApLwXWLA7oLIsmC0HTkwWku
YwwYY9hI1JtZLNBGYDPOkefINsBbuAMx5RtEkmQ0OIVUcK42VJ5MJwgQxT8qTcw2KaSaZKkwVr5GukWcY5hO34
BYrpAp5ngHjxdAiBqSAfDPW9S5xtbRa7jSsM91PthnyMD6ytcotLjj96pvC28Kvf3H6NazWo6UPT16JvvAM+sJ
S9IWR2BXPcaWDmYBVMGy02jdTYX9qXxLaP9VSJWJb6WDhMEur10eLMFnaUzCB7Ak6XnZgwR6JZVMrlAP1c4LwT
N8tt2z9Tl9f6bOdr/xU+JmwT9x52vvIIy/tENd8vP3plubj8ATT8Ubk4VGsVxPJra5rkpNEq0XMzMo0mUUjFbO
yMkuoJTNLcgjE8a3U+8ZstUtbyX25X7dvnpRJLVnjjCRnXJp1ijEtNWeS7V8P9J/qP8rCY5bWsR/FaPnrMdur9
lius36ES/2ChhiBwbvs+ay8grzqPMAeXWs7UkdlCZmCI1XKzpkwcRpv29GjCjCOp119BXqUND9y6CuPfa97582
/+7/qO+oHK/60bk1/+Dv7Nz645ndvCKP/Evi1vOMnM6av62r2Z6Vd+fbzb/+2sODn7vI7b+24JWvMlJefevXoB
L0/knxc99NdaXHN9yXtpwJqvIrFXnkqq1GsROH7zoDSGx0/Y0oXsDpp9SjeKEk+pvSn9+4t+V7nK6+KO05/SXz
4kYdf2nF6I7ZKft+fkG7wzPvwGtKdSD7Q9V2nq7surm6yQdgkpW5wbBqz1y7tze2L67t2nMlqNKXmuCex/u3II
H3329S/HLP95VjKYH3H1E0mMr9vT6fplvSkfHxJclqcSXPMc+gcy5wki0IU4QpxEp1kmTyyILXAMXnUpMxJWXl
KXvYVEzfQDZYNSRuSU1gFFUUDNVggCZLBCiPABmkwFsZBupRhnliQNzfvK3k9eevy7s7bnnc8bwyGwE1DDWsYf
7Zhp6Py4KtVT9ywaVPTfXMPPPafv7rhx20tr3pv3+x/yvXU/b/9l5Y90txnJk3yeFwV2dbJD2x66Pnx41+aNq3
h+oU1uSOu2Hr7w09navUdc2zDA3ev/+j0V0Zc8xeSZeKznF/8KnlrbK7Dsr9xp4ENs0zxYQ++6bSrGYnjnyHjI
Id0mLQYJPK61E9uEmeRV6Tf4/WreO5DUgYLyD45ldwkjybv4W8vrvdJ+8kcfKt+XZx15jfSB2SNvJrY5BTyDD7
biNd90pV8TpFDatAXfyVMFiaLJtElbhYPQBncJ9VLvdIReYX8a0O64X7DEeM/cY4cZBLmDcIriY08wCSQHOIov
LJZ0FjhugG+7x+QQcA6d7++FhHu2/oacP9xfS3h+jl9LZMk8gN9bcA69bq+ZtX6TX1tIRnkqL5OTvmmEJseWsn
VIx/W1zZiGfkLfW0n0sh3kKIgmZGhwpHv6WuBjHLY9bVITI6J+hpwv0BfS7h262uZjHHcqK8NJNUR0dcmkuPYq
K8tZLbjSX2dnDvb8b6+tpLWORn62kZGzVmvr+3ENOcbpcHQ6nBgeWtUmdScpxQVFk5VmlYrJYFoJBr2e9udSkV
Hc75S3Nam1DKoiFLrj/jDXX5fPj3r6HR21OPtal8R7FiulHhbz3GwzL/CW9+pNLd6O5b7I4o37FcCHUqos6kt0
Kz4gu3eQEcMps7bESkJBlcm3CYs6/3hSCDYoRTlT52mbScAtAQ7kGoUhWiNRkOzCwp8uN/VmR8Jdoab/S3B8HJ
```

```
/foc/Ws7BGA9MigHBlUkRv19p8rcFu/PylQvgOF+Z17Y61BpRAu2hYDjq9ykt4WC7Uhz2d+msxGhwDXVqGkokQ
2mcOkrmVTTWBtRMp5z3h55tkAu2pTKEciBCvUo07PX5273hlUqwZSgWSmv84fZAhKs/EFFa/WE/0loe9nag6E6
UHcXCY6gx1LNTiQYVb8dqJYQGwwPBpihqLIAq8CrNyDRFyGirP6an5uZgewjBGUC0FbGjlv0dEdReDldJTh4i8
yneSCTYHPAiPeoLNne2+zui3ijjpyXQhkaaxDDyA0pdsCXajerPyeOchP2hcNDX2eznaHwBFCzQ1Bn1Mx7ooAN
ONHNzW6ePcdIdiLYGO6PITHtAJ8QohDVVItrOCMIzcZxKu59JTbmDRFqdCTScjGZBMKxE/GgHhA4gq7r4Q0gz5
hBtiCk6SjXVcULdrehYZx1gZmjpDHcgQT8/6AsqkaBTiXQ2rfA3R9kOk68l2IbOxgRqDnb4AkyOyGxKPYjO2xT
s8nMJNC/iDAw4QUcwimaIaLvMKqG4B2jPlEirt62NNvl1rSEbGCXeQXIGO9Avwkp7MOwfVmwlujrkb/EioXyNq
cFP272rMVrwuC/QEmCO5m2LouvhApF6fT4uuaY6FqDeMPLV2eYNU0bI548ElndwNpZrsYqHmId6mxFJhJ2I8RM
ZSomhpEiAK8zbNjwC/UyMjzg2ZK+jbbUSSHBzysQJ+9nfWHFYtogwRTK7xMLDjz7nD/ND3cGwL6LkDMRhDqMde
0BzWNjmcJWhZSr1eGnyYyQxrJ1oA6aTrmBggDH/qihGjOINhTC8vE1tfvZAkx0xswWNG6XVG1VavRHE6O8YpBP
mdXHv9imdHT6d4TirlDOnSXg+q0aCbSyqudmYkbxKG8seGCsxwJC3eaV3OQqGcdgRpMxVP5lTDSKFCQtZ9Le1M
Kbmu5Xy6iqPUldd7llSXOtWKuqUmtrq+ooyd5mSU1yH9zlOZUmFZ371Yo+CELXFVZ4GpbpcKa5qUBZWVJU5Fff
Smlp3XR2trlUqFtVUVrhxr6KqtHJxWUXVPKUEz1VVe5TKikUVHkTqqeZHdVQV7jqGbJG7tnQ+3haXVFRWeBqct
LzCU4U4kblapVipKa71VJQuriyuVWoW19ZU17kRRxmiraqoKq9FKu5FbhQCEZVW1zTUVsyb73HiIQ9uOqmntrj
Mvai4dqFTQWTVKHKtwkHykUvEobjr2eG6+cWVlUpJhafOU+suXsRgmXbmVVUvctPy6sVVZcWeiuoqpcSNohSXV
Lo13lCU0sriikVOpax4UfE8Jk6MCAPTxImrg7ID89xV7triSqdSV+MurWAL1GNFrbvUwyFR96iJSs5uaXVVnfv
6xbiBcDESTrpkvpuTQAGK8U8p54yLX4XiMjye6lrPACtLKurcTqW4tqKOWaS8thrZZfasLucesBj1yYxXpfPLb
MT2zvYOhGKndQHL3MWViLCOsYEbdBAsepd7VbM/FGW+rQe3lhp5GtVyp5N7rZYE0IXndWDgant8iWUJI4tXHS2
7xQs2K8dOLfXy9IHejZVIS72+Lj9mwAhLJcEwDbJk0h2I8EjHEtge1GqeEvG2ITE8xaKIQ2Gu9LbhscgAm4MCi
saKYSgcwCPd4UAUk4ni7cTdcOBmvQyH9TLFJVDiEjAq8eSg8R/2R0JYpQJd/rbV+QgbZrWMcxLowF6tXRedq68
5OjvWKkSV5Ry5Lxil2NHlK5TyjuuSW6cL7WUvTx9EtT5IuZg+iMb7IOUi+yB6dh+kJ/lmjikSqxnDNKjxhoVeS
q+kxHol+sXolahmh0+tV6JawF5Sr0QvY69E472ScpG9Eh3UF1xEr0TP1SspF94r0YReKTF8B7VLWM8xSVyudon
q7ZJySe0SHcQuf2+83C0T7Qgql9wy0cvaMlG9ZVIuvmWiQ1sm5WJaJjpsy6R8kpaJeorrFy2oZmwXz7+o7ojGJ
b+U7ojGuiPlUrojmtgdKRfVHdFhuyPlUroj5qyDAmWg8aHnbHyUT9D40PM3PsoFND6UNz6De4d/3NBEY/Au3jT
QfLzkn3dyVdAdWBkoCGAGWZUfag0V6GlsyOSMlJIgCZHVJEwCZDlpJVGikEmkmeThtYgU4mcqrpoQQiElCBMlE
fwNEz/xknbixN0K0oHw+bgqJm34UUjtAK4Iv/Pj1Y9nuvDbh5D0AqhOH6DqQUpdSIv9s7UOhGZ8ePHMJ6NYhqs
VeK6edCJEM8J6OTY/P+HlEimIpQO/QwjThHgDCKfg+SBS9/JnQ/HUcSwR5CiIn5XneDr8bj3nMIJ4g5xqEfI5l
UwbBD08hhZ+QpM1qluCyR5FzmeTAvz4dPguhM9HuCBewyiNn58Nc7nzEYcfz5QnYIvpIWaLsy3OnjHd+rl9/Ki
lIOlGWGaNy6NjhmkePlmNMK38ZACfhTjfUW5PpoEwP8E8gGHtGqKVoXLEfahzkA+dSxr2j6mGk12zmRdXiVo72
5spmXIJH3pBEXL543J4e8dlDuATyldRvsO8rJ3reiXuBdEC/4gXJlkNx9fOscW9P8B5auXP/LpcyzmVDt3qTt3
umrU0apqPaf7s5HwFufU7+PmQHmEahSBijeo+FtC9wMtxaJqmOs4o52KoPzVzOOaHGvYYBgat8a75sp/Hq+Z7O
QleksMtx876+DXC+WrGM15dPsqjoBk9tJ1jifInMf204KpNj6RJAzzGKbC8wviPov9q3s8oxnXCdkI8anxIoZm
fjnHj4xJEua814dMof6rRoOeh4NSjuRk56+RYNJ10cx9o5Vknqmumne8lShSTITzIKzVuO7kOnQnWYet2bk/N1
jQhg0TwtPMccjgH5CzgGUThmLV40HAHdK0Otv75pY5pTuM2NODRUc5X3OviEnVzfbRfEIVYNLTwrN2hS+hPoOj
j34yGk1+ZJlYgRDPHp8HE7Mf8uE3PbDELNXPaPs5xQOd0No9Oj86dFzEGeWaI2yAxF8U1cHYm6ED4qB4NkUGws
ViJaywxBySeU7jMXs455bl5sK9p2tBqifc89gzyKqfotm/n13j+uBBbRHklYpXTq0uUP0hT5zvLdLJary0adab
zFs6jT/ekNu6n4YEdjVOmU1+CzRO9LlZBvbwiBnjOaON3dEAiH+eU2asjQRvLB9VVjVIsh3q592i+G6MxVD+Rf
yhTjEuqSxD3MC+30YVzMJjOUH0Mx5tTt3cbPxc4RzanA9YJ8zzr5Xkljje2ExnwyFi8DK0efj3P+bkUMUrdXCo
fP58zTD3MGZB76AmKz2LVNifBy7SYqRxSX5p4vAcTeO3U4yDmJ134NDCMxvxkFddzhx7JIfxo1cvLM6p/4ESi3
TWeYzt02Ehp5Rle4deIzqOfe9K5/CSW64bL3T5eCTq43RP1NZxWaYLmEm14sbEa4VkzVqvj0RaLJNY5tA30HmH
9xGCMIe7RK/F7uW4xrR4yr6IDWfXTzFTnlqpJj5GoXg9bBjQ1n7g5nWpShXeMTjXeecgS7CNr+bMK3FOwj6vFJ
/V4V4a7ZdwuxfwJe57Do3EJrhnGarKY49Jw1OI3w92AOwy3wu/Z3UKEr0Jc7KybLOU03IitDjmrxjXDvQh3K/H
q1uHYiVLcWYz3bD2PsC5Uo1eFpzw8dtg5xovGqQf341QHc1XBKcY4W4R3tYh/vv60GHFXcHyMfyfvj9i6SudT0
1wtx850xDAznKXIUSW/Y7uL8VqDcHVcn8VcZo3bKi5DOT7XZHFzDjRLaByV4rUGaTOIeciXh2uBUfLokE5uRyZ
PGT/PqC7kUBpn1bqV2TqOJV/XpcYH03/9AOU6Ln8lfhQuvwd3PNw2xYg/hjfmO/M4BsY35dpYzOUr5nqo5hRKO
BzTItNn5YDH1SZYpZTri9mNcV7GKRVzjdQNK0kMW6J1hvMOOkBhHpfPzTVVyaHrUI9uhK8Y2NH8sYLLWqrrWsO
p+b3mE5UJ2i3lMjLLXo9U3bpPFXPdDZaC2WkJ5z8uhWaBYv27NEFncetX6daN8ePhlD3DaGUJj0U3hyrmtq4bi
JFyHr+LdM4XD3hYPAcs1v2zeoCzwfqNxVEM7kJyh4YrRnuwBcu4P1XqHNYNaEODoOfBq+UuN9a1Zv6eEx3I24M
rd2LXGO9GE/tOZ0KuTewEtCw8j8O2D4GL72pvS1rNir/rJPZuw71hx96OtV4+1vXGuw8td2vvRIldr4/351oPG
BnoSoK8DwwOdCbd/Gm8pof02Ulw0Hseo+zltd85QCtWi+K4tL7Sy7sFRi0yjDbPXaHoWW+GIV7vNSrdfB3VOxM
mX6cOy/ZvHvI2HJv/nG0DZVgbxGQZrnNI1H+Y2zukv0sFuIZZP5mv4w2T2HtZXCdMA9pcrX2I1ePex7DNJkOnC
```

```
kwHyxM493FdU6LN6BhNyvNVbMb1+U+dLvdc9os0D6KD5kFDO69Pbx5Eh50HKZ/xPIhe0DxocCffnMBTfNYRg7y
wCepwExb6uc2VlLPmSvR/50oJc6X4hOG/51yJDqqwn99ciQ7ztvZFmCvRYedKcYk+m7kSPc+84LOZK1HySedK8
b91upxzpXi8DZ4rnav6nnu6pL2fa53EF226RMng6dLw043PZrpEz6NdJUGDX+wpE+U+dnY389lPmegXeMpEh0y
Z4u+6n+WUif7DKZPymU2Z6CeYMimf2pSJch3UI9YFnFtN28X4/LObHdFhbf55zY7oWbMj5XObHdFzzo7iM6BPf
3ZEP8Hs6Hx4P93ZUSyznruinD3xoRcx8Umc0lzOiQ+9pInP2e9sFzfxoQkTn/PNHS7HhCZ6Fn4XiU8aKKfD7vI
v4d9cFXC9rMTfAs6bj3dN+bx/DeHe4G7s/P/mjP+fZe3/cN9KbiDD/PSJ61xnPlbhZCr8PRf+VgT/2Qt/tcJfV
Dihwn/kwp+t8O+9cDwX/nRXsfwnFY71wke90H8S/ngS/r8KH86GP5TAByr8vyL4/dE6+fe9cBQBj9bB+78rkN8
/Cb8rgPdU+K0K7xbBv6XCO73wryr8JgV+vRbefhF+pcIvEfyXa+GtI/Pkt9bCkXnw5i/GyW+q8Itx8H9U+LkKP
1PhX1Q43AuHDmbKh1Q4mAlvFMHrKry63i6/mg4/GQUHVPixCj9S4RUVXlbhhyq8pMIPVNivwosq7LPD3g2581l4
V+l54Ue5T4YXnl8kvvAgvrJOe/36u/Pwy1xl43iV9Pxf2qPC9XnhOhWdV2K3Cd1V4xgffscLTT+XKT/vgqV0p8
lO5sCsFnkSmnzwJT6jwuAo7VXgsBXao8O1HrfK3i+BRK3zLB9sRZHsvPKLCw99Mkh9W4ZtJ8NA30uSHfPCNB23
yN9LgQRs8QOF+Fbb1JsvbVOhNhq14aGsv3HevVb5vEtxrhXtOwtfvflH+ugp3b1km3/0i3L1O2vLPufKWZbDFJ
f1zLnxNhc1fzZc3q/DVfLgLxbyrGDbdaZE3pcKdFtiIGxt9sAE1tSEX1tvhn1S443a7fIcKt9vhNhXWqdCjguv
MrWvXyreqsHYt3OKDNR6HvCYXblZhtQqrrNCdBF0UOlWInoTISQifhJtOQkiFoAodKrRlw0oVVthL5BV1EFChd
S0sx5sWFfwq+FRoVqFJBe9saDwJX06CZSrcqMINKjQspXLDSVhKYcmoNHlJEdSrsBgpLy4BjwPqBJtcNwZqU+H
6BSPl61WosUC1ClWLbHKVCotsUKnCQnyyUIUFFTZ5wUioyEiWK2wwPxnmqVDeC+5eKFOhVJwil56EkheheCG4V
JirwnXXpsjXpcK114yQr02Ba+Yky9e4zoyAOckwW4VZKsyckSrPPAkzptvkGakwfZpFnm6DaRa4OhOmJkPRVRa
5SIWrLFBYYJELk6HAAvlTzHK+DaaYwVkEV07Ola/0weS8FHlyLuSlwKSJufKkYpiYCxNyLfKEEZBrgStUGK9Cz
gjIRjmzU0DxQdZJyEQRMn2QkQzpqMF0FcadhLElkIY3aSqM8cFo1NRoFUbhoVFp4FAhVYWRKqQgQIoKdpTVXgK
2tTDCB1YVkpNGyckqJCF00iiwqEBtYFbBhGAmFYypYPCBhA8l9AAH4C6oIOK9OAUEGxAVhD7Bt/5rwpX/HX7I5
83AeX8y/gvbpQKHCmVuZHN0cmVhbQplbmRvYmoKCjIzIDAgb2JqCjkwMjMKZW5kb2JqCgoyNCAwIG9iago8PC9
UeXBlL0ZvbnREZXNjcmlwdG9yL0ZvbnROYW1lL0NBQUFBQStEZWphVnVTYW5zCi9GbGFncyA0Ci9Gb250QkJve
FstMTAyMCAtNDYyIDE3OTIgMTIzMl0vSXRhbGljQW5nbGUgMAovQXNjZW50IDkyOAovRGVzY2VudCAtMjM1Ci9
DYXBIZWlnaHQgMTIzMgovU3RlbVYgODAKL0ZvbnRGaWxlMiAyMiAwIFIKPj4KZW5kb2JqCgoyNSAwIG9iago8P
C9MZW5ndGggMzE2L0ZpbHRlci9GbGF0ZURlY29kZT4+CnN0cmVhbQp4nF2STW6DMBCF95zCy3QRYSCEVkJIKQk
Si/6otAcg9pBaKsYyZMHt65lJW6kL0Dea98aPMXHdHltrlvjVT6qDRQzGag/zdPUKxBkuxkZJKrRRy62itxp7F
8XB263zAmNrh6kso/gt9ObFr2Jz0NMZ7qL4xWvwxl7E5qPuQt1dnfuCEewiZFRVQsMQ5jz17rkfISbXttWhbZZ
1Gyx/gvfVgUipTjiKmjTMrlfge3uBqJSyEmXTVBFY/a+XSracB/XZ+yBNglTK3X0VOCXe58gZc428Y26Qc+JUI
u+J8wK5IC5S5HvW75AfiLME+cBMmkfW05yamTIcOQ95TzzngNww45xE8rl7ZM5fYOaE8+cZ8i3/CZnzFxkt5Pb
luBq8u5+VC3X1PqybLpj2jBs2FsTvT+AmhzZ6vgFvv5oECmVuZHN0cmVhbQplbmRvYmoKCjI2IDAgb2JqCjw8L
1R5cGUvRm9udC9TdWJ0eXBlL1RydWVUeXBlL0Jhc2VGb250L0NBQUFBQStEZWphVnVTYW5zCi9GaXJzdENoYXI
gMAovTGFzdENoYXIgMjAKL1dpZHRoc1s2MDAgNzUxIDYxNSAyNzcgNjExIDMxNyA5ODggNDExIDYzNCA2MzYgN
jM2IDYzNCA1OTEgNzcwIDI3NyA2MTIKNjg0IDYzMyA2MzQgNjMzIDUyMCBdCi9Gb250RGVzY3JpcHRvciAyNCA
wIFIKL1RvVW5pY29kZSAyNSAwIFIKPj4KZW5kb2JqCgoyNyAwIG9iago8PC9MZW5ndGggMjggMCBSL0ZpbHRlc
i9GbGF0ZURlY29kZS9MZW5ndGgxIDE1NTYwPj4Kc3RyZWFtCnic5XsLeFNVvu/ae+2d97NN2qaldPdFKYS201K
ggJi0TUuktJ02PCoykDZJk/SRkKSUtiKoRxEBGQTxMYwyDmdmDuPR4jgeFGTQqXp1ZLx3xOP16HzHGXFm9FS++
+D4zUXY3P9aeydNK6LfOfd+537f7TbZa6/1X//H7/9Y/x01ERvyIx3aiTBy9Ax4o46bllQjhN5CiMno2ZoQLv2
sJgTjj+BzKBDtHTj3r3d9iBBOAM2e3v6RgLjrybcR4uDRXBL0e30jz4wsQchaAhOLgjBRIw4q4fk2eC4JDiS2/
UXNrIfnnfDs6Y/0eJfP+20pPJ+HZ/uAd1v0NFMPzLLgEQmD3gF/7Wfvt8KzgFBhWTQST9yK7ryG0OKVZD0a80e
bFIlyeA4ixH8P5hi4yJ8OhgryzGKOVyhVao1WpzcYTeaMTAv6/+mP0aA4YtFJ9Bu43oFRCwqjLWgXOgrjT9AYn
X+ZU5ELZt7nCe076CRTDnQsuZhKYMOiM8CnBtY+AfoAPB9F3XT9Mj5Hr8fwOXYYsbgdRu10x1F0Ei/jOHxO+tB
dv0E3oRfRcTLmz6HDQNeB3oOrHrjfgk6j95m70THmXbQd7UNxElUon9Hw74IuYdTNv0uv/4Zuo5LJXJh/V2EBS
WGw8zRwPybNM+VMB96MAxBrY2DBcbwSZu9BYW4zXGX0aqT2STaw7BjIl+1F59jb2HKujDkOcoiMc8D/OLoJ9A2
ApivhwxL98afoKIY4Qzb+ZXSL8haFjlEotyMPrI7hGuYxRT7ajLZjD3Bohbl9qJV5D6QgC3pewXOYZZBdMI2zp
W7fuOO764X/1FW4wD7jUTAphXHUPq4fEU5eu9a+nsvju8b5WeO4VDXOlRb/4esW/7DAvqp9vTD+hqtR5ura3Ah
znethSJ5gGuZdjQuInwLiYS7AH4MKoES5Dh33OlK8zqj4jSyHKifOT34Hmc5Pnp+syjQXmksLzYUBDl2J47wrn
4iHlYa//o+YohxybgtCit3gpzz0PUdF9k61UavEOKLnu21WM2Yi6jsykUVXoslVlKgyDCVZGrOJQyU5KjNbOMt
0abL6QvV58mWi/3wHVS6/eDG7+ovl8GW6WOVQC/nR/J35v8rnmI01tTXm4tpCazFoI91rrIUwVwj3LWfOsM1ns
IJ8n7n6wpkrl8+EedTXd/UKy/X1fYnCYfauq3eEqc3XPubWQgxYQONWx2xWZR1Fqkf5/bqc/egHGXt1m2ep1NZ
clMfkGudRDQGCtwCJC1cmJ03iBdPFjLoqR4aQ78jfkY/fRm8zb7NvW9/O4zcyG5lCqyWrpnrR4hoDU1yEaheim
mqkrICxQsmtvfLnJx8f+UHBIfvnL38qfsJk/fG/M+x5xYH7djxoYpHqvQ92bH9mnKkRrzALxfdOvPjSmRdoEqN
```

```
loHQHRLoaORwZilHVFg6NsowdlSmxna/SmC5dqL5yAYCbrL4yWfXsfi0Deqwat3rWv4Q0184CAw1cjiXmxUu6S
iXAADzm2NNXPz1xggmFw2HekcIlDLhoUTaa78hE+3WK/QSOHJxrbVQDGDkAxqVJCQnwzXOCLWJjmY2l1YtqF84
BE8H27KTRVgtig4cefvjQoYcfOfgX8U+M7dNPGZv456bPLlz49NMLFz7rAUtFplr8rSiK56idGoRUNRBHFuawY
wP2aXVa1sewOriZMKs36BmTwaA3m1ijz4IsBp/eAPmkNem0HM+p1KzGxPPcaoOWxSjTqMZ8xKxTdmNkMRkNep1
Wo5LSzpihsJrOv5VdRwLt0vLlk9nV5uy6KsgOpYn/PLtaaVJ9Ln2kJ/7z9JmuomcLsiR4Mwm8VgqvFS4JXsc/6
VmNljfksgVsgTbXUMqWa0sMN7PLtbWGFm2TYT17m9ZjGND6DHdoE4YsFaNgMVar1GqlVqlXGVRGG0RdFphr5Ww
qm9qizdTl6G2GOWgOM4edg8u5Ev0cQy27EC/SLtQt1ruRm3Gx9biJc2mbdY36lYYNaAOznl2HN3DrtV26DYZNx
k2mTeZNGZsyN1k2ZQWQn/GzfhzkevmAKqDu1fRqAzqf3m8IGsOmsDloGWLjeCs/pBpSb9UMaYd1Cf2I8XbT7eY
Ry9qNaGNhsZopZmjmqZkatTRS2I6Jx8Xjx8SRQ2zhi8xixvMkcytT9xTz9EHuzJf1fBiy7wT34pd13G19fdTPt
0AP8Q7EWQ56yFGkUmfy4JZRM+Kz9urVo+Yj4Ct2ATLZM1RlOruhygbxDfXh/KRpAnx28WK1+Nprpoum18x1Zkj
DJSZkYkyAmIkz8SarKcuUbcqpzHUgB+NgHdjBOfhGqyPLke3IsYJolhALjMAqNk4liY160QaX5EXGXEwz1mpOZ
Yt1BVNTnYXfCfOtq+478O7Vcz8ZZ4KLjo8u6OHKxS/aWk+Pi7nhMPaEXor0GZK59DHkrAUVojbHnCzOZMXIdkh
fcEhpHs3cgkaVm/V72Qy71YTLIJVrZ+Vq5hWR9LpwZflymswXJkmSmS5CeIKdalRsKt5RPF4MZZAmMakvbO3CD
FArm2iIUykI5YerGRcfWfTTsYn/9ddXx3666GkmcPLg3gcOHHhg78Fw+Ce+Tcziy5eZRZt8bHn4y9+EfvXmR7/
/8I9vUN9cBt9cglNWg7Y5ZnE+XsFjn4JfjRUcwytCKiaO1Eozh1it6fzExJUJWnUuTVBfkBziIE04SBOAVkOg1
SIHMNUi7bWzUoKoBSQwlepKDUfg11IaCr+W0FKawtpCM19bCmXdCtHGie8w/fczUJj+6dd3T14eDUs1cQ/0na2
g5yy02lGosyCjHiVYX66fD+mHzRa9LhtpmGyFyqAxMRnafNOlCXKMAZh1dSTzL35xUbxoArUBWW2TpknbpGuyN
Fl5iIliM3E1Z7XAgc7Qw6WWWcFI4DKKaN/7wU8Zg/jbHJ24i40MXbmJefXwzh/ev/8It4XR1C3/4ysfiG9nw9p
J8dgAM3H4y/sfPv4AiYdl4lruJMS8DhWjNQ47g8z5WM8dytWPGotmjwqaQ5a9uZuFu4xHSgrsSJmbXVtosOebU
Jm+qoScQHA+Xpo0QVyYJidp8a1OhcZzO0rHS6EKM5LrJU3LkjGbYbWwxUVzyqRQZo4d3LvvwIF9ew+ehMpPAuS
vf50gAQJHAbsydPaNP374+4/e/GUfhIj45uXL4hubfGIeZC3JDgQuKIPyNO7oJTX4CYvVYnhCb7DCbYXRoFQpt
TpWv0KlUmKWZZ5QKBXoiTykhNsKqJBZVlsum7ciK8u62mpQIk6Vm8VjZjQHH8k06rBq1KzJs1lJpdYqoUjrMjh
1RvYsCLEPJ8xSqYY6feVbFOuZczQSdR6S6Bsg1GbRUJsFl1yvP+KQxmBFeYZSNM/QjNyGdWidssuw3hpAAWXQE
LSOGEas96A9BouGUbNqrNYrjLyNz1Xk5TK5bC62aW26XL3VaLPZcq15c1E5Y2ftuFRbqpurLzPMMc4xzTGX28p
zy/Lq2JvZOnwzNKNuptnQZGzOu5XpYrtwlwIE6tcZ1hjXZq21rc1dmxdmwmwY9yqD2qAuqO81+I2BvG3oduZ29
nZ8u2JEP2xIGBOmhDlh2WbdmjVkG8odyrsX7Vbu1e7V7dbvNuwy7rLca73Xdm/uvXkboGozmcmKTUJAK93k4s1
6/nk/8+g+8WOmZY9o3fXnv9wv9u9jSsWX7mY+2MHuZmr6+sRhcXE4zKwQX4HvQ8wZUsRZ0vPxl+CsVsHZd5PDp
lNGDN0aFFErGasRcYYMvboEmbWFWRC8pL8jeVcnFYvJaqhmVdlt2Tuzn8im1UxAZhMqLGakUis9MZdfhORfLL4
pfviiOAy9XQdzF/OM2C7uCDMtzCK4Vl4+TBo8cav4I/GomEDstWViHfMs/zEUktko5CiabcrWYXV2Vt7+TLVu1
HAk84cFOQuyEFZkLTKW6e26qgI4Wa6Qg+UiyasrkxOmi//4mum35gzIqucrhc1CVMBTp0QBjZ0CuKTYeS6eHc9
h0cbMhYugBlstCkgzgrSZjspqa2YzJB3PNVRWNDirKhsWQBX+9c9X3r1ik3JNx4Z1/JaKxsaKyvp6cbTvyy3HT
hUWfrD51rE7k2fHHqgVGjQX3eGozCvU2pBJx4zaDqr2ZxY+ULxv9v45ezM3g02sjUeFC4r09lwjKtPyuUVZ88q
h1kGpg4oB9SFDqnYXxItf/MkERQ/cYK4j3WuJaV5k3ra5Z+dxo3PPMmfZs3iCm+AnFBPWiayJ7LM5Z21nc/Xki
KxNnSmZ5uSxM6cWTC6UCo2yuFY6Gis/333vfYn4vpPHxGM3He078Z8vvs+wB+8ZvMu0afMv1/7+z0zrH7aORu5
6gDl99Xfh+C1Np4/+7fPukXt83e+Vl39Aa/q1SYipTyCmDOhBRx10a6wPqgrjQwwUFTi2GbVGrTewyKTRqFczK
g30Tiii47pVeo2aVB0GqY281gjdnVQzLtF68bXlAno5JcOgjdCCMCqWwxw0kTlMDpujnqWfZZjLztPPM+iVTBZ
TxmBAQs3ShJH7HybAvMCc/Bnz/IGr//LU1YsPsQIf/vJebrSv7/Jh0vHQM2ox+PEDqJk2NOQoyjYrdHoLe486B
0o+OqLezCF7Nl+G7cxSg91UpqvMJalCjn9y+sNtEkp8Nc2bquRJmUtP01yUK5+mz+3IG89jUxGaSyM0l5BJfUx
hmsvMNDTpCWEuZt9l1j8tPlbzZN+Jn/6E2bRpdOe9R7myKw3h0+HIG++xdX1XH3rl+6d/wXxM7XgK7PgEztpyp
sxRD11HSaEJevPZBWy+YILufLVWJxRyWcwO5OMHy7J2WH25gxmhss3zBK2uMF+VbbSyRqQqtypL8suRkZ1HA/S
C1NtkSHXBdOES7XbEi6n6bvgchvINXGWaD223QyvMr5rfPj86f+d8npid4SGN+IZV4xZa3udRfOahecluIzFXU
64t15XrywvKhfLC8qIl6jrNUu1S3VJ9naFu9tKCpcLSwrqiJcVujVt7i86tX1ngFtyF7qIuTRf0z136LqGrsKs
orO7X9Gv7dX36voJ+ob+wvyhUPKYZ047pxvQjhrGCMWGscKxopHjx0fnj81kouYvN2VL2L5xTyVRA91AitWlSW
2nJKmBmw3HNFRbNORPnT/3q2VNbnn/1H59hDJeCf927xXzil4//7e1vvPCbvxc/Ez8fu7aS8TDHf7xvV39i8TL
nK3935ncL5ouvPL7vzrG+0eWLb3rpifc/JD/aQd2AvprdTOufFbkd2VidMWo5YoIyATUP6l1ZptFursqSXhfJq
z19YTz/FrwyZqe9MmbRKMqCS4qiafUt1RC/TspafWVlw788ffWzcWiGU/Xsy7nQCHf0yX3+FuiBZ6EtjjnKLJS
hGVWhWdx+2xZQyqI6YlDrmQUIjvwytAh0s5qroFFbfqEagn+CtGqkdFVDv0+q8sVqUpWf3T87TdN8qmk+XJKm/
7AZbWbYm2cz5NCTsM6HF/LCVBtfU0ttycZbvExnW6+HKRffm7z61Pj4+Au/aLi3nretat1yYHfflaepBfc8+3y
ODdHfFNmuRxcE9m7bZFz+r6hARX9b+91/1T+U/J2NnDyqGgX5cVE19eMbQsoBMT/957gZP89ZuXMooLiMtnCnU
YDdj5Zx/wXu0Nmq6pAGv4Nu4QtQgM9Gl+GzB8bLuLuRTRFGW9ixa8u4V2Ht9WuTfBlazI+hp2D/LewXBHNUhEb
R7xgF08Q0sUvZp7AK34af4yz8LD7Av6loUjytzFD+SVWhepBqZIWEYZEUQSb0KFJCp6dl/idYQn6Cy2VWpPTel
```

```
LIBWjR4YuRdHOqTxxjmI/KYg/GYPOZhvEseKyA+D8pjFTKjx+WxFvz4jDzWqw+it+SxAS3UJGWZkE6T3GtGnOY
JkMhwang6o/mxPGZQjtYrj1mk0o7IYwzzd8pjDsaPymMexifksQJlad+QxypUpP1neaxFS3WMPNZnztEtk8cGF
CxIyjKhnILkXjNSFfyhIRIdiYV6gwlhbk+5UF1VVSN0jwj1oUQ8EfN7B+yCe7CnQnD29wsdhCoudPjj/thWv69
C0+gPe9cOCT1B72CvPy54Y34hNChEh7r7Qz2CLzLgDQ0maTr9sVCgPhLpS39OH6/1x+KhyKBQXVFTK82nkwQig
yA5AfoEE4no0spKH8xvHaqIR4ZiPf5AJNbrrxj0J5ooGdGD2JKyQZgb9/uFbn9/ZLi8QvgWWldoNFObQTOvIHF
OYaVZcMM/jebfjqowQ3IIVBQSMa/PP+CN9QmRwEwuGk27PzYQilP4gDroj/lBVm/MO5jw++xCIAbGwzYwGGCyC
4mI4B0cEaIAOGyIdCfA4NBgL0jpAaUJZSLolxH39vREBqJATggSQeAOIPkH4wBwEYWkqByY+QRvPB7pCXlBHiD
YMzTgH0x4E0SfQKgfMJ5LONINQmckkBgGzIvKqSYxfzQW8Q31+CkbXwgMC3UPJfxUh2kb7OClnv4hH9FkOJQIR
oYSoMxASBZE6GMSlMB2KA70xBy7MOCnVlP/xoP2NBl2IrMyEhPifvADUIdAVdn8GaKJcsA2SoBOyNBRQcPByMB
XNxA3BIZigyDQTzf6IkI8YhfiQ91hf0+CzEgY90NIEoN6IoO+ELEjvlSj8cCStzuy1U8tkKKIKpAKgsFIAtwQl
2aJV6JTESCtCfGgF4zq9suogRoQ5N5pdkYGIS5iwkAk5r+u2UJiJOoPeEFQhaTU9NUB7wjhPxDxhQIhEmje/gS
EHgyAqdfno5ZL0JH88sZAr6F+b4wK8vnjod5BqkZv/0g0GCebSIR6e4BJnOxI6hOfKUmKOJ8EmLc/jcEMJvK+p
C5THEHFwf4RITQt1MGkmJ/8+zxKSwZxAibxTTJF/BB3fsmA4UjMFxeKUrlYRGQnF4QikrpFFDbwToucM91+yCb
CdQj8QIzYGgmlFPNvS0DWCN5oFFLM293vJwuS/cB5hmOC3oQQ9MaBo39wOi4gbirCfcLQoE9WuGh6XSmSLLyRZ
+ORfpLZ1HXEUV6hn1QQyJckYdTb0+ftBcMgFwcjqfrx7QNrmigoWqCivz9AlFrpEpraWj1CZ1uTZ52zwyW4O4X
2jra17kZXo1Dk7ITnIruwzu1Z2bbGIwBFh7PV0yW0NQnO1i5hlbu10S641rd3uDo7hbYOwb26vcXtgjl3a0PLm
kZ3a7NQD/ta2zxCi3u12wNMPW10q8zK7eokzFa7OhpWwqOz3t3i9nTZhSa3p5XwbAKmTqHd2eFxN6xpcXYI7Ws
62ts6XcCjEdi2ulubOkCKa7ULjABGDW3tXR3u5pUeO2zywKRd8HQ4G12rnR2r7ETDNjC5Q6AkFaAl8BBca8nmz
pXOlhah3u3p9HS4nKsJLUGnubVtNcFoTWuj0+NuaxXqXWCKs77FJekGpjS0ON2r7UKjc7Wz2dU5JYSQyeZMwUE
2NLtaXR3OFrvQ2e5qcJMB4OjucDV4KCVgD0i0UHUb2lo7Xd9dAxNAlxQBDlnpoiLAACf800A1o+a3grmEj6etw
5NSZZ2702UXnB3uTqJCU0cbqEv8CTuIjWsAT+K8Vllf4iMy99XoACqyWzaw0eVsAYadRI2v0EJ0ubb1+KMJEtt
yckvlkZZSqX7aadRKRQBCuHkQEleao0OIZ8gsevJIFW4quciRbJfLLykfEN1wGknl17fVD1UwTkoJ5EeEFJPhU
JxmOhyDAxH53It7+0EY7EpRQb309sO2eErN6QmVPBCjsRBsGY6FElBMBO8QzMZCo/JRHJOPqpkWECkz9Y/541E
4qUJb/f0jFUAbI+cZ1SQ0CO3WgGw6ha8nsTRZQxNCL2XuA8OhKasQNDfs1yqHQ32hyhDUqG0V0WC0Ui6UM3tG1
ADtehSNoBgKoV4URAkkoLmoB5XDvRpVwVUDo26gEFA90CRQHD4x5EdeNIDsMOtGg0BfASMn6odLQB0pXnH65Ie
7H/ZshW8fUGpQI4zCwGEtGgKKHqD1ApdeSinAmPAXgMsgfEeBphv4hoBOgP0RkOulazP5dFIpIRQATSNw9X3t+
tfNr6WjOIwjVHY1aFuDaqfRfx2XAN0j2ZyQ8SEYJMCCpagSLp9MvxXoK4AuAvcYWOWne2PU/grg4Yc9TWnckng
k/fJVP5A1grGf+soPaEXQMNASr/yfwbqC/jvk60mWMPPCKF3nr8aVBi34d1xE+n9ErF4f7SmbQzKKAl33Uh8PU
FT7YC4Cnv0mXYhl7ZTfAOU2FX0S7yBd88t29VIpgzTCfJRPgK76U9IkD0vRZKd6RaiGg3R/VI5wSUIEuCZkD4d
oVEi29MhIJ3kmqBbTY9wLVD00QqIy9yQHQi3pLkWSH+bjcgQXpUVJEfUc2euj9zjVqwf2eGX7pBjsgagcoFwSd
CWJTwBG/XIcz03pOCWBZDbRPwG5IMU5kTiFCZmJwncEpAxRPae08VELEjTWumE1QVeTMr5egl3OpR7QbIhykTA
ZpjEQpDmfkJEZoHPpFiX5x6ZFpaTtEMXQnuYdMh6g/kz6eip/47Db/jV22FN2VtK6I1DOUj5IvEMyqtO9f2Ork
8hJ2kZTEZ2YEXVTFg1TPAa+lYRkNgRozRyULfSnSfTRbyLDTu8EiTBQ9FB+Ek16HPfLVTLpoR4q20c1DsmaLqX
Z6ZF3eYFjhFaGKR+k16IpBL5aCQaBPiFnQ3wabTJXphBLrwHp+wRqs1f2VHeqbidjTUJDquTeG/gzQs8YQfb9A
L1P1Y9v44sEWB6l55ZXtqhiGlI32kswGUnpP0CzL0RzOVnRiO4JuepJM5KmBFNfms/Toy55fhEpEl5DwMVL9yU
t8lFNib8G09DoBTpiTVCei6XVUC+NHil2kzJm4hP/RpvSa5xvWoR5qY+up8GNNZkubyYu19PRLvu9n+4L3aCqx
+QK5Kf6DUzjm5yJpyIzmTczTxG/XO/80zwwTK3y0f1F1zkXi1J2z9xB6JOnblFatEm50zLjnOmmeR9J03VIzoe
kJ7bCaug6iPnRNorzoJzRUbikU8xLK6s/tSPd/5LON86YIK30Ar3HZR39NKK+Pl4k665Xw8nqEKWajvD1UBXSk
Ev34b81Z+O0eibP7KmsS2YU6SD6Uz1ITN4xnWOURnQffPfKHpPOxUGK7cz+4/9Gxfp6q7rlHEnI52IghdRK5KJ
y2lArPBE5bfDkQeugn+yga26YE6Cf64CVtfDUCLON1C9OukLWi2g2roMx4diG1lBeEo8O+Ca8u2CG8BboM3laB
fStwIvsdaH1VIYLuHVSyg7KezXMtsDdJdORHQ0wswaeybgZkW5UktcKuzw0d8g+ooukqQfmp6RO18pNJSY1Ww1
PHcB/pbzqBN5uyo/ob6dIkXFrSs8mWVMnxYhwJjwbQKMW+kRm18C9Heg6KZ5OarOkbSu1oQnWJVtcVAPJE5JGD
XBvB9mEohn08lAtiCSPTGmnFhJ7Gul+InUVnZU0a5O9TMZTXCpkLCU9CP5rU5I7qf0tcAnUfg/MeKhvnMA/yTc
ZO82Uw+pUHK2h9jkpDm1UQj1dIygSPFtSlB1pXmmgeBG/Ec0bqSQnRaTzupYkuU33zvWiIymhmdrnoki1UOpOw
NEF9O7UjBSPbmprg4ytxFOKeykmWtLQbaA2Es9+F6S65JhyUuymWyFlCNF/ygrJA075uyENsynvt8rebUj5uo1
G2VdRWUdz0UWpnNTXnSkUmmj+rpY1X5MWYUk/rpHjsy2l2XR8k3mUpPs2tUPilZQ93YONNJ5aZA07U2h8M1+pd
rngXOuh7zuJVN2efnKnd49TXWl6/2lPq7XpnYBUhZsp7cAMuqlZqT5LZ9bUO096D3e9kyv5liz19FPdb7L7kGq
39G6U3v36aJ8u9YLxVFcinR+RVGcyTFenznTpbXCAUqS/78WpXMmyIXnHTF5Sf+ml3QKRFr8Omjc6oWa+IUbpe
S9JGabjhNyZEPuGZFoyPzrjrTg2463qm3yQtOWb8I9Rf0fld6oQRZj0kxUy3xhKvp9NYUIQkH7dGpjh9anoI9y
Wopl9KMGgN01zn+xx6ZcyIlPz7/h9rZLi3QefSqqjj3Z+FbQLj8Lc9I7ym35nRFj+70DuQLfO/H+wyN9Jdqfjo
9dF/FomfnWihH/Vh191cBMl+Nd6/MrLZfwrPvxyGT67Ef9qOz6txae0+MUXLPyL1fgFCz5Zjf9BxM+L+Jci/oW
```

InxXxifFm/sRlPN6MnxHx09vx34v4KQP++XEd/3MLPq7Df1eNf+bDP52Nj1XjHz/p438s4id9+EeHDfyPSvHRb
Rr+aCl+YhV+3IR/WIGP3DebPyLiHzxm4n+Qjx8z4UcfMfCPluJHgO4RA37EwT0MGx+24Id3cocN+LCDe6gUH/q
bKv6QiA8+mMkfLMUPHtDzD2biB08yDoeaO/B9DX9Ajw+cZJDDzX1fg79/ltsf2c7vP4UfuFPLP2DGDzi4fTDat
xTv3XOK3yviPfdv5Pecwnt2cvfvLuXv34jvd3C7Qa/dpfi+XWb+vtn4vpPXzjqucbvM+B4QfY8P/00VvjsL33U
Y36nFO30+fqeId/Sb+B02fMd2A39HNd5uwLePGfnbM/GYEY8exiNmvE2Dh7cK/PBlvHVoFr9VwEOzcAI2JWbju
IhjIt4S1fNbRBzV46iDi2zHgwMr+ME+PLAC9/fp+H4T7t/J9elwn4MLg8jwZRwKnuJDIg72buSDp3BwJ9cbKOV
7N+JeBxcoxX4g8l/GPh/useJuEXtFvHlTBb9ZxJsq8PdEvFHEt63CG7bjW0Xc1YjXi3idiNeewmtE3OnDHRb83
Wrc3mbk27fjNiNudTrqcYsW3+LD7iIV7z6MV1bjZmzimzNxUwZ2sRreZcONDZl8Yx9uqDfxDZm43qnl603Y6VD
zTi12qLGD4NjJ3XwYr+AW8CtW45uWW/ibVuHlyzT8cgte7uCWafDSugx+6UZct8TM12XgJWa8WI8Xibh2oYWvF
fHCmkx+oQXXVGv4mkxc/R01X63B1ZJ/vqPGVZU5fFUjrqyw8pU5uPIsVzFbw1dYccVOboHaxy84jO3zLbx9FZ4
PRsy34PkObh6oPs+Hy+dW8eVOPBcUm1uFy+BWJuI5S3GpPocv3YhLijP4kk5cDNuKM3CxgytS4UIhhy/ciIUCM
y/kYOEsVwDCCsy4YCc3W4NnO7j8YjzLiPNKcK6tis/txDbgaqvCOSLOBqHZIs4yYavFwlv7sCUzk7dYsMXBZWb
iDKDLOIXNAK9ZxCa4meqxEfQ3HsYGWDOIWA8M9DlY7+B0ItbCg9axpA9rgEazHat9WKU08yoLVpqxgq/mFdsxD
/v4aswBM24BBqasBjOdGImYOcn47tnHzP9/9g/9Rytww798hP43K53LZwplbmRzdHJlYW0KZW5kb2JqCgoyOCA
wIG9iago4NTIwCmVuZG9iagoKMjkgMCBvYmoKPDwvVHlwZS9Gb250RGVzY3JpcHRvci9Gb250TmFtZS9FQUFBQ
UErRGVqYVZ1U2VyaWYKL0ZsYWdzIDQKL0ZvbnRCQm94Wy03NjkgLTM0NiAyMTA0IDExMDldL0l0YWxpY0FuZ2x
lIDAKL0FzY2VudCA5MjgKL0Rlc2NlbnQgLTIzNQovQ2FwSGVpZ2h0IDExMDkKL1N0ZW1WIDgwCi9Gb250RmlsZ
TIgMjcgMCBSCj4+CmVuZG9iagoKMzAgMCBvYmoKPDwvTGVuZ3RoIDMxNS9GaWx0ZXIvRmxhdGVEZWNvZGU+Pgp
zdHJlYW0KeJxdkk1ugzAQhfecwst0EWFIgERCSCkpEov+qKQHIPaQWirGMs6C29czpq3UBegbzXvjx5i4bs+tV
i5+s5PowLFBaWlhnu5WALvCTekoSZlUwq0VvcXYmyj23m6ZHYytHqayjOJ335udXdjmJKcrPETxq5Vglb6xzUf
d+bq7G/MFI2jHeFRVTMLg5zz35qUfISbXtpW+rdyy9ZY/wWUxwFKqkxBFTBJm0wuwvb5BVHJesbJpqgi0/NdLe
bBcB/HZWy9NvJTz/aHynBLnGfIucI28D9wgZ8QpR86JswK5IC5S5EPQ75GPxLsE+RSYNI9BT3PqwJThHPKQ9yn
MOSE3gXFOwsO5OXLIX2DmJOTPdshr/iPymj+nhaxfjqvBu/tZORN3a/266YJpz7hhpYH9/gRmMmij5xtsNpn6C
mVuZHN0cmVhbQplbmRvYmoKCjMxIDAgb2JqCjw8L1R5cGUvRm9udC9TdWJ0eXBlL1RydWVUeXBlL0Jhc2VGb25
0L0VBQUFBQStEZWphVnVTZXJpZgovRmlyc3RDaGFyIDAKL0xhc3RDaGFyIDIwCi9XaWR0aHNbNjAwIDg3MiA1O
TEgMzE5IDYwMiAzMTcgMTAyNyA0NzggNjQwIDYzNiA2MzYgNjQwIDU2MyA4MDEgMzEwIDU5Ngo3MjIgNjQ0IDY
4NSAzMTkgMzcwIF0KL0ZvbnREZXNjcmlwdG9yIDI5IDAgUgovVG9Vbmljb2RlIDMwIDAgUgo+PgplbmRvYmoKC
jMyIDAgb2JqCjw8L0xlbmd0aCAzMyAwIFIvRmlsdGVyL0ZsYXRlRGVjb2RlL0xlbmd0aDEgMTI4MiAvTGVuZ3R
oMiA3NzMwOCAvTGVuZ3RoMyA1NDQ+PgpzdHJlYW0KeJycvWOwLV2XNXhs+x7btm3btm3btm3btm3rHts+/bxV3
1vV1dX9p2NH7MwcY66Rc441MzNWRuzYJATySjQCxnaGJqJ2ts40DLT0nPiyFjaGLk5KBrbSNIomZi74/4CsUCQ
kyhbO1ib/i4UiUTVxdLKws+X8d5yQo4mB8z+AsIHzP+HK5i74Ai5m+PiM+AzMnEwsnCzs+Iz09Gz/DrRz5MQ3d
bQzhCIRsrP3cLQwM3fmxP+vXXxyFUU1Cioq6v9GGDg4OPANPf7N4AubOFmY2eKT/g+Bf1BXE2s7exsTW2cufKF
/YGtrCyN8M2sPe3MnfANjYxPjf2moGlibWOGLWlhb2NvbueKTC1H8T5V/EmWg+eeLBYoEX8zE1sTxn5L+Y+C/z
BK1czQz+Y9a6NkYmfDJzZ2d7Tnp6Ez/oUz/RdE6mdLamjjT/aNJImJrLGRn869snKCg/jVY2MLRxOif4j3o/p9
uW9naudl6/S/Y1MLW+F/S+MYu9nQqthYOLiYSwv8n+B8I6r8xMxNnfBZ6RnoOekZ8Ewd8E3cjc7p/nVLZw97kP
0iGf8EGtsY+XvZ29vimBtZOJj4Wpib/bKC8nAxcTfCdHV1MfLz+78T/PIJiYMA3tjByxjc0MbOwhfpv9X9gE9P
/PJYxcHa0cMfXoqf9x0N8+n99/mtP55+JN7aztfb473BZAxsT/P9V9H/RgoJ27vheNAxszPg0jOws+Az0jIz4H
CxM+D7/U+q/TPi3Af+ByhtY/J8E6f9bUsLW1A6f4z/r+MfA/1MLPp3rf7YzPvm/2pkC/3/o/5OhnbOFkck/nfJ
f/ahNz0L/Tyv+s2H4f+3T/8H/V7f+/2vQf2kJ/Uvpvzvzvzvzf2Uo6mJt/R9ukv+nm/j/2OmEL43/L0OtDRz/d7yBj
YW1x//LiP8VqWbyn5fk/5eShLPBPzUI2JpZ/9tofDoLJ1ELdxNjeQtnI/P/bLX/Q6jYGps4WlvYmsjbOVn863a
BT8PAwvD/JJXNLYysbE2cnP6Z0P/gTGyN/+dp6URsjeyMLWzN8JWc/+lpA0fj/wL+RRu5ODr+Y/F/zPI/Y/99b
GrxT44mJu4mRlDOfMHIj0bW5Zp5P5/yzV1k9JQaKttE+48cTgDjXyV//lrHUze1oj+UKUXZDJqUzeGv8DbZsjH
xg8guDRFQBRswHqRCBiwCWM+chX+nOsysXSMrhkpKGfWn+jQi0hVMhISYjH/O78xHqbV9XfIQt9tKxTHz4qD75
on2PHFloBuoaFLNQ/igY8bIgoOspdn2+wHiPwNIqoTg8vhhIWDPbPPFAYEDobv76vA4RFCsnnTc+9AX+LZWH7x
pHWAVhYTL5ABgPf5mz+YW3NVXOcuskMrnzbzwV4QNsBH0nDck48SeBd4MP04nWJgNG/lpnfl0p1wNnnwpF9rzE
TBZY0g8hKYnViWS6WmuhRv7h7zrB77anmrscxB+fM3LPgE2QfNDvJ/hW6CDyvqBT0oAeDs39/kxbI29kCjWw/J
Qctsyes05RAlJH2CkBkdceh6juCkm42AUGhZNlRPUN0XPyjDPwEVUjAjW06mD/Bp5Bak3itygvnmVhmgYfoXVi
x1XiGBXT6B7l8nfrkqrHB8riOMCjr2qndzgxyJKtD0AojWwmZ7mKEX0bU/nK6IhuJqG72kxgQn18VfuoNx6OgF
/BfmXD9aWY+hz/c5azHidHQH1V5kwxufwRAHPp8bWfeo6GfeQFMaXqk6HAPcr6av2txuZrpA2dM6ZHXHRPCTWS
B+BamdGjcBREwDTY51nxC7n7G77aSb0GAxVWrA7eqdwzNtpjp2H2yujoHPcgouRS2+o0MPoF1usDUSUNh2O7i4
f2nU4BYAbtSJuzLkUWaqxLcDNinCCdBNdudcFP+p/sTX2gWnLRRKA8FvN6RAnsFHAtSCFjxJmg/4xQHYbJhWiM

```
nZMu3PM0X/TyJMu3688CUVcrk8FvUgC4o1kYjA8pQO1pVKh9ad3hxcM76QS6UL7OYwxpJyVxNchiJgtuu1yu5T
OCKC74jlxIonpDLdNsGL2k1Q61k6iStOV4Vdb3Os7WhGdxK8uajqRp/9LQY26Qhya0O0EEmYViQWm/83CEVooO
7qIHAQmbEPLCeFNurhrbV6mEc6tKrbgaeMS0PTMygLAfC9dmYkMKSUTRvlIGmSb77eukEPPKJ0ytRHHfWBI5EP
4EYlFhgG+0ItyQtl9d11cIhaNbMmKx6Ei7gVhhZiCu/Y8XrDaullOked2nI871+SvDwUzP6mJuNWgfLZeNhO4e
i9xBn4w9hj4R2XP5BlNjGO9aXIs4Q8ztMT0StTezbm8KifGOE3I3Cx67AnXVcSZoCilt/cGcPwJRDyYVpBEf1T
ispP/mVyKGFrpOl+dis0A4gYkkAVNL42yLVUEop6zFjxv/iLTzgohkQLswjMrvJ1U9KtHHU/RNABgUQ8iRcNTb
B8+ea63o0+ABdD3WSZXH4QjIo/dK2UWEaRa9fSPkmqNESyanPnq38U3YsvVYZkktDceOXG8bd1/Fxdt2r66Th
OVnkMwPnNweZbXh3HHO73mD5rIs0Mf8sa1rytgiMjUyDj65U7gIxi43EuGn04J7HOuYs3ERPDy3MzW2DGJ5CRl
9vUbOc2fWtwIgudIUcrNQW3tFNHHsLk2fI3jr+UIlqXPX1kG/lLKkt5R+3X6ymNwfhmtwLa1i/KO8m95NFy9o+
Ao7r2CdbSBlesXFz3C0E5XWuR/AbwWnaAKAbIeAu2YszIsNy5F2LdA4IZQXNg8LREETsQDng1PerjyYCBpt/Dz
yQJzq2pbOrPn/j54PHzP40IYlck6V5eqhrGbRXIeuBQkm9DvQM7/V56CSPsXUlY87HfijTeBU2C9IIkKq57lID
i2IdAWOITDYPQQllE4H10t8FwXtXzt20r424DG4IOA3VIzxDUwbV9Kbw7zpdZb+hvU2T325sPf8ynH0yD5n2B7
Zakjc72fzf5oVLabd1qk0UfZRymLhLV8UgPQWLGVZHeDE+lvMmfV4eytYwYizFMKZNyBdiWbGbQKPXJJfpbDR9
sqMx/XJpz1lThQzrjW6UuAPq9bHAhyaqGQk2oViftB1009A7Qe6AW6uzbm04dD8AjAmPymJ4cyjVL9uZddP+W1
fp7S2Ss8KHDrM3jtf7B1QXG/lVkqGp7JeXSMk8hI2070PujxZ1/IUJsvZ1qBohbuR/qoPdOmdlzvBGTGsqMMdy
CZ3MtSipoKgGGkrrsV2JyzpITcmXvu7uJGRDTqWOIPVbivYMsThuX+N18l3U5r6cYP78vXa16qJa5vjBNI6cuZ
AxPpn5x21CP51haBK2ojamc9r2Q0pqzV6e7TLUoT09FGtocxfOWhj1NdT2dBUlPdx9b+dxeEEkNrYyyCzH97Ed
CN6WJJLjDrfbp79JKEVF8N9idcDlQj+XX6zOU3y3BIPD0vMLlZI3bT0Nkis2rHSicWjW4LIWQo/7jSwk40w9wy
+Hy0bKS2Aa7tA6+h/6oYPo3pUtXvVHYB9opaP371XHRvZXwpoKLql5nQ1vPb6MzcnY/fg2i0To/FXyz8FpfwoA
k6r7O3fdlUg+WIVlyyUceuChRyMN75mauo6EqB+xldTjTbUPMQHTXCbpjbYwAGkiSSAQt2XunQ8c/uYh0f1h93
eMByoZAsEq8d4Hw8y6isI2gd9t9YFqeMHYs4Ni57h00p0jU9xtosoxIjDqK3HSEYV3v7PPDQYGJx8mShgeHshi
d5sTIqTiZ1taMgjuLcgE2Ddna93FFiYUg3AhUxAMsnUuXPE7dy70lCHunWk/tWELSpmcxYoewA4lgYfBduOP9p
F43JWZpfIfSvk+VZMr2U0RVfAdxVbBO9TqenL91ydmLdYZa3LR5qVgh6SlkaqhI7v4o/0fO4gDP2+fhKVRB1Vj
6GaMdZq7RxCz1oD1blSGnZfw7hDWYHVoEotF/b2KWyuuEAkjIYX54r83UzvnK8OE2T/2MnncIgrPTX3qg8nWu2
ibK9uiBtGVVGT6UDIM5eD4CsEKZJlyJ9kod0pza1w36Cock53vDOrVaA9/hK7w4kYhTrvj0La/XVNXPIm7Qbx6
iynn94nt2eGAtE1aSqOD8BDmfrU3HfGcMFTiJuOERqvMkXCYLLXRkPCF7pYDY7l78T4LzfS2N/mWwoh76SPyWg
7xAAO6cJH6qAHkviXuqK2HkLyOBtt0+mGYnKnXcX38WpEhMX3KmWlEZdAlYgutq373nJIcq2qfVmA6JmNXWP+1
dJgK0BLvZ8JUFUYiMEebadzc7N94mNmcjaMQyE6YIH1VrlmKX9RbUUEUs2yisMNJlkFW6n2Go4E8eatcyeKzIw
45O5eY/w3yUHpKq4KqhiACNjjbOmqjUG+YBoPuAXxPh5qq9HL/pT35WyDC5+wtEacvB0m2lZMSaePYmSezbJ3y
islfXZeGsweQIZwZqCFRSGq2UXNxs+9ypmJVHZ92br5l8FdO7FY0phc3SVUdy30TMDrLyqGqDp1OJdozgGGHhm
qBal/3D4iUyjfVwo+PKg+hnv7mmm2i94EW/N07FyEQAOcFwTijcZVSHaVxmgyzIM5aQrqpNSaPr+oI4lGYLJvh
6QROnIgn4S49EdmjDMHtyNfFlEo4oPFz+Sxg15cH7XY2RiZWOsjnq1Z2XYchAIZ16Knl6OtqGOgAPe5mxRPFbI
PzxxXgOZHIKftz9KYlxynICf5DSMs+JwXWM4HD2ekKsBTFXWsNFXglmP2ukziZH0BSUpC6HVUDLVOfDiy9LMZ0
/HvEHc6t2QCHQmHNjA8AKVeeskaSW1kfoF1v2lBRVvg11FcWZTGaAintthklK/+pe5kpq4uRibNDEGAbXN28OO
K+tjzVLUHUWipRmNvkRVl/fZV/WBtTXcauwhaM2iG0cdqz8Pehb33i2QEVOp2dq1itvZRjDDW4T2s+ODrVtoac
0CUIsY31uuEJZfkPNvMvbvYJXXmTkLJa5tYuiCzLnGqFiId2V+Wa8ReuZXkjRPvY3jZU7i418/DpHb3oFfB1Lf
uk1nIb7O2aAtv7Dv7gLUbtRQSSwXL45Dkm5hpQ4+sFOq7jqmNBrHYxGQ+8Sf50YIXWicg3u6WUimGD4ZI8c7MV
3Vhv4tzLifSrRFoQDCS7iBGQ89RY6JIzqJ6wDV2L8AiTMAUdKkLuVX1MH9f53Su4mxC0olfBcwttnSheZw69a1
+HMsHnQlr2HVqI8iqGkBd6sd5+CYMpR6xYDNyWGab+EJ/LNSRgt5vGEos/v6djJ3IoVWcpcjNob0o2XwOBwb5m
MgX8l0n+B+8ywTUQG/h0s9MGwbeA5Rn5L3HNeGl2Jm9YQWFE99AF6T+/3VhxO4EjcIVOZvqUAlPKu8qCuWlV+i
J/xj5LYmyLY5Oj6DV33kTTq/R9cwfE5hzlyfBgyCvrm8BDCwKrPOPiNKTDiDakPdmhhGjIiPC8n/j6IhUuEreu
l6Xv5+DTqvkE1HkizXwhayOfmc8E4mGCEnSN1G6021gMIEfuAvLeb9LSfse/uk2xijFljdADciOFmFRGALGFWh
0n3EnaLwlpFE+jjtN2FuDcw6bMXjBRxBOc1L79Kv1evs+8IKEEz35N0Qs9NONelsMMPEf1cVN2yelTHppJ6ley
OmCpjTd6lsV2KEGNY4d8HCWrDOgE1fGHdAcKofEmwSW31G5VDzYmEVH0JiPdplQuxoJTnRgXzI7sQoEZYZepvk
qMKwA/ZRx/qutr+To2zpPRLZgYzdGj6g5zYwDnx5LSTZgzvaQxBCSZKtUZcyfg2tBOv3Mk+8IFnZdxSbpwyBsU
OkClqQ7b3+iu7OvmNzy4e4zVevq37aqIfsDdNIMYZ+xawv2znrBYYbgQWB5h40ncsSzLQYIdxJHYcULmpT4g2X
hZ4fxsEGJ12ZV8P+qCZjsqOPBeY2WDIpcnkdlH115Ydi39Y+JQzMPFi4XHTAdhpj3poRLwCL54/YnZeKvpMH6T
+XDyISByOfbFLLWwRoYpDGKKtsItxWe0lL3Ybsvdd0wfUgTOjHx/PICiEl0+7Gy1IM4J+nEupktMID+nQTeHZX
oFfC/6Uw9LO3sUwJ4cwP+RaaXrq+5uilTR7NhTrXEODlIgTyinVxPiib5WIoDlP7Scd778AZQeULR1nw4yP2uT
DyBAgP/IeIxt4kOEpsRkP1OjpirWEEQMquWF1Z1N4YSD1KaF20nVZD7EZr8SGIjNzIa1kLNVM4LRavTJkhg3sK
FZWgBWoCncUZByswt/07+BA5+xz1/W3qatlBayR4s02yLqMr/QSooqN2FGqkThaAczStdNN6/1SO+YwhbZu6Ul
```

```
TuNRDOUlTzgsOrHP37nqc5hUT/MUtBZiFz1aSd4418MOFi10PFFBMwVgdLbQwxr0Pqi7YijGb+/GmL/Lh00BjR
Vf5hpaVEDejQKpYbjMLFIux10d7ndhwmqgse5srOswm0mSSsy3M+a2xWTOLwcblxdFyaZMHfjuqZjEQlWkx+0M
wF+RqQEGPdJYjeWXTRQZKJlezjXwBwQaWQcBYdZTKwyeEug51MhZRXPO1UEvZkbYppMPXX+1pXSpC32m5NPT7L
P/7Y39AP25ySwAuR79IwFMqBJtahLdFZizS4Vz7tFD+4PEqd7Nvo24hmEInuMo9YCTUScKFY32sEkREPvDMjIq
LOmkMWSdWCtmtcFh2ZTLcJoE9HBpm/yaiN2WZ5YBE6HRcw4vtMFr3hRM5gNArNmTbbVOapz/1pe/bOgE32Hhza
dd3hVUe2xEVfyswAyOMj7KxsXCV4W5SasnV3iMudr5y2ID4hxOv2Y0gcNNoaGEjC1tg+EGm02ncgWfkhnJNg7o
IJL4/0Bt1X8Z2+mEIZf5xpK0j4RbdnTy9Vph6lSCdzUkVvLFMCp38CnfKy6YEATwqegQa/GkHw7kWdyn1dCIEm
NGUlhmTyO1oQ5H5gOfgeY8C7Q8OMsDsruRv7O+NoTjk39hLJzvBWiszElq8N8orwAUdzAw42qDEGPq+H5mThuh
FdMNoehpZJmfGWbghj/2xYwJBrvdB5v6OMKdP5BiXyAi4r2XLRsafYl32u8lTjoZ+wSeANMwtwVjdOv/ecGj2+
dUeP1dH9baVpVsTBcQ8P3QCtwlFDecoXiqjCkjbyS6maLk5MXeRaNfhsDArB+ccv1QumegJ18GaCEdI+71uSPD
x9zNt+4NFr1o5dj8T1HzvqbD878q2X0LZ77yzog6i5J2k6Te9Ww8F5uxVGfZ0V13++qYj0GEDJs4IJe5ziuvfF
aeZsqgC/MXWcXh+OBmqHAcqH4BO8wpRlwcBSFpgAsbZtldFEloa5OTtRevgxvSk5Fca9toRErKLKb0f1KYQMfT
MSQJ3ZqosnAgUbaSeUabAH9v98p7cSXzahoa/MDqCg9L4LxCKt1NHa1sS2kbp8cd51dlPXF04InRey/e5hs2L7
D2fZGyQqhtZ+2y/SgNTAwct4oNrL+UyIX51DQC5fomqoyxSpa6ZR3IboG1wuuDf9GpzJKDUE86e4NWPuxv9N+U
Qbl2vs/5/KLWCN0osUq44aqSJH6lxG1wnR2E8+15JkavDhElUjbDzEBGpzmrFc2nWfEoFO6Kxm4MxLgJeS4YIk
4mXDrjtVdKF3aeVdyyzpmh6CYGoFPhv3HrrBUCt7mt2cp1uR1GAkok+NSXWeX4Fj2ITPv+uQL2JXHzL4tCPwmf
1uUjXZeNqdyo7WjJGNUS/UV8T+Hxzsa/+UqsGvxSvoNO0Co+W04+BvXuHa6lrjRTfKPMm8G9knYIsHdUF1TquA
d6TE5Mb2wFyzeDIbHfiJXTioN9aj5SXa9DYKnqWQ+Ncd/Ic2hN7j+GWtl/thb6X7ezmUUFhDxi2rzo1CZ6PxRb
6XUcgmtESefRAMGz5Li+H+rOwdw2BOin8FqmsbUZjEYgKiGTeQovNbX/q9a9yY4RPGDB++o8g87jbzVQUw6RW5
b5wuJv4oEy3OneplrEIhNEIjxIdq4XxCOg6n8Jl/hD0oQEGv5p9Q5K/HYmk3gXx7zecUahR8HeREA+j+lXast8
gKjTnzeTpksYeqJbBeIb2KDKWS+RC+pozW+QOONy6o1cPNmNdi4ksxq/KxVXoFJB3Dn/8EQBL10emMBQxMt7Oo
4xFD7luesYzJw9QJZ+iiAf3KWe8AX6NI++bzXH+MezG30p4ycMhCwHOqWKPLlrfCcelP9/MSs1loPfHooDw7mq
StIyjtcnNPAYVRx5qrkNmfRqimWZTBrq72uNyFXBc9El1hHOmuiZpOXVkf0gD33QbrQ0w2RSoDw/KV1MSEqX6A
RaklkeIHpugXGy6KAawgO5oFm1dlkNRyatDHMBbjedFg6ay7j0xVnlcucrN2ACQjeMnQE3ZeL21bRxMj5PSVTN
39b0u6RDBLz8DHzqQ1A5A5iWFTCBnSUvVsP/xgCN2hZip99xqoQ+OQULCY2U1v8MntC2t1PAB2RGHIRkAh6JAr
cHa7b3lAttLtnGZeecf1pjDdyj3ssxjslPTunJGkJYb7l6Chipa+Z5AQfWHIPQxOronQhrYpSG6C3p9t3/kDCU
Gml9/Pjn8GLEm4DE3GXINipTgE6OqOyvs4zF9oL87TRG3TOm8loE9/nF2drmRIXk08gdEmGX/mwj1J3TaDIDdF
w9RLA20o5EwR0Zjm95xw4dWte6JMvXepJatcl61eTLRS4RkHf13ECUZ0BLXuuFu3aBdQr3sV+ctVvP5tvrHO3K
n+7EzKoeKKnZQAOMW5mVIVF7788DL5aHvGznqEc1cfwx3dTeItVTwboHkA53SjXGQd72g8KpZStl1Nw3ghVTOa
+zGk4hzH4FpRVgE2r8c36AA8DeqG3FhWXrjZIAOIiC5kxRcsHnOmiMLquS0yJ+asxiO362pIZmWAD8LdMKVZSV
pATRSAy/kLxRusVDwpxwcoO95LlvN8i1OiR0uZb+nW5nVtHP+wC8K37oMqwEbpOAeQKKWrUZ/kaqw3xyYujfcS
bnB8rsQAQdChUnrBKDA3GjWkw0suTp4wi9YRlYUydDjZqZHnB8YGcSQsuRsV4scXf8I0GWCd0wVMHnHMCkbcI8
ltGculC07ZY3vw7WmuWIFCstYaWQZHf2p5e/RzF/4qTRhHRSlTemX+vgpc0gs8ZwvcTA8Prw/nzFgmQtvsYqjZ
BS107YPRQG16ZdI+zxCOqsDjCg5iZ8QR6t31kcM3SS0jFEj7TjmlK1ysItrHgooUdQIz2I5vpsuo7MaEfaKcBV
wpyd3uKUNuXjTgXE+kJwcwdFdiSYn6HD3r0rGTND30Yo2A33mOqC6IuUYT5gAvjZhKmbs0/ARtUQCrIFyiC47a
buHdPeVVG7gai5/32qMQAsf/rx5rmkhUNmeNrmScMJdOIajvB5SsGryNr7YFH2stHBVE/lT+nN8SntpTdluCbf
+lRbRNyhburqu54IEZRDOgPeUbPCb74oDGEIAPnOdf8QkYcmbu7n8ZHuvhDeO3BtkgXab6n+FuJtVY8ndGiP+6
uSrs8b1psfPJb2Gemu8Ircds1V/NuBhUhD3wLZruSwwVqXPifrVrWaUmPHKrOUYTdEMbLfSpisJmMtRz5cJh3R
4iVcoDcIsevxnxcxIS4OJIhOgGhdjNTF4bTem8QAQrgtkpSbevyTTsOe9Bq57Gy21Xi9EMPgLueIm2RraFKgKu
2Fa4iQiG20yVQUJl7hNexJ0Mrd5X7tupgd+xHBqFvjY3sH9XjM4u/KWT5ug+8QzwnJIhhCeoQFgCvTim8t1n5g
4DvhnbyafCLnE3yOSOlJSJf4y+GafDuDt3oJ2H9iOva58eNoPV0e04eV+leXdJgwH1rxZvdDThkhCTKSdbES33
TpQScMyRbve094S9QK3UlN5m6UMsYPrlwfyim/FxIDkbsktgAwvE5qrk+2EAtJDyGJI2H68IciAJXk2xaSm61j
nU/xqlbpLfGvyg1Kre6R8mGF9rX1r+qpTF/jSB/FFC42Zi/Rzd9VUfqhL8iyqh0XFteoFrRZ1/OGdO7kijhMB+
qrmvLco//y2kqnMbWI+cXYXNuf6ojfcYoXXEP+4qPLEz83ZFcYdCmcdvkNb24k6O7ymVgC/9M2LIB1q+oXKPWb
QTz0KIdcIYaqbUC44jmAbd8z7g24U0wTv1sfn2c4X7QL1iaBL/cmaZeC7rmzEmGGBeIzjmV4Bbr3l4ME4uAYuS
HRmz1s+jFDF8En0fCR8cF8eqr56XBTxHipMhQx0N20DuD/h7T2kLlzg2UDXvmKTiAsWKTFIttuMqPa+eHijvfr
FJuhTflXpBx+A2W5fbKhQfgMbTQG+w4+UWHM2KcSLnvIPigiBNTHlwn6Mt5Q0J1eYCvv74PrB9Qn8yK2aFc55S
JUmkqdEGSVsnt0pYpNn0AtwySS3m57L08eGs3KavQg+rjGylE0SHmuKaIFqA7wS/jJLUSyi1/wnHKJEX/sf15K
N8Z4WBQF54Fc94nHH1WyKq1dr61kWRpFg91EfnjEEwm3dEW7SiPV9N/JadjWQ+FW3bs1S6EInZAejoeQtaA8ZM
eaarc18Yy26hokz2gIhfVLil9eylvyInfCoA7tZgagAFYO/fYfOgXq3t3RVB+Ew8haL1zRfGF1Rj0XrWj+lamF
+3BsM97TkwY6I1FxzfkNQlyiOJDh1jAo+0o4SdHE5VaXf3kd+e3P0F+CvqHrDVEFts4YwlPbZwneIyzvmPsqMy
```

```
obPQub8rwSrvNB6Js4SN7suXajjKabd+FXtTtQiJj0f9GCcnVXBssRjSlcpZKPqcitbXAHDXp9y4SrhC2uXO62
dbgCfs4RVx+YXXn99HFoCLYCpc+NeOWgQ8Sot82ICn0BzXmYM1KmfR03D8SYw5V7uq1O7c/+QYE10aUIDWdcdn
BhBt6im8VItUXIVf1IwHU0H94kSK9Ih2kZI5gGaSMs6JIqGIJGSLR6RGxr5BSW6JRHs+7w4y1UouVOl79kYIOG
DkoiY/VL0/8ojDrfm3Hjq3UXvLHMU4gKDekPOzZeO6+CtQ/hcP++GcngFa5FQNXAEPU0FjwqrAf2pPZKP8/ksa
HZUQ+psm1ltUuiGWQF98noyCwECO+LemFddCz84hdUVZNu2NmTTAOnQ4vN4A2fpOkgSJVeMCLvO3H50E8bMVlT
GOM2E1WbYeS+VUiVjLdHcS0MBF6OK1j6rvhaB7C++86BrS8gJ3lLGUQxl2QKMne61crWv12IKni69fgdbP9mN7
X0Z9BwXYf7DkyPtRQLstL6MqhSutxjnXVpo7+jiyGKHWZbAe4RwDW2Llv7WIEA22BNDz6hhVt3LEtUR1dtYZ1z
w5SSWXIPRw4ghzJ83oeTeAm2UUZTtjMGxOUOyPTXT0BQGwOtLp/pmssPuF60XfHTqjqU7ykv5SHwz0HfB10TrF
Kso5fxCvKhNAuIN+PHCntv6Yt+pt/U5OpNW+k7wnMNB6FOIo/NwBUeBQIeszGUxDLamL6LYG9i6vWg4PMlTtOr
5ehtBcJwUB+Lg4U12xE74eSOXBBMNdZZV+j0etyfDoo5T5uNmgzHRgVRF0z2PukoD483Y9LNzdh66rR9BZWVKX
x6sybsOXvvkLre73HT9foQOHxkMIF0+mQdKV7NpsOddz0ww8mRGaEGVXAejhb0MfO3We4Yr+lDMFUWcEZnjtzM
Bi6p5C2AvtMb5+OoTk7BcbGS+JjM83mRWmDNRDFWpo6HNRwvVrEOf42CNw6JPLMhiM3puRwbobrXbXWg3pGiy3
CF15z8vJ3rhBMGpHWtAy8Rv0KsnekeOpam3WBKbzbeDTTSQnak7VeJKQM9aA9QF/i341D94kz5Magq1FtslJaU
zmQJWNE+GtEtCG8Pk6bYKxTVmbpsjYTrxPsC6Zna5BEPMxZ5YJO6UXXuBKw871sQVtdLjTknoECN77sztxxXwl
gKyiLeuFzG6UGdQDrcssiELglc/f9puzI8tGTsDjeDditM0g/UcDe3TQ7JB7KSr9N5Y+TH8I9F2LFaD4iiiJuA
BZq+0PJN30y64w5/jU/hZAnOpi5kgj8bl8Ct3AiSItqmFMp8efBJs3ckm5ZOiQt/o4G+KqSFw/aOafvG8gDYrr
5DFkHE3wGJ+Blw9abkZI2kJfFv3vaQp2n/y3a+WuVGHuWOHCCYAvnfqmSOfqVKoaTSonryAoaOD5xKYX3HPMl8
2Y2l7nri93fSsP8KBY+eQbUm245TILExmBGqRReA8JN5lwUvYykwLNfNSTAf+k8A6ILPugFJRH7xcAIZuydV7U
CfP9DSJbWh3FiaU8/Ca7XgLg3XOZVgtYXy5dLXeFIbeTwzxCZJeIxtQVtPjZt+8RFtEWJihY/cQbA8Vnkw3FQA
cxEsQE5iKQ1DO10FgOvzds9GBAUYkn+oWkexvWVn55km78W6lLVv/dfUd6olGiTiH4iuNAD8dcqyI0jTcwb2Am
JWTwhyR0uoMBV2Uc9XfJrpKfXrl9dtCSPN9YnMcRJ+JLVgH4d58hwdpUoo7paXURs6MXIxc8Xsash0mTJx4mgp
wPqxrWFG6TGjwZpbQLFSTjruU3fH75BfW4RkVPin5kqe/K8D7U2rRllTd9m4cj9EdIOlmlVZxGuts70yqvXsAX
t6d8/GXpUU/w5A0zZzeU86U5YxZI0WbiRBpxohRXWO2Lc5LOAu9ihDz1twJLB/KhWq6jE21GDEC9/d2PMTZiPg
glO7bOM4bN8HBCyfZMyXK2ciS9QjfBKfCNIdNpu7YCov6uInGpe4YI0UQBnTak9Rqlv2GC6/78o4GFeSp87RbQ
QMwSj0lu9SO6d0KJEhYbtLbgiu4Zy3Hdu0KIdoUeZx60bj99/14LWSgyxFbOlx/B8NtPxfdicajL0sdKsdGvji
B4Kr863egIyS+aEHTA/IOz0pseDvTHY6xzbPk7FiRopI04Rczvek9pXbnNeONQTL0FGa9MlFP7KPISABH8VnTD
vgLMMvMmmpo1dGtDla2mkjYNsOa6AdRTOyS+/Phaa7sNu1vC/GZbOJTxp+lddU2SPbFkJnS57p35Lo9V40eX/o
n7HNV+0v7oebHSGGYUibF76nEqoQ5jLCK/phEDAtz3sEfeBVBHyhAnQoQVjPZhJkxxjSjkgf1Mlkaxb/eF4a4i
xwqB+8bTP6k/F8er8IIt7HD6ji+puYnGoMRo7m/gx6CXyUwsS2SbJa3asZ+yo9vsi55w1d1Y/3zAHyuaOZ67D2
H8U8IR4VKRSnAC3Q5cl4qEmSBmDbKPHnSAtQNxJF/vpr6I8yaQmD3XY6tCgIumyE+0z8Rzzod9AIn2PXWOi+9U
rSrwvUF6M9rfXBPHgfxK97z4it4H1+TLa+XAf7aIAeffTT9NspaHlkJtstD1KWC23JOsm/IsxY3OE/k1KfNi/V
SqVT07UNGK6rHzOdg1nmtegGagtU32xvToic48XYHl1Cnkp+upv3tSXgSJd211NgjqSE2lU691Jmj0MxkLxiFD
MYPbsKtPXep+BKotfZ4uBDoSUBuZg2mA+Q3ZkdGmRVPwadvD0YFz+R3sGwXL/Q65emGcEvb6rUkBHdJeF1N0o1
VNcLYHwpEnWUUtNXjrerd7PaG4AzneGvDTXZHSDD2936IiDfiOmq4VD2mkA5wOqOSj/TbT0w1QYMjJi5IBZXvc
ZXWVvyRSmRCf7ojnqObeQguKAbC3c8IrQodOoM9i0/yoE1EymtSN6GnjJLsJKbWukIZBkdUGQDiDLytiVoHJ9Y
FG2sJTz5sE+qgRf3SJJYZQLR/zm7BnDYNOAjVNObzxS2/We/PgMtMkOUYXC49m4cikU6axnK5vQx7b3SuLoioD
HDL/SAxyv6uqWo0TdstZTJR2cRstN6mpxGNt6KZQJqTliRbo+M7OkMIN0ZsT44fGSOk7Nn60zRXoBXnYgIHH+D
8BgC1mUFt/HpEzbHsL/jQ6HTt7+TphQKxvImtNwTFSf1r9de3sNHAWbp1gJbVTKtRDMQbHOr+ADy9TjZY3ouCw
2U20PYTt8mBFl3j6b6k1B2mj+ajH+XacsGeCTX41gAoCRo28GEoL2tomHZWMFsJudSPw3aOvLt7CN/t2nnnkoE
gzIw7gWHGcz4U7AztxbGbyvLkGioeh4DlL86gaS2p8qFHQkCaQtADTkcalcX8bY0A4/iVh1or2NuoHgFN5C6IO
eBEULOeHnwDItKbBFoBEHxji5V4DKyxcUz/5sxN1p/crC93BcszIyPPl6l8EeQghqG2RnsJqDGfQAlRRgi20rc
H4l+Jya2yGVfvSFJrIC3BbwuqMLJO2Pt6HL954WHJMI1rDhuXFdsPrjODydZNiQfcBz9sf6W3ra6nPWgHa5hhm
eKLUvjuzNU/dd4MYqVUNOoyia2Y/a5Qah4q54vOtQAaFyHt6xAtiE5IsUtHAODN7NX7dQmLuS7SP/46b3v6tk6
yZRStwmokQc4JtizRIkUMfvoEQxzmCmujWIrCCdzhxr4gPkzbtzu6R35s3/mHkIlxShV3wFs/ZHrKLJPlfWa3s
jW9UUy2EWAHBuCQ4uPFb7HWx4GmGAbqXfuo6tgz+2t37jMMx3IW9Rz58CFqqi7qOG5TvRhAGrGU4pyUAkzUXnt
FdnizL9gBZElraTUV3FbR6I9OBwbpxeiB/T83A6g9CRIoiJJ/oJ5yqAR6dlgEhVb8jemIpTv4c7y4ES4fnhX3L
M1FqKwbPMTwhpFMokC1+lzcXuSELw6pm/BTIC/wYy5mCBjohNGD8xulEYxGUTx0cDOLi03BsU5Pi4x8mkow6+u
nSj2oILuCqRTOXv69vLiICg5Yg/MHAGp+7DB/RbD8TjAjeFC0Ds/aSpOKjzqF4DSnZaYtLDbUlYAJSKXClnI20
eKmNy+vVdlI6tpA51W7jev3kzjyRi7NqvPIw77N8LXZ26cGlC/CJELdFwFSjizKW4kYtSnTrZK570ntpJfo9J2
3rz5RgaSbV4CLfv770WzBdqjAMpo8i0kNPCVFmhdqgvBrBw/O5NJ0qOfpDm++9a5XX7tI9vRYII/W7z7mEZSz0
r6mCmKiJqkxcQSp4vxLE4UguKvDXZT1ZZWDKiouhH9xsQx/x4LP21XHKL+vbP0hRjwLNcvURqSVk0daAo2PGBt
```

```
wrVPtNqGfl6VmHyP8/nmvOFebkMd6aAFeFJryWe7V7taNteES+MJDlfdwpFMDzTwG+W2o/udyzwEiriHPY6CNI
EFpXzzXq+S0DfogU9NFM/bd3xuSqAa1WX3u6Mt0nf72bJ87T0RdbQjJnnhCfblGYGsabmAvhZwoVLlRXhvZBVP
fYc1UHYH0L1uGrw6tCg+EAJ02VZl1Pkzz49/dgp0mp0dZKmU8M3p20q8e+6J6y17NICpwaQ2OHBq7doD62hDp2
7vHl81s1/qYSYFBPaO5GMP7/Ox46wSAgJ9afkPgYuqEX7lAjUdBRuOue72Jvmuh4wbKy+owjHs64rAnNzMqjsX
5JLTQFJU+qjGglszQxwAMKHGnIcgU24Wv4PYKjXweqOMCCSwI0IbLF45sno+bQI7jP7Vv7E3i7MjI3XWn4O38a
a1je9nzYoYiCL5oOQK3gmbOdQWncYoXIR1scE+EJUbFEDSH5UsnMmZtHXXYE5xC9LFzu/PFiS2lPKnRNX0Ex4x
zKVSoOrn+Bc4HTopUoNgB6LkaN05qTEYA1tpVrgOt+aSZeBmALOG8pJ9r3Wsk9fkknIXlk4CySfXlmPet5Qt8l
rDcHXcC+wc8vftgQ+V0K6npxL2tV3BeCVw6TwAmIQ9D8szQhDGuN7yP7xQZIbDw7PRw5yVbP/SpACCz+hMYvbr
A4VV20bTqQoS0iS1DUXezOJBGk4I6FPz37wEdbxK4Y7NaRJAiHpjPz2NgVO6B5B8k3IL3hUGXLbO7xQq/GQViQ
Ui1N7KeKL6qu9JWnvl6KetoDpz1n2G8ZSPdbXTkfCAR/34LF+xX6RxVKhj3wxcgrS4j4gZoCuBS77YM+HMSH8e
UHOYGTBPedkrFmgPv03P0zFzOmW+YA+NA59OMFyPQgrhSYyGLfSLBH/m97a9ZaTrZ7pJS+3P2EC67FAfDi3Uwo
FFeg3+1206rP83ydm/uv/y6es08YQrnx49bLl7EcrasgJBu8iFJUOcgzlzOilzwGhjMcps81F6rRU8z0rwyuw2
Adyr3WDrLZXpMcegYk5G78PHaCrgxHV4wqw5wE4nZ505NC2NqS9ubXxXDyEa8xoXg8foBbFWG281zSnlPeJ9wf
Q2GQKugjopDeh/o8XNsdgiFEs92q7i30qNXTlp9qMSSwdRNAEagzRqnUaOYs6uO02lwJpuuuR6mLSpe2vQSD4A
8vwMrpQwK4KhWJDoi4P3kjSzRrRfBG9H8uYhEqqfKwt0iK/3+IQYybV7PK5jUGMtI9IUow19d5FFv23vzMzUbx
JrGaG4o36ZdyfiDGFhJHsg9HQaLCJwxFVtzBQ8ESilDDKV9yF8QkS7RR9iTRcKdgcS5F5RfFdwApvM8tISwWfd
zWwlDFweaC157VWP+nhfErn6M2qn8CBZottDDgz5xQ0+ALECw6jp17Td5C7F3q/OB/DFVswADzje29QapKdsIO
uNpnbTTsEEB83jedLPVVwG2bELz/Y1PapSkNYFdJjmp/XprBoGoULq79sdBcwI5msBderQThDmoyXV5+KsGDY3
0GeAknaXjJ871GRSQ1qjT/wMNwSX9BJrQ56gHIoP1I1NQ9znGY6oHDyov4EE7Z8E50O7kZ+YU7X2OaddCIT23b
VZ3JzWl5K1hHCZ7NqQm1y2pGgN8vYJ8JAuW3ywCSeSS3nl/ef66sHFKY3+vX9FbO3TrZXBPsaQA1yn7GnZHiUK
U8qtC7NnOieG4BWrYEkIk/BssnEpnc3RiF3gOSIA66CIvqL/b3N1jjoOux8EWiAHdG7GTMlNeihIqcfm8p1cmF
cHQzD2kRkPqIEEjNeW8l6vlIr5l2z2vkCcvM1h/g6ZNl0KsfeoPtpsmzW79wTCCB3upTWGEJ5Z/u5RhwnPepTm
hsA82xtcNQfCxONEB+mzbwUdX/Gkqg0RevVaTU0XnFqyr2i70CL11U1zY1bwxZ33t3xRpK+uXnIgOyao/nz+L4
wJPVHjbh8/I7ckLf7g2Bt5uQRsTbNUrJqf0oC+RfQKDqilwbf5At/kRxLWznbawNuZ1L9PtcrFZBr+pWmVIvyx
KqNuUirOlBsaOXVNYxGi865GM2YLK9/IkISk/IoL5BKxN82ZKM8OYmyhPvKCtqzIqwb56BdQC3ZkBCu/ftQLKW
/brsJpFd6TUh//j3WmB4pneROhc/yZuHydGYnlJNdJhxBxNinJk5vvZOdLct0JBsQx+/0wF/5wDcAV4YEDpJaI
Q885oa9szdu7GkZIpT7aAc8DsmYtrFN1YjEdOkcm5TflBOZIu0YUZKACTfSl3g7WdlR5+W0km6wthmMQiRMGy+
RSvvsv2OarC5rjQADHFeCEm3Lmk3sYN3RQW0119HNewz4P8tHeRD52RFjKu2CFb0bMIr26swRZGcmlEe2WP0Kw
2rYlDjHR3EQcZn5mY1fDv4M1u4UeNHZhjLl4rV0ySN/ZQW8EAChD6WQ4cyLSTG8iHOsysNMejR0jd0Qfak8FmL
11YB9Tm2vXm0tevJ6aGQxFkPLYrvFBS+rYPshuIlyYqBU6aO5mn3Xbwg1czBrmwognUxL2OFSeqiJxTdrSN9TY
hzuAI8vebRtPinpTaYJpO72S0ocSdUwbJTvyBcm5XfRlF2UAGOEdZHUEVdx+hFOZ1FeeQ/p8rALXnkM3UORwW+
GaIJflXNYdlSz/53IDFCTkc+eQZycyUdE/yEPLLAE+athQWNVufVZQ5OPBAl+Nx7fbpHOJ89WEotm5qSkbt+8F
XohONt+J8zsPGvnCnkXMG2ejBhVVfJu0gio/wNyuUwh16TbJ9Dgw+vJcCgbK/Lr2cpyES2TteSMvRuzvIW/hwm
kkUJgy9IBL4W4lgndDX84Orxm1sxu61vSA0g+u+1HDzBTf9dT34aIL4MGKuuSjhYCh8b6f7jwP4pQhwUFyFQVM
ajMNKWpzDWU0apYhOS/sgX8w214MoSZ7jo5ulck+sbMQMCr2RDxV/kwZdsPhKvlM2pMcJJxJtLQFGFxzNcNNU4
9wjxGjGor6ux4aAIrbDLuXN6+g4jC0qTw4UZZd2qgTNLm3A4U4dcceYku/wgYdcJIjtGNLmiU0PkX0kF5AfL1j
7w01g2d2NJef/XGi3ODYiGZl4A5Mfj235e/WBbkMWJvdNa4U7CkLK4jx5A5xPoRIR2fOkbMyyvJdX0hGzB+cRQ
kHaZLFTTdunUGLNyu6x81ir/mmXO3qbTEypisefjKgE9g5JhXmJAtWX/YZtNw0ZeUwZxhCMlbfkkeqXQkIi20E
8165sMQX1HTjATI2LIvpweyJHCSyMseHqba2Qd8r+mS0vhznc9q63ZJAFjllkmOnA9oYCBLhPC/1ow12euoa94
gNJq83bHFx5w+x2Cd7nGyL9zhTHtFEKghuvRtsr3SvfQeq4lXqIYWcnc11fI1cf4SM5KGQ9epeuXYOKwnm4vMG
FXD4PtyjZbJXLTnLH0g8MtHJB9fZVV6ZcG7kv+1sBaxDf2gHt6JDFIu3/Vdk3WqA3NCWpimWlHQpjWA6xWL6Df
I571ZM8YV5TB9BFndIJMguCwn8agCuRDT2wsRLCjD8LH4fYANBsEmMO9U2idy6tlwIqWz9Z36UQKkLC0NL5l0P
zQ11RzpLNLHqJBQAoiuGgv8payNsRsWKks7jHMomrt/ruu+uz7rp+vXCf7UKmUeDmzDwK4dRLSuIVy0siGwJbI
ygr/ZxPK3vi4J6PKaYnq9/miotTW3Kqny9FpFT+txMp0/2mfXFXlKxGpKaG/dXac/qkeHy62eNuKYzhlrDboYe
ofNbnYTudFHTGMU6ZR0c7Rz/JgUL8Igaij4zHQIPF0oxCREwRmeTsOzPkRC4RVo9XMbohX2Ssw4VrGEtAIfAJX
gEYOiYpphGS4/bjvp6yhvuhzXV1o1PUu3w/keU+yBomHTIdcj/3dO2Ac/u5oFfLLF8BYLst6w24oqa/p6zT4Ap
Muq07y+Bh99tgW8mf1BOvL+BanBWVvq8BIkF1kckbIJ2UVkDy2DObLWcxs9AYs/uuoqOQi1qgCGDgJSEUbVbMG
ql3nVjwSbCDzsH4ndETSldNVPy915Jm2AwNQ456NH88BzELqDYeITEkcGhqnwfhXMUIh1tVCz2GU4uR2U22N2H
y94iYCDoQ11seGtKhapzZ7UyK1gJsHZeDN+2RaA5TD6D91PmU1oL+dYwE/9Uy0euvVFHcWukfszuP4abBT//Tf
a4ALsZSVj0KTqhhNmmRrxD6DRF2154qJHr7pAn1c/2x3vxuGN9Sg5mUU3YUROvOouyQYD1/AtxAXLUWR5wu9KM
qqc5w+8YXOedogBrxHEBnjY+Ez5EQJOu7+1dkf9TaSjOaUoTALdUAWK8yqz2AXe1m/g/qJstJLu+mdGWbc8zvY
```

```
nqCuk0BmpYGtTeL+A0h1D6gZ1XP1HLwWIdGY14y21G7cz94NYxHj7OAdyorL8QoQLSuQSgltPcn+FKuLM+H9S+
xQO71UoNCysksvJUihIHlXbAgBb4L9r/N90sW1+h80bLXn9yWZwO0OSmVG36fBTMQgvDWTN0+WFRg6A1xz4YCM
hA0hj3OymvDBfHQ8h0D6zJS8yPHEjxWm3PxGx/Xy9ME/HoawNe4sv0LNuhf93aCZNcUHeVzFJtZUA3knyp0gaV
zQsXDf20RpxDGqRgTp11o3HcJW2HsbF5mMk6GE01SYXMxPh6tzmEP1mzRd7gXQ0W39l7eX7YLu/8EW3HN6jCYE
6xZ5B3Omg2ad1YhCQC3l3TlrKJZnaGjiHzEviGbH9TwKS4jt5RTE5Gzx+9jDdUjsGyoDyrLxykI0H3TV1NNRrI
CrZ6p1Oakw3IYpNDR/TpUzAQ83dlztL6vHdAldIEN57iC3kkpC2vIuizjD9BT2E23eH/XCKNswbX1aNa+T57Bm
IhdkqGnYq4okG2SBm5sfGtYLeAQvG+TLXjEnZSwRIBzx0OMNhe/oLjxDp+ognY3MqaXF3lVEZkzCG+C3JB/aQe
4XaPJwEgSIKj8qjG1as7CudpeE0RjCWHBH0xGpr1knFffwzB24/eFXU+3MxTDBUdd/6rhK2SCWYWhNY/HyXnPD
Y7XbyNkucp7PA3dbuB7n1hl45w93c2KsQavR/6YO/8MvpakAIhODJdP2w6nTIdfctwQPLG5PW0a7FvEK0etx1j
guJzPn7BLwU7PbkhrTMMoReWJiZ08NKMndexgTvY8YCkkPupYWrZGAuDvfUyiD3iR/uGBOTRF/p2usdgO8csaH
q3wXiBJorPPqP3VazxWpcYhhu3dXNX9zImS2eXB0zIQYLlaJ4d1yY5KYEbElfXbx3QcI3/gdmR4golflJ90NZ+
NXuV7JXbSoTCOcCb8PKEQSuy69OrXPJD3TM9Nh24hMmYnKDon7AAZ6ZVti3pgNwtzh4JExloFjyVl3Smjnd3zJ
Hk1qdQ3Ht/JT5Nx9yPCoj2z2WLOio4WDQpFOhp+1qFlbUfKKvxz6FjvD6vP1fQ+3CG6em9W4ov6hd7FPr/HwxB
PkGu5s/T5LTZYRCvp9HOXx4QobUsoQX8jupyZql1B1GUyFlDcC60kkPnRluF8ge+ysT7o/93RzPDY/dpRURCKR
H3NLjbvsCOmSit9Sjkdm8fuXehJ709CAxS+27op+EcNjcLCtI/4tdWdVE++TlznbPts9aluH4IqZ7Lv6WomSwk
joJvoJ12swssWzOPVaCIhpdTYCpn1xWQ9VGLV6a/7LOc5yBtv2UTgCQYW3eox91jFuCAVMLGBpX2wIx84aOen+
4OxodXluTOyhEslRcIiuMpO+k3p4A+Fa5kDw1i+FABwrEQpbpoT1wdceUDFDwv4B3CoFJ+VKsuVEx2/Z9kGewF
lL9FF1RPrrgLAyT7PxNHudnvp7ZmP9rxrOR5x6geVvzG8JcfQiMsflR4Yi1CsJgHl+gOdTvwAtrkXZLHnIumgW
PyYYMxn7G7z0U4tO3OzpgtadTfqKGyHpunWq655vXYSDafSOZAW4qJisPZgnhVaRbptogVHUqRJmk6wuw9GdJt
1qYZkrEOUTBm4VXPP+MjqgpvJkxILjzuS885NmL0lU/17rDhYJrjFoorjmU8/a9TEaQoHqQTIZuRjbPN32hKEk
ONF4vuEwE5PPmaHs6QnihZf7HXJIVoe9fts4JrtwzenII/n+7Md0ZiCWFcDyhYteJCBfHg420G5Z/TKp29jxkz
Zh5jCQfUusDSwbU1loU3qT7qCrZ1eXBBo5vopvTjNQ/b3S99enDgQmDwB2JqKfindHooiPrm62nRS23oWCXOiq
fRv/TtXX7rRtQJguh80l3NUu6UcFdU2ZJFqhc/ERNVSrAKJWpALklgT4kQ5X/+sCniPyah9m93iQMLzK/qsBnh
Uu4YTRL9GbDqBKqw6Lhs6Uty7g5L1LwMrzgdZdcgUfGaDcjuLhwGIaNPKeLaIiHJuRsk7w0zS9E5MCT1SbKMNl
K/LxKGnzgqvYwfmn8s890IlnukvYqrEanT9/WMLV+zTqUb+9FkhUWX6wkq8ffRKplFkBU3FCVLL4ZRgfluW0eA
Nf1GnRLlBGOQg4Vs+YuRq3eR4UqRtI8J5G5BbjcZMgr3knU4hth+7WzSROkZgKvQExnBPNcxWBvynG/uijkhSe
dF7PqMgHe8hAuYN+Lj21BVUriUN3fIl1ORmR2jK0zxNvexTLw/9KKOCom04Z+d8vzF9kn7r5Wx7VBPUBwJlDCm
lRodV9Xlg8sY9YzjxhymBgTUCXx/ErXVKemffZ8Co/A4UHiRnJvQAZFjOO9w6Rjmg/LJCH+UX/RWibih5iAMSG
ewsNJUz7hq2qpHA6CSx29DIocnU9B9daWi0Gk27y25TGXYxuJAZ6GZP4yUGq9ehimZFDjZp/jXSmQMytDyVPi5
YHDWnTr6DWDT7bWtA7iMk1Pzd6q7UMOY94bYPU70GxW7GEOU/uqshZp27nhhPKkHKtNf2QiVLrltErhgGtmPex
EEQuVlhD/UYlE7uZi2BqjPFSWcY68WGFL9XOc1j8pdYwr4UcNqZ5RY28LCa++ct+8mpchQkISKwPnIrmEw1iSn
VNcB0IN8bBzSC0yEKia/24rWOHoxMsfdfm39pWAnPHl+VyyLiyutP0zsCG7mOxRG3eLUeMt+p1NLwG/68ryymX
V7vxRtKnQIZ3/u/85M/cyq87PsxnWZZ+m9RcR0FsEnokWz1bj9VL8tEVfYhGKiOUyUoo3dHEwOABWfnReSeWX0
ZvuqXzyr9HPo+wtQXJ2694EW0GW7UUf0mtkhd7CTkM6iVXW5NwnXFe7o8iujX45pIr3CkwhkTAG/k5ThmHXsqy
VbTNy0kEGjiS6DYnHY0RWktKtacucuci05Z1Zciy817g9tMMfhdDlb6BaTlCmGg5H4yJOVrCXfMNbGQAe7csyf
VNauAvGMp59uoTcpv85Evb/7dCEuxCa/ItwrI8UkZf3ia5p7zwDYZ1fgqXjwYfZPGJQJs3B7ACYsj2/7qcosz7
LDMxZQEnBn5kgxtDIf3gF4I4HZzzj8SDqP/Sq7T4JYBNUu5Ir88aNZEwN8XfySWm5iskO6VTDZreuz8s5fKMgM
3KL9Zr/1VHMQvPijxPkGmUOkfgguUzeGuOxqf6uC+JbWjZbxnaRcNIsLC2FMwWFPEMzlIfv4TP/9SmcahCoECg
MLH1FK4EVbFtomAs9Rod7bdO9eM3blDCCxIF32ce4RILnx+R/0NZ3u14XtqOJI1v5kdbwxeRZ4JvZ9l5cdLrXN
WKLsxJl8PHc6m6k7uog8EdJMjyfC81ttb2luUe6YOMdU1W99Yrr6YquUJl4ICJfQWzfbtNDzfb7D4XJYxA4j9e
zwHL0SbjFLMB/WXn2aEUx9XdURubbffhwrsrr7tweNxLoSfqFwcaiwik1g+FaUucKvlIJZZWYHvmP1H4MrmgEO
LP1hky6OfRNv/4UJGfGoM3caoieILjTSwYMHSwswihvobeN4N8sPImkjpomksq/J2USgUK/Vrq1pi8cS6S6rdW
0HYcMhmch7wotyVfIc8NTer79r327zIJjn4EH320V89B8RSEGyyJZHn5JGc4uPMZ+AdyOnr3JPzuZcSwog/f0Z
QvEEzrUnqzsT9vMFtv8Jq2K3DT3huyJdDlLeZIZA60LQdd8seLUHulvdkzyKuWv4J2UXP6vXJ8OUoypLi7TyTf
VGcWnRWApHsGxwKn4ZEr947QRmxrhT4ln5S67kMqiP/i4DNBnluXIie6MZQxkp9eVnIFGdRU0fVUdu5IK/t1L9
SfyH7asWXFRWQSLACcvgvtOMIN96ZBfqp/LNC5QSz0XRUQWJ/Z9NdGqavT4aODqyeL7ybqg68rhyGX+A2aj0wR
Qnk7cJ6OcgLfRXfMdYEhSq0AIoiXq2PdhHxn+ItaOPxR5uppvSixnk3dslwKu3G3IQEFZQ9GcRhsaIc6Rt97GA
8FpvHs8oPcv8wPutunLL0D7f7zGjzLJKhPpW8lFY8gs2mVZtr/T8ZSsC4mjIzyn3lhx2SwwQ7iN2AaVm9ytvTE
Mh/UuYUL256pmbQiB+7Vx2clnz4OxxZyGG1K6hedxMcOLP4LqhiDjT8+5YlIu0bMsmSND2irrbo7VL5Dtof3cT
i6esykpjxvZ9sylBUcwyadgk6kqr+JiGHcTfUyMwLwQfHFiOZwCfMbYowuZ90d4EUopWUDOTQ1zqMb471kgML4
Oi3Ok0ngISnDBU6I1ytZOEH2zopOfvwjCo0L9aC6UAt/oDPlFf86pQQFdj6q6j2EJKbLoDlZHm00yhg0dqKvGh
```

```
/Oh4gUnaUuLjWIvNDnHnei7T0LQONrOHUsJObEUwe80mwtabjDgFuQWpmOE3wN0RYHAyFXhRKzGNVJJvju0Sji
SSCufJkYnu6xDNR0lxMy+1U06JNcU2hU2sN7HzfPH3p+gMWlCzw6I/HVCcaYC9XSKBRqxRz88V41uTNsmADfMN
KMRT82id+BgGyRmBAtLo9l1h25tk5sXx/qRFwxz2X/vLH6HQjwSn7vJBJ3nXPfasTMu7NVcdI7u5jtFqYTh4Rp
1m6T3h1mJdfRt7n6YdOWY15F7ikRSokmB6q0UC/19XS+CvOKWOz7laooIbSOguozGFTtVB/1mIRr7d4IZrChtl
V9U+YZ3bHcgeqUZnjTif56yBK9mhI1DqxiwEBXrnTopF1N4pLLC05awYrk7XtCoMz02YUkRN0bXrusLvUpX/in
fz4l+9NUqPNGZf4+ZxBFZXp+XSrhlEcPoJ4sODJ2zoXxbFizDCxZ9Xv7lwE5sH21u58ZobUvR1MUnwo3B1oSIT
KwcC6MyGps8fZuuhbTqPbW2PDnORHhzkZewg77sD9/cvLO60OkqoWdQn19RvsgIA+p+OlZMnxB9lOw7wIMJI2z
rLcq015xx9CdTauOaKB3+2zyCECPdVykWilZYcjxe7C6LUsi6Sba3Bpo7HLW8Q42khez7dbMLrui8sMBzTJACL
jAe63nKj8hgzbCQk4Vm8vG4jYe4HpO2BxRQdkLV+RMj/ZWTpQlgIqWUucNzxU3ts1VUqbUIOg6apetmVWprhDd
jXx3+8pFBl2dlkZ6FDOtVJVsY+kCc5OCF0Fg8pOvE6Nu9IUiLg59sGMBeQmjFO/EqBjI+iVDpw4Lq8qot/mr0a
PJeXyiRI4FfG0x4lOKb3pcBIDF3fh/SPyqtYrUOcsgEPFgK0U8jHrGqZsd3kIqm0Ruyz1obnRs9KaGyuEkaOVk
eJ6ZTynyHjM/kpnxEd9bBwWRR61lvqMYvTE+3fTjOfR6soqZMVlzFoYOxLYi/M+SRFYrZewm6noGzaNPgwc1zX
Sfg2qI+PsCsdYQef8bUeXY/cxAQXQ8JSI0SXZP1yxvorEXG1CFU5A3GNeI7jVwfrWTpdTyLrain76ffHMOmkb8
cdFedbXIjeBV0klkGtT835F66PCXe7v91F/qWFpEul9yAl3naVrjkgU2qlQTa9Hadrk8PX1IFOoXeYhzc2nI2E
oyVEhy1xlZ2R/geqSMe5kXaUKJeNh7XRmxRhyfwbaUZljekGfJJSEatEYVLaBusXzdoakZPJ8dpZn5Q58HSQFb
/VQ1fLzeZWzujRFB8l6xm28H3NHc9X08Lt4uMbxP2zCgDokFlEJazrSR3eCgHsmkI++2K9JJZK/8/hh8TgrbcZ
5pW6SQ1P9IYEmhKpcOyQzMgTAXh5WsdCwv1jNEDyQQtrPpEb5RJmDrK8QEhH+m19ke19fFAmJ+Tl50//Jx+U4P
FEF8LLvjjfIlCpsDCPWMkoLi2UQsLSXQ8mvZ1j0RGBDWaH/uSefdRSaCDkF/1A47tA5oYtE3iMsk4pNJGZPIxl
HdEu9O94xpISOGIwjbQqt8Zh8zwQUV2N7QR3LhIuh9544W8uQvsw6cfZis/Co5TzacJSbQG9pLYZjFD33SYA5N
c/OOxEXZHZVGEqFB41NQUkSB6Tj+z1QE+pXmxJuekjNwYRbhvieH3UGw7Pchct4RB8NjviRi6d+Fa94vpx99o1
Pi/t4H4v7z/KEA49Od2v9INiU27mH0AcSic/qqCgBSZAL3cYVN/BC6O4lEBNVSr/PXdQTe6VXe95zZvrcDIzuD
7NWNpV97Hwn5IqJH8BVudjqQrY/03eZERgZWYuJP8xIs9RO0Z/8dUYBwhlZiwpCgDf23Mh3S5r57VaWR6SBtCQ
lZ+w/yU3h8zugH+5f6Nv+KrDt4ZJw87sJ6ZpEdnIl1ilfuST2dTNPWDNBvh0LSOFe6E2Td4d2t4jHtE0RZqEIN
B2Bn+52Z/bJGUTfptZYooAoQlfyPOuC78GlVEV/w130NTMUOlPkKGq9I/g2CJbYR2ePGE+y2lXOAZud9wKPgkU
2P90fo/zpeL/sKNEVSX8DOVSGagdFzdRK8WSLQjOCqP78YdXyEgSNsBRZS57NqkpKJT9gC/XS6NygODr6SyLH7
1oYsVtQv5WOxbEBnmSBVgMN48tirkN5LDVnKKdTNdJalnZrMeBRp3cCPCCqOk/dv5SlzEM1Zok89GLi65jGTu4
IZXk64l7JAwaV2JASDB6azFw5q3Rq3e/NxVvzFum21Qn7h7BkTXlXOQBLUSvJuzcjqhMcbJ54uD5ApNx+n/OPM
78FPjNnYHXDAyYNugnnng0MNI34aI1GJjBlSrmHphsYHaqmrPxQ/dAVJjxnywMPsGKQxiPqoPnuV5I7vkY7OkT
LEfgBDM6U0g/D5IrdG8HvBCY+OnnJld7vcRFCujO4dM4CaliHEfdG1DrQSgFz7+I3WNydTfzl9k6cZaezal2wO
o5/wZkKdzT+qGeCqAi46OgX0n7zN4rVJKZbMAQV9DQCDaUtz1o1d6k1YdRHWS/bWQriNE2X0yWyokvobZnLLf1
CXAzfeRErXTfQfapPeYdh75160IoD/4hKBk5QcQwrbW7XVceQBOpilRXnshUl6y5d1Jc77VCuC+QB+R94S1wZa
woAKtFprRK6B28zTgt190qsDg8q8u2s0wPqbnHaaOr7hJk4pGBFXqhrpagR7zi5Db0TxpjnfY0rKVS5al7lUMI
iD1oRhZ03AoqZtv7U9CGuc6amN3PZKn40txNPgbjjagU39Wo+B5Texp5FEmsiugin0X4Q/cZ2BbjtYQhWoOqce
wj3R8q3UcWItcKUj/UAjtKzqaFoNm0yUczxcJe/OM01YqesN7KLJTpoPgeQ+7SMkD5cL2YD34S0ciYiBD6qagn
5x5lgi7pv+wR3WlQzw+zfVFM5nVKQnW9mhWi7r/oIPj4y7uY+zG0Mt/fXx7V3/FL3dQl9cR+fbRjR4KZP9xMSp
r4MqspA2hwWy/7dt+dRRut5LnpjcpOPI6FiXr6GQS4xBoTmbTKh3ANTun+nzsT51N6whkTUqHXWSGXQr3vV7E8
GS+XV8mQZRb+A2FZ50w7hE9K2KmM4S1FMrOfGzuPTsbn2bjGxZoW2gx4DDV84Qy1ezonX/twtk3RLwUhGt3NhC
o9aj6R9W85ygnxAQ+/w8U5wPypTssRCxwWuxTOsQMLAtUEFPkozJinnoilyMRakyhHLL6SuABpCiftOA2PWwJX
U4xDiT2AR3wgXL2SlITZ5ktDf6rmTVEtld1O3mMAm3dLnqda237IgZ4+GaAO1yB4MqQHsa7LbIJj3VbIwST2Fr
ATGBOV3YcgpZFZJGte4RIIzi9f5J2guTp4e/3ZXhKM+95uHIp7dcnbDJUnThs3BvNQbuIfpBcXMpWRnSxOPzmz
Zbju2tRc/mqvycj6YJ62jIp+NAbdrRAEP4txx4Nhn1pb9yAIbeYlkSbwPZ3e/p7os578qB6ByWpqZuwlxO3EY4
fcynKeB0szrAehhfHOqRY5pvOj9gc7sHb0/fg/NPpPsTfRebyW+oQ/av34MKZnlSJQexthBMnpQbLBhgiKiLrz
o62lbGaylGGv01WwHvTk6A0AEyRkPH1DEKnFjLqdTEhgOdWWmxvZDaHSnTs0bVauoG1M9F7JNtDjS37Zp2r5Dm
lnPlm5NlwLssD+E9Ax+VZfh0g2vWbheh9ERWg3pScl7S5wJWOkQBAZZrC1Z8qzJvHm56iEZ3YfVWSivZqusMGe
sL8UvtejdBws51zhZ9HBIBn+pCUOEkZ8nZD0Asq8Rcd5qOJc6UjPFZKJeOznTgc6fJHTn6xifsfT1zRRg58E18
t5T13h53gd6dgWJTDRlExvbxN/NfDudvrkRp2IxRNpmT6RT5FWREbIQeVXMii3RBI4Mu4+CXRW6/sMgNlRNQom
xW37C7LmmANd5uDY9BxrOrZUdDh0VWP3ME1Ess9xecBfkkUhp5JajOpfZ3Aolr0IokNj5cL9He0i+ZVwm4pdNz
mGiuvF2Cm0G/iXlvYqHeKXVWN4FkRCVDFUpVWRhAVqYyZSXQbQCkt8SvKaxAdMPAg1vAb4gjd6lHseUp3Jgje6
FFJ40v3cBh0sRQB2dbRjUmuWILhPfOG8Eg9vbPZDt9C4N3wCom3xFMZyfeiV33CBm5Kcu+fhtVtvgCWdfbxGwv
DOyIZI5UNHv2BQFvXNyo2OOM4+MLXogMwqqm7GMTukq14yzh9jGvWhY7mMMUf1F0AXUPBNCjif3Wh/+F4W/us5
2DWknxVtZbeeeyuxJ3CWi0u8awokpwKmBpbs0KZzVmeZn6rDTHIXgiQqqqnXpIXaNCxDa0AnxonOsbahMhZWGz
```

```
yxS8GX7DLt6F3IEyG5pY1XifhmlwtawVFyqUyxuO6po23c77I8RFsSCIfe70iccEK4wk6Wtd6GWiJJGGnuoP/T
rJzheazeEaSwH5q51HUdzj1rW3v4xgE7dHYbQ+r6td60D30IO7dKZ2JWc3/IZeWwlJoiyJun5SONp7q3m/MOel
TWuxw6d9JGs8AmOFjT8aK4Kl3oS1ckrrbSD50+c0CTJbih4Tt6llJR0+ikq9GpwMkd1fM/bhyavWPLTD6bLPOB
QydvMnrYJcF58o9+6HunNywBU7dUA7Zem8dmIuye1rT80H35aOvMBbDTvdk28GPK9vAW1uwqpDygLlTnSH0KFK
yq8FNT+a1Qq7SF7R201MzpQFs5oeBo3RbvbPXgff1+CH/UnMvfzWXp1wFstyp9G/6V633kfg6a7VrlKMpzdvb5
4xArnmdExrqKmickFAMDz7BZbE7JN+G6VHgdzwFRwNVU3sCmQjdpB/5GRUD5154nS9eglfurn8WTUuGrENVsfN
BbvWBIR2tLWCQT0DX1mbjuVvRKndjHbVTFf5bPRj1YsHInxWo2B8pOeswhtXwhAAy+OA/RDaCi1V/EL1M5DqhV
sMN9P+NT9ZfbIcZnXt821XGiVpR+oXAzzq/vM3IVORDdYhGwSpVPIbvwjKyc+l7kc1SJV3Wo/rdLPj4BZ7Xbtf
krYYzPH1dBxAVHXX3IOHHmoCKjKFhgJdeUBmBNIsy7dBpt50rs8d2BZA0wpjdXUiWf7FvthI+DACxkPF3NX7K5
9g0oCg01m1+nELLyT9m5FCvxd3x8y0MNtlV5qm0826L4vgVRZJLWkZgvt9tbHcT3s4wp7Riz/kbgqxqs7ENXgx
F5VjHiS8RQVe1B/HzP0aDX+9rV1RUggPGgomYk8g5a6w1CSw1f9X0pkUNrUZw0k5Vo5kWYwSJOaGSC4O26YtQ0
+TSU4wGFp43OCIC+/9AsenHN2zsbh0Ma1JKXMtwOqfoG2l6AiAWlyNILZv4SJ1zFXV8HiOWbf2HVpKK/Sd2Oki
7ftP6xcI6dua8AL9Db49drC3xWFhtXshhNd7xuDADIhbvPSJOx6LJBCc1DTxu5+jFByDaP08G51l4LZzxlkLD1
FQ8HtmRrzCiszj5eSIECC3fr+BpGpK418n53xdg7Jho9CNbqzUj/4EtCTtx/xyeRtJXGHRpsyETI065B5VtWH8
8Vm5zeSe+Jxi6fEgdfwKWmvuMUgdBIWaQ5GhO2Ha6SX2+C+BQWuhZsdyaFu/HcsuLpDhUVg6BxEuxTjahMDODS
4PrFZtSXmJ6qYjheIaPxhYYZczRlGqfiA9a9ap1R+mB8aSbzNFK4RCMbWJ6hKFxuflJgAgJMZgK91p2s0aV1Jb
iPdh39v5ZoaWH9sMBAqdh09rEZmPQhB6r+9H6y4pQbmvSOlzzJdfNk9XzHq8k1/AClKDyupUIPiYa8JtO+OYnS
rJnxGXmeBHLke0IkpEjhq2rc3WmecujjkcthRs4PdHuoIUc0tjAua3iN+DgwtPfXnj04p49NpuyidpbkaOzyUr
fXX4BWpOIL9PpWSOVTBmU1+Eh6OntrYjvzO9mBEP2OvwqZeT3oZDRfqOgM5tPfeIUivSsoLuH5es1pk3bHSr49
V10l5ahmrFQx0Jl29mk9O3blXoQXVW+/6iIiUV9l7zrpuCE/vkL7ox4KnnTPvXUOVqQdXHs8orbwl9tpGL8+w6
B2gAhO9l2dEtBsBhvURONpNwcAd3/6KzV1vu8t01UdhPQE8tRSdkZb30SnOjghtedUmIeEVeq5ozSvvqy+d4/B
fQy4x+sg1aJDB8g+Ka/zl85P+fDupb39YEC8bQ1A6TKDBLAPYhvznMpJPujAl/ja9/TkQELoJSq7eSx8wwAZfw
/aHRaS2S/Lf/mn/5uu17F/wWEeCz0Oak1cFL+Yt8FcEO18HtU9ThZ4UqHqtSPi9SanjVdS2gFeAtIwNPMpGs0s
1T+J42Fzqnyl19zGfFwaGHyuCeZTWBsk/GHAMT60MvC9hOho2kryQZ7XANAA1yla4pZSUlkmp+yi5OHscqSdqO
7fgAOt+NRqCi049ilS9o650ofPiHQGfsJAEoy5oZWAF2wokhTdrzSTJlDn5KiE7OzUzfiiP0vpwi8WCgr057e+
30XxOEf/kJJ8StgR5clxXAU+z69EDH0kBImr2sUm6T2I9PzYtKhwZRf5WFx+svlQ44qcUwOSEasvKBmce/oUaG
SrQpXMSznW2cwJywBreuvAQ1U6wwLutgOJZQHicYpeQhUycI0KIS2Yw1GMdnQReXyAWW9JgnRX6wV7t3rnD/Ga
kn3KZpUYo6ZdIrByFLEDgPpj5MLHIC+MbWUp5O5crB7eSg3Kr7IUFuCTvyKKVb2wLgzq1iihPYYDgLwpauJBGs
+2kCyTX587ADPIWK6Y6nue0G4L/wHWW6qVvhl5MqNNAunmY8qHWRDyBiAqri5IGtqK92Jl1JprzYgd0gz90rR4
5Yny/hbTKvSw5kHzckAXQ0u1mtYPQJZ9RluW8aXf1fGMk3twKZubmKqUNCIYsfbVexTz0BvaIFlNwuIKhQiwVx
5t1GvP6JtOzdLoeSz9juJzqYblHQOnRR7CRrciUDKvq9nWCeCxqpwuyBo0JDLG7sC62HBDXMZ66J0uSmUsvtx/
wZozrayCSmEYBDGhr8ixjRPKCHAaela9TnEB2F+0HbKx21yKk1TnobRlCqlVTuopherz6qUsabFLdyZ9w4iEF1
2qRH8uGjr4DI5Mtu5Zz3fy0mgzlY9xvds06fcpaVbTKwsHWKrqJ7aThWuKpAuLlOw/1sBFmcJrUr0rdROIAuL+
YA1A9YVCK3oR/yPfXfpzGgKpBFH4GEWVYvvZdA/TrcxcgKMTe0uX6W4Bpj8ygzaFQNB60JPnF7PKsu+AAIbL1C
Iaw1ZgL3ZNot0NEEkzRq/giGkci0vkdBGqrZSiyr/PZ/1A52ciCAqqEKbL+U94ds567t4pA9CZRCbmJ+44TdAm
xzZTfV+rzDtyaAygJDRoDaiCAPWxeAE6zvE9BTDjgZ+1yQXg4UMhRzPfXMZ+RKxo09iVM7RRivfmpT0FrO9fbG
Jvn2+IMc7zzES3iF/8BLeT50agkqU69ww97CKSYnb69egbAg+/Q/ntXEHUiFeF6JlaO3riq3+0c6Yii31Kp/CQ
gR1BypOKU13LFFtJlO/CJ6MCubKM2G9vAGe5zervFlf9oCxvDGudGx6Wk/H4szgPDfq3N2otAYxpeAX4m2q6vy
dAS0sbIlTIL5UHf/AlbiQ+AIn+mRwTgIwg2e21uybbZ+7jg3LlgtVaffRlzinTt/h14l9BoyjDfCGEj5kvD3gl
zEn4o8S+MvpzXTdxhDmRuw41Ty2RAEyNbAoHp5tGLupoZJw/JyJODxnQ7mMIow+pLdqEa4fIxFzEy6+WkJ+jO6
1H9mt1DwelMi2OjmrnZJiw7vdxprxki6Q94nGLNQKD29S2rHR0b1vIaXevIFB6yjlSR3dvh5c7VcFiNHnQhDNM
vJIWnCk9HZxZN5JDOAlRYCF8zcG8YhKMACem55UomEPqhfEySdXuUGydep8xCh/qvIPuU3BmxJB79RrA3q98Ks
Anu3qsoyV9RjgnqzNkskre/7djaR72hG3O9VREBEVM3rnJ7gsmH4r3vLwL+hvuhAdL5O4TNgv0vjpruDPxIPKR
zXDlM5g6UZeeWCam3upgf3LVGMAh2tynWFnQGKqD5nR6AW8TulwxmK85VA5YMgzN36j+GsjsNwN0tmzgtsFXDn
O2wJc0uv/kUu7YH3jF2OH6V+7wl3C1LEm7WZjRL1PBasBkpfUGqT/tQUIh74VNWj+ZKDU0Ub/LpoBpSXgEh9F1
pp7fFXdU/8XJ+r8gmVAucsSU10UTWdJ4J+anJBjCHYv7ZJBINZGryaWtYaj4Iyd74yp62jiiHe8UWAh2458WP1
0kewJTUhX+/yiQBPomMJdAXn3brQEyv3ikYxlSZUWIMdCofzGDavDOEhCLO29EWTHeifrVwiNe0q1R17mpfb3o
CRBPLHPebfof9SjmW/6jFGByyQUsTbkGSXoQutBVl/KrSyo1P8cIlJt986uZajVrUcyeif9TT+8rhFTDBD1xwi
/RPO+4xGQZNCb1I4Q/C45jOJpvpVYGvntykDPaLSb7vRbiWii5SNUAD74LYxFEj6o+aIMKWbZPU5UsuUzTNWgI
8ODs+eCV+BEgxcpXYcxyYb5wdAZ7LrxjllIERkBtDcBxOsuiFnR52uh+lTByGqzfmMnFuoqR2Sm9dBNqi9c684
PMhdCFZA+lSTgKmIf1z5/KJNFF4rYcuTarKcF5y4ZlJFbl0GSM+nIi67mxeHdxqfORM3f2hRxOiMgXFDt66Nda
```

```
bQAG3ZuWwkXYBG/OkitlglCvGcoFDrOv3IePzhsxlpTDLVupeCcJbwcZrcfxKh4lP8eCiKQmqUVvamcfLH/xgy
Fju7B8hJS1xRsOpRupANDeD9UcESAQgr8bmNic9VYssu0Yo9igVery85ux2/o3NwBF5NRL4iojbagpDvfOsX0y
RBcYfTDOwIcNsZip0Ge4vuNaYOiKZz+1f/kPLDgrC8OUzQkfguMnvPkWqjqwnnkUmeNoSvxk1/hVSst1p01Hom
GQZDIQpnMV8b3MNJGMHD3AxNzZaKe/OqfAHH5krBJmLucNPeqHOIH7gPKBHEYeGOHnBZbal/3r7rGS8h7+nHBj
kpyfbCcB35ElFLMzJqB0G7mfERevDnGRErgUj6qHlWUAIF44+BJEshR4iC1VhCK+SazG8WKpcTemmxv2AweLj5
8v6sUE+C6aZ4takhWdjAkB2qBqWjo/0HaO87cz82mRn/8vA4CqpXuLqCgPvRotnb96sNS+IR2z8iz7bZkzjYpn
E5YXP0QsoBsk/J9DIo22YNh9iAQpeieKjsyr9bzZdhDUNFrQgo9JnYhTAqxPK+SG9Yxot6LeMK6ySaGXabeKA3
9SE50E9scgxZZiRmjwkONqaufOnr90EdC/LpMEw5zbDpmUO+MtjGjFk5BTh2WNia0wjyVVUcbsyMXWdQY2u4fC
1hh+g61xTrMA8qN7dWGSZmbr8DIZOGRc02fqAhsc7Qu9mUy67NPsjXwmVRxPKoTFh5aFtDX6V7k/efZxKxC0q1
x/y7/fz4aEniiuy/l1HSoVT+GX2HA4rrkcZ5jk3lOm2iuros49qLd3xx283l4AiaqZYx5DFaw6sfKVjEIj3hmo
pwNnzBcEYM+k6vnc+vNunezsveNxzuQhWgKxF7irapQL9nvzltvk5Cf5uTt0/jamkzCvhM7gxFo0bnozJmgPje
zQpwrT1Up1yr74pW/tAXouNQhjrH+ONVBJ+akaYjeeuL5A7Xc0/M9VuY88bmipASz1DgvaDH6goa2WHX2i6rMA
mZGkDnxjTxv69dEJEd7m7BEZgPWXM6CVyyWbcSiRsQBxINGA37C+ecwfu3ZN0K5sCBgrtx8eumtmk5hV844LpJ
RpDmQmX83zUrmEaSpAPwTZk3DfsZV048Xdht8xWikyLbqCRD/TupfpaCBiVq2EtGc6h9WUMZvAO7z/QNksxTbs
/JUJrV8jlmAsyggUMThP2kzsInRPqg9Fulq+5o628k8hUFR+UVmBZA7zg9WVyTxikmGlBJQlxehVLeT8HjmhxL
7zOrruW8kQXXd9+2FFFa4uo7hUWJR/m+Cgezik1XkStJoxiSQ1ppv4udCba3phFH76XabyF3Yf2CbgkbCAZpZk
SG7knIIxTkNXKv7fQUSScTEzVJ+IhnGXT2jAreX8ma6FpZop2uxM4/KHxO/dU0nWDl283Bsacb6OD5QTX292OO
PXShKa44Wg1ZJYkaQeeR8YcHxTsXQNVAjLMZOaZVzbY93F0Z8f0WUTQPcq5nf5U6V/p0tdFplJ7RMJw8nB/kEl
GBaj7I+cwy+DLoGmZKABUL3RKPDP2/zxIGxtE2fgK2GwUGkaCq9VMXCtWnzEnELmjPEOpEkkC3F1po6HZq65L7
L0sM/CiJ+/16XP9M+Wv8juAjPIkFec0M0xGpwQbFKere3A4kYjDn1lwhszuRZMrjfMeXWTkKTnb/OGe6giRClY
7p3H+V1v04ZtbL0+5m7G7lxUgETXzCXI6rMVMWD5hgPaZhUHlwjrmB3VAIK/utO06l+HK1yMX5I0lMhbrwBumJ
AlkZcqMLdAM/ZYPeq/f5infbcrG8WfxAwqvgZ2ZW6ZRV72kQLCp+AGrywN7egQTpBM1QkOq1Am5D2XiPWcdZBM
26S3becz2gf6nQOisSeHEZZFYvJQqPO27CPO7AKZcE/ir10uzACePxpjiRAXAFKuHnvSf9mayfuKcmRMewR7J5
S7VvSIIlDX+0ZX/9axuBSRCTbT/4XtI8U62NOQFWIP7RloYyyzeNVPkL0dwDvEZ3THTn8QrCcGBHn+MvcpRtEU
xGmN3mUCBG5t6rJImMLG+c9au4jefOrBlXIhopV/mHwVLF1KW0gzJh9n5otvdNKh88kAftfbF93ZO9nrk9NJ4p
URCtUyIMHZsCh4ozcmSKi8EtMyVZPAvMzGh6uEGJ1bdr8fxxesITrTT1mNpD2LvdrrV5GRx9A3dlCIXKiR92Ki
CnDqcobc4IwBUAptrhVeAMNDCikvbu5aDg5goZReENfAjvoMKXk7lGVoSaQezFjIunzUPMPd7uU60mNAUI59Lh
X6WehMBDZS3uztgvb0F1J5JHo7RhvW0tuzgHryfLWBOYbSrI1O6kKgFdmZH2Fw7CzHjkLEzYEqp7dmjViYRpmI
MAy4v3fn7t1VdNdRVylaXXYPHDfVPU/jAKgt/5RKe4poi7UmyOjpGKu460CJarhAQxIiFgJ1CFOygKjFnu1JXq
LoacahUOpkYACZvfe39O1Xp4pZ+f4ySzHkzLpNjfkTa1McW0ulfGx0rvIuqTuDaIpYgMSrQCHCzGRnFboXvGqm
WMhag1Dd3x2KbIoR7b1z7WkiJOOuSwPi4AHC6Caev5y1Io9ddUdal4LppaaEiXlM29LC1ZqyJMOTge+5NaCMNU
VwQdXUIXMbdDnU4IdwEIr+Rq6xMU6Xhh+5MBOvYi4r3n54eKu4vTFsMdcm01CUG4Feg6HqkMeG1atn6xMgf0ur
4yW5tht7+5CNssesvpzULv2VdPrSoWyx9YiSYERPnjYu8wtIIamPK9jlN0EvA2M1IlhZVQVNwFnMLMN18QrO/K
9nULvZMMlAZm2LPJjNzGICGwVm5GT2ms1vGgop5GGe5lsnV7/4bNbOA3Z/PfORLNdCGRpKwmRrJsQ2xYZS/YS1
tYuxBcS9/2Ym60QIVn30GAX1pMNgdlepDWkbbAkaOLz/D6qdRRQ4PfdtdnKNqSQ6S7c075dvoE2YtdZZ2ASy+T
SbSM9TvYBSjsQzngo6td+cS/Pv+PniUVGZEBxCdOls3WRBiCfnXoK/xS82MJCte68aGOd7UvSBv3fVIfQP56m3
5yjj7KArYdJMHeMapgVDG4fAbzRx0BjatzO6x++/Wc5pLj4r8rujbUKeTg7RlCFd89YNzZT2XD/0Un3mBk5jv9
/lm1FaxrHT7PM8AJ4IXy8ORZJjHfUzw6Z0ZXuS+u0X+8r8m4JNdYzSD5P/8zLO5wmIRzPSKBAUSPkJEwGYXZw9
sB/vIUsxxHJMvfBndOBxNPSF3uu5dg6xYzdd8VoLOQmv2HeNkACBXj1HzbBrwKdPeqpmZPM6Jap0xpeHZwPrxQ
fsSAK2bFQ161JEfOBzW1Jq/BdFLa/QxsLxzCE2WurO1fBtilAuV5A0WxDMokVRpJTo4LZ4Z7q76VCi6OfbEe+g
3lrhUvTJ+Crx56qOlTaAQvRQ7dxtV3QQ/h2wGwKtJsTIMk/XoxOG3PWEsNy2YeZ7aufFMA8IV3pE2JiPeez88c
g6kqS9VSSW6nygOPsIYh9pV8vAZlKY0THORwb0FNLl+mrhM03mXjMVAW0OaMUt3Z/qcsFMJ04xNwztnrZTLUv/
wlKqKYT7zMccXRGLEL2dyTygogeXUGstwsM4IJUBmWOiu2i8/LfgKRKOBlv3JoW/QUiMQX+Ci05yevTvclA6mn
PPNvhPH7oC+K95h0LHzmS8YSs56R0bmXlOR2K2IdYMwZwC/0B8tDFf5iVAzbYOtptRRtFJ2/rlMVgsfaDT3C9W
ifMuGVGQ4A6HPqhBB1bo2tKDOXl9s6Yz92fVGFeaN6+7Fqep/CxZY25H5OIl+bGP/w4qaH/tGv6Ip/+rroBpyF
CHL3OZpENzZnbrrWtTppU9hEid6S+T09lhpHX10oNX4ykSE07zRD+4o4DHHaSjlOZVOWPwEBKPcdEb/arOsjRf
e21/9ca0Q7ktCu6gfsjGrgKkhhgyjVARDhffVa8IeQTN7yfcmCBcKH+90ZNx/tfbjC0oUhpbIzTLjvEMpq5FoP
wpEUJ1+uDYqKF/+MP9N0Me8zfGvxXYCdIYX1Cl6Q/N47IeKSQm7VnZLjnS1c6rpQRGm/aiBTClMU0gQOiTZ2RD
FTFQFuVCPSHtTQC293aOvJ23t4/cMp39/VYq1w29tnjIEBogZKris6alEs1FHPMcc9OAzPghrSEmprbbSF8a61
c5e4xmsOiVRuuOw3P3oSEWsVRkL82JD8mjbjaaHOMw47Duvl7PXEsRd2YC6wkPnMLDZXzTAtg/QaAA1blcB2DL
on+7Y0fmlfOYjfBcVJqsRx1xNLfUL5UUTnLsShTU+myxBcVixgQIYZzzcvN0uL+8XMqE0398hAeci84xp051cZ
```

```
JvqtFgI4V5x+CM2cPrcP49bu3495e1uxuWYd2atPV0n9NUNV5Lr9ErsX66oteXSF0voccgEjgZQDtnLZ3fyq/E
YtYsg0VfDO3tJSKaR97vwe/f2+bQWidQS1s23h8F+/XopWZHIYGJ/3aXD9Nss8ss1WZRbmFevU6mpuE+hgX34N
0HtF1+Ch690ST8/lKFzueFlvqU7jilk0cMTi+eRCAoE2GB0s4jMvjOhe+Taa6No0dVLWsdthZcKojoPMuNvrRO
9kcadBHRPFZagP9nBy6tmBdRosP39lN8/wB4irFNv8SIegObqGhPCuLNVf6Pyrschlqrg0PYpi6kyXf9gsTpfp
jpI+LBgV3YXlOC5QpAVXWEMhSIrtLbW5igo3VqvTxnvuMltwkoTwn/7K020JMgNmZe8x/VXnRZF83tzm5K5mX+
mfBFlxTuMCKI/cj+Vr8ZlfZY5xXuvM5Nv+X/uBzY4VR1D6AspVo+5mU5Hp/gNtloh+Mr6UmuFzeVn7095cfXZG
FnDSn4ahN2dxUA7pD5AX7M7YLMMRoMACsNQP5GSGl3gd3bJyYw3inm2IbU6y0WRKbgDtnloCQ/vqLgwxGLKL4S
zSKiFDDoJiemYup9r6LOJF15fP2QGw2DrPSqo/M52s2Ri0VFwIKk10G7hEwHdHgpWbo3njka97o8TzxC5DGwcm
ECRx+s3DDFGbsn+PlVqUesfDl8t5dcsxRWc6Y+4m88wZ8Rx4oN9eeSVgHgzk46e89VVFIPFusTkrE+vV43UNgt
It/mQ9+6apnb4+Ea0fnnxAuTgncTVI/uF3fZq3wha9mR66mKLlxTxuy2NdqFN60/5HllWwSkD5+xF/PtHhgAxt
iWr0zw15hzcV6m8MSQmIBVyU9f7EAzGgOsNATe0UNz5U2n1yaRumORoKg1O0J25KpH3V6brkg5+85DReNMtMb8
b0TsNg9YuQrEo2EilkArugSV6w0AaEVd9BAuGtTMjOyTdGZTVq1V+Bx8DCuYTt4K6vX84NkDOC+/rWXYHFWrPO
IBYecLgsPHbKg8ijs48ycHoXeVV/T907nYJXxfHU/km7Sg+t95JUPkF5kzGA+DrJhoQbuQ8sTakS0jZd+s3P/0
wvG9K8Joa2sO1z4Qj3mrVfIpUOm58fZ+jyeFmcvkG/+zYwaJOrzxGoljxbPl0GE5oaXLdBpeyvo1dk9U4IcPQ9
Wp6XsLDQOEstcZEaotFc8STcgLWOnnOUmsvV5aekd1uYVs7XPm1cqg0V/uONKlQsmwzKvHjj5ZQ5FptRUHoWRn
2TgyI6UhJNCuDyUDZALFcy1QbYyzK+kyeGwjOwsfC+cnPS64Ml4TWHIVEx8b44s2LXJK/KhRxEH+0Db/tHE91T
m2sXMpv11gQh9juSUKLqWDHcjdKQ/Mi7GFJatrCrw19k7QtUbv77FtAtI6ozBm38sEhDeLVz6pIbp9OqG1+GAH
YQeW2zRQFYhWGlZLCdgSCP4wbfZ7gdUUiEm7Dzppule0M1qZABQ6De4yvx5Gab5fHCEvVmnxuaW1HqJDPDvyA9
yqoq2UFbvE29P0FlguXAR2lJJFd4n2mM7k0X6QYTWZtaeV/mMo4DcdRpjFNXyIXHL0ECUUeY36pI+i9nl1VMYg
U2BjX5hnqr+K9V0UrOrpmB9KPm6kA31cC2ze236nAS9I4TuOrakI3CEae7iTeVX/Us37lax0poMsvLYCu5QmHN
cNe4CfvAC4mYy8IT/nNl09ByN9aNRwFi6jya3iyaa+saGgDXcFILTR2O+Nfss7q9/44inmp5xvEgfO4DeN/J7z
Hr06jkBpz38WHWRkYzvqSTNOB2P3VkDaFacjw78jP5fQ6MLYCvK4KesgKqI8R2nEntGX1mrxUv+yHfdyaAtAaL
1+GZGh3msp5Et2lMLcPmGVmTXQ4kxRcQOChOXhsC8KUqsus4PBSC4ijWiIiNiVLubd65MDZvhn92pGSzecojhd
cwrvZYCahwWzIacHbvOfyEBNS58qSvQHFQ/UAPXJiybqmIlQO+0i1N4cfTMbljWrkxqfNqcDeQQ0MAraB+O7Qm
fh+Lc8amcMWlwh+U6BpAFhfxdpzuoAXYJIAFaz6lUpP5AeYFgjQmjZpfjN+39O1qqb/dLaUePYraxf0Bk+6BqT
ULwBK3rkT5QCQA0o39rTTAC7tvBqTc6Hm/EmYI+Dz1ne8vJUc4N4SmtaAAJd7oThclJPK840e7chiQ2UUj0ykL
FYQ+eqkFTRpCGZSWOmc6CersZhuGYRMlaTzbl4oTSuVQQJjqOm6ecto8FOTIhA89p5HIb4S7Eh6T3y/5Clvmh1
h4Xq1OkU2DbPtHpXkGbb7MkAWBxhymdSaOj3Lg4A6gdy9kSZ9I7r/0bUpQ/ET/VpSdopvUBE+RIyzk/M9A3YuE
a07vyjw9Shs20g56QnMWvsTEEIP7grDFry81EsNVMe50wTusIk28jr9cBKmHHUHRPClaBv50KgMITdgmL8MFNE
oxtXnAkAV15khGtNhrjG1rgIJdY91mODAP51kSXacue0qzGTJDYSpYkxTHReFjUYcXThy0izJO/XTTLRodV9+4
U6/MmdWGtlBUCsY9OgtYIT0oxLKhrJCbWkzfoIigDSOZKQjV3pYo9OzsLEL8vTwHmclDzqjn+zeF+TSYIAduY0
oH8fj4ByJcdwkDjSb2LKqkJn0VnRj8Ry+OjsTUyUAOSPWDZahDQcHSiHUYdpBsLry0rYyzGU6y15moQOsrPHcW
ZK6LfdJ45gvAtqvwnd33yKrWnH54xZfVin9+lDC3K7UJh1xtbl4u85bRHZBAtImH6cVRmsj6N0px3FzJNz5ZQi
bOa9GztW0jf5XboD6u5nBY9rwANSzUeUYALBsj9Zo4IHBid81nTmo/ZLz9SoX5j2zTpLfkGMXanGwMrE1lVxSa
++nOV/k0rBmJWEGAsPSLeSxrz2jBERYiocwqFGi3kUsFeBFQEWzYlHTAv4kSBXih+m3oKCliw8sof+LngKvFpj
uGUvRPUqEpAS+Zm0RcWCd+0xFMraIc1GXXm2YwWQDimW9tRt1WNx6JDSoRHljYclV3HINVB1qjthRfoD5l6k+s
4XZ9bt+CeR9E7HU9fmVTlQttPSLsfqUcLqLCMfi4YqIv2W2TZPHhgLjwnS07nm7VnFhg8ulxmpAKsI0kGOXn+R
38pphjsinASsH06Uf0Ow4lDj2HqQPP/+PoF4y5AjWBob4psqFjaWqK73ZszhwsOt4KywcqB5HbmP6nqBZFN5+Z
+ft8MPM3gg9lorUGMFLSIufrLb9tKzLzQhrgJnuzzXsBpnUeBJTyTiNXEzHadjjX68EkMkEgAfVxqLKks+aKhI
pFp9yEmw9DdMZVQnWAf13+R/YxFpEW0bjdeQIHc5ci2Ci5lcFoORhVULJG6xc8zsSXPPab+W6U4fhtkrIjt1em
MyQAuOrivW31Qo0RAzMfrsmqlTxqdZAJX/ttfv0l6Wp71DdJrNEXZdV4NPQJkmKwzhNpteuu3XUUQhWAmDaJ9d
t1q5fxAVHcDKKtMpNc7w7jATbwXoglQ4dPuCy41I2MLpsA3DmUIHcVt9aIixGiWO25w0CDFShKwf0i/YoWpa+P
/fu6CH1kNggRMqSsfPGwDbiI4jIEBIHmWj1oU5XM+5AIbwmtudNi4vKc/2GZ5Ep0X1EmAJGfggpQk5VbyYoP4V
aMZRq9KnhWVjq+YMzGgvd7Kf5PDv9/Zjo08mCRjaMuh730TcX9AaEesTYYqvTWkdPThxhDdSFVniKKGQgYohwv
/A3ni5SAZm+pXGcLXbljtPjG4v9C23UoDlY3CM4K/1oop6WT8XxJcNWhsEvSm9d5/P0Wn1hOvZhIyYLtgvf1aQ
efWZ1HvBsfv3E1h7VJmUlzYHXo/Fb4O4ktRgTtz1RlCAXLuD7M0/sgb3s09kFk8CYVZv5x7cw8GjavnqNlfJ2i
fyk4I0aFmRCPerdSgsip4IdmjZSQYcZB8d8C6g1c76qyUm5UQjflNcZhmzlLujQ0qPTuJBDpAfPyzscEjYYsHk
VuMlv6lsBMJRWZIOfGaaqS+ASfsSRC7gRejICvMHt6yyBcMkI2R321yGX1I6QztX5ELvy/cDjOPxqAugodocLN
yod9mSW3rpun0dp01+Q2mePFMiU/pCcHZwXJHJugvXGo8dqkWR93DU3YQm+3YGy4S6tZX+9CUMRkQh+THiA/5C
33E5xVE4IcqHIbadALYgeZz4lx+kNq578DcI6EG22YG2XdKWlsVXGn2znEQHSZMXQ1BkTUxxGCbWIFRjUs27EM
9JcjdSBzIvulZ0UI9p9ycEipDDNqPPd2a6OQMdvPcqUvTKAeNxVD72Y+qUpqTlE4Qxq0wx/JAIik0L5iZdEvcQ
```

```
AAFb8/LvnD0ZI3fuI+gix32m8TzRejUMFV/zMTu0y+qzsnSxzSkubpSHpqrQZHHNN7x7Vh8QZmshqU+rK9tHzx
+Swyc6PFs7qbzooZgbHDOoEbIu1plWCJIKWsyeO76E1Ud+rsZ7qaSsONIhFw+3LoIcX+StilvtccUFynKVNHME
hGweRQZ3dRjFHFBNzjmK8M/KCw1ecQwdUvapHX0U5QlJRlOpZOKIttDQy9lUorpalBIs8auY26Xx+GsIbcyhzB
6ZFcMzrdllgUurGkn3ebFDHdW/wl+qdx6a6YBgclomsBZS2s2DiKOX0bp6nyrV6XhOpQore+gteSxbZGbFiWbS
nl8OIt8hpg4dR7jZfZelwKhs33/4TzRINv305wIFKMMcG8fG1Z8zU7y0KHed4Czpsg0A9HoWzJHOTJ0p7BQnBm
S/dMV87eDMK+viPYUJJPe4p/LMug0OBEmxax7jdro8fIDA7YWQsyRLcDflPwMzVbXAX5mPwN2Zf3fzqNrs1arL
jkEaOynyrHomQ50lA7U/2pFYna1DcEAGKmUFYfrjTUkJ80GNcmxbVVAojI7cMPWWK+hkLkguc0omXQECqeRwdz
KxNpuwKPF0kvIxWoJ1flbk/oFeU+mmrnSUxyZ2dsWAOq8FtYSJIiDjdqnh/1yQS787fvfNETU4tQyME5ihYq+e
VWzfeNAL1UlzHi2JSlg2UbTvVddjVREn8wRSkRvNRW2Jjf44Ke61UzAecwwbXhI43dKcM31TporCeky/l+XTBh
b88gVaB/NVPM0ysEdh8rNbgXcXxV1q8NWUyOa5oEYbOwROfYGsBLFlTwdn8rc0qirkvm8TjqhvYYlPOmnyvGLP
upMCwuoUxYGV3udB89126kgkDcC/PG3dWwRZJlr4E87SGBX0BkuPrer7XLmLX4WldQ+gqEVPvq3/EKoHRVrhlO
JNvRU6TJtKzQc7+3svDu05WRXhCtyaaOn68b5T3ek5t5AAUrueo5jqCxqbKMaS7tBCl2QBDhB8WjtqoB6pwH5M
M/AbOLOuD88W/W1Q6ejCLV9NTqLEogMnFQcet4G4nl4aso4hK5Mu/coqubV8brlMCzFjCFRBfpckRpCcKAruiE
FZcwyuAqLhTqyX/+08gDoSiWXtY97lECO5Ae8YgV7YVRmVT2CU5fbbvHCRWAjOMHynmNPNvjXIjkw5+/j6X6lL
Kb8mSITO1cCSXycOc3vPvPOTnaHRUsjj/VbNUyCyxpnBsBUJZ9EaAfSfXQq0Z5f4Na/r8NMUBpZfUtY6u+6ZmE
d7/Qe2E8f1ZJxL2BPw4e34Nfia+3UIQVyALKfwBIa+G/25A9EtTQLYxyWXy2DNve5kY58Z4RQvDu/t5PPFDd6B
Itk2qHYakvoQJhM2JsoO4JHv+4zdMHz7RhTDUr/EYw39kSIdPKSWkFACjkNDatTUo1EYoZpTSaCMczKuhnOcDa
Ylom0UN92Aq94Ow5oSNyqDuVxran3uSVoYFeiDqDxam9jsF+OhhvLE+BmZBePXyWM3bY8WzfGXtGtZlT1J97L0
3qOJM1zg9WGbvDCeMgLWZOEJcaNMN4rP0TQ7cJXon9+TBJNzWi3p9kjXzcuCaWJN2J7HTLOD6G7+HFjELI7m+l
X1cCahFB+lWB2v7aE3wFc24VXv1527RX3CIk2zA6XrRVSmWzh9qkkZUqukxUV5V7Ce898CKmTE49exTGa5/Uw8
G8fojv0t2dZinLYuLv0vieu2x1cCi93FKQzik149O4MXmoT9Sb53uC1Aja2CsXJVc36rQ9ymIPy35NQP8phgSr
rs8gvcK9WMTut0/VEkjicgK6VWTUgrrZ/It8AZ/PEyB/wlVQt2gHA3yU17M7VIZ0cWCib2JMnJwrAljAAlOxGa
ecYAJwAb9GYab+OYbknLc3E00rDqieU+ZZL1fgCithioKy+G9XJoETuCA6/Tt6BLPZmvYBI0rJsT0E41Xe11g7
TE56Xz/Bt2pSTFD0dvK6pT5x+HBAP1GSArpwQEwBjMYSaZwJspJpe9lNDjpzJj078w7sUqPjwTum6eFQayJNVM
IiPpCvoD6o2p/gP41KB2saXy1ki1fSKWgL3eezYhju80erX8SGqR2ycEf520x75JsW9qzxpJoIDeCgODdX/TzU
U5MMngtTcMRUgFVpydwBsXRxICIFawxaI0cMxRm7WFoXqg+oR1/UWQg5nElhrhHbTQf6rT+gZlKEtCZmWmSGsx
8UsF2pOiF1Fzgrrwlbp/skjmByy63TqCraeHkHqInXmJUs4m7wJFJvZPfZhxBwzGUkM2FVCx2vVwNrvzIT4nQP
kYUCn/iXQ8N0dP6QGCZVmCVb4ay7xKoTnkY8rc+KSRZgE8kbyKRk+DTQAgyeeW83Yd8ijJDYE2HEUAWZkgqA1z
2RnGZSlJmWBCXaLh9yu9moh/CjiWwAfnYQbnQiLK7UFu1q7voiPnuS2hMTMBLs4QjvZYeS/6ktD9yOC+NzTRVE
/ZUPHD10fCJS2jnjJ7gHkk7BnhA0KrrB3LJBpZJOGJVgdNQkLLaZP5fR87a0yX63vTjX7mCK5nkSJ1n0yMVK4W
pTKwv6DnXD43fqMkWnkoK5mOxy54maTRP1NHJdhdnwMlm54uD/xcAHoDhf8hGotnd8h6QgmB6fdr5yVEyv6vBR
DsCtH9B/xK/FhvR/dTWiTlzbU4DEN91i8eW2xVcCWs4W+dcpOoWPF5bEOp3c3fPzDCAfJoJ7yzr9dEWCpnwmcu
/Ssf+iY0s9Xba9XisBJLI4TImzS9VPMMgV8V6TgoabRyKJ+SlEgNbHmvJa1fEq0VVbN1pK8edE8kdd5vTBsftG
WURDIEyWelrYtJVRm2bAUlvaikwHTU5Yv0DVIEdGdb8errwElJpnJgNKURUMyla2GRR947nQMj13qH5bbN/48F
t/CSRlViFu1wkATmLBlYJCpoZTYID0MeaDq1iJl36L/dBck7y5n86J8guXpBkocm7Z8lhSkHuLQsSfSsdu5Erx
BWWyt6PDLVM4LPdllxx7DO05UTqXGCWNHwxNkdZY5JyPhNc8RFBYNofHg1PS/MrOEvvlzpqa9h/pKwf/2LFuy/
rlhF9iJRKnYL2cXxxKZO7UDPvbrbOu2tym8Rf/Y3scC8xkXpKxMUeaanXjF7hF6habzitk8SyAccEXa4QRkI+S
h5NVxh9YobYjLn7joyTaiJ1DfkFqodiE1rwIFylTLw3s1MOLf5keTUx9jDvdh0L5NPN6YeXkCDUibWy3ZRIvmb
px1mtfJZeax8RWvs1wpDwbfJ/wG+g0eOhMVGF19aBvDMfZOrijdyOe8taPszig7kmvsuvlXjMpra4N94jXmPkq
1dFIOJwpF2VA4XhP0xw2SRytWH0vuXhSdxuhmtL5TnLAQDWv/VpFTqj2BwAU83ZIq6OWvt66dOdfeG3K6NWk7a
Rkk5aLiGe62fan4KLiy0BBigyErUXGpO/1nrX5k2JP+NV04XxRmeNxcBph9VmVNS87wBRTsAij+etKLRs7+sj7
/6w3+2j9aWFDYN444cBERroUBbR0tE6HoHMaS0bajKbwmObuebuCmppv+8EhS6uCr0BrLGqEM5WBzRlGezkn7A
BqT908poGvme9mOB7VV5n7mEgVigjrg0fngMEpc5EP7Lw51XPyzdASY4Ogt5jX3MIJiyosECgHncUVgmnS6hcD
tLQMzEIq0+i/thh/Ye+xfCMhmYOe2k4xZElOT5rBLn56oQWCqQFtbRJkZ7bPC6kNtaGlYprFFMNTVENx+maxpk
lvFKPND6Y8/7KoIkvck6VOecVcDYPHwuiQ9HMfaRZx9pBoPfQJGB41pVvARvW3ddGFvK9uhLPvcdZ/Iq8WNVBY
1meepnWglW1HPBs5d8zg/Fbli+1kTEZR0kP3JY6GFjMKkbW71CWAXJVRZlw3XwvCM+DNCWYMeUV5sEsKDYy+c/
x82h2PlJkzQXX8xKQVk5aifO+sc+fUEz7MwCZzzs0DWc29yQBKCxQKDzd0cZO3Qg/2gqzeGAI4Psin+hMkRPim
KM6TZxWNClQU9bvHBcT8OrnwYhpHX65Q3cFQPR5JgS8aQyvZzzjY2XuPxBkGHBaEgvJDH9n1A6KlRzkzkqijBj
Qhw8i+/XGRKL/STfsOqAr9SRW8lctnlKRYirLbk+8cYkoSxjLh00Ik4MgvJKiLDEuEuArFFLUDnFqjwA3lwujQ
Www7uSmkPCiKz27wi1H0614XdsW1SGGaFfMm7bj1zypjwKrCNDgNPaNwomTWewkYHy6aboNq5z4lsijgtoAxKC
sAqfONi91xc3Bh5l4dslqh3ICaBMyZMIB19tm6Xg84pNw/ogN7oYLD9A637lLqGSKzrZk3m3pTnu2AvzvZdYJF
```

```
EPRIEZOGXT8xPXLLibd8G5ctnDtYEVAHJ3ldiX9wZ7H3OcWczchN9jODAXtj/m4hOAwjBDjb/BWe2eqN7tc6nN
r7+OR1ooRB4DrdiQ4mwy3DAH4h6FbgeI18eNNxjOelrzjs15eJ5E8O8Y5IiKjBhs47+XcnIjvrjPwfu12i7k7b
PE/NllI4EAG6GbO+xB8UmK03Xohk8RjviZK9EHtjMaZX9dvh3/aUEWTsjK7LZJZ29V61W+206Z7MioCRsPqr0s
GHltzXG5Ih/n3oj7hkOoyMTuSsZ9O9/ywWSLFG5HNFPW9BWQB3LTtDV9eJttg2u/3bk8lxPB8wpr+t/vKBumA9
9ZDBr+PZ5CI8UwczK/Rzf5IdFmpKHrJfKazpplX/rjcBimsHievlN2eSzqeL7gaXR31DVR/DgAXcrVy03n0coZ
qEn+TrMMKcuyCq/hY7SsWorC9ExMzn6aBw3oSUbHYb8cK+m7OlzPnntBtIi9KSD1EaLeEcA5v9amzflSmz22FN
XzMKTFxdatxynHiGXhWcvXNmSprO94iPsbIviGTswUAJUPWaA32ABo9DtahWWZ+8LerNIAdWN2ppE49BwTcj1y
F9FQXysSN5d6B1q3nGedHR4KDRSrg6psghh5M7hIlJF3z4AHQJA0jlKeyne4FJ+HxOWmkM6pbelrqv0n1oocEy
xCnh6XcLy5+OweVplrT9osaI3palUCHdo2nR6dVc+Yws3oKdw5w2oUsJCVQxS+ZkpNIhgcVot1IV4EbPacegyF
38iSrvyxLHlcr0ieIHaYre7EmieaKsHJoaAkqCsvdDS/gABapI9TXym7vNR11YEBl4QKq9ACupO3mRO18Gp5of
r3TuhAr5ev1kNOaxdCrDYiT8jF8vV51EPq8cvb3xCpmVHeRDwcqsNPUUqSJSdWfChpxlJar6jApz5rnlvrPrUP
y9GPNRSUeAr2U59Hhr94CMFq+wQp7Fi1HuPg4ZQldcVXUrUMG0WBDlCbKsCL/oaWeEIm+/DAefepMk+Osy04vg
C81IKY+LR96uCOHrSSvpwdBPcx1RV82Tqoaqk7DVyqJxZlvWBmSa4mYRlhrkW3PhzT4jUOgNBe+lcnZK3MjkeU
WfD3spCdYDW2wdSBr7IZswn9cvhqapzJGQwuUo3nSRteVKEXYyo5+NT77Id6tKpFkOKjK/ETwzvFI0jxGgvG04
2VcM7i/XBycnZkWBx4Ewfbxj9mMIAofG4h/kgn21AfEsmiO2NvdMnd4n9phLIQi3lEix2ML+hfYqZ+TBK6GX9y
Q9TUEB2uOTsJsE//mczfGVLIOxdXosv++LjiyTSnMstKs6g4Sj2VNPBZUmBw7vIPZsPaArTUPw8TQuYFqXF2XB
eVhptFf+E8D04SWrlTL+cabmuHFuPwkWK/hLRBdzsfFOBd0a2ARK/DE6vf9jIqxB3qsyanVQ9zIcqMt0wlqtIO
N/mZXcgpm4OPtCp9q01SRo/ZAa2KYeZdl45bvDXojd4zshVWGd6d8AI80rvhmHuGzn7T0xSxkHIm1Z7T9+iCqX
HJ1zow4G7b1AdwTkpLxpTgHgn15sfK0+NPXWG1Lia229J9kEtk9cEVV2ysiCqFnDHyFT9i0Gq78pDxFSxJ9dNo
vUOm1+7AYaTx5fSSDw40JIQta1/sg3YEF28W38jAfeRgiwwURd6/SjW1HZhalufJVOc6B7wK1smNOzbRbcYLxF
Is3yN78r2kwxMRdLSfdsFg2VPiU7LbU7opbjLaJoSUx3iet3OrOz7abh/T2sE0p9hvYC1vKrThT1zsd34g1F31
RqDN1oZiiJsLWHRHp9Ua2jbInupE9Nu5wqSudOEbmr1ISw1ZROUGuelPozW8+YPLLQ+PjxuUaMOzldkELNOcUC
u0UX+MmRWhvqE/UPlwM9O68soqWpgmjxUdFTbzupGmz+ApIsYW+2A2DI2zsMeGkrb+q2zDu63O5zpIEvjp85ca
hmLQZWGYdjILbZNp/CwDNW9djmMs0a+25PPhozEkUblK9/b37TinlTLB2EZUwBmOmR6l/gIVYQSq4WH439ecPG
Y3UKaQozIacmq04MKdVAfvGzNtu+2r294/50AnFts1mV8KyOeiVugLWfi25s9B8m5AbsZyeKG52C7OHFuLhh6R
rmVIOXVyR7cvQnm9pH8hzlPUSU4dFnH+qc8nRbx0hpalFWnwnbVbJ9Mn1nGOuoNYIttK4hMO4l3SAG4wyL3nF6
ViIhg2XE44Wv62UEUoUOLsBULZKec0uBZqTc3ZB25pDcJI6Ops537adg8N8FzQ3y80pAs3CvLO6FoVYe2hKQrr
jlUdFXjc/HOvJEk6KjJlS7PAa9B/MdaQORVZKOtylZedTaJpgBQk+mt9AzMuM5nJ8xExTEW06gaLtuv5SZe/3v
Vp+l/u+aJIUtSyVzXPNos5QnQMosanlfMcLvIErfAUMkvba5y9gh/H3BhYty3e4NMAjjHFg1tSkt4BBHXlI/Bm
/Cya12DjPA4R2o0lxGHIMEHw7hEIGM/eVzpqD+gbZC2VdgZiGj84LtR8EX3gcLuJ/oKmqmlhg+G5WxbzrzuuuK
DlfUeVWVJUEsv6adwLRbn3cfocsi49Wu1jqsXbpvfmOg8VCNJhHp/AgliKI1AUkyLjWsL/9aEUWtGmPkzUx3Wk
afcigiJOzzxUOC/uX1BEbG8mjuem6r2vZ3KY7Bco+iwhYzEXZ1r1TI52XbhAcq7Kv+W/whAXm0qwTU1ss040p3
c7FEjdhmP+njCSjaykegWMHq5RjjTt0TVNt6quJtwRqrCWdmxgHhM45oUaXr3f8D10iYalPAZlnL9UDSH3XGq6
9OItSTYtiuwbtmWh607r8hvz6WSGZfv4HUHD9TZ0qvvw5TW0Ev8P7ME2yuxtjtpC2P18RrPG98ZY0RGK/15gyl
0pTa8eanSWEenj6A7NpfATZcEJKYUpqppN9oQWalrSbOmeosHnGrWK8s82g2ZI6wk9TLcGjYijQo1zfMNkxi3l
stRASTgjTMfzDPWdIh+Mq6hP9b/HiK/7dUcDZYGXFxNs8aRqa8OvH9ZP0mYCaykEol3RSSipFMH7HOn8AnmLt8
n7eb2MYzp4e9nZLB9wcbJ3Haf7pQzTXgaXShQ3hm+PuJpA0Se3pvBTyI8fMzzxzQuApyiNaBRX1eExnq3ziNjb
LyLjdvEPbdpzy92vjoWCauD1ITHHo4YZ6u8CAl3t8w/gP08AexNQ9ePkTcewRCJ1cqS9OMA59XayiJffgUOf1Z
sLwdrYytLQWXpPfKB7rFdxnZZfJv7Scc4VAlEVyKy0/EjD66NEoOPWXDTzU5Gq8Qn68ydtu125nalDeC+oQEMC
+uOzVgYwiZquoq0tL0vFYva5YoDmbjmip7jlAmetnV26KdYtuuYWXn6Rw++vLQJpL7inzD7PlEHijVYg+WWXbC
KXwDBAdPLT5jc3OVRZUnS/Phka6839bn3A4ckdWsHnURI8vF1FIbw32AyNRYcCwtwoRFcUXHbo6vyVIXnJNEnW
LXab9/jBmP/+BuBYVZHbpZk1cNRSFiyKSTEaeqo+kxGQriaCzl3Mbx8Yoiuc4dkeUN20dVjiPWCc/c1KxGUeMS
lKTelpIrdcKy61r2pvQcUXpMc2Bwzp3vYO2tDlI9WiZGF1D53736NNTQPHO/d9Qe6rwnkU/H6fTgAVDlUkvNJD
KUf7w7mmdj2qu2y13qbmwBp6E7n4Tzi0TlOchD0K4630+AyWhkGND2U0ZieTbPYHdwJBWi4/z/HSyYmkk2E10g
0EG7RIdbSNxHohXAozp07FGR/aKxqh6H8PeVz+ueTcdVVzc3XBecKPpX+k1ehMtga7+S6wBZ2H0W92p7eNisBt
MCrrvqClgypkcr5Nb66fmL7hJskPe/9iSe3cGXhFXHwDjClRiIVA+1XiVGpfoFC4tvv4WnwbFPQlvDF1zcULjK
/Z5i3VGI//SMqVCfjMGG/qtzlpMhDE0XiMF33h0Ygf1ZtMYHvtHQUEGashAIcKCFbLs2JOG9K9EudNj4ISYuVP
oTY8MGh+mZSfkVcVeh586bHih970fvtmGMVq35v5Rhw9sknRgo9alOOa3f1I60XSVwKy+DSirz1v1yCor8fuSQ
t8AkPExmAJGJpPDkAJEt4/DsiEZ1OxTs+bTJCrx7I0ry7n14NgzDi7wAhW5dDF9T7FZlSRZj3jUIrg/SofGqCC
/sWhxv/+uO05zhGxmEokS6T3E71gVXodAkJWcV7y5+nOg+mquG8UHeWO7fndVWGoSdqwdgbzBEeJ40kcFT16XF
kcVyqBh/lkUqtvR4wttxPYHHrJJB4BWdfqbTKCF8/Rx/K/hY+lkw0ucbk4r0d0brPAcESNMcQHyLsoHwfNSS7q
```

```
1yZffknujolAxkwlLX2DDUBxEYPlZ0VAisRrz9iQiJFdIBYZAMOPBCLfo+HuMg70C4mX3haXrO74bEb7Ecn4VR
qPQrE0g5BbgfkKlMlmrfyYC8ULRT+9IYEXOKescrMEMmBr48iYTdEAOWWEIVgiOiMW45krWiUY+9ljS8CskVaL
Aua+KkByIBjvIz0ieTkqMIkgTYBdo7Efx5KIxTWVTMzK5Y4OQZ5nBubTAIabR2c3TV878tJPmkDba3SpTdphdI
yNUEDXUoxVZfyIynr98IXqgD7UFiNagR4KHBPUubgdci8NBoxlmQnkdnyNtaxIyZASrJedwptpUGC7ALb/a1xt
xLBOS6b2eCwIqNPhwfxP6VIy9EUuVErYLjisb9mpyTX9mQp64bMNkANKiZXlgHm7VtuOuQlql5CFHRaCCJYT4x
kYZzx+xAz5rFCrE4hklbwFdSWpXcqsUQK8cankHw0umNeccOGo0wlnncB+CfTmm6Lp0tMOq9GwlYUal96EK2nF
Zw2B6CiwHqRFjksydUJ5Dz4dnOcJPunFE5b1EEpplwG+E7J0rc8BJfhV75BsMMdZchez+gvrQ44OP08hSsXT/k
SiX6XRfZHIGLRelfrlBFJn5lWMkIgLYueoDPTSCQeJVfd8xam9E8K80bPI6yzSrnf41zrjU9Ul7a7Gn+04F+Yn
iHyZMlgklHBSzzToNhCLp+4N2R9MAUX5dSSmSWYDjIgM8ckGLgPdNl4V1Db/lnfISPGEJU96WsjUVgoAP/DW6+
8zKW/PenQmsQbTfckHWthP6WgkxiLuM5K9kmd24cLha3dgnmUTWA7+bj6Hy4Qo+ki+L3isTDBxEM2sDIHw1G5f
9P+ixCNvuVBCh+7YAi01DUJDS1Y8ts0yA/uHmqrjwQRG7T47oBSwWnH94E6GNsGM6gS4MmFaOBD7RguE7duUj2
XnbB4i7pJlwdMcZ1DSox21822rF5zr5cH6mRzaxB/Pi8jhaPsLTlYx5SKKTxB3UPK04LorJcH2PPnqkV1F7ff9
oQ0z9IvgVNF/D14jHKnGviiKHhFBuhI83cf11VOZ5VDn2HAbNHQCjv926+WvLO56M4opOpJoa1WfU5IR9d6DcF
oMt0oC1v+VfALZ1d6GxQzEVT3LcBuWCntGvMOG0Ak9kes6WuqqrQ/aKMgv2zLBIinG4Iy+Kfw/6APiMzUQrooI
616dH94pICGoNQ5JGbtmj26TDWJ3agwVmUqpSTtXG8omoDEMqfayzlVHr8e1gEWd9umn8oXJzcymmLqLLR6zaY
CtLWVEBz/+CK7tLxf+Jw9ZmZaNSOtTWhg+mE1mvmPemBcaDjSvczu+FPNlHQlTHpA6RYlhpuK7M/DJBKhl8T0s
T3Kc9Sckl5CZJ6pR+4o9AZDUjuAXDkOZ/mpw7R8C+z5pnp7qfVgN9PiPDChvwGuxrgueUUAQbY/hiEscWkE3ug
IHZ8gCJeD3W0H3RPbAhyUAthHD0SbhgZ5jY+vejUgu0JCz4gZ/2DAAA9NZaOKVq56fzNEl1wpC8Sg456+2KTgC
wrx0s3vkRya/iz7LCt2YIKi66Z+L0ER2xqmFRiFFKevGE4uEMLSuNOifJSugGExHV2zKFuCnEoJ1wJXpVoXf5B
KbBwMmpt7q6up2uhSllylsMSkjk6v4V6VgR5G/O1++kNDwy4t6bTeNk96euWf8A5RQtO9t3llLwLwmFCcqfw3kJ
f5xBVUXaby75DPqM9ZQJj/7J+/1ifU0VS4j31ZOG1Qmv+44hnV1ALOJJq/EocnHcWk7s/dp6HDp62KWH2XnZM1
s56vzFEYVqb2U5p7tsyAzGPIGZfnUlEBpeTyMSyMc+sdnTcJPDA3ieHBqfUFeEaeqrrsPxv++sA3xxcd1CbMKR
g4EhSDZFu9KBZYnek9BFiPjobkEaqR/3WfEJkv/8D9AxCM6TRshrjN0tFFo4uYm/QGyL8O88GPxjje7pRVYIgB
mxvzAXcNRjZPOQcZaBBlRoz8Y7uFyuAiTk4Z7qLJpKtDD7OBQ9afyHDnUG21QNAWrAy6zZvDl+Rv6nkyZKzAIb
kTcjP/+YJPh10Z0SPhxSD26+Cxstj7F/UoQ8+mvnVWFGOSp9F22GsC1RfGTMTwZCFC2b1eQKJtVzln+5xTjJ16
V+DydMRaQ/q3tVW3jSNnZsFapqzNFJ6YshPjzlfPZRfRTuk8eY3XOAsqqeTlA+AR0q6KlPKdv9RdnS84PzSaTm
dMVcAZplqlb2mk7A6+ZnpHmuCuWitILz6KW0sHF5J5WJT6VP1B6Gk6EfOxQGmG/ZkLMZLg5XKhx2S9bTo7ZRe7
JZJCduHsfjVQflMvxOzbiJ87BwHSsqB7qE/x2nklRUzyTshi3vPa92lSEvFXnZ3E5Ib4tTDdW8reYuqAZvCgfW
2k51QmBHGGqbY5lx/FV30L86MPssiJmIJ68fJURvNSz18nMYU73i4mF5GCUc32SzhjFv98bTbwvWkRgggFMgJe
vdZDjsa0CxakEEb7AjJt4eIX2c6hKmetlyO/l1sgkWxtq/JmSnjoK22Qm9jODpzfz/NwmYpA89a4msfIfM3qDB
fsib4A8cRGHWSRDr3lnsidXcAnRJYwM9d6YJD+QX1dH3SDlIY3TMTWscDcedYR1OIcnrLlrk+XQXe6LWGHW+q1
xqfWwuQGI+ZSoRiZhpoUSiRbrcRFkN6XgNnjQXbYKZ2JUbpiyZ71rDdAZ96ZBXM7fSnT1cEAmA6+BDDVkFJyh9
OmCT/ddc5SHfWzbUmXjXt59X5x7XUgnqGcrfrGJntcrgyyLptK26GYPlFep/zKxuG0D6T2zJZpemA6Sry/i1rj
rxq59aTbUlL0WvH3s3d+GX5cPVdnRK/r6BR9WZV3Ebjzzndgprq RnB8qB23cGNyxHk3W7p8IFpP6TdtbtHxsuy
ptHOg6UTVa0bqZ5ZkRAEpjYTy7q2cs1gvgKsuFmXtyrQZUlei7HG+4XmywrgV8eE+q+U3CqZa51GK8LtL2lczL
kmaZSeJl9KKielzYfsApgg13TCcFqKUITHg9196x/pMFnMP6qsE3FbTGSmlGtLpGBlq6U6kuhHrOl9VmWD25fG
BoDpL/rUxjX+hjTfv6eymo2NytrFV5+eo2J7llLfN3TsWDnh5nHT6h/uDKkZZ2ryF4r7xOByx8VmQXt+v9ZY5+
BMTNP3CDGEoyoK7sgoIHu9CkXeuRAQ035ubIF8TMQ/D8B1pW/6zNGQXnmbfaMPeWY6gd+anpr1I89usFBGB3oj
PIuP+kQIdhH8f7gfdaktUYPwTjm0zqfxkyaooJulrb4OmeiuRDY4NVA0di1WxWhk6UCyrOk/BiyJ11JnUaXSwB
Bpi1LmcvxHonY5+J0HyYTA2SKeAJoXQxORGYMv04c66u7esKYjnBl+1nW/LMzxk1P9LRlEnY9Qx5mer2qyb2nU
GGj81OzOcLNM6D7RFsdM11r3sW05vxhaFPgpIL16EpE07vIk/Kv5I6ezPdeqsPuX6fa6nDMO87ZqM/lHVcNW2X
8d5Y1ZjIMjXJQBCt9bvAP4zBK1QQSdmB7ZJGJMeYjysEDaEQDHjQYjyVaCFNdFJbbLvrOTrmvFXWVUWbbZ6pvF
AZI2BH/A8VMAV+305P34Bb0X4Zxq26qSDkzq/FYoKSIsWxa9C68Z5SHCBUuWv81aiDFh9IHkRoYBFFhva8vleO
NGd0u0rWHWN6g8Hd84zUnHOLzuN3hgTEgWWYRXNKAr91jHf/Po2qRiQMQtD58zxE1cGSEp7RSrIqdcUujxHKfU
sHqYNhduEETU11lIz4au0S0AJFmcehkIdD7x011WixJ4RK2s+N2GC/KXa5EKUDUadLg/bD7pWumIK5i6qW+z/w
aA6jOmdlWNBPIfCibiLxYJjO50Qha2uzi2+9TfGFckWrgssijWKjGFZh6ugYM1DhVyCuf74/l33Fmi8v+IrRTq
PueqjrB0dkXnwgC6rk5d9gFLy5e5jY8rcGiZCrujX7ISVUEY3RGJ2ivR0bJEwjM5Wh6yjv3xrMxjyp4Eg+2DXz
apIP1RkUApAfb3hk+5tdlvCSZqxVUdasDisFocBjHvedpJxBJUsJFq5NxgQUlcFGD+oKkQVf91D1dLWHR39gJT
lT+dLh/3sVpI9p5LIZOtqgt3dUP/5MSgA4NU+5mEyPeMTBcf/0Rc6PW/rKKf6kVIyPuQ7qYd0o/6D5m6Httzci
OIiZNLb6wQpyDP0Ry0ratr2WtxXLloJz6mN+HiiyVuw4TPUJidOR5v80RWIK4rRGRYSyW/P/6BHlXgqPsmp29q
wsX4LIVGv0RzLpLpp+8ZO7oEkiRYvPSXYTm4Tsw5lIbmy3BopVswVzj1KnAG80F3GQmp/OW4E0OVtG3NVBbDEW
```

```
m0dFjTB2wPJfpBW2AZzUGu9E6SnUqoYFucZq/PIXyjHBSYZ5xrW/eVw+DV9YJ8AN7bwBTZhRB33LaVWjSPUYZm
0iYYcXrzC9w1HmVqbsqQfVYsWnpg7a0AqjBWeBdFCqAhGxpCBjEc07GiazO3+YUgMY41oVGJFysCKtL8ANVsRg
NUqGy0ZKQULEuqXNHwRpaFwVQRedYPNCAScGJeIpNnk6P0HYTw5nCHCbkIYYYJAvlClJeVHNR1rk8d7DN7sy4X
Z4TXXqi4rZhNY0/w9w8Mo7BvZgdsHpehquLQpdvEthbqnT91ds8OrJweiwzQjDN6VY3czwrjY7lC+4edvxs6j4
jeyMpcAhdRLW3vziS2rZnLSLhVQZKr4A2RV28DbHkeVlaJyxCxP8T+V9ydDF+v3aNcTWaD1Rtq19/Zsnlwmmru
S2VjC0CHB3K82xxSRNCI/oZo2P5hXdbf76ZiruYe3EdXUg0kSdwUy4jLaL9/IuK+2kNVtd3aYaJ0XzUCPSuXX7
e57POgxfktFDHakuGhD7bOt+35y0NxSC4qMMFBEaEwcWbqH+rBS7lY2l03M+BYGZ+24j3GIOIFr3Q9mBQFlrA3
Cjbq2cGn52zNFtfEWJDZmFoF9yHlK6qPxBa2vGyR0GVHPx8hWEfMmWSXS0FjLdGr1dMV/XZ62XDWHi20QLnuyc
5oHUBStcy3GNQ+/7ORJtnveISB2tSEUBwh5ZBAUeh74KBx8phbvfJ3vOM0zPun53yAoTkp04VUw+B/ft3hCNkV
LO80drmBUBO+9f6e9d0dLqjItrnPKoZWvflk3KNELHBbgt2LvvG+inko0lfKyeL9Kvi1RhcNC+tzMkq9Vt0Fza
AheInZQdEaJ6AYDY8bEwwzGSf2WAio0zPRGrGlCj1mWHxFA6SHSVttKPhZPaiuxeSuXZR/sFElCB/yxKo8M0Fw
jbJTn4qGdt2N1+lWtRsjiRyL9uE4aEhOii83yy2FQzVslme0V374r2Pwygds95mnOsyjlNAMa9BIHDFnuXemCG
j/KBMJanjC2mQSxZsE/yHdOUXrWC4DBGLC7onxty+cfNTd/zmCIhBv3UYjNZNc3JX/cex3Nk7CNOMMCGtpjGgw
Ehh+FXqWrhJUObrg5pWKZ5wVeaRVsRwS4v4q2ASCekUm65gtxKoM8IctK/FS7ugDb24zxS9dWEGty96QVgo0ym
/z1VMVleV16qpOY8Ldul8xf9QTIBoJeXgEB1dv8ne/MjNZU8WAQ0KRLjpDMobyMJb4MuvqzK80lhZFrTcObX0N
9ED+TvzUMS6wX+HP2K9eJmf2kH0US7an98uWh6vozqKxdFpKj3NgxZxK2KiIIt/KWrJ2tvOe5ts3kUfUCxOC0U
ClIW2Lu6BlP2+3MUN7//mDR6HjGT96qXjtHFWFQKAc0nhfw/tbzDQnB+zAW6dE+kRjCKo6tnFgnVYNFlhtuQxB
Z3qh23hNiQB8pRww7oEK/R9aIOaDyj3Gn9iVa54Y3HgTOwakNvasHks/wr41TRTy4bD95lJsPikC0QrsrDd1Ln
zBPYVbLL8nsvRjFI8TE15JaZyBgIxXLLe1YdPMnSJe1oN97CwcsuQjR49nF9czch9Fh7eqyXcyA61dCJBor7RP
7PryDYrzn+J1nFO7j2/MsRpHGmL7QVE15NuEB8sTKAv52UC2Oyw0Di7bpkJcx8vyulRJL2nkjKPhGaPx4yln4a
8L1cVe37Gf3ipEfkaOXAri2YWi40GS0Ep75YHYTEs8t0UMISKXl9pGfDzw68+nRmYq6TTASJl48i6atjNshKDv
/1sTmDSpi9pvUOTHgKnuELFotYCldLSBBHTRjYa2hU4RC24kOp/+kl0L+t08GE/IZCbMMnWH9hTqRxZteIRH5E
/18Jp7vxGYwnE1TGn0OFaqTS3ICrLWK5yVRa2ME4fxzD1cJRb4OiTcDlGskcTSOcfJPKhFnx6ysCeDYqmF9v0Z
wfrG4YtpksbVqBDkS0jMMyd7SCU9QtmXXdlNMfxNS0iX0zypKyq7VLXQ1RPnG74mnI9DyItG+uxc5V688qqm0Q
eybZsCTlIchnAxQ15SMDvh+UuPjsc19t4r0IdyMCsPXjgbXYSGVC6VA53DaL+QsrXJ6fnmGH07fc5hlH830YNc
Aqo/peJaMgtmXIo0hygeQ6AGccZfRj46UJrqGYB0o077h9KjmGMYTq/S4XAT+g3fachXHq2gjttGGHXvO9p1jM
PSTlZCtSkCGdX4iqXDe+L9GdDlYs2pD1lsvAJlE8wHDzgXbpHOIbbbafBTzskTIQ7R9uLyCmEUU1kU0/D573lG
vR9TaKfW31znDY7ciYZLOADNm/Ec3IHRbbkCyiPD8D2t8N84ySIcHMxMcA96IDakMT4tyHswqUez0A3T3bRbtg
dhwTev7fcZ/ZQLGmm5i+qOcmRjkcEa3fCvaKFiW10+WIr8iKxlLyGh+yr6Od0/HJy7u49J836hYwSisod/VQVD
gs+E5+mYtjNJSDbitp/RgKQmPNFaLTiCD1gU+XMl/OdocrW7bOlvPTDhY8AqgtR6np9kzIp7KgSJvJxhVNEaRy
3YbewVfBGFXWH8uajYhJWjucKHKgDBTkbTu7a8YFBxGRJttxOK9hbZ2Wa9dsPDFF2T1BrU2tSv+JWXcRi2K8/H
GC8Bs6qBvoGQs+Egl6VjQFj8DSBLYrHRg7BfnOlOeCSkJxTH3C+ideijq1rqx6ExQys8yi8ZJvDytx8GfHKqUl
Q3R541PNph/typhfwK0v/yGw2kYwsCHBPJPJRvFnSlYSBMhXMfvH+fWvZi5cNJ4mlzaIT6MScNaj8CvrEDOo6o
6kOrO81mrJKK5gQZkv7deXVJLLUGQU8u8mkkaYfp5a3kwYe9wZzLtTkwquJZ2iNr11GOvnFWC3HdKfYzezFbks
2LBf/wcPognd01xEJRTu9VFxh7KdqysqGWnrd8X2FbUuzIwZRuU6UYcRKcamKtW2jlJbr77zClRdtVmJzdb38s
Jx+g26iOLtODO//XgbUqiaOlB5y5vXJ/3zzmAyEppCI4cbbLNfSX70bGzDVuWixalM9VfluI5NI0AH+uBld0uk
yaGXnWLCYfF8rQQWw6zyZCI3QjO6XsJHHXjjcWqpiuks+sQVjMIBdNt7BQlm4ESNKNWWZ/NxCmMTjHI7LITYpr
wk8rN1MzQlmYopl+2FfwXtZoTMAvFeskIrmZTkp3nJsMKAFz973ml04HOgYERrb0sEujqbQYUpSg6SASC5++Z2
KqBslYzWalGsCveR+BDTTo2XML3093Yiv2m+ASFUMM/cxR8BzAJZNTiZAVlVo3GPO4WQ/wRQNZ5DWG6xp/bLKR
FrJinBkoyyS4uHMI8SkoYhEIOMNOSzkeiyu6G5ThxPXyJCpffB3JMl0h7ELuMwxeBzcl0JWpAP4p6N+KqDWSuW
EMTZJnL1gj9JXGXbIfy2IiU9mZE7MQGSJWdvZIt6ekpReu6lZ1dBeXOsgFiuEKhBBob6fKx1akcwI9y+iEO70W
27ij8b3XbkwCP0oZbwFpswHr1XPR1oHIBJjo79HCR+EakgBm970e+N5tHzcBLuFhap4cyhWJEBgnNYzocz/v6d
X8YeD9TDN4b+vvQJEdSA6Ue4f7gLEihuHwZxn/tOf7SZ9emXPruzUzRVxh2G3KaMxPsTAEJYPwSd8L4c/2T/f0
djBvCG0btgeKX5+/nGZVm7tuUxJM6yb4patp2knNU7hruizozjuzMj+QPjRXLJnigPmjmNAeTh6nXXecX3v2Gv
m2Se9bYo9ivpDlzVpSjHwKvwEpBo7leoD79nrV6jUSU6kXpTYd6O+SZfclby8mdniOhANxBUHIHuUBmEOt3JRy
o/hxejRlzpJycnpaJ3CYhu/D2mBZrhk7FTMwkzoPUvLz6KBzvHyq497FCwteOSc6jmNS1a5FvYE6+0O8HBwBuB
nAXTWDKDCRNpaLlINsxkTKlpgQbXuxO4lsNpjhHHUcmhwhO1AvBDMDMz4gPFL6YdtJ1gPmWg+HJyHHVeZQP6X4
IwdreiS3ns/jMafl3RM8lkgrQPEtWLrsYHD+CC8j+mrAT42JhyaMwMxKhT7+BInRacDQQFIsWXhwZvu75Bsg7O
7bS7iXir5IAmqtUzfehtMKo/Kk7N9fiKLe98jM6y2siRpKq/z1I2/kdmRfnZZSjDs50AAY6/pTXtjMlsv8BISO
1I9BRaUONEMzlkCNbTm/HYyFKCHjd8wWqCQ+vg45m+DV18xclb7XLUvQRUE2EbqflodPDAyphMhIwrmwnMGM95
tcgby7rpI9nLXCprZgZQ6qUxCVXUY9Q36wmmBiNr+0SzS3kYNWrAdVoWTVMAFGr04M82vwh552Mz5/aJpAR01t
```

```
4PZYO9xsNNLHBi8Y7k4pkvQ95/ebVTuMkbONf+yt5PtJ4SgIStmXpbkQAlgpPoGViZjYwh51fkYRjBfyhyJs3B
SV/I8e8lIhlQNqV3uBiVqvHTjMSTiwFs74rX/4lBPLsz85B+nI4QvyTS0KghZyKzWXJYdyE8TrMAvzcvjMZ0WP
AOBxIQTbhzbnV8nI+M22re9y9cktxRSrqucyySj6ndDQDJqB8bphuphev0+lKZJctlA5Fswgp3e71EiQ+UbEOV
cJJ4XkPTbwYnvNADkaw4Tn7R7YGeizweEr5EJ4D+LSOlp29w0bWXZmiP0v+ABecQdCdtF/HkGRvCUzkLusdppO
ISzK2Lig6jonlqMCMOQyTHhJ3eHDLgWn63K1ZuzdYAc1qnno076KN6E0ZzpawRdgv+pAxuiCdYEZF5XZrFS3HJ
3N1wqY6TaecL/H303D+rebkJ7A/JdiHsdZ9xwupZtNLS+alkvSZJy7g/YAMM3LOAYamTqyOQ4LXZNtX7AQVHNY
VWiLl4Qu3boQEpIEPX4QSpbSu9KDTDpOX0wN/MNNxPpXjQ0HjF/c1dUet2fWHvZXtnBGAVdn7tCgdQf4dKXfP/
MCL5Al5rYbxwOqbsZPsl5poP4gfmy/GKGsXIeLJ2OvnrNnptLQd2ap4ciC+p8odTdmokfETnvqjUkpASqJHrh2
KPhzxWmZaDESNXLTAKPhWL+RA17NYH7k9ZQzxuFxXCB55rzR1/W9lSFxTEctSy8W26D1WT67+tblYBvhbctLTr
BEf2PlnGx0zAAFwoRyxDe9ROAkMEF7sMhgWdeagTkGDdwohWOWwFtaFH0mhxTBp3zR1Davm/f5EqJRcE0Ny7JF
ua1Uwv/UiSsAA4w80aSJcvRoDVWJnzs9+1AJpOizby3EEZCjxCADMOVeV64sJ3SikMLgfE1lHV0fRIuQ4U+rb8
OGNqVuy73++fZLq45n8vQOyBBLGB8Ovt1Z6ymBivJO0aKcD4WcPWN4kjvsPQwPHGL1xTw6iJD3pAiCAR7VH37Y
+IZJ+kBGhdXs1E9QO6drx0vIgtuJbU/qXnTOTqURodICre+pkhRG58iYiktvHa0zWx3GiDv2zVNLNFCco121z0
QCvWBtImOTrj39FlGY9x7yEl/Vy6glPNY5+847C8ajXjuZROSgMWmSGFXShJaDrTzUlqHaFU7m7JaDRor04Rs0
Mk2AW6v+R/vnsFO8w+HF50gnGaNE6NwMy6L27A+DPtJb4WsHju+egaMQjDPyJp2LsEHHTh7RaKXhhWpqjkdYK9
4SwB7x36XaKj46KAcOlHbP9F68RB33Lt5ErRrdJY5OwLO9JV2pKRwFnv+App+PJKWRNYhQ4gTXtAjReueAyURC
57OYL2qtkGk/LOTbbjH+sRWry3lIHTGamgEWDtVfbl4leq9dOaBMuLo9zKbTdxGWao56h42XdV2jEaRSFucOPZ
YfFNLYOl3CcZGoSgcPwoM3Bu8iqScaXLOxj6OdqBL6m4JObBc3QOsDDSnfTw3VcrtReEwgiWJY+RUKFRovd3ML
9vYtptf+sz9ty8gXvI2Nb1Lz9KdVQBLJxvYwfbODZT7qxUVpMzufB89//8XhWqTY6QNDm8qtVQg/OP5TCWplWD
39CVA6AvMvQJQ/pvUMHrV23dCCd6lPVEGQuek+UocSex3R/QFHHl1a6d1J67ZjPcal36w4ahi96mTPQvb/Idkl
5yyDYj4v85J6dZ71vUxhj/JYZGLkCn9UiX5feKEdRKQHHF7vsRRUEPEb8sIrpDh5NZNT0zEodsWVV5O9VMLpqD
Z1SNJ9Ma8Oqbb+zThfrewGVH4eeexXGSzl0lXh+/PSsxTFNbPXurjt28gHdi7SIQsR+SSuIFIrdEBZqc2EONoY
EbjB5aun+ClyMcsGGH7Rum6jsrJqhaMZt5Re/UwSzwdRwCKjUORHb1qzDX+JZOnHigSNtr9Kq6z/92efaS2dhG
z3mL7ZkbOjgwjxKLxbqIlW7NX+Ia3L+gZ4y8KoBOuupL1vTxXeeA54V1IUC77z6sB2QOimOzjdn9/yfHDVzBKD
/GwKpIXhZ9QMGh0qrmGcMP3rnrPlDnDEDI50ankSLQsd+/i4pIhd1Wgvh2m4GS10mLquHMxrwBy07kXSCT3ggq
WCMWOjX/WQJtv2xmc+uZgRgsib3bxzxHTI2D92U9QzLbVwhLi8tHSHrsLGcUk8fv89U4L5yaE3XcSmMF1bp8U7
xmLRWv/9ixGMeNmkZ8a/9NSuYsVuzbVt1zmlRQ9hQmhA0drkdrJsugqeq3bDhlU4jT3gdHRYoQSg90gzUbC+Tz
oGKIY7wGxLsRo4oPzr/VcGCluBEiF9MuTI6z3zI5vxmG49tPSNaDJ1xHqGCiSrZqEV3Ol0vK8t84N+aB11oXk+
3+Nptt0QWhM3xxqwTTWMbJ0+6I7nEqYdLOlw51qf3aNL2nJ/cRe6BeLhQTEOAvnrbKMfFlNdZ/ZBRBhmSIQNJf
ChAOFQ/N+i2A9f9kNljVxziAllNSDrwpHcZoNJKPptSfhD5pK7+EKDNEFAR3xKEYzjbqgHUFXsYefoaidRs5St
9Wclek+OLSJAo0bwo9n4fubVwTJBGj75oFhC/zF2CJ8VlNHyErZgOu5IZXxGtu6cvwPCMAlpEJi0uQeD8/wqyQ
x1KZKZzDLpUT9nj3JOWZfVCbyhMiyZkvrc9NU+w74zK6R3KLJewx7dv+OCFKXhK7ZRHNXo/gfZfNb7u9fgkp/X
qmdAtVurJ5/JDCawwb7vR5qV7sHGF8vTDZV+/V/EqlE/Ea0HXxNhyzl3MbGDqmYnFucU4S0sVjpfzYWnK8K7be
rQ3ujaunGWYsivqvO+UNMN5IODVKHvuEQNSxZRcnw7fUMq8XLm3T8KUg6ESwFOOtjZEUNFR86kb5qhv1H0kIVj
f7oYtGlsSZyFEXOTQ9JmmT0aVf5n+dpHQNK6spRNMoiiBN0CEg9kXmEkaQSy+4UaBmfxpxuLNJLwSqJPIRB35U
6fF64uKA/Tofa43OarHRufpFFbHziNid0aegrZJHgI16V9fRtSQU+ir0JbswrbtmT/cwX3l2j+tK//InF+zgya
E49L16JMDIQ88vzR4yWIJPehGO+/oskMeqqMn19nM7cpVsYJzJyENMrIx0LmjuxhtY1gSlH0S6YAbB0WJi57L3
FLvfsfP/TMZkJ44U1yqp45t6l8qY54d48S9yyx9TC1Ia2ZdDYjR51Z6OPbowE4tzUAmnzNH55HvQH3BbFU7q7Q
JAut5DcnY6eqS03WY/RSmGd+OZ5U+xtp71bgyj/aXe09U9HPYg4wRp/SScaR1CggdHiuz9u69NIN2AsTaFhj4a
XpaWEkwi66ZeT2KRI8s6qhmUigfDeiweW3eWrNPWWhJMP0UCfrM4VNN1XgdWKu/GUdhy8mG2db2LvHU0pp3p4+
Yah7TPRlFFkyuuZB5feMPYIEG/tc9jvoTi2ZhVVUELybyt/D93ISZw+dqt25OcHVbCW0+4ajU8UJlPVRKfKeaj
Zj5LGEVN4ho70PEC9+peTPggwg+WvACGYmTW5MD2TXALzs7c3S9pCIxwFLSrQqOK3wW/PtRLRiJyRj7TETEqe0
cPQCGxw7XgGKCfddYA6wnX/TDZQxp0LUz0nDtvESj90BtBL+KAXIW3/KLH2u8cJKDkDg3TVvCNNGK0gXT+bhVU
nrC0SW2/2/Jgvg7EJxL6pgd+OzJUDY0Cx3dbFbjQtHBIxnVUdK7MTKgWSv56Ttfru2UDaSabQ2+mkEqZKcHKSe
SJVioJYVGJ2Q+Vx7OPLUPE9/NOvTo9ResVqYz+HGxwefOOutovGeQMW16IL/N1/Dhj4ITlN1F3I/8oAd8tX8IF
8Kvz7NNbi1n/i2IaE8cdmGT95MYHxyd725wzwg+4l33V6LWx3nYU8N01lPG3/mgljm3jzdbw87vp2XmPuEKzYR
OARkJXnH9wOGJ/dA5PDGt2KBNUICsp6GsBZLcinIJEEMkLT7bidO5o3KfmBpv1ECtiK7yXS61FMtwu4Fh/sR6o
YJ71goMUnIIVDb+IrGf2Af4iqIZ3wjzQwcxEjBZy0c1PdnL1WUZb1vyrpfvTfCeyMe5YjjHuXxxweGatush+iF
pnUZezeWZd56iqwNJXixiAvpqNHG4DMRAMoZVBHWbZsEozO6GGH2zNKlfUlaRMzTdfYcKbqOSptTUPRqDwAodW
xL7tw/wZFL6rqADVzzauEgE9Tv3BvDQCmqAL++2zSB5ld+NahpJ78TuAuzqhW+egWJsryD0Q4MeB8dfgUPyctr
kPt1ptIBQ1BHQC379KLVNBSyHI5vDZ6p4p8PkTu7LIJf6itUNzH/kh46p3ktPTqY8vUN26LJ4uJvUliWSSpigb
```

```
Y6vpbO4i7jX33QKZxB7xiZrAcpd+1QbZ5jXw9DxkNl2VbyjIB67uFutAaP8VyAvjuYadY8JBXiGdKdGbRqzobK
2p24e6s7NheOeFSi0gDaNVW8a69pjiDje+1qzWBixA9KUDqmO/LG9JcyZucYZ7DMdeM2mecye1ehSASN5SbGrZ
LGiEBpeDzcrY/LC5a0eqeLZ5IsYGb85zdnpSIXpiddF2LUtshrbC5dQY0ZmXAlY767cNh5tIjpitVkfUDJB21J
0wsfxsYvWvHfO0krbbBHyeXQlkXV1jLRF+o8L2fedIkv9L35HtQPdB7FiVoOT+RewJQrsca5DY35AHXMNLPRUd
7wGxUD+/5BkJ+tr7x7lBzjejMrymNvVRYej94ep6ENxec22f+PcS0+lOOh7ZxFrbIEBstLsXoVFqKhRe0hW23i
RmBlcRP371AmPc8ZTsjgJZY037qQwdNZX8Zh5ouTW/lLW9oHufPpnLZLbphSWzBJ/FCONWUoC+TO36+FkWYNDk
1p8rNxOTcCnbQm0N3tQkGfWF5bCOX57ccN+z132nGdxnQ4pJ1qtnA4GcTMfyaP9+OXBO/P+LgP/WMy5ynD2W2i
D0Me0ZIL0YVmbX7Uw6hlfLmqV0lKl0Kb9HtSm3JFp4m98DXO97lYcVPHER8qiao05h/KBK1UfCVBF9HB75dgU0
StnyThJug4Is69AN9X3KAoRjqylvotZGsTXt3fAnBr0jjDiHfv0J6izX2YpNaTbNHMgRCpVZiYibUSVCGRpRN1
+DOE2CTFcWgCFCj1Awa6AnQ2rOH4XzrhBOo5zXf2s+h+xRBQMvytwdeOZVRCcr6Yb9tHNwMvi6BMRGMUgMyOV9
bNkA15DoJbwGRUWpZV4fESWDuiosLnijpVVlhCx4nmQGVdQpuBi2FzU/eBa/cZCgpfK+pnPQo33DBzu3vOPEhq
Dox0OzInTf5IEm1BA1xVR54Yn78Qeqfvf1KRIZCeeTxEMnWdnYPQvKJLsTdCpoEgQFZGVKQ/EcVN+sRpqTnqZz
bTLY3HDjvJsbWJFsZ3BgFxpZAz6Ws1aZg9rej2cHsfkYADHNCbIg51agHany0opQamRcxL8XCvFRfLkMzr/DVj
HDJ5NFdqwTPxlsfEHlIyJr9yM0RiX2OepSW4z2W0lPWmuV/1LpaSdHwnv4dd4oZCML7nXlmtrd0TjppKVAVZVI
ObjdCsRX02rw+dxxv/4Xln63QBjV1GHKXfxVaZZ2MIS3nNfhRSPQnf56BBFpugwCNcG7nwhgNwDHE51Gx/BGx7
RQQCg2GzHkVWAZ7EHetuzXgDR6Ape6BnhZCDGC1QLdEV8sBU8S/9K9mbaZj0tH5FwqDhxd98d+tL108DYfCam2
IhRSUEiDR8VMUlq0rNUcBJWiY3x4LSDXMgS9yNLdEO2Ja/1ByL+isqEjXJKYg4HPrgEioCa3I6dAIPY67/icIM
wO3CTS8z/qwgkd1cRj6PZ0eDnFD+kivvDbecJbHs2DCC9xUTFr5u0cPpj+1WnOfWYvr1DxCbtiLCcVT6qy+jMu
SO3io430P+cclXuyLFFU1zsM09Et+NemzZGfg5MK3QifqA9RcTo9+T4+oIi84TX+DgjpWeqrLltJf7CIH23zQe
8VXlCZdgs4E9XJ+HkobDH144Vehf2yZtkS9zebc8H2EEhvfEbQz/+CarzsrAJwIILfB32/2IPYULFkZBHjEMqF
RbsIVk5kl8cruW70WrXQ+zvKUiyjuvxj+5ykYg7lZprwRZAGw1V0YYjQ108Pf+xrRsn+XtLkfBGEX997Pq/nyK
6PYXGk0jIQr76KLKHRcQrCOrX9fwSWqaQSTRQLJcZLs1rwWMXkOht4HlzFvJOAnmadYUA+YIWz3SuINO55HyXl
tLZZTVf2i8fIy5q3o0fqDAjgWugiIZ0lzOMs0KPcb1bmmP+M/SojLshUq2yhs755AE+sSrlNOlqTdREQed5cHA
q0oxO5Q5Qel+UdQdBVMbbdgg6Ig/CdNNu5Jf6YZtory/4AjQAJK0rtzBhn8ni83/dS0F0hF+iRmYNWHeKmoU+S
KDDRVkPqYEMSKDh26kKGyhvASbDtjvV3mUwirQq8sFmdKdK5bCwMpa/YewlCQlxkFT6K6xQJ1Zkz1JUlQMDCpK
ShCJarpkn3Htn3Q9La+ivWFngfiBVXXcopztTf3SRrsZuStHmtjZ8JuI5n2Xxbdvs6RDfodRxhTMWAXs0Fq/aB
28ZTfN5yaqJa2NdA6yklV8OKhfqRDWDvNOshqcNcVIaUPEgasRWSxv1VjkowehM9YjNhQ4TwluEhgSQwL0Evlf
/8eG5aQuJA8jHGrAF2TL43W9N0YkqoepXs+8Ect6F7bbnl6pwANhVUynq2OfF1uyWlEHNDrnwoXugPYFBRxcsf
kxxDeoCz7RoG8UxaATjxCj6wWkYINtIITeZCGX/kOCx393BpoSTVWaFbQKCTY60GnEInvvsyq/f4mnzhIb4pqk
juzWFGyyqZ/5MFHqrthX+Zeh+rjouPr/japA7IZwczq3b6q1fu7nYevCes2dGT8wVY4Mc91sLf6bh/b+nQ4aQl
PItHR8mUGyBa8Ztlx5v8cM9KBihOQtnvE/tPVnbrIq1n3zxvNUT4WHQdcSJAHxusXQT+7iFiCaLi6lkrF6o1Hu
SL5CM5iP/VOQ0hZN+5+m/SCo9c4vhHBw8pN1btguSCXEAsiRZb/fUZB2f/C6iVQ8w4pihSm/f9BfWSWM1FVW/7
URPdMkbbfmgvY2p+UmgDP+DaFxTGQXA2uaeNFt+c1aVKzO03oo1NKkuRXIq7tc/AXqgMscNM0ZEtibzILWvZmR
HZtcLf5/YG2mkntx3sooGuW57kAiUsRsv7XoJjGChLByvTLBnUcan7RSX4bvwVRKiT9cHOSFSU8BnHYMF38F/D
PG8Rsujp914j8d3/SuEilV+Yc1Zbifw3qq9M7sKfm2qLJmA1J0MTNPf+IHWaR5VNn3qRopQRXqErr1/ExuCp6g
2IXcDjTFBteGvigiu7+fF77U4GJPQvUrtxEZpQL+l0DYDBvmbMRRtooQ9d2+lLRBzst7IThdjM8jHICEcAGNHT
JS86ur5yAtVHNCvHZKVHfy1Pl98APc31oXMctxV2SdcOUgv7HRqtkjjfhbxM72O5JL2bbkCMMyX7WEZ+3zKDLK
53KRQ1GEI8urwcrYzsav8GpASE9TjpLWxjY2bQrnUVXJuMvVG5sUfDi1TzZ1vdgU/CsnPdNY2E25AExOALia6F
URf1lST91BJsmlD6uYGnys2BoYO8BRLg3/pjLyaID/KMxnYoR0UYLjufkptDnwe3OS4chLHewtp3Mz/AZHMrQo
VpXb7RwMY68zJTnnusUonkBggCn8l6QGJclN0+knGiX4mkqv6fDLPS0cI9cibaeF7kjy0xK9GUPrebt7aM/DMF
MWWKpe1NziyjYrdiWjvIq0LCyOz9SeND2gx+5CsNnDZrfDmaD3p6iDRvEt0ymWfNWKoxGnLQFv4W3N1/MNpKVm
PzpjvrpghltYTbgBZJK/OwyJpakcodRWt5AERC+Q2S6I7rfTYzWOtv/9h1lQosfOqxKOrBbH1Lwn89jEptSQmM
8QWpBvDOUAAgoCvF6SP9IFE9pXcA2yUs5rFlv98/Uw0Hmsi9/TCVHMC90kQbUQkfRmgDI7Lll3H+1aCiMIgEgk
y75mlIMitFQHibIH2A0IF3Sc5qclzOk8Ym3j9QUYRgTDxi7nQSKSpJPv4N0J1YSsxQ7S0A57RIHgOq8zitam5h
zavH8UawU5PIlLpm3H878LK1lSCxOGU9IB3ApJSlZxZZ0j1fS0dnfZDJ+ssdZEtTTH8YyENDyRqfcdLk32WAPb
W+l+86KduzdG7aJvL98/oqMiUDowKWgcbVDjv23M56ajfZgRPmRFpk3DfdvfHz41DcMmfeQ8kZNkCL3npsBxpO
DqafdRB8+KpxerPbN+K0BZoi8j1RsJuU/G276FF6YMbrFCvscO2oqXerMcxXJrb3Uo1vOBA+IUAkIDcEj0wmsY
xAZp3tnz9bnnJWlMAoikDS9HyEf2lO+nzQZFasLbnzQhf+UCklKFoIEbm4EHKwG1rPOKKQ1IcOdIYwF2XNeRN6
v72/b8WOz9a8dfmfN6fSWeYBMQyneUvXX5GJZ/+DpHA76z/+C37UqrExwlTyfJmkNi9lH4MQmHHilHU5UQPcWg
+RGDjxFmu6SOQUBQ4uJP+QdLkc/vDsVKHw2KksAOtDndZM2Fi/0fUj+GCM4testCEU0BVf8JlPi9z7Y9tdEGw5
RfCDSMhDC7IMjcSe37xHNS6hP4aJN5aBmPlqgHP1voYIyz+XO0DixnNSxTWCqXmxLkWFfMbDovzUZtAJ2OlFIp
```

```
wgD4qY+ppEza8rBsi3LqgEAb0A7rTB8d8XzzE22A6ohyZaqCgPEaxBpGwOsa/KDDJCBEgAyEIQe1oAQMeXMT1d
Dd3FAZjZlEaq8mWyIkZd4NeWauju5gAzVBM18djiJ1Fzpc2V13FyQcStN8I9ODp6GpspsAWLay37ZnikxjRwpi
Co3cpkDpoynrv145b9I5U0zBJtCw1v7dMpBt+eyhJnGYd0ROXs3UX9rVXRUekeymUQZp1flmuHGf4jzK1ImN5S
eLNEwHEdmiQBGVtg4COQv5Rr6LsrzK0wXsPWglDUAeKOUv+9nNRSqRVaH8rfZepkh4kkGfKMNEHFo92FrDaBU/
z9dPmC1gIeTaekLwdMjbHQqfo1AR6dwz/JM8h+Dhnca7k1BkXfE1s6r7JCvAok5bYlu2Nm0yfiktVe1GyIr+9q
oZ5MNknyocVj4gzhsY+XSK63vvehB7n4C1ROXtE9Z8djJtnxWb/HECCOHC8BDkZ3kzfy+ZNReW1e1mSnRefkFw
/1CTR2fhD00q6S99eA9e6jTLvEx1v8p1vsRnRhF68IzXre4W2/Bnb6ppTNAunCigdmUJrnwn1tlU6WdvFhMbjw
xoDhRiBqNrqafjEvhUH5zCnbQVOntxp/A99B+2WpEHmBbg8oyVKodV2CQmjE4e2aii2oybU4wx2SsdSZY5zwly
LXsNrigrizYYlp0UUr68SalGRw5bKSwzzKEsAbNQO11224eDThcwE3jicJcrJ7EYLttODphVQfFZUvCR7drepG
daLnLxP3txMbGUZTl5sU4+RX01nyI8jeQDt6jOe4Tsjjnh0mySp/OtUcOFJ2ME4UsVsrcqfWPnCLucy6X0pyzW
XsEedsnz+kt3znYtzPTzTaHXqPaJVFx2ewt9L38sznS7mXkfWgWV9K9dUKCCkD6TfrJuhpOpjesYG+FlZhrPEp
BD2tConchM9XCyhXWQJALOJP1ASLVSRHkYZnwjlhnxI4PnKZ/ZyVpyCsCl9JL6P3ekZLOLvA1PzaVSkBNGXT3M
jbOia9FOD/cConJEsWX3WHscD4pRJSr1Ynppzm1WPdn9kzxDlOKz6XX6T8Oqg0UAQERqSAMbGKpBPGLxdggTlS
AGH+/6iQ69wkLvz55sRG3evvJNuOiKWoJNAKTVwTGg+JwA/N/swcQkRmRUtzjh3H2rlYsl7rhgSpRoOF6ZwO/9
JcrIJ+DGRBnpXb8+3L1kQpcM8PiwXfq2IpDoGRuw0XKwAqgyV479b3e/8TzU6xQCQ0YryzKTlZ5Gg6hpPHkei3
OYlDQLp8uZce/3KUXUsaNSVwAk85y3Sje58G/QKfm1fIJOryKpF9tNaIyWmASdM9SAkaeADS4JXaHTAfy1mT4j
Err31TXCIvIN/ucyQJVy02pUQQQh2SsGIYAYtGOwulx9qMHURd+nBWez+nlQnU4nxsQ8MfzqZjp/pEgmUo8IWY
KJk5CaZkYr/HLICHe1WYLMDK5b6rVObCauRz/XgubW9/qDL3bKsC96OyPDqIxw9CrGbxjeUGZcklzQqCMMz/YH
bHCKK9rTPSZmAbss1aR2d/MH7LVEuMySSEajr6rlOKqsmkhZxSm90JBIsouwiNhJmqWgBpVYVwYX85g4IQU5X2
aBB7oijefEvJbuZIJ67c6LBZZand2edfGjs2mhzvignDDjLgajrRJzigslP9chqrz5xbnHbCRz+kn+M8eCfJ9m
SehnYmvQ0E3Xy69ifAAivYNSU31RdWChiTb3JA5UkIPyGY9yNfWJu3JmoR75n/+9T2NjyV+BVKSkNndUNCiGc1
wL/rsDDBQrSpIzZfVUvetGxYAcTyEMQGewuDNZauqXVj5igwo5NdGrTVns3RVG5VLiFCVgAKqETW80B6JlIMIn
x+FGRvdIwH7FktFRMQR7wzkQEmwXWM7Zk+ty8BsRbw/pNr9Eg43q9rcSy/qsbpReYEkvczgaExwWG7ODK1Yw0x
DavfaXMwP0CbOZ2qjk5zsz4Gijck5h+IAq+2nFj0qMgqWt5+OSPjVLZ7ECaOcUZuL/Q0Bhkdc4fCmUa5ekclI9
fkjH96lj6AicNAK9K6JhfJlmVjC4Q/oFcV2xx3vafIT//1R14jqh/hMcCUke106C2D0m8ss0pRIS3oO5i0eMOq
VZen+xamVXOjgC63EA75wn7UutB4bRRZ9ATZ8kP7KVMCCnxUdyPFrgsOMxEJchQ5oPjn1557DYSA+W+otMvMBy
t4BnoTEt4T09NLEBmnuI97isofkz/1QiP7aJxzTdszSiVAMGcw5YoNqxRPf1m8Y3PUEvSPogysmMSIag//NKOo
LwivhmacKbfH6uhvL720F8rnO8edI5pWlDRkfNjoCSwC//pYg/6CObIznrjs/ihtMzXNOggbGP1HQwahzauuWs
LvhwMJ668HOyUnJalyaMfY/mInaoDGIFzjYn3CF1C+qgE+YMl/rlBOGxpWNrQFJoPW0uvK4l1t/fdrNu5wYcqQ
adSZNNPlVmUDyuDcF+x25L9DouBBrZbEOkaGh0sY6A/6pRF+kuAbFIPmSV3ZcJjKZu+enFdGiZ/kolz+1xU84b
pyhupz/MiQiMeyHeVmjCvrS0VvBdRMv9gvEAUNclxRQ1WM4kwus2U0KqX9L4yFvD0kpy0p28iWVxxip4offlQr
UQWgPE/B1gNX7pD0zwTh/upAo/jUbUA6fPQygeo9O8m+LeddoCfcEt04jCTIP+nq/ZvuVd3zf93BYVar2J5qR1
uNyTPfA6dX56YHRT5I7Ts03jJG8+riLRxFvVoJIfkI5LtgpzEy7e9Z6h6Gvigzcb4ebqJppht+39fj6Rl3VWzC
3wFS/Hmha6x8J91onuWV9LIWXDu8YX+cY8s+ZwRDriYGMvW7D154AkeXCD2EgNyacZ2xyLaaumUbCOI2JBMaqb
Amcoj/n7U26A4azt6DMax2BLzv09MZCCRnoNtSEhevGXzo7PlCyWR+w90+EN/B6tIYMcVxd3ATJqnYuOafqMYw
pTygjsEt7eGvuTPmNln5ro9itEU2aUbLVwD6LHbmqJpsZDhoNV5nACk704Ij4VR3EjO6Z2hGpo+R8EKF0uXuyy
Zf9jWdItfthBdGFqlEXaWHsKWjEBhN8fm+Yl6it+L/sVTS+YckdvZfyHNbPrpwY2Rt2bME+oQIzo9zuJ2LCwEj
BFyBxpxLLMjDGLfOyNbtT6LeuCUfMQfyC01+9HxdQ2TVKjr+2jAfLIEWHpef/6F+TwYDXHc5hSLQZJ4Zk7yZCc
ARzCOnsvqVokU7zq/X3VagpUITSZz/1OAVpZP3BaRkjCjrweBJhuYTrYcoOwLyJ3qFAitwfZDyRG2Ma0fCcZ+0
1SRv8KY78j1zdIgMGgKSUudh70S19ICuPeWe7cLTf30IHW391q4sjbrxNoZkCmqPee9be1D7LyT/ZPlQxeJJDP
Qxfqvcdmm+vduNkryjqk9Ngzt2Wv7GbvXQ7NbJKfbPB086JMcI64DMD1EUQr+CBKopZNVYTTCnQxfBqUIwdgoI
j6g5F9xV2JzVaWNRIsavxSnnMfz2C65Ln2W5HRppkh6sqy1xL4LguyQxqSTI1TRzI57FuZde43EvJ4Zrllhwuq
XGWzZ1vee/+84QGINPU5LFWBDQyPyQkB+UdkBh+5kHRLUertHrNDYRV6zCmsU1M30379BcRPAl1okLuzLMIcVl
7EGkjDw6+aUCzVkQASkOGWoQEM1u1i8NIi5Nak8z7FwRMQRsTQsopcDFARHwC3uwP70PvQZ803rP6JX6HVE2Nf
Cc7oge9P/G4dlGOEbGEctA+zcDf7T3Tjoxmbwuh+u25rGkCH+F2qZTXLKxYYbv9/atYv8NiGHw+JHs7pAG3wQZ
+pHbulkSGAiVcA7UgSvvi3lftnLQzvCZt45Y4SWHJH1EgSVVX1siJaHW8+O5Zu0fl9XTTRRH+7AJItwYgZ1iOl
/2SaSRTqh8cdC6a77Wy+he7+7i3h5NKD3CXHrV+awPqRW7tkMbA/p4YHLlRsaO9pUAHdRsVacvAleZSox9q06i
zO9PH95XT1bQdRBw3RhrsP/sVWQA7s/iXG13WXHQPiEFk4+4As3YMypnqM4fNWCHngPwKCfGg/w8Tx5KtXuv6s
j3RAeVbhzR657iKlJbG4qgAJETIgNGqXDQ7auyaBP4Q1XUWzyod9D5Mz+xMq9o5Tgp4zpSiYWG13G/mC/qwVlM
sp3vOERh44tRhjkR9oJXa6NHn4GITQ9P3MF6gpEFXomILXDlMNRb44qg5UGJM26tq44peBWELCxGAD1j5mlXdI
B4sb15x5RPiHHbrRkwwy6VfSybQR47CJgduZiMUqIfWUKOAiiT536FnjDwvgU2rzEoDiFnrLuX88trV5D4Na9O
```

```
S7cN75baGwSgmvIYd252q+R2BFfBewz74NnVMbPaadwB7zh6/qw7+qmN+Mzb/oayEsxuz/ZGExLO+whavX/v47
kj7qFypcCdO9TGZBCwP7d0KqrtHwp601y6pY/XAo0y7xUrxcRbm3EuVmASuO/mEJadtK9W2g5McbA+f3mjsRHu
XbkgWKnApCcxXhvu/3OZoNneM3+e9aXqqzxV3AIzZJf8EHAKsepcVHFc1gHLgvVTfOOaTOXHIk9S9FkyaBt6xO
yIsoHCHdyrh+KpvdT6/0pSpV8PGh7GUu/L5O3T8e//msyMVKFYvpz4phEmS8CRUdQCZrizyktNLhP1A/octiRl
6RoJH9UKMC71aCLu8BwR3uov6IaXKzJrlhd1K11WEl/FDZiSedOtTw8V6CLnFeKqiv7cGWd5ctkvkwV5K47K1w
3JtwGVYjRuplqpzWgtWH8oWHvnuj0LeVSblgh6vCJU39f9jlD74KiLIBCaYI2SWYamKHmi1eAVLRJssTq2Yhd+
0yB321WJlZN52nKSn2HfZWO+HAgXsOFnGouo9FRvRqkrKcGreUrrKKBjDB1aER0bggXJE4EF3uL5QTN6fjF7yL
enadKq5Lew+z0ubVraWY0yhjkbHR6sUjusHCb1DoWIWR3jITwlumr/lAG4ltf7euPy1shbQhDdtHkdRmAsSsD6
o0ORMMh4I9QhSd6PH/T0m6XqpRMzbw5UXm6ltEmbG8hrZPv6HP+/NKORv77tBAELGP33f4H3dvmLYM0mDMPqGS
rvYa5EuKV1Z/JGQ6InmPHo9D6p4xRX2xykxiy6PwriMsg0u1pm4iUlIC+Bt9HkvJE4o4nkAcnim1RpDBTGSLGr
lmOVLtp+8Cz9PyAVK3AlE9t44AQGneLFpzjBF0hzxkImIqYRgpU7aeB4RWp/cxyybTWscltGE3G2HzPhCcDmEa
rrQmoyC3wElFyKd3KYy3bxYKEAehvJU2VQ/TnHvJG9pNh8uNm4Y68vcSbR2Ga+n0NjFGVRC44wMzPtzyJO9+dR
9aDRm4H7XMoTScqBGfCmlpvZlc+TuVYKNBkhHKnzFSsLzkoFzfseJE/ajqiUoewp7WoO7vR1kLiN6YVvMwW5Ig
NtukrlItx33VIc+x8g8krQ1KXLfi8ng6WRp9w666Adlo7W09h64pqIIcxzWQsjAxIyME8ECV3bmtdRPxjpdYkE
cvJpaO3xW3l9sVM3M4zRafrI5u6oFnVw5kZ951SKy+ut1t42+fJ79LLjvCARwccUArcom04JCwTvG7d3R8zNr8
PJhYIH9Z1a1jFnGYhb4YSc9be371yfYGrYVfzJ5vd2QU1Ul52iD/deXwdCIIfuJX37KUS43WWpd4Q+IuYNlMJZ
u11S/iAQrv3KUTUOtbKYuDt6P1nK3u5DL+U3ACdQ/c+s+9WA5oKDfyK2vRVIVb60afrXCPWDV9MwCiTo80T2cT
DUS9R+mWnHXF016zC9rqYJGFdEFh/FpHOgYARhDaVqEAOgDWDBO996JDN96Moh8ujNqkaiwmpQ5znrPlDnBID2
T4oglkigFJcRXb2vDa9fmRwYyoPH1y9uqdpDZv/w9KoMb2QveZoRJbksBnWV+JKnQVjEJxZlJV6tDe8fUG+G9M
FnB0EIMy8VpgIJTpIzQpLOEJV5bV/OSR4wYLQ04ULhlzba8Q+oAlmCBs0r/HUiD8r0a0kGX40C+3MmjbDle+T6
J1z2p5mAQdcykrauMzsrDOzaeuXOhV9Z1tbdJfEkVpLWYamtTPBt6RID/+JTn1PRfwWR5ab8Qht2PqzNoHmykP
lX9BbOx8LrqJ95j2wDXA+3j/IyeSrrGDi+dFGtpHhGKZXfcLvuuVaPXI9tg0oRtByYNNNhJpoeXkzrxTcNNcrx
mcK+OsQfqMg35IUuWfoSkJyUFoAJIOqdK2mAC+iPCjVdoxKQRikXUpVGdGZZRo/qd/PMWrFKaQEdNA+h8000In
GqRo5889+87sWy0R3aRTxKR21rIpL79ILEqrTbMoP0vv/NzVkOrEVKqqETjVTwgXnlfqUb3spzye3YqLCBH2Vn
nQTiH7uKW3428A2DSp0pCYFYCp4/3fCYB4BMOoVNZDQYcJcygjRjzoo4hTDNidzaZrnHDLknPhnSK8vuLPOUhz
wnEFhx3vwJQCiA2rzA880SIAZHWwuBChicR7Y/tE4nq0za50LQqmji0p7EUE/+oAEIeGv2YIR2bf0t8scVl5u2
Q6I/Me0xcbCIH0oxgEg86Hchohx88HiBTfsj5M2rK+AYSTVUQe5PYPZqBsqUKUQzx1i3/gjMGkjqv4k5JmHqQe
Z0Pp7iD7/KNQV1tP9tWs/mxXZefqqzKpbkfgvQ+k172ygBX51eJnQH+yrWho4PNi1eTxJGqvoqzePZzoj+vhad
tO8OWfGEG5g4ai97pRyoqbujQNAsS/mkRWGrZ0Pf9NSbs1/TG7DQfCfNytF6vxk/lLZaakYNYOs+fwJpHYkhfg
xdq2eSWLIxPoa1VHAtwn7fg8EJ8y70vjKkGy3TixIJz77sQqARXK9BdrrhZfJ7iSBm6yh5o+sLd+4FB1EqRhm4
x11T5e5qMbtU+0WRetpRTDIIhUoTP9IlsUSAmhXfVuX7cnu2E2HHReTJa4Jqz1A9Z5G88ZMpVy0OFOB8I+QzBf
fGGKbYkYPV2xXcEOm3vDgjIaea72l0bXxOKxRZea0GM5OxJKuHGH4brWZ/kE1zTJLiT0BUTb8MIkpCHxNu3V4P
BkXZ+xyw8COQWNn+npFX2t3VifEmxNWn5pS3jlSPlPOtbaiNSSxVHyQQhKAI2QxCwQcj0WzsIGjPy2dXtj6koU
hrMXBKfT3ffzVCfxXW+INIQImevuVZoq55n2X6sPH/z8UIb6UiJEWnnwayaKFslrpsGdWyaG8lajyQ602L+E7c
oMIr4MyPfmAjHMPKl3i3aw74YlEV+m9cQINVM8dRSpKBNqsTn6VYH/HbHkG3iIBnihT320wlHrurWHfeLj1e6H
jR9MKvSd7QULzstAhWTSekxc/4zjXgFeSmO8UcfSDucwP3vNIeYjELkrORl7A7k0k5md+e5zPX1kZMuwrWx0jk
Rqfw6CqSi5i7ftaDHHd1aX1PdVtS4F5ntP4Pz8SqhFeczTIZXnBLsgyBP8MXOxnomb0B5HWFZeEAI6PiMq9uCd
giYXPx1HsUWgIFZY8jZ2Ic8j5n3eJCT1gSDzu7VNs+9I4aTw3gBbgML+j6s30767JWowxPBfJMeV5YzzvAlrPo
hgu+AsyxlAtynl8tKuzeTUO/ACTXATOus09fy6jEcf3r9mOL0SXvAiCxTT76CaUHk0nkGxx8Xw3IfwCmsXn9Fv
psS6SlBGrfXA550ory+x57+wNqYi/AStwsfwYh21yVsVWUgP2ZyufzUtTiPCjG3oU70pKmoVP3c6MQgtEK3nJm
9TSTP9pyjQ8IeJEOm95jtg5yZflLbzNhuJN221rC9k2pnI+MzMquRlwV6J0T+6gtX+WtT/KWGM8Z2Wv+80Fqd0
XKs6aGG7GnevGsau36CQTmo9rDqYDGbIaeOLcTol+V+mjxx1JEQBHDf9V9aIIloeJ74qobSVkbZGBvhFSOtnF0
txGXckCcwKnxWkzddDbQUUR58O95erHaQpeF1MbUmwxjg8t/7o8JVSG47+vXD8Cp7+0yGOuZO300zd+AOeuWeX
+SF6k/2M8ru/yyaecAMjOxDUTEcvyiM4RQerL9C3OflcZB/3DpjOoVAlF9NG4LJR7lK75s4fqmG8PywTz7PZB5
VQ5ExekaOvPNcsovjLvBGbOKeG5FURoE3ssI+1m+8RCxfG/lLZX53WSJjRYXLJkcrswbXNkbnBCOtW9IquqRhl
I7BsVhrsldwgbGHxvAQ/o1gzhWSgku7BTFxKlG2gi8TIru1iA5HyG1B8naBmkYkkBWkKTWx3G9AGDtZjK8jPi1
Y+uEoIF9DP1gbF9yJ9W3TkdtptG/ghPoy/4DexlrkHfGuaYx/hyoJBOP2FTKlZc4Dafl1Tmd4B3MJkZJKIeXAV
KAbfE3+StfO5ftKCZfnmItW1Z8gXJ4bP3B0HOicBolB2Cgj7+4J2yeIkTK4kHsj4jNCftKRmUDbfioHcfaEuXJ
GpRz0tOya+YKXWq5W2jnpykvS+7XpTxyRATbfDlb/O/y0gGRwBcYDQqnR3TAK4ycCLEJ9zijuxcOW+9itC9Wai
JZoUt4WFCp0f/qkwyrMoAFRtmCx8haJiQHuxLQOZIP+t/oJpDlUTxxd3UfYxjln3HWWXGKwEwqkeZEXNepMKL/
y4/kLWYBWwHFfJOwR6+4qgYpw+IsKNwpwC9wlrtfck6slMVYpDssUOhw0hLQGSWXjwu2WwDdx6hOm/KvKRIPT3
```

```
l8reEgZWzG850WYLtfD6RFCAec+UxI9O78TZyLXbEmmOMxlKTOPHRXWsYfQlN88PA2i4AJ3ECMbnIQpN9fUFQ5
On//ewj/Y/BumnVO2wpieFZHDP0Pd+w8TiTBjXvXYcXXooVlg+aimqxDdz/bJHdeDF9Y/SM88LjOMC+eTLaHor
Mh3CsE8G61Fm55NANfifc6SXYb1x1mTfp55dm74nSYAZDs/FZl3RQkiuWRHn20gjQJwDS1EMAlQGWidI20PkjZ
btEN2TmGX+4NNTJnxZ43pQshpxatHvRib1Mn/9ecso0D8YTgz16mVwu7WBx40ORtK0XuIJIMLrGsm4YdkGQjtv
4Sw5lw0hC1aSVrIvdT69l5GMzb7dzSuhtyEU9Si1/W9Gw9Kf8g5NGzFiN8vidJXc8fnw1oJrpMVVbRx0pDbE3P
aMo6VDuHKKXi1JuWVAqlokCqk1PAy1QITdsNh3B++agXc6DFDAtN6BthXnN9IjI5y5pmDXmWpPAlPV3mnG1FKn
BXY7maWyYr+cxRrvqXSIFIfZK86SpVwP7KaKU0iD44g8B2tISIjvoZPoKpKFON/8lECWqwJOKFnrsI9sGQ7sOG
Atqe5dUsD+ja8vEa8EaX1PunulWAHg+1IFUtCIZYIeXd4aDYbQMk1YyCBDQrZDLwonzC4DA7vYHsatOq5tHSEf
8YpPA6s7PcX4GQNrXy4ZHVhpYbSTAzo8aG1HTHxCwDm5QYTH8KxQZmJkNVZCpXcrkpcmbIDpv9x73C1h4LQrM2
HB86Z6WhrE59Qxa21A74CM9EV41oeDzUjaVhFi2u5+imOjkdT7L5KGhsrBTH3OdhE8njt+8Bw32/Infq0ITKsF
wFgP0Rh/OSqrea+GGDioqqQhLsd5mCbOU7pIBlbIVU7Pufs7IOQpWP5VyF+k0cX42rGgSDOoPW1k+lDU3mdhl9
VYajtcW0oCPU0LoxytPOHL6K/m1X5NZBqXW7iK46+XebiEYp5IeRXxrdT+dPVEcsNJt+44KW+YYypifQHIQ/zE
hDi3Z+2ty9Yd16BEkYW3cDCmMLMLKMXkygL62mkbdSVHb5bFrgz7QKRUQRAjjtaayJcSfld164V1KAxteNPD/v
z3cIk1LMmLrGBHha+e0eX5GdBeqjp1EFOEpZ6XPVWjwDPaUWZ8tjG3m1Ocij5sGxbOKnX4w21MjtGLypbKWvMM
av9OxAi11pbVaWKGf0WnCfpnoNrGW3JMG1YpmYkuw/5cDsjyFoNRa2lZ2ksz2jrk1aBcyLZFwZjsjEU5sIorbL
aI2HuJQ1Kng1DrDG6HPDVkekHJdYn0RRGunu0usYGrh1q8o14D0rpXM3ee+goUeJxRv0TEwohircp4SVA3txZl
M8RiYYklMt6YddcZhL3VNipguc1VJWz51KFyyYw1bms5jnXfccZ7LZN0E5+4hJpLHPk+Uq4tdOfX8j6duuNn4s
BFzZPIdQUoN5klKnw4nozeCmcrg3qkv2RjbSnSPPSBpnm8SJWJ11RCacaP9Zzs3WFfTZ7963DP6Hwjkoq3pkvj
f/zv8fU7edoSD8sNXZnq9XFho+w7gu4asinukTzMLaEuuJo4kR6o6cwlyVvZR9mJjOCVV5S6g0OlnOUY9osANh
vHdAqEnfFtc/I//Y1ZwZ4P2wEmREl2G3nVxf6GaWuGibmhtnsixCUbWTP+n4kL9pQoMmxP5RoBuiZ//HwJsbZx
A3BXcOIzIpzhcsD2BvNVLx3ZOmAXV/8iS2tOISmjmJXWQogU3KNrj8rcmyFmf+oqT7SezcuPSeSfgYprsIe8rv
DQ3M2kEodSSb0LQvcPbaufSu6QzGIHRBIcQeo2ZwVZXYM1IwnUZ5G8AzB/GvO7+WYEJUzQrVKsOQilL6yknl3f
+5EWs81qEQcXWXE0rTmnyQ/uHMNoS+4N55XwrBaRirOcvaCPQklG9o9phkrPn2kOEg9U5009sKAJAeWSC9y9am
h4Wb4OJ/22BQDkVA5nDxF/Sgoi0Q8e8SzC+Q+5zeM5VKm6qfqFOKbKZOUfDqdA32Hqr2x8gJoUfh4n4o1462NE
BQ5itLJ0i6+5gIBhHiz1dNY76c3Et6O9EP8i9Euu2i66ELbw3oUlniAuRx6DHr0JLIjWUQMW7+nomArQje8/X/
nTplD9+k3NJwz02EYnoY6jqwskb3RgN/3obeu0finPHS9/StBPcC3EV3vNpfR3O6TQ3AniEooIGes/dtrkLfAu
7NiWLEz2UmfnxZGkZ7TCrx2Mx7X0wMx3uRB4EbjQKZhpcJx9nCz9ltFjQkEJ56fWD8/1g4JWZHBw1AjROT6VKi
woLH2sRk76iSRI4RrOm0KXOyDSwCEpvmszRNgeaTwPqMGTLhfu1kY36z4dJF/AhCo7B6tgadeotOp/8CcbVe6+
JH9/ClwiJDfGmrf/xUhYFFfG4hyGjDIXlngol9aABiqCgcPS5FStqpT2kr1qrVJouPXEmXE9I9gkmqqEcjbkRo
hZLnmQCZPanu1YDubjeBqModqNcFx8egbhS70yyntqAw31bnHElFItBr/sAS7QznF2yhmd7PVOpd8r3nIToDC1
+fmczjxZtGc+ATQ5bq7njhG9p6PL9gIMJulw+EwgvoUQzxIgtxCLjJodEYteQMEHXbCYZy7bgVxW+6ZmDt/4mm
DxsDwWD+KZiOyNaaW4+1ro0AQuq3o35cY436QifJpVu3SN4miyHVftdGrXFMpxmDiAW6uLG+0Pi0JaIYLCKL+b
Uas5ARCGb3rg8ftTG2ddcJWE7mfsTml/AZerz5KY1lqXKB5v5ARINIkiXAMiw/G9Cxf3rie1FwJQjHuLpFr991
zNwIpP4NCPKVW4jCZ5TcRr6AO0AMA5lxrLboLlq6W/xpo0lq8bGEPOzNt5bkDZDy1fT4CGkr+TpWEb1x6M2JDJ
EjvSLbaIaduLZDeRw+/lTOs9lHRCDwh2a0a/7G/wqL+7JYiTKF822JtdUYVI9kz/2ya4cPMDoCat1A6eMlzo3T
wF2wfMIMr5F3RsIj+CrEZPyzAmq9EsiE3uAMk1gXTMa0LidhIF/Sb6qYagpKudYjgeOhX+8Qhp47d1g5unUxOB
/un2F+nUKRwpiY4CiZ/hZD+IYule8SzvY/OwNRSqHQxOKYVXQpFs/wXObR33jfSMMa9O+XDIEef3WPejWqL+pP
gE4dzbS9FmHE230DUBzHTU2+iJSEyqNUxrDGBMfl7UYoIY4ZcM+NqK7a9Q/kMoRUFlfQ8Ej79+7XJdhhJixkUf
4mExbSprGJ/OuYzdnDs5aMShSc/rvKJXP2FXrRJzSVePTw/XKbLS5l947ZoSqSyY6Fg+EbPQpHGKclsUX0dDcF
SR9fnbOmF1v8ozeeIqxhqSa3zHUWQER3t8JCI3HlQHY5Ae09bT/AL0yhBcVaVAh6O3uKRpyR7TEiD9FtrKYSsN
c6spqwI5t9KmlmGVl+185vG0bB50fYQNEz5TOzt6A3vfcAzPtmn5ZkTaeLHI2sdINm5y1arec2a5R8vfQPSFSE
itjrjrEjyDLKacepCh8ukaPuhEI9+Rtmilin4LGkb0PYwDMore1AcrIzWSAakdg2j0uHNoihD04EXUXIAgGUdI
3DxW0vNGfWN1Z+pFc4b4GDrwJ+lHkgIJodDqquaJFH8MkyY/K4WTmUztc+vkMWWybn4tHiAnW457iD/CX5LU1D
VQI/nn4MKlhMeHFiBgR+IfwOqD1b0t77s7t0AX/AOJy3qCpiWKly9+D4GvW41JMPaH38ha7ZrKrWtz0Ym5TA1s
69OmzkD/LlXvUgGKtdhFA1BRqxLKs3Uz3XR6KfcuQjtRSBDf4/xUVH2+q2lz2aAjbdOpgfl+rsvDaT7lDR/Ez9
OikYkPYq0seC0Rr4cZtDxyeslCjvjY/M3g7dMOTd9utdIIF4b8yWBpy6hG46w5JaZhYNl+yJ7cXiKYb55JdPI6
IFAJeR+h14jnEtEBlYAkJHQ2iUymA1lTsQzjafWbB09Mb4TzANElKkMIcCN7D2/I2Y14OG5YxeBrPV5z3h0we4
/wzl9JoURPAf2XEQdpay3xiVKz3mOsMemHWnWkpOxC1svLXvbmoa2FYOFlGt8YnCQip8cjnprE0R/Z1ML0RczR
2/ZtXUb25cGLrnkg4YZueJNOKGyW5md8YVL8fYRruN/TctxScdr7kbT6MA1Sg4ZZCAEN0ERv8iilQwfHPE7OkR
UwteT7HNNXsxhmuin+Y41Lp7h57mVpIPabrQUfMUusoifZGOOIhNUQbYzjWsowApOnJ1hAvji2ent2+6Z91wGY
rZPGlMpukO6KNibSa+q92nwaJaqtibOgBcTyDevbYhIkZM9Jh/aCtsaxjGctbYJsJKWlqd9dhpjcxdcGZ1ZLQp
```

```
kASvuZCcXpNswp2vWaZpLVlIx69hiyH99mFW4BQAH56tKLGizhrP2objyUWEg7LzuBhU78pEHufFerpC06WePc
7yWJmgF/BWmMNrTsaqXBAdqgHdA5/uq2ZC86yy0SOgfPXOsf3q13vSS5AyUrAxuAtQeZLQMs9od8OXIXWXa6HC
SlEBTYry45Nuc1A8MZve6A0RBVkIJAlRASfOGPyiTXs6BXiGIEHMyKflB1pThGDKyDCQ3Bw5rqWejwA0SqhJeU
zqSK89uUI9FSwp1ShDAEQ2Wrunk50NPuWVERXzW6bmui/iHUVcJs6KMmll/EXfyr3elej6+jRzEAoHDPcJaenF
cZEDFb4x5DNtesi1UI4twUuLfeYOIeUWlpoMT4nAfpVGXeWd5GXgMXGGtqH0tGNWG83obHRa6ju+ySlkxg8YNd
N6GTBDXsDWwRxf0VVY+N1E8+bxFj1UUjEXXeaFudQ0jNnt+7F0tEMTSmJSsVkS/IBamieaNUfZMyuxZdnyk307
MtQLEns5m9VixNnLHbcXRrsaKuJYRHO3R5nbF/gFNnMTxl1HD+ywh5jaoBGoJsXeGGOU2+0citosig8kVtCr6U
kfeDb8zuk4/lirb5s8he635JloKSMqM2qIlQIdib27iqeMybCsh+5UuJG8tE51KOJU2IGM0NAZ7UPYeenrNlBY
OnvnTPgbxWcuLCNxAhaSFFJ/BTSuGcovkL1ow15J4qtfTG3rkNF/TPFhYHxN3m2qJcBB10yMV9el15OtH5Cz4S
2w3837bPjUfPw7YpYsmhUIHqDDVCtyT1zvJyrzoYvvK2Fl0W5JxZitqDCweANHIm6k1wrWBLXg0SYde+SqcMN4
lQ4er3Jd8FXH+5kYqYTvX2wRUNeUGmaLbOLoNW/rRWxvLrDbKP2aFB1LLRSRSLHcQX7Ccmvm5zMlJPgZiDqbtC
ylhwmDEH6MZ3CEag6C+aXiDdgo4OTYFRQ4R5h2pBBTjxYz0qY/JPYAkfDiJhf9UNNbExbPZkhDVli6xrfdDTaN
4dpx9tnirlW92vds9PiB4ciPfUQMH8GMr0LvVWzr20qYQdtWQSO57iTMfguTQYx59iL8EIDYed1TVxHvXsl9Ov
CQuKmP30kEMbdLShjoTB7ivk645K3GbNSKRGKhZ4SIc7JNjM/hTFsBe29VXyZQchQnms9J3+UG1TN6HROEr1/Q
82forLgePuhJXL9oriMkeqC4J/wBojxx8sYXA3YFkwqylndpv9GcfwR/L8vWqV8geMrJ4f+DtfgP3DNnSkPcmq
9r4eYEMftNb2/q9P6gbpdLPUYqAUqUklPgayCHF4ZMQfD5wLfMhBp/hMxTMtPpWWY/nQ88oIniIZMjrS8bTmsh
g2h7a3RIHTwLccIKVV/lk5Ejd1r2bCzE31dO2QPcZt40W3dAWDJRagcZ6C2GOyoADpZ/zYdUFEQQ5WpDSNvE2Z
0SdHifWV4Hp2Y/Osy2G9457kHVxuK4k38zbZN4kdDeD/3QeGIMAh/5xfXNNFMYh4elj+QtDiJw9buo/Z/LINhX
VaSWT47Tl/VQrjDsWjImHcsSfhgRqSt06GG7ltcO01BeM4+Ti8Addhn6LxQ0S2u63BShRxgDwQ58J5ImPVY3gX
UHHS9irSMmenbIPvV9jeAfHfzEYeCF+zv5mMd0XEHyAtNporMk7FkDcg9i1iFEf96cvsdLkMcK+Nmv9B9XXH2Y
XPeho2/h94R4RDVsHmaocsOCl/GG/kGMuUsysiYqKIvFwa1Zq9GdI6AtgW9iHb7N7LLwgK26fM24Mvt4Xsw3DO
X2CrzV2cU524OiNtKwgIGW+7tpUIc/V7G2ihifPGvpQRY216Dy3z6gF+cMnmloppvU5KDyn2d4a3Hg5bTj7BBf
GQU7nixR7EkTzeI4O6LpaiVt/zMSKjZGgbnyiuCSnAAuOzGQ/0smAIFlkm1JGbuL7N0sdqiq7mfUCbGQ1O1z2o
ghTF+Suf/KZjXTs+7TlQfqlgnVzJTY4KhyYDzAStFqE/SfaRo6WsWCuDUVyPDwf/u4VQOVYWq4c2eM8s9KaZL+
tDYE3H4x7rQ4SgvcVEtugp168a6nxZNynSKycz+rxOrhILt8NJvYn1v5cZwML4b+66f4hNKDgFpEn4dUkcZ+PL
7wTdWJhVV3iuzXeirPI5KhysGdeRkYGFCB8gmXhd30pO9fosaeoYJhhMujm34oCpLsoPLk2WhtbRr7HBPClsZc
aYzWFxbMiBsPDVI06V8deM4xnUVIL1GjinInOZJ377hTfb5tFEHFqr6HIubxazAVfqIOqZuoyNAAFTNuyw23fo
O3+DfAWhD1jnB1DERU0AvPSQX8ERclLztHDw+rqFx8EbJ284G+veCbGPSkyIMutSvbcHTZP0flBrGPjMeX+i8g
gn8LtrbHQiVr9nQKrjWk8LQDvVTicdlgDNpuK1Rj/WNfDXy2a7bfa7/Ht6m1YZN8cOSI4nWE8iVdWExNYKn5MV
DUtlNggoMludmsmOJICSuU0dALGDn1Otm9VYUSJJ2QFcnUD1/bI4PCWRuFpPDlXlTOIBeiW+Bvgscw4DBa96GB
fLEdlL9p2+LszZPkamP2I/Q3exc9NcPMndbRcphtTLeZlIz1NAwXeClWuMWvaFrKbf6rgoJ9w7mCllQ/N9Kt45
Khh0CUzCJDXYq25jks9FQE1tAH0DT8nX4bHhOFAW1xgDXc8Fd9PwZKjGT1/twW1s7SixrV502noL7AL2/jWbbO
WRqZqI3wiDcnF99OPVI0as7nvh/wd+xUqr1FWRbpR38XRuqc5rzUtXxFtzJQenQ2qfK0Rs585AY1linpE4w505
djnYTVxkHLmwPOFFRKMMBN/WCY4br2vPxJhtYj7iykl2A+CC8gH2CzzFM+x1As6gymgjswhSIzPbilCJWKFTYo
3It85YpH6npmMVGEr5gAxQ+14bAJTHQFu9TLrxfVSd+1YFGAs+lD7KH3qQxf7dfeqZUSFv6u093O/ZT0f40Bog
2ms3LiZGiu82MqJWW9TRzlqKARzW54T+9zeyj84vb5o66LTt6yACfUeltZKHSSX/fnQLQGSFEgpFgPIzOwU35C
a1UCSVZ6CRPm8iSSzBNqlt/p6DXrHdSkt3JgSUN4uO98UIjEgwjlwNUY9MgPJZIen0f1zmvpaHuVGwDGEKriPs
H8EVOfAj84RigbGrqtkqT0QZejl4B2CWjSb1A2t4aLljwkJyZBsKH/qvpQExOqMhdZbmia6xHMLNI8/B55cZY9
6jwtFtLsTyDSrTGg3qC+cAkcL7KfBdFTrJUiajohC0u1zkjdnEtzqfOYngCNBmMCQTSXG7aIaN5rbqnjgsSC03
TGUjgHT+nq2s2AmNZxD7JtNM510Bvhrlw/ALmB7I2W2yKxr+z6WotbarDy2OaSWLrTn93nungo4LadjOrk+s9a
XMdOojgkZ94g548OB2YrSAIkgwXNEhG3mV/gRktT1T4BQLjsicwL/z1EKEU7eTOgBCc0NN9CgDDEpeZpxusgUX
mh3YNR4vppgMJ9NmPz0tlIWkPpi7YoqMLOGEQPXzHUPWrShafE6JJL5gfki5d+updFYg14qxpY5v6v78anLf6g
tRQCLCvqtCsPvQ6m+6Nyubx0RxSLPHxjgi5+qH91tfWqu09S+zJI5FItlizMSG9tCcKrqOkjUDDiTf0vEv1o2v
EC+7C0v5ScKLmMuAMBUrDI72izoxmS8msrWEYZrmWWtxzSCeWI4gnaY/TtsjfCVj6ecMRiK6iDYKhbhcWfsMvu
+M1QiLkVLjA4uvJkYZ8BMFljaHCukeQTGFMREIbEm20USO26VJ2BrODwvsxlp2RTDDgJboummRa1Xj8w/c7O9K
MhUvV5KlrWlqsGRpl/QBLqXbdy+CmQS5z+wnPPWEFvudUJn7uZRQO+upGk3/gXBzcBZufmMOJl5qazaKfpZAli
mddRd9zj7B5CpS+VXWHUI+MWVaOYujI4D51Fa9wbLzHlol+6cmaGzo3U23guLn5lDiiiJr4m2B6jWdtCr19aQd
4Ud8DRgWyYZQLgvTgtUZVggR7gqjzUeii3FpjAh1dUJ3HTsz9Eli0bSgxFZopGZJImXhio3uTGnHhCvrprERke
Gz/QecNLx5SvNppXQ61JhwQlu0hbqDk+l40Dki6Sb6wBvXluPxi8nKjH6NFru9c2Btl1QQKfNpAhoC/ONNfyUL
ODJfFJNYndliXGeRNHJtPTPrfHE6R6uWtVETACURbtkpE9fVYpUTOCvk7+ybe2fgkI0yWwA/dBDtm2+EBczLm6
DLGLJYBcB5loOoKENhgLKOgKZ3/oBmaaUSb22jWiC6ov60iUHlgQbSe4+GKDHWtSk7luHJLPEpAyvrqYfZe4gG
```

```
IXRqv1ao32+cc4NMXmd91kZ919CsMCgstjK6E7QNnoHEugMJbRO4M+zKCUuO42TtFyL2CzJR9wne7sXj8/nNJ/
2PmeeFQ8QLBACdfgCQq/ItJPJuLh4YIGf6ASxLwBwZGhF4fNd/GhjQEOhtuNvJlD8HETIfUsuXZHBZ0zSn74Bb
kPjVQapugJ78eL3dTBxKx0/7PoqhnlZxB9ymv2fC+Cvnk6hhnR4EVkPN1+t5HPSaQmv0D4+JKX5XLHZfGDQM1z
/h7dqBy/AvHhPcxFmJ93fVP96LZRWVnQYDnHKTLR+uHz9DCsdbvAP4mm/g9l+derjho+HGUseR3wlPRhOlyjzh
346FNs+YHBNkp8GIckpu+8XKZrReXCh0ZkaknG1dA3n9GpdPvL+gT9SuHppyo4mwCdO5aDIPL3I73EfM4pl19p
uaC7JTfk1a3EGIVSPxboBpoH1Mng8fWlx3y0rCunvVarYqGTMBMZLtNXKI4xLWILGdxKWAo9kWOaPX/c7gQNzA
sZ8NYTqJ+Dp/rGHqVTKqKAkjSK1HCVsMl17y27stjwtqFWQgFn0IALHWrGMj6qSpFRtbtnVu+TKbPlumCf40at
fJPWGXU4K6UqJHpLkHe2ooN6K8T1ndWLxNCjfmc+ApI1rGbxPwBoZG8R7rJ0rcgcxCEqNAMhs4BHval3WDoJ0p
qUrNCu76WwAs1rMhebXBvmKEZ/9QCaVy5lwRFXv9z2QCMmV6Wf295lDUKJOSwB4ItvpHfaJrHvqeVbXaGZr07Q
P5S2mIZ1sXAubUvTTygL5v3rmViyHzssIgpMFRzyYsU6dnNM/g3axWTb2X6BUOqFMHMCQ1xmxziSLubBsAVOgQ
eUKzOS9vgufTqdeNXJyvvUziHlSsvHYBiN+Dz9r/7WGBKTesnX8zLdKsI7yxC6p3nzFBZQUtmZGTSq68po1EyV
GL1wDLN9yMN4Ye6n5zT8h4wQQCXWzHPvsRnDHLLDGfkGQ9Ki8VnqZKSMCMpOFCrOivxGU96nNeCs+OTHlGfFdB
lMJre/yH31di+uMrI6nH0D8EKjSJlxpoyRzDEaIt0AtVZFmV6sRZoGAU8OyvlBtzfRCSwdBqbBav6miQISaFTs
msJhXJwbDlNVPlpPq/4NyguY85FKbwAW8k0H35CCn93Rl0kNV6muVxDWATXn/WwjUqxHHJ2C20sP++5ZxEdvAu
wyv0BR2vGQC8VEqmR/MdAEvKNcdU3U5fUZaLpRJqBpm/gHlhio0Gp1MTWE4GwM9NEJz1hySlNiNTqaLBuVC+YD
2hhutzHc1oZ3fWvP+4V2d+V2xpWdmsY04N9o/N6KvGbHh9+1RfdR8mu7LxAmvaLLlPSS+xCJUWE8420RMcKry0
jD1zdosPg2RkZzG5RfUwCO+rs53XndphnLsXJTR8lz0gfpAprHL8WOYMUKKfihi18VMp0p3TdKrMNgFUJPTKgY
zxbxbv3UD2qWrdjbyfwBVW5BxnLuV55pYP1t2zdy1IUj48xQ6ElHb+z5R8aPVAeLKwuGBuOJh0KcqpFn5mWzAH
Qa4eW3inmkzKq8xeygAVzwBxXwHUJXBHTMTy7qgcyTgo1Uf47lj2BbUL+LK7LIvAAu5M1sYyWm/eTHcL/XzKTa
HsvNqaZfWCsLsoO3DgxSeroiRtepMWfvFYRmtxyahd/XCIz/81ckbiz/UFMd+mLIopJ3ZB3lcs+yX4prjpsrj1
/cUt9BDlcEjTnURiBu/QUXR5QUuOykLLJRvN2wVJM6yl0QhZnqf/bLqp/TmbpXtc9a9Xf8ALFHg41mow198cZE
kAmvYUyNUy31zBcNSGtNuHdEmrDAcUXLhIRO9fORjN5f2fL6KU/bCivJ4+PLQRmArgkUP7RKWYt4sS1TrnYXwA
XcgCGxwTclg19iiIQDvY+t9M1CVURc9hwj1VBhvDmdmoevcWDr6FbH+GZ3/UYPnX55dDGeGndKE1ZBSD5w7hNZ
coqRgxHtux3TzNNuBcT+SowPCVqMf02YjDwKugYxk2hQM1RkZI6PMn8171PTgEgPEx47vysbSiSWrVmU8kqAPF
0vM106ONaq4aCYH0tkNz4odduLQn4DYDvVuIKZrVtD1v/Vgb/DMGEO3l3rJLHUEXPbxdpVzH8XBpKy6VAx2QZQ
SwF1uT17R4qgC93a2aCc9gQMT35fSfuMQHhXji5mqI9lJBtIRCCi2qdGWTAGRqvwCbFmFvTzDTOu3OtpuLhI9y
R0Qugmu093D49UynjRWOchUJ5mY2c3DFF7fU9nix6C2adqDjy6cNuX44Lichgx5y5PpMJnQM5L8wTMTkoEz4TJ
r7fZNcumJatOMlaGXw1us428MiH4nBHsC6nojh7EMEyIjv1h8v2XHBtxCS04cLZzPbGT2CNM3jtRrn9Pgb1QXu
HgFPEh2J59fPbbaY4leGfijmVcZ6bxSIPUxncv/Z8CfxQGeoDnKRyj+EGvqmzF/cTleLo7t0NIvnj4yKI489Bv
JwXQI5vqNjmxnZb7ZzTdiUMgKxj7Njs2Nix7aRj27Zt27bdsZOObdu23dH867zCuZs19QB1WVXfTSm+SrjFsD3
s7W9bHbp+4DvTdJh+Ohtvo7uy7V77dqydcWRICcOLLz1pbvquYECya2TJbNawpzsh1TmQiLIFphjpl5dSg3j+M
aadIUabUkLWYitp1AU9q+CCFaEo5tWOVnTjjk5JN8nje2DMnzFMk5EM28yR+/wmiDVLS+r5VZWWtJURZXkMa/l
uhavnQgbmmDTbBZt52zbaW+yN8TDOvQNYBH7MkA7ZPw8IHpSRMZhHZohlLLt57K47JmcGd7e1Q0ZK7XK0SUP4t
LdKF/shXc/gK3HsKEdZQ2lgH6J/9EmnkfOUztsAZGOrVLlhn34/UuEyVSfOT67xoXreNWRU+B/1IYSYbwOIfLS
lsOp3YAcyxlznyACUCn26470wvjuvQV/AnXnVvD/JukjbExhvHtjwMIgoAlCoZUvlI5SWwUtKnYlnk5nrFOZmB
YXtwgRxRqhQTv8oi44cjxiu0tJNkQreumFVgtCA2WU61blamQ+dLf9m4EahKLY+Lw9yIBOzuKBG4UsktAjDxjn
F6tcrbmFpziqL0tfCgAqPQSh8e6KqTb78SC2+MbjEC61wBWXtaUsFl+Hxv6Ip/G8XZG89QkcLRt51bGeFuJV+l
R6/vpkQvDg3JHOakcO+MwOecUwSzlWEEDEb6UeNGxrPbyhRqnPua/zY0716emwynGocgoopzska04YhjBWTMud
gerP7oavOEKBtQbytBWuXLGacBmLaust63cgvrnRidkXGDZiWke4++AwMU8EGw88F32YIhsOMW/vqb64arKLFP
BvJUVWoL4W2Y5OZK4FNLvVLG2sqbqIONK2dOBF6qsQD9Hc4bNjoPZP8aanrb3VR/Dxf3cEg45VNA3TRC5s7EZw
pKRyTgnAR+/7CvRURz99PvaYfe+1Gx7jy0hTjdgIwOP8Rb17UfGy7i5JAsPX8Xn4F6HHP7XO/b9z1Qcy+JGgzn
jpo9JiJh0HzK3CX73zEJZz7oSZWtzn8ZADRahylM7gCMXL80enRJtcQbgUho1ywAbcnkzu8yTehOfBHKCKgD9W
qLVk4BqMUxfa9ZbtGzYAZdRkWwFjNQ1J9UP8Azbtx5Lz6B0MPlpo86FRNRZIH2m8O3y/4rq/ZcMA9f/LcIeYc8
v3oLNWS1U0AtdbyqbkzuWCZNMrATrZqEZ1pbxI7nshUnotMwzJDV+qi6UTAacnz1sCduIZzBNo8ISVPHqfhShi
QmOzdpy1qPQqcweHRbLZMm8MBazeVPSAgA1FWp0QVSojJK3/jlwrrgNAidwk/TQuF9B962mjTapFY2XBjk9os5
T3VtkOgEr+xbgN6AcihkhFQE7bgMF00BCODcM5V3wYGcOGlx45x+VJnYZW+657aNi72lgU8Ofkyp05ujf4t8OV
TGgR7Gyleb60iAzEwa6oDeh1TH7ICTZnmjcr+GFAbqSKU/WawgfrT7I4wiM38vt4POZXLr1B/3dCrXR8g9Wgzl
SvebUGUW6dcSQc9q/H62WkCPjjmppQASGubyFIIVfKhv1Ts16cmTDKISSvaY3Ea3QZ58Na9ZyhG8tfM3WkYOGU
l/KhXg75De1aebbv9RfP17SmLXcZNOv+tsplJYwpfMOziF+Krd1TZxIuUftGcuMDiBLsjM8mR15Yedh70ChhXq
1pJ4K3ynaJ7wEblrghovUM3LOySCd8p42S/ls5HMUW/Q8qwWhtzIWkNFajtk1HGi+iEyu2BeeMyrH0ErHUQr1K
SIcMcYv24Fa9mRCrt7GujI3fkWBPX/PIE0blY8uhylp+D0AZV9mnQYbKPzDiTUaItOJQnPiXHiqP0fbu2w6Jxo
```

```
JETCNPGHvI26kLQxK6GBT/+EpU2CarkCieMTsyswjNbeTw89ffdKeTIkC8voWGKvO6UfZeR3MFnnlfakcSorhW
jfP5VGHYSxu42/du6CI7vv9pG5roscn7gE3XG0vnnOM4u+m99e0SNPlXSkYhpNISxoTJVrIU1ZFTPQtJYX8aF1
7fEaWug1kXBy+K1R79Y0yAtWoHrtxdvsPDHfga14fz8jb8Aa+RP6u/cA5lcfR4uQQ20D+bAymN0Zk7RroYA75e
yoDkUEO1A2Ninhr2pwCrLJJ8k7+lqZSZGulaJtP0ODrm+D1dPSFuW97qeEsoikbwPiYfoGhN24vrrNI2yvHOlK
zEoNaVm+GVzaMSH/czzvMWCeSQ71aPrZ34f0jCbAZxBm4uRLsebE5nyn8w65XTSPTGfogw39GA7MlW9+prtHkR
NvTbQhjpDrQUIEMI/UMdHh+8nK5oE8lshlap9lOuy/XKmPbxWbeBw5Ht4H1hoU4cfyeT82BM0N1xwQ+0ymqkDJ
OFYyu3xTj0A/AxQ9nac3kKktdhAj7HtIHq94xBmhb2k4V9SP5kfOgZRzkcou6Zf1cmgilYYxlRIbEoX5sk4prS
rUX0qnYfMLKlfkcZjtkJCoSNdn+IeKvBkr4wgNz+F2DsRkCtmEvIjknAF72GkPdFE+ziynUiQDOvq20mW9c94z
HeQMjojaIlCTa+NDmwLTZnOM4WtVGoLVWGIeMB74XKEvELGrg0LC16F4LtF5nxW2KOW5T3oJjNrzSoSANG2Sks
CjbRinh1Wr6c6Xi0nFLy4EPTf4FQqQAXNASEXoC8FnAmwH4jDux4k1IwXxXk9p/3svY3NETp0ABVaVh2GZroBw
ZFbX8iJdaSBw7r7nExxf4WbTl6wTuk61T9y0Hz9VNV7RruIOPcj1COYZQ6Wn5vnTvQhsjemDcRe/36we6QkUX/
vjZ9FWtUFOFlCZ4cKlUKH9V9LEDrfKaDt9obpsLmHBbctceaOKnXi3PCsKGRO2Ue9kAbhRmIyB4huaDzHNN6RG
MxhTs6lHGngmcwnwnl83oUv3b4eQZRLb85tsiZPsxuqmJVsZ2zqGvFF8lVEx0IfMiParg7JGLs8EuHIk2B5k2u
ycR8CvDXChwNOKnY+XNYpo5eMVsfrofFteee/rjIPX3k8ss8n0jHXx3j+AVxoMw415+J3vvDpE4mAA59B5ELlw
y8B6LRS7Cke3uavfqLerXfuKchv6DgHYXG1n8k8bPtR8x7YoP8eFYpKB5trg25fw3CFYT9jeMxOzPkTlLhCmYz
xuQ1GLtm4yB4dlEzeVai0pHkdzed5o5T/hjeLGM7yte6Z3KlZC9QGwCy+IuO93FZg+8arayIrnHwiMC4y5NODO
m323M8jH7juj1LwvgV8cPFBwwK+bkzwMd96p8HSXQfbPTUU29rFfaJnIIPQnu7wbX93pKDkcrj9guJnvxZSVnm
F6dkTv3z5iSRghha20CuDk+gSB/n7923/lImTWw+2XBhVG/zE4ed7estYFMDS/BbRdUE9yKN0MDmszJmTUL2GO
/YgF+Oidro+lUCbc9x2W7c3eAqDZi7gOKhYj9lcYdgOuH8zbGzufzunuOjcEdnfZVgDH8m4bC5+l/Wo+HzKeae
2sWg/Y7oRy5ftFgQVoiV/Iu8fTV2gIl3rZN7G6L+F2hcltlIPMVDckMb096YNe8yKUAs14eGs2bYtUhtPPUrTI
eCnypGSFrWxYaldeoivL/kRp4TnaUgewJW5KJ4l6mWZGr1NPIsLjjnw1b2ZqpEj1DKros4AVUvKw4lFzlqWr0S
HcLgM91M5ZkTVBUEfEgulaEWWhIkDPsVTHUviVQXp4JxsilxUb1+Lip72Led6czBvP1qUMDMY5QbaSSrKjfwgI
F9jVBXySop1D4mU6MbPqcC9phk1mJv6jeuhwFIRiMtzHVUbXjKzNYcP5o4MJ2LU5ma0mt4R6dp/tEZkHHq3hph
F943wubIepD6FN/rDJ9PZ8QsuUPI0BxKyv6rOzAG1PEimTkBu2bDzHdeDwW3hUA0GyRMEibzyOxKS6ijkFbB7u
QB5a9X/OPB4JCka+Kj8NLbwhEcys1ZvTgshj69wyho7hPJ6PjOW5uKLfztkZJ1T1lchhMt84+SznaLZYYwMddV
W2kxzFG4SKwdtnEzyJab7+dQDr7CSsNyRDPqwHX9MbrH3nuUkUnjcl3YQqYNTgVal5+WWR/FdFcOcK9ku3XP6h
0h5/qFBgXJj0F2Nlu2R0M54sd/o3R8paFjbARuNiZCjFsh9l86qu3TauaHdsbapy8Yi071HhXP7aOh0qYqCJ1o
EXG5StxfClzJBVe8B34wyH1b3Rjph/NTOE9GVnDeq/sAn/BMwYrr707RlUKMpLzdxNwDwiupzTOrX09+mjrHlj
Q58v9R683eEJwtJUJp+/3s076leQ+TrA3Vqym0hwggd5yZDpTSMg787WPcvJHIMTEWb7QusG0DZ5vkmlsQXjq2
Dh1bSsPlUisaru99aaWwIMeq8oFRJWRV7XlXWtukxjsbkAysnJH81cOZMKP9FbtI+P+NCMOsdZ1qsSjUEQ+3NT
e1YPBWLNt0q/SqylC6FTG6LasbmoCCgkp6UOmrd6PTGBT2zrAriMQOxtWXqsUCtx09uoF0ve6PiV5dOww5ueax
BQsvqcOron7I0dWSg9WsgEBHqk+gZPo+WjQ3ma/Pxmu2vnRmjkK/yZjdtnlSijwy6XZkUkinScAnctXXDdfLZ+
uXB8Zu3CBRGHbYzcTdppQVI6RrqN0JeMNsHIQmRfBB07nMByI4eUcfljwzOX21j28UnkMFPs/f/IP8E3T974E4
BmYT1ApG+crVXV/ZwyED0OMmZxAnN/BTkoNxLc4E4aNrZ+IraqYJBnSSNIMwWNQCiza0FRCY5FW7gDFimyZvmi
NJkUMEkYv4j0mQX/PUR80jkwlYyEw44W08euMOPMNJBmWJoVeFxfEmwdW4xIjHr1hNSbn6FC8i81AT5+0geORN
YEK3ABwovX4LegneUY4FgMEH4wZX26p/kgDm0LeMkJzfiSoglxU36Fy1uxRKzN+f2pYyyqNpV4sEhC1/oZoZww
2vWgUnhcOztzWqdDJGQWOOu36ZmeS3d7m7JGFL+K/D+uuhAdt6gye9qR57SmxlGtvgFWEq0F3htZCXFZHdQ/n7
1Wtj/B8RMONOWK/vXEtmTQwYOG2z4sdGzzbLW7GrBq65YcquO3ZnSrTJ0g1O740qA5fL9bFnc5LKqJz77yBFk6
xyP8ceKiPv0CLfy6NpW4Hz+JA134hIxl0/m5a+zFNiQVA0dMRDYmk8jMIYxE+QiCpbi9e3p1cYMcfXdL41ZMKH
OgJAtNHTCLnj9YQiXxKXFlWxrVSmo4LJuksKFS3N/6v3ciJJIf7yB7jEGtjJFSV8sXAEs2DBI+5zl5QdXPrT3z
bm6kFCmxVx9sMT6q7e84baGtXzumtUU7d8KbBdPRLjwHD64NjiRM6nIoCtDAzozJmV4V+EzAcxafOUfoa9q+ai
pdfTxs7G5hlzQTA3SSrCGQhP+auRhxi/q0iUQ0eCnPwcLI+KDf3PUvfB7Magfcj864IbvlyIdwhJyF59ihLCEJ
zn7qxWjiPkQ6RNR0HHNeU5SMVgBK6Jkc8Vra2N2dnSSIfmFdlROwTElOlaQtUIiat7uKsuu5PkSVYjoqv9g3Ak
ikz9rT2/KQovrQpan83LhDN0oGC5KodpDycaApvzzQRDbPx9b7R03ZV2Cm0LLsF/sPjultJmtsCL7xaX2VOoSf
wO6etu0RrThFU8Avne91oltHS26zHsSCFQaIqRZSfswxTC2UlGdXK3Wu4VU/G/J97cqlTi4+Vw8U5JYPha8Zi6
b+k4wGH2kOuBT9vlxCWWxSUQUsmeSMH1P7rDG19zAcP2+FzQJxjGhpC8CaYYoRf5D8/XHFN1RmjqBCpqfF9dB9
ucDTozCYnz0qfcxMTB5MMb9BpWACInuxek7x1WztAjmBCKjJz9aGIYe6ZZvyN3QDCzscQuwkjrfv9y1qarYtmd
/Xqv1EMrYslru3bCl+icKv3klsnn2Mk3Z7uBkxSVfP6+uvkgyJ9RfZD83Ki3S5haHiibMrUvBD6iIycLHn9FfS
SlKb0GGW/9gzVrskJxqc618zF6NPI2B91QtzuCz5o+1wBPyM6d5y82DLI50d3zDweR0ig7oey6bJolwLIRl1DT
XZ0sm0/gJODH+VZIcsz6b0Wx0d/EW1n9evL7UNYUgTJHFwaE27zy/CoLIUxWAsLsasWtIosAzfCsUy5vLw19h8
3R/KrTH1d/dBg7aZKVDxoopfGLDpdgJs/27Fs0myCbWRq6spUxJf+mbLTAs2F/hG+dq1/Q885DVupB128ppXJR
```

53lPSNxji45v0rwwRp9K1t5kQntLOH9Yyatz6B1Z/pQqYulu/6o2ORc7zRrU3Auuq5FqOk/yOqQf3dJzVI0nUv
DSS7Y5L/RBOIWVQEUsDwLLQZHOAR8a1EZzCThfDDjCCQBuYzpHDBESJfymYOhwtVYdaQ5FlL2c0DhKyde+fgIm
jZ7hKX1DsLyC8ryo1C5aPHzrHxzYy0ZVsLPdUttRae4vTKcQ2z2J5TALIm2ZMKjap06U9JwfNli6a7KNPd/Wwe
JULs7eZby+EHCXV1GlMCLvA4LH0ZCtRZC5IMD2J/C1MoKgE4uOXY4mT3MrIRGQGvkzwPwpUcY/HMOUawX98cLI
4km8cNBU7EQI5CsczpNE7AkD9TSujoje/kowtFiCTX93JbN9cHjWJnm56TJu27NMFShb4RiIZxm6On4prfbr0V
NI/9wULyp5wLFYIiupVMXpjtv6OxM/Ro1I1MxcdlVzFV49L2UhxW1+bulk9cDw+Tu5CWys7kNfXWHewpTjoJO+
/rxl2rXJ6/dszlNKT5jyFjqQJGSZKoU13JrzeuyWB/zW9wJTPvdfgl+sdJhOzKHjXNlXbBXSCOWzq/Dfnyt5ef
6jkj7tLVZQ6ExDGYbhCUFBEypgP8v2lgKocbTBrqHz9dFOQYA76qsH+pWwbLkKCQcfWFA3J28UvbL3aVulBCUH
dSRmTHFm8eRRn+F5ozp1i3ztEiHj3pOdpUq59wd8J5msgF9bEf116ivoTWmMVOQI5geZtXuzPspBlHDd9XYJDY
v6HqKJyz8OJtV7Dqn6Z6xuXa6YxS+QWtrdibeoecoxX74hdA1H73Kn8GIP2ipkZCxkaNjegNWI4XijST0/fbgt
LWXmMJy5Nf4jbTYT7U8Ii1GftDjhhj4mAWbuRlKasoD1ED8fv5UHEfpO+CREqL/qjhB1PMk13NoLdhsg7MIxBE
8f+74g/9hAO6zG5Nnlx65lDr06c5lwFQDHx/QVgUt56vt9DEjtSlU3/HZtc5gK4mG1HWtwlE2fc08sJCtLFU9u
X8ZIYcvpaxp2DXJPOq05ow5erw+4fE9ifaEV43r/6A58M3x+E7JbJVHW0e4xI3Hk9mPQwlASgVRQLvneQiFGwt
PyhKm50CF0a0t40nA/WJX/N9OX0tZ7SkqgCYTY196o9QvnpjgXA95Pq0S5OlgJEB0J2WAy9pIPkaqI7uzRJEaU
/iKuJ7IMc1NNJEFfg4iReDl1PxcM2RCMsIDmCDT9jmlg/UEkeWPBHtk2Nyxo6kaVTlY0ifZSkQUm1FleTkD2pm
szaEYZXeVXatcu90PEkT6HVbxuVX5iqQ1fK8ErKZa7FsKNES7pP52NGb3K0ZsKDdNbflv2jCM58Ye6P0M+YJGe
DJ4VGI1MxzFt2RUY65YvEFNjVhD6aWWJtSScFKM3IrxuzwcI961tde4QU0DXmi2hc2u2k+ea0iB5zxpDtO8voa
WVtLoUadG7FebRHma+vjGCQakVPw/RN71bso5swcF4WUEHxj5WxV4G2tovt5RTwyhOY0vwmB6+1YokAsXA/wxe
IAYb+BXsIJtTLEt23cVyiBTwAGGmSW2GVMAiq0sBGHk7ru9pfkea1TC/r1hOK5od4udS2+hshiE6CvFGH7D+Oa
XJShBWO2VbB67MdvX0mmsAS3FezWnYDd1YZwMKc6ai+ln4L9b3LQ3UV0q3EDuuyXDfIezF63ZR/xaEYhptrPDB
tmysCFiLTbPQaP9s089pclRxaecmvS9d9KGu2UbQ1lYn1xPJmnYO3H1SQmNNI77i9oHk1eeQRlA7m41LWIQo9R
+uBj9RFGft3OVX4022POPZ8ree1/c3bcW7CSsz8QIVtUQSlWXOOhQbGnPx+WnRKruvxwG5YiHY88H3nlwi83Gs
qep86I1VBNhnPYvMC0mo9DcPgcZI/k6ehIsTG3msKczlsqk+jM9lpmcDJNNhC+XQ/VIFEqAN2tt404Y8/2+IxC
LeOXg9YHG3zoam9oTyrmyTU7UxybplcdfkCvGY6CmdIpdM8M12t34sRCDXwUhM2q7dEaimPHoB99WXjb/V4y3J
CnBllUv2JIY8jytD/WFs0KDLLE6QELll84gp2ae1BsH5f4hDXnKlvIVE+ordukeyfrm8N4hlomKmTmCVI81Zvu
FEF/1mpKyZ+XLFlUpm47iFIBLqOaup01i6ZEGd3qGsb2JSqHP7hp6ohLS9wSR5DLkl6UhpOMHRfXc70h9g4HeU
NcB0NliNQZSGslKKC+mQElJGZnYdv0o/OQVYaV3lufk6triP3k3WnroTknO4w+gyPSiZuiFigR+iP4zUJtkoHF
6aYYDZNe+47vL42kqwJ7cdLS84rjqSpg49NNbyfdsPxAV6E1brZ2BWNV+vcnhOn4DGOKVeZ+XHl+ifAZJPkFmz
GJ/EzvWJyd05eVTrrNFUBEEEGU5p89XqsR0bov/CvVXV6r92KvggqIN/U7hJR+9iXUvr7jGIucHMDkkUhmDZKX
eytxbT/ZGAdvZJpfkKBxWuIn58zxbBGMVHAjnkqhmxG87NpQjExdIGsRrvddfbEwavEzTahk6PyN+HLCOHn3Te
HP4mYtfomeawq/n7WX3c5AEf5edNxtl5CelD3UqnOqAcxE0S2Bx0I+qbAhvo62AOYlqWAcI7ayuyZBY72QPZLE
l6ano9+nBP40lYyiE1dPTdhUDTJo5MKWB/XELH+nYBoNTRCasoWYPFCeYloRRj4pNhLKUSGkIfuQi9H0QYHqhn
FAuiIx26QjKeDZ/TxmsVQLq4unTtmQL7+vSWSgKBj6Z4DC+K03RN/CjTNu2viSeossJPEwpRraXkO8k/2OWNuQ
cZIwKMi2vqc3T5fMgN/exfG2Pa1/TlCcJy9ZA5k0BAatdc4xPyewGsIFJOZ9cjqqsbd7P8gqMDOrtyFnxXm8Pf
vjszKU9DcY4Q5QazCwaZXZRwgPPtTCXuF2+/hzVOdFVLfACoMQkzcgIer/TxN5O2K2uQY4IcCb0M6KzHf5etmW
oYVGS51/K7zbVC9MKE+a9c9WrtZh23yq0OUGcTZAmoyUvrQ4sEW1ptmfdDPnYDGGvU8Tzw7ssUdLjSighHu656
lcfnh2RAv151m7L7na7gbyxEitJSVm9MR6h+jOAnc38UnWXuiyvCjv2EAM5eAuW+DYgbmXZUAeHnFbSExQfFkW
aUIOT+dL/em47sLCm7M65s5dxkbpzcIf2eXcfyJ/gDGibT7Sxotp4PcQQxvJVKchrjctRoL4UWfetLUioCykzF
zgsMIwUUXts9s3ysu160Tg3lFcy2DPZ8cuydpPrfVQdBOLW08u1ExLarVVNJhoux/E9CWa5Fe2LvgEIzRC9sE2
LghcCzxF9xjq/dSgsnr8xnz5T8kUT11/WH4gw/ptrFSc1aos/HHuwNQt8wL7SvbdBON5w/pwC7oyN9gTAmlSv7
bJQ84o/ZqL7vV/pGRoo1V0wuQv09tROF0GCV9WT4g33mwI+j9nT2UyarM+/AxSNaIz6KzSv6j6rKDOBpQV/V0W
plpo8fbMDW+toBWH0iwl6kYE+5jKzn++GCXa1Ah4CTtHXdPCLLeGdiNpJm03NPqHD+PhG4pW7TiayHJ8Y8Lt8A
fVqldNqVwHqKnBa4d++WdBEWf2FG+r4gAbmJpfi8RhaEDd2n+PFp3mf0xL8VdHq+XFUyZq1daX9Amzev1rrdX3
vnv3MDcYMMnoyesLWC4Vv2csD8KcO9qGKzQVGAyL5H29V12/pc1EERYbJemq0R4euBTYGzwPDTngEQX4VXAlty
4hc7fOaIPr1JbK0Nn2WdUtADM6cqx6+1g7ldJxdZ+eOSKG3JGK+3W0q23Ct/IAvW/lbcPXRYD7XbnApPPRZMGK
50nqjTc6+8KZBQsl/HNlxCe/cc7nlSuXYWhJM+/uDWcBgubXKWQ5O0BGleORBWoGVxHX8kjfsJSAlQlwQzGlkE
TOGKo/+1T0zJcaVYDSCkccX7l73eYGiKaMKNW/Z1QGjpT/38SJvA3FMGQ8UO51E9IKj2Sgxa0PRRQiKG+NnNdI
ymH9gOG9UGwsDymtqjjXWeq1Eq2p4KizaO7vOmm1NK2nhwzouoclcs/K7zkuKfP7j2c2U5iAkxQxN59anAWH8Z
bJdyLVzweuQpyfa1zpPFG1RkbJyGjYdQDDv9wmXdcVA1AOlt5VRVvLXF9qJv+wK7ogruXPDGiSC7mXT7ThCt3/
9ZPO4Ru8qUvQsY9rMJW3b38Ch/F7O2EED1K/HIbSYu4RPnn8YQwwVcv0eCSSb9CjV0utQWKJpMEzxAZbFQ77v7
E0PosZwDRuxu81ZDg42aWOJw3BfUmTIb2J3itvbaEoeCfFrbUTWaZ7D8mVaz2M0rEXqOkYp3t8T+4uFqkn2mLL

```
WGk5/N3VnlO1NGgwBdoBIQAiqLeAtSX4cG9FRHHwpeBkzQRZgQ1wVPAZ4ILNDsP3TcVMr8er2dfNVBOpc1xXwc
hEJd7wn9bUD0CLpAkGJmAIphXsNoboGyjxFC6Ndn2Cn/0piNnt/+W5sYolgWsWTyxIhfE/wjrT+QhQiDtRBkW+
ziFLR7Ca0XnRrKApoSmMjy932WRmACm4Ct4ocGfSZrWyBDaDIsANpnKTLiw1AmQwGg9xE+8aVFBE2Rpr71IRuE
rbLyP381vCfScvOb0laP/7INX+449nCL0IEvelrtJ8M+RftDGuMis95yL3Kj1+tk3WwiaxJZMiQpMk6DCq3Hu6
JCA5+nGo8YJv6cKPNUkgEc/CJV8ZU9nNzSuLqexLDPMe2y5roufDskz3hmpRTKrHZUAzAfhvTfgkljHz7z4ksp
yrN8ExjyFCFS2KoCeSLfYi7zO2Sh8Mq0udVzF/UoM6xbjSukHU5RUVWvh7O8tM1kQf6cMmdyFlNCclhN3WX5v9
S6bhapjL3dG5gsLqpRiDXY+OF/GXsENxfkPO/GPfcWsnp1hH5P0ZXDmBGZH/4ZMwrxPA57dDdZbmYquVLK8N5h
vItQCk4cXkh1mmnRMqY8nmfS0cCyQEt3csvpvPuP28yXqSzUQlR4ekJtzq5M2+7I5NJxGSorw6QgK0ltBa8UbR
zda+RzliQT60dPHIuVXm32s1oWWOglnymkikHnd3KnMI/K5jSoLQX5BLGsgVqNgcnmnIxwxgfLJuFvYPbiXKFA
YkWozpbDSSqmx8e8lGsvjweiCR8nrFpbxv1SKNqda/ZiJCi3tZFa/nLU4+ZHvDwDuxgjhFPB2G7kKHVmZVj4Jf
nLxkykOI1LXKq4O2kZlY8vtNYFqUp+fVWYsXq0SYOofAdWIPqgEYhAVTxop16UQoaF06va64tgzkc9PxPipLpU
kPtZGXctdZbQFWTapaiN78x9pVEa4H8vldTq2nvJqsLD/6PT4yFoVrHYUJjqxM6VSTdU81LvneIcxrf1jHR6sR
m4DtqCa7Mi544bUUwi3yrBOfVR50Na2I92OrvGC4S6uFmb9S5/QF+gMOfpoT7Qs580c2zUVS7YtH2HpjMT9NiC
F+6bLfLc4Ub5QUmrZFBtkmuOyrufTcKS5qwYDjZ8t981O6rDLnLRclmj1o0fxe23o4I2OjpOfS+GpTorqRSeOG
k634xpVqwzoQtylORCpWfYCY6UBDtl0yeIvT0p4OOYOb2UpCbJNoW/OAIdCm7zzrmqakM8Xxwow1R2rxZqa9OS
MB+l/TL7qqBZp7zBJMGa9p2wGM/zN0FDnC4KG0z/NrLK1tJjCSTz2J3ZrqkRL9YWOmlkaCmvfDDNp0zpXZdGrw
1/ciNY7+t44v2Z0c6VdjCpkAkTQxhiRtx0y8S6Iz/Q7owc10UxGaHK4YnAjssCKnYbU7So3Z8DPGlKXKOrMOGK
hg0IxvbsQ4M2nUghxXiDa9CmQqqv5fvtYvbM4NRwA/AeXWQdEFKd4215nYmaOnyUM4CqGPlGUzM0rB4kyrKIJM
PwhTJ61+x/wBESAuj0FNQlWLwYFOnGGxsf1E13M5bHrjiAvQZAq2ClyoXcxvMIYwd56IM+j8WxOI9IfgwUKVH6
DU0y+VvKb43FPMxZxVz0x0B1Jr54TJQqL8KSQEIMG3OvzEneFabnVDqgNrNEqkDvGpuNBxChWww98c5YOrk3DX
lAjdytZ9GNgHOEV5KmAWpTRoyE6yRZ/jqy+eH8k785r14slw31UmfAQfMUGUFGgtNPPcueA/rWK3QXXnbutydv
Yis0mWi+7wNT3HWYFoyhMZyK0pkwb4Rqo8G9fjl4Lql9EgcspH2t7IDL1kJ9j7H9lPksxIUzSX8x+dzjEBGCyL
tn3ObKekA7r1fXy6bFS+5kYecZNcG+heJzjdOZzwKvch01oGMgB2Sj8l9z9KknP7yxZIxtqsfyoq0AEx0+/PfN
bHN87ZScN7r6JZDgVQZGtYFQNHTIQShphK1bqoy2i/kM+Pl+DjC37kwCtWZIld3JMMjwUG+V1/eB+Zo7g3XIU+
rYx2g4o3/BWF+r1Fj/jkCzVkjxLwte6xW/9M/LWu1aHq6TTvndtDeUQ5uiNECpkbfRECMFNzBgdWQljAEudkgT
1n2T6Z1tVmCPxUmVkWJ2zdfB+WebRs+fNL1UJ3mAf6fXv9BwvHfCM7cLuX6nH3TOfPVZ5Wnnrp60Kpn+p7fLcc
vrkVsuAgJ1k8x8zFc2Ixp685qWkTsou0ydhWt5IiXo+UX8ASHKPI5X8eCQyWV+9GSuh5gJeqZlcXbevmCQ68SM
TZibWKbSGpdT0wzzgIkh0ZJTAe6++OrWwlI5c+CBc8jsacomMa1auRWN7HBvrtn/n2v7+p13ifItCRZ6Sb2XlR
xwttDxBjn3sCMdiezc4y4EkaVeeUsPh2rfZTmR5o36XM6OQ8uQWgsZtK+XrjGqXWH3KshNKkR16S9PdKr7u3Fg
jrzl6NMh8WFw5omEyOtgl+afIZAlDbsB/mUEYxCA98p4OogosQJjgJTyPmyZNRV1OhFkNQfV4WtteJ+I1PWwmk
Qsu3XzqFxrVMh2P8UhzNW+o9H2/6By4UKSWyiEtfGcexcmgGbOYMFpq31TPOyHlrTXYLCCP1YpSBT9DUhBavCe
APtpPCrT55dcS8QUm8suxozvppusA+5+Rq2bT49Kb+5Pzx5/Pz7oOcXN9RrSAQWtbIMCFJffxJV56ajbX2Nb5x
+iaM2jhcjkbXYL3IeR3g9UJMwaILItsld2EBVfBtoLtGPfAP4xAVQ6GjlalOtyqI4SkK6Tq6asqtAoT5OUnzc1
54ZA5ZZnCNiikrfvWrokAdcr7QitMoQLpSids7S11DaRFKl9EePuQqpn4FcL1bzdxiVfUckzRyuyPoyTHQtcOs
MDVuAwtSchWYtksdPiHWwP2jhqb1irHHW++xaKaFApyuavp3kp6GRTl63kbr7ZaXndKFtQ0ZZIiDtWMR38v8ua
lbT+btQRKpz8JzCe5+PcTwq3qoetSsBen5CK/zz9TePTti54qVDegVo0DoT0EzlL75Ga4VVx0L0TuQLwvkn/FE
0feVrTpV4O/uUVnbJuaVLWkA6kPDG8xJ7C4zLgLrJl+cJgBOiT/nq8LWlvIPqnrvBURUtoisyKdmxaueYjMA1A
qPlE6OzZjU/YFxkqOLly28S3TU40/BmxffNBy9RG9z15NffCw5CWz/bF2iPFCIauQWsRoB2em+PhG04X4jTzXO
/epaP7YCiRedP9VDXwKurfaamqGpcnY618UOEYbf9ue3GYhHHdRPEr59Km6crazOn5nY3z1cUrxJxUKbFuL3HO
0QEFLycLkHEoWyyd5qPkdGIVO4FK8OWFpkkcFZGxQL7NPDmlMtgJS1Qw1zwwGU4SUihEgQLSDksWsAEflE/YgS
KD9Tx85L0jkJLvEsBY8oL3cgziQNYzGsZLnT2viNptMqtmqMNFKCQd7hImKt3bZThyqhK6D6SKdLzdXH7iklHS
AEj7jHiPDwCR2JQ0YKfSIZhfvgDYyia4Ba6g/0SKEv3H7mOdzRxRFXaOeri7wxk/7+F1qm77qRmMkXaU50EOR0
osjMJJ0Y+nyd7jCmMmBDz/Ge4m7IPhL397a5fWcuk8aE6a5JtnX0GTtnIsHuILZsmAVK729Nc9YMNMUnEgX+0f
uUEHYN7Y/FjVv+y1I5/3FKoyVslVd/ehwW6fLJYnRCc+fxtNPyKcVURcg9zOLJntnDw32D4RynfIweH8aItPGO
0Pg1ah91BYd2OBq+Gtm+L0f4nKV+mo7wQeDLLPjjIyyfso+5ViD4Pj7RFzdiazJ6PPMsz1PC8lImVYO/4absBG
n/GH5qqgXvq96zzpVzyF8v0vDGrc9a/baWl32HecvWYG9teums88AUw8PSlddmeUJoICV8ApTqDsEQVKzRfdIt
ijhE5GBNWILTE5XrzFVwbIW9ArsXRwSP58XIb05FK2doCD8BiP5be6+WlcjQYXmE3yx3WG4CKfRuILD8vqFN88
XrxJsq4/qZl603S6F+vj7Mho9RB1SPElH7rb618AeeH/bO3R8G/ROr/2leChokfR6NqhGpgcZPMKH8WmTomhwq
TsXw8f84vKWL3srTEnM971Vt8D5t+p/geo07XOWtTbgyKtSG3sC4p7i3OqQ0hUcTmsbL0/9ux0w+rk1noXW+TC
QfX9OJNCRWoZ9oX7YYWGVb9MJ3DzQn0gLxg85OXBrbnOTflv3d7GhVjAh2kC3PlMkIOqYDzot4c9m/ZNvVJF8A
paK1V4T0Z49idBPAQXrlnyhEkCiOKAJeRqAKIPDuHeTduFTGjRdE8vaX+U70yBsYKGF/27SbFMsCiaumVS7KsL
```

dI1rR1rAkjPWLqrTFG01BeRVazU8rkA5adqpo8xdF9zPmcfHSPuLKk+456CVNb68sLMj0hsa8zzGBrd6tbLcx4
gnYnrUkUunnCuNK/5O+7y6nfZwCa5OCFanAVOQKUorr8v66dSfN7Ur1UWohyxZO5NBnkp8Rt4aip8wKnDWFjbA
zSjhgXJBgMLv68tHEnCRV/e73H5PbcHqsjKFFJU7kd5hgNFj4RaE0zXAARYSNOkzdD0c5cSDcBIWvFqfXl0rVg
QeiDA0yqBtETtbVows3Sg3WW+951b9qw4WAGvKNtWW++XKkK/uQ4AudwvYjNWYgx1qYqsXcNVLYLAss9IaIevY
zLZXXXvAHVgY3Kcj9WXdjOpQQ93oduNKcVd9EvzFXEptDfP/VkzzYNWHC8/VFFKSK3n5xVz9p+KiA5Ywv//0Br
uptRnXTOqOubgAPvUFswQKsxl2V/cjMhR7FBulTWIw5rnk9mvzKvf4mT+Zp4C8ia9JSXyZN4SglwwJ61UQjqIN
EXUbjr7g/NzDUn8SEiVQcSZPZQ8YEvOMBosvRBXkiKMdoM0hMkuPsw6psqCJyYpl/0XfRvPU9jPVPGNtXEzwBs
zVgTiYQbXbycaY7Sp+OnkKaUEchIn8xARSyr/VztOfI5hn8Cx9K4k56305+KzKyUbsac3RVXSxsCHJbOnFQiK5
7Zcv1+CY7CFMQAgsW+rEaiy2vDT9Y1atbOyqqtG1Z2+W/qWZ6+ehGfoVoHCauEPqPAoP4GCv8ur//35unyduRC
i8ZjVg2+xUWSOg4ncEFnHgoIhJDWGJgCjW2ZRL97EawzTxENvoQontx76rF4wHwO8qKddLw4i/0852cdVQtdFO
BHI9jdIL9wR30p5hMzC7pnkYMg5iNirvRMqw4AhVPpzG3XLPQw5ziR7zy/aJN1oj0RTf4jdmyRlwW7Iro9bfr9
9nYNHDo/nVkDCBFP8QX+34ojvF226hgaSK69HBFUYOs0RYkxL1T4x9RMRH3V0qK19Q3rVQ3KJrLaA22KCp3GMK
VZNNwtJfn12oy2y/ioHC+y9b889Yq75ykYapr9B+gkxCaW8b2LOVQCGEDoxojGurYVXsdNpTjClxkmx+xY2ux1
Eom+d7Jka04HQpqEK4BhUztZYoHu5Mavpk0c4KqVoDcZuYznK6abCYBK/kclCBo+hiommWuJe8/KYZ1+n8EEDs
Te41cJA8ro3+zT7hZv4/MK+ItY3BgHK+Rupmj1ml7hJjg0lg7Eqj4uRi+vCCxeg3kzpANoMk8cHNMyPx4V/XRA
kaw9iEpkh6JUpqSFOnzFbrmuepkJH7qqRYNXBcwazHGvq60RiUeNaIioWJWKUxWe50IykXoEnOkh5B6k+y6T3y
AOHotFFMuz3qRSzNE1Nv0V0UGPOj4dyiLzpD4iQwAxv9P/Z/B/w4DYxtTQycXe1tDJ2uAt5Ops4u9k6mvpRngf
wA0vbZhCmVuZHN0cmVhbQplbmRvYmoKCjMzIDAgb2JqCjc4MjEzCmVuZG9iagoKMzQgMCBvYmoKPDwvVHlwZS9
Gb250RGVzY3JpcHRvci9Gb250TmFtZS9OaW1idXNTYW5MLVJlZ3UKL0ZsYWdzIDQKL0ZvbnRCQm94Wy0xNzQgL
TI4NSAxMDIxIDk1M10vSXRhbGljQW5nbGUgMAovQXNjZW50IDk1MwovRGVzY2VudCAtMjg1Ci9DYXBIZWlnaHQ
gOTUzCi9TdGVtViA4MAovRm9udEZpbGUgMzIgMCBSCj4+CmVuZG9iagoKMzUgMCBvYmoKPDwvTGVuZ3RoIDg3M
i9GaWx0ZXIvRmxhdGVEZWNvZGU+PgpzdHJlYW0KeJxd1s1u2zoQBeC9n0LLdlFYEmc4MWAEIGUJyOLeFk37AI7
NpAYa2VCcRd6+OnPU30XiY1oafhzJlNfd3e5uPF3Xn6bz4b5cq8fTeJzKy/l1OpTqoTydxlXTVsfT4bq88/+H5
/1ltZ7PvX97uZbnu/HxvN2u1p/nz16u01v1Lh3PD+X9av1xOpbpND5V77529/P7+9fL5Xt5LuO1qle3t9WxPM5
1/ttf/t8/l7Wf9eHuOH98ur59mE/5fcCXt0upWn/fkHI4H8vLZX8o0358KqttXd9W22G4XZXx+M9nTV3znIfHw
7f9tNq2OLau55c5N8wNcsvcIgfmgCzMgqzMihyZI7LNua2bDfINx2+QN8w+npgTcmbOyB1zh7xj3iH3zD3ywDy
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/AnOhOcic4EZ6IzwZnEr4vf54nO+QU72rJzyebvjSz5ApqNF
+ICEhaQuIDkhZYNyMcTLzygiQtIWEDqfNy/5GnHjEUmLGDItaMHZsyVG9YBOreefTPNgRnHZzY9Y97MRmc0OnO
j9E0237AO5sobZjQrL040MdOZ4cx0+oabF6fPy5sj4+bolhsa4x0bndGHzhvddji3c1u7Qx+6pdF+buQ4Lnxnz
OhJt/TTxxPHYe4Wm4/vOA5z1zPD3A2cF33YLV88eHo63dMvNwFq9j5X4xtiz2vnnt7nbXxz6Zd5cfzAOj3qDL6
Wxh8Gw1IHxw+sM6DOsNTBtRtYZzf8eZPhAYon/M8Hc3V4nab5oew/A/xpjOfwaSzVr58Kl/MFp/nfD5Io2CIKZ
W5kc3RyZWFtCmVuZG9iagoKMzYgMCBvYmoKPDwvVHlwZS9Gb250L1N1YnR5cGUvVHlwZTEvQmFzZUZvbnQvTml
tYnVzU2FuTC1SZWd1Ci9Ub1VuaWNvZGUgMzUgMCBSCi9GaXJzdENoYXIgMCAvTGFzdENoYXIgMjU1Ci9XaWR0a
HNbMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMAowIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCA
wCjI3OCAyNzggMzU1IDU1NiA1NTYgODg5IDY2NyAyMjEgMzMzIDMzMyAzODkgNTg0IDI3OCAzMzMgMjc4IDI3O
Ao1NTYgNTU2IDU1NiA1NTYgNTU2IDU1NiA1NTYgNTU2IDU1NiA1NTYgMjc4IDI3OCA1ODQgNTg0IDU4NCA1NTY
KMTAxNSA2NjcgNjY3IDcyMiA3MjIgNjY3IDYxMSA3NzggNzIyIDI3OCA1MDAgNjY3IDU1NiA4MzMgNzIyIDc3O
Ao2NjcgNzc4IDcyMiA2NjcgNjExIDcyMiA2NjcgOTQ0IDY2NyA2NjcgNjExIDI3OCAyNzggMjc4IDQ2OSA1NTY
KMjIyIDU1NiA1NTYgNTAwIDU1NiA1NTYgMjc4IDU1NiA1NTYgMjIyIDIyMiA1MDAgMjIyIDgzMyA1NTYgNTU2C
jU1NiA1NTYgMzMzIDUwMCAyNzggNTU2IDUwMCA3MjIgNTAwIDUwMCA1MDAgMzM0IDI2MCAzMzQgNTg0IDAKMCA
wIDAgMCAwIDAgMCAwIDAgMCAwIDAgMAowIDAgMCAwIDAgMCAwIDAgMCAwIDAgMCAwCjAgM
zMzIDU1NiA1NTYgMTY3IDU1NiA1NTYgNTU2IDU1NiAxOTEgMzMzIDU1NiAzMzMgMzMzIDUwMCA1MDAKMCA1NTY
gNTU2IDU1NiAyNzggMCA1MzcgMzUwIDIyMiAzMzMgMzMzIDU1NiAxMDAwIDEwMDAgMCA2MTEKMCAzMzMgMzMzI
DMzMyAzMzMgMzMzIDMzMyAzMzMgMzMzIDAgMzMzIDMzMyAwIDMzMyAzMzMgMzMzCjEwMDAgMCAwIDAgMCAwIDA
gMCAwIDAgMCAwIDAgMCAwIDAKMCAxMDAwIDAgMzcwIDAgMCAwIDAgNTU2IDc3OCAxMDAwIDM2NSAwIDAgMCAwC
jAgODg5IDAgMCAwIDI3OCAwIDAgMjIyIDYxMSA5NDQgNjExIDAgMCAwIDAKXQovRm9udERlc2NyaXB0b3IgMzQ
gMCBSPj4KZW5kb2JqCgozNyAwIG9iago8PC9GMSAzNiAwIFIvRjIgMjYgMCBSL0YzIDIxIDAgUi9GNCAzMSAwI

```
FIvRjUgMTEgMCBSL0Y2IDE2IDAgUgo+PgplbmRvYmoKCjM4IDAgb2JqCjw8L0ZvbnQgMzcgMCBSCi9Qcm9jU2V
0Wy9QREYvVGV4dF0KPj4KZW5kb2JqCgoxIDAgb2JqCjw8L1R5cGUvUGFnZS9QYXJlbnQgNiAwIFIvUmVzb3VyY
2VzIDM4IDAgUi9NZWRpYUJveFswIDAgNTk1IDg0Ml0vR3JvdXA8PC9TL1RyYW5zcGFyZW5jeS9DUy9EZXZpY2V
SR0IvSSB0cnVlPj4vQ29udGVudHMgMiAwIFI+PgplbmRvYmoKCjYgMCBvYmoKPDwvVHlwZS9QYWdlcwovUmVzb
3VyY2VzIDM4IDAgUgovTWVkaWFCb3hbIDAgMCA1OTUgODQyIF0KL0tpZHNbIDEgMCBSIF0KL0NvdW50IDE+Pgp
lbmRvYmoKCjM5IDAgb2JqCjw8L1R5cGUvQ2F0YWxvZy9QYWdlcyA2IDAgUgovUGFnZU1vZGUvVXNlT3V0bGluZ
XMKL09wZW5BY3Rpb25bMSAwIFIgL1hZWiBudWxsIG51bGwgMF0KL0xhbmcoaHUtSFUpCj4+CmVuZG9iagoKNDA
gMCBvYmoKPDwvQXV0aG9yPEZFRkYwMDQ2MDA2MTAwNzQwMEU5MDA3MjAwMjAwMDVBMDA3MzAwNkYwMDZDMDA3N
D4KL0NyZWF0b3I8RkVGRjAwNTcwMDcyMDA2OTAwNzQwMDY1MDA3Mj4KL1Byb2R1Y2VyPEZFRkYwMDRDMDA2OTA
wNjIwMDcyMDA2NTAwNEYwMDY2MDA2NjAwNjkwMDYzMDA2NTAwMjAwMDM1MDAyRTAwMzI+Ci9DcmVhdGlvbkRhd
GUoRDoyMDE3MDUyNTE0NTcxMyswMicwMCcpPj4KZW5kb2JqCgp4cmVmCjAgNDEKMDAwMDAwMDAwMCA2NTUzNSB
mIAowMDAwMTM1MjE1IDAwMDAwIG4gCjAwMDAwMDAwMTkgMDAwMDAgbiAKMDAwMDAwMDU0OSAwMDAwMCBuIAowM
DAwMDAwNTY5IDAwMDAwIG4gCjAwMDAwMTAzMTEgMDAwMDAgbiAKMDAwMDAwMDEzNTM1OCAwMDAwMCBuIAowMDA
wMzMyIDAwMDAwIG4gCjAwMDAwMTgxODggMDAwMDAgbiAKMDAwMDAxODIwOSAwMDAwMCBuIAowMDAwMDE4NDA0I
DAwMDAwIG4gCjAwMDAwMTg3ODkgMDAwMDAgbiAKMDAwMDAxOTAzMSAwMDAwMCBuIAowMDAwMDI4MDQyIDAwMDA
wIG4gCjAwMDAwMjgwNjQgMDAwMDAgbiAKMDAwMDAyODI1NSAwMDAwMCBuIAowMDAwMDI4NjIxIDAwMDAwIG4gC
jAwMDAwMjg4NDIgMDAwMDAgbiAKMDAwMDAzNDQyMCAwMDAwMCBuIAowMDAwMDM0NDQyIDAwMDAwIG4gCjAwMDA
wMzQ2MzUgMDAwMDAgbiAKMDAwMDAzNTAwNyAwMDAwMCBuIAowMDAwMDM1MjMxIDAwMDAwIG4gCjAwMDAwNDQzN
DEgMDAwMDAgbiAKMDAwMDA0NDM2MyAwMDAwMCBuIAowMDAwMDQ0NTU4IDAwMDAwIG4gCjAwMDAwNDQ5NDQgMDA
wMDAgbiAKMDAwMDA0NTE4MiAwMDAwMCBuIAowMDAwMDUzNzg5IDAwMDAwIG4gCjAwMDAwNTM4MTEgMDAwMDAgb
iAKMDAwMDA1NDAwNiAwMDAwMCBuIAowMDAwMDU0MzkxIDAwMDAwIG4gCjAwMDAwNTQ2MzEgMDAwMDAgbiAKMDA
wMDEzMjk1OCAwMDAwMCBuIAowMDAwMTMyOTgxIDAwMDAwIG4gCjAwMDAxMzMxNzAgMDAwMDAgbiAKMDAwMDEzN
DExMiAwMDAwMCBuIAowMDAwMTM1MDc3IDAwMDAwIG4gCjAwMDAxMzUxNjAgMDAwMDAgbiAKMDAwMDEzNTQ1NyA
wMDAwMCBuIAowMDAwMTM1NTc2IDAwMDAwIG4gCnRyYWlsZXIKPDwvU2l6ZSA0MS9Sb290IDM5IDAgUgovSW5mb
yA0MCAwIFIKL0lEIFsgPEM1NDcyMzRBQ0IyMTJCN0NENjVCQTk2MUY5RTlBQ0FFPgo8QzU0NzIzNEFDQjIxMkI
3Q0Q2NUJBOTYxRjlFOUFDQUU+IF0KL0RvY0NoZWNrc3VtIC80NkY1M0U2RTZFNTBEREJFNDg5QjBGQkJBN0FGQ
TQ3NAovQWRkaXRpb25hbFN0cmVhbXMgWy9hcHBsaWNhdGlvbiMyRnZuZCMyRW9hc2lzIzJFb3BlbmRvY3VtZW5
0IzJFdGV4dCA0IDAgUgpdCj4+CnN0YXJ0eHJlZgoxMzU4MDkKJSVFT0YKDQozOSAwIG9iago8PAovVHlwZSAvQ
2F0YWxvZwovUGFnZXMgNiAwIFIKL1BhZ2VNb2RlIC9Vc2VPdXRsaW5lcwovT3BlbkFjdGlvbiBbMSAwIFIgL1h
ZWiBudWxsIG51bGwgMF0KL0xhbmcgKGh1LUhVKQovQWNyb0Zvcm0gPDwKL0ZpZWxkcyBbNDEgMCBSXQovU2lnR
mxhZ3MgMwo+Pgo+PgplbmRvYmoKNDEgMCBvYmoKPDwKL0ZUIC9TaWcKL1R5cGUgL0Fubm90Ci9TdWJ0eXBlIC9
XaWRnZXQKL0YgMTMyCi9UIChTaWduYXR1cmUxKQovViA0MiAwIFIKL1AgMSAwIFIKL1JlY3QgWzAuMCAwLjAgM
C4wIDAuMF0KL0FQIDQzIDAgUgo+PgplbmRvYmoKNDIgMCBvYmoKPDwKL1R5cGUgL0RvY1RpbWVTdGFtcAovRml
sdGVyIC9BZG9iZS5QUEtMaXRlCi9TdWJGaWx0ZXIgL0VUU0kuUkZDMzE2MQovQ29udGVudHMgPDMwODIwNjcwM
DYwOTJBODY0ODg2RjcwRDAxMDcwMkEwODIwNjYxMzA4MjA2NUQwMjAxMDMzMTBGMzAwRDA2MDk2MDg2NDgwMTY
1MDMwNDAyMDEwNTAwMzA3MjA2MEIyQTg2NDg4NkY3MEQwMTA5MTAwMTA0QTA2MzA0NjEzMDVGMDIwMTAxMDYwM
zJBMDMwNDMwMzEzMDBEMDYwOTYwODY0ODAxNjUwMzA0MDIwMTA1MDAwNDIwMjMyRkU2N0ZEMTRBNEQwNUU2MkV
BMjlCMzUxNzM3RjQ2M0E1RkI5RjFFMjZFNkQzQzgyQjcxQTZCNjE0QzNBQTAyMTEwMEU1RERENEN0I3MTY5MjJFR
kEzNEQ2RjY2QUE3NDRCQ0E1MTgwRjMyMzAzMjMwMzEzMjMxMzQzMTM1MzMzODM1Mzk1QUEwODIwMzcyMzA4MjA
zNkUzMDgyMDI1NkEwMDMwMjAxMDIwMjAxNjQzMDBEMDYwOTJBODY0ODg2RjcwRDAxMDEwQjA1MDAzMDU1MzExO
DMwMTYwNjAzNTUwNDAzMEMwRjczNjU2QzY2MkQ3MzY5Njc2RTY1NjQyRDc0NzM2MTMxMTkzMDE3MDYwMzU1MDQ
wQTBDMTA0RTZGNzc2OTZFNjEyMDUzNkY2QzY2MkQ3NzY5Njc2RTY1NjQyRDc0NzM2MTMxMTkzMDE3MDYwMzU1MDQwMzBDMEY3M
zY1NkM2NjJENzM2OTY3NkU2NTY0MkQ3NDczNjEzMTE5MzAxNzA2MDM1NTA0MEEwQzEwNEU2Rjc3Njk2RTYxMjA
1MzZGNkM3NTc0Njk2RjZFNzMzMTExMzAwRjA2MDM1NTA0MEIwQzA4NTA0QjQ5MkQ1NDQ1NTM1NDMxMEIzMDA5M
DYwMzU1MDQwNjEzMDI0QzU1MzA4MjAxMjIzMDBEMDYwOTJBODY0ODg2RjcwRDAxMDEwMTA1MDAwMzgyMDEwRjA
wMzA4MjAxMEEwMjgyMDEwMTAwQjBGNjVBODI4OEI4NkQwNzIxREM5NjcyM0YyMkNBNzQyRUFEMzg3MDcxOTA2Q
kQ0RDA4OTlERDg1MkYxQjQ4OTY4NTM5MTQ4MjYzMDdENENBMjNBRjUwNDc0NTc4MDUwRTE4NzA2OTI0QjFCQzN
FNkI0MUU1NDAyNzlGOEEzRUIyMjcyMDFDMzNCRTA1QTAzREU4NUY0NDMwRjY5OTkzOTMyMzIzOEY1MUU1QTU2R
DhCMTg1QTc0RDc3OEQ2NDVCMTdCMkIwMTdGRjI5OTFBRTRFMUFDRTVFMjU3RDU2NEVDRUVFNjgzRDMwNDFBM0R
```

GQzI5MUUzQTFCNTlGMDY2QkZGRDAzRjA3OEVFMzRGMTJERUFDMTVEMkY3OTNGRTk5QjcyMkI5QzhGQjU5NjlEO
DNBQjNDQkUwNDcwOTVGN0FFMjg4RDZDOUNBOEM1ODE2MTk0QjAxRTJENDZFQTRBNDJFMkE4OTg3RjE2NDExOUV
BOTgyRDNBQzNFRTYxRjc3NzQ3QUQxQTE2NkI4Qjc4QzBFOTE5MjUyODE5Qjk2MkE2QTZGQ0I3OEE4Njk0MzZDO
EREMkFBMTIzNDBDMzU2ODg5MDYzOUZBOEFDOTYzRDZGNjkyMjE4NTgyM0RDQkFDOUUxNTFDQTFFMjBBNzk0QzZ
DOTNDRTZGMUNCQTJCRDEwMjAzMDEwMDAxQTM0OTMwNDczMDBFMDYwMzU1MUQwRjAxMDFGRjA0MDQwMzAyMDc4M
DMwMTYwNjAzNTUxRDI1MDEwMUZGMDQwQzMwMEEwNjA4MkIwNjAxMDUwNTA3MDMwODMwMUQwNjAzNTUxRDBFMDQ
xNjA0MTQyMDE4RDMyMUJCMkYzMEFFRUVEMzBEODJCREZFRDg4QTdDNDRDOEY5MzAwRDA2MDkyQTg2NDg4NkY3M
EQwMTAxMEIwNTAwMDM4MjAxMDEwMDk3NEM4OTRCMzJGM0RGOEM2OEFGMEFBMDdCODFEODFBOTY4NjVCNDQzMTl
DNUYzQkM4RTEwMTAzNDY5N0YzRUY4NDM1MEI4MjBCOTM5NTMzMjFCMjJGQjVFNkNCNzJBMDgyMDJBNTdFQ0RCO
TEzNUE1OEJFQkMzMUNGN0Q2MzRDQkJGRDMwRjc1NUM2NzZENkRGMDZFRjUwRDZCOEExMkQ5ODA0RjJBRDRCRkR
GODNBQzUzMzJBNDEzNDI0MDVBMEU5MkRFNTg2NjkxNEEwQ0U3RDUxOTU0NjI4QzUwMzg2Q0M1ODZERDZGNUZER
jE3RTEwRUFCOENFOUNBRUNGODYzMjIyQUVFRUI0NEUwQ0M0OEU5QzZEMjExMERBMkNBQzk2ODJCNDAwQzg0MDA
5Q0JDODFGNDkyOTIyOThBNzlDQkM4NUVDOUIyRDM1MDEwMEQyQkI0NUMzMTREMUNEQkRDNEJGNjI1MUZDMDg2Q
TgxNEIyMDE1MzFDQTI0Njg5RDU3MkY0MTE5NkNBODMwRTM0ODAwM0RDQ0MwMDIzRjJGMzIxODc4NUNEOTQxMTM
xMTE5RDE3RjU5QTEyMDFFQkMxQzk0QzkxQzUzN0M5QkQ0MTNCMzg5RkNFMDU3NzZCOEU5NkVCNjlGOTY4QjJCO
UFDQzE5NTI2REU0NEUzQzNDMzE4MjAyNUIzMDgyMDI1NzAyMDEwMTMwNUEzMDU1MzExODMwMTYwNjAzNTUwNDA
zMEMwRjczNjU2QzY2MkQ3MzY5Njc2RTY1NjQyRDc0NzM2MTMxMTkzMDE3MDYwMzU1MDQwQTBDMTA0RTZGNzc2O
TZFNjEyMDUzNkY2Qzc1NzQ2OTZGNkU3MzMxMTEzMDBGMDYwMzU1MDQwQjBDMDg1MDRCNDkyRDU0NDU1MzU0MzE
wQjMwMDkwNjAzNTUwNDA2MTMwMjRDNTUwMjAxNjQzMDBEMDYwOTYwODY0ODAxNjUwMzA0MDIwMTA1MDBBMDgxR
DMzMDFBMDYwOTJBODY0ODg2RjcwRDAxMDkwMzMxMEQwNjBCMkE4NjQ4ODZGNzBEMDEwOTEwMDEwNDMwMUMwNjA
5MkE4NjQ4ODZGNzBEMDEwOTA1MzEwRjE3MEQzMjMwMzEzMjMxMzQzMTM1MzMzODM1Mzk1QTMwMkQwNjA5MkE4N
jQ4ODZGNzBEMDEwOTM0MzEyMDMwMUUzMDBEMDYwOTYwODY0ODAxNjUwMzA0MDIwMTA1MDBBMTBEMDYwOTJBODY
0ODg2RjcwRDAxMDEwQjA1MDAzMDJGMDYwOTJBODY0ODg2RjcwRDAxMDkwNDMxMjIwNDIwQ0EwQjQzRDNDQzFBR
DczMjRDM0VCMUQ0MzhCQzVFOEIxQzI5MkQ2RUU2NkI4MzU4QzgzRkMzMjU2QjExNkZEOTMwMzcwNjBCMkE4NjQ
4ODZGNzBEMDEwOTEwMDIyRjMxMjgzMDI2MzAyNDMwMjIwNDIwRTE3Qjg3ODgyRUFFRjZBRDQwODRBMTNCNzJFM
jc2MUQ3NkIwMUQ3NTQ0Nzk2MUNBMEYzRkVBMUE5MjMyN0VEMzMwMEQwNjA5MkE4NjQ4ODZGNzBEMDEwMTBCMDU
wMDA0ODIwMTAwODBBRDY4NjgxNUI0Q0Y2MDUzQ0I3RUMxMDIwODUzMDE4OEVFRkE5QTUwOTk4N0EyMTFDNjlDO
DcyNDY4MzBBN0M4MEI4RkEyRTRFOERDMzBGOTQ5NERBOTBFNkY2QzY0NzBDMkJDMzZBRThFQzAwRjU0QzEwNDA
xRUE1NEFFODI3RDU3NDIxMDQ0MjY5Mzk0MzAyMEIwMzdFNEZBNzQ0MkI5RUUwRTk2OEM0QkZBREVDNDNDRjgxQ
jU2NDg2M0I4NkEzNzkzRTBERTI0N0VBMjJFNDE2NTI1NEVCMzEzMTc1RDI0MzJCRUE0RjA1NEYwNzcxOUMyRDd
CMTJEMThEQjQ1QTJCRUIxQ0QyMEY2MEVGRDdEQjYwRjY0RjdCQTgzRDRCQjY2MDNEMDNFQTU1OTYzQjIyQkNDN
zcwNzA0M0NCNEI1QzVGNkZBRkMwMkZCMzhBMTVEMDcyMUU0Qzk0MDU4ODZDNUU4QzdDNjEyM0M4Q0U0QTRFQUY
xMUE5MzczNTlCMzBENTA2ODc0QUYyQTU0NEFDMjZGRTZFODI5ODMzMDlGNUVCNzFBODkxQzk5OUExQTI4NDM2R
kI0NTQ4RkIyN0FDMjUyREYxMzI1NjFDREM0M0FEQjMxNzdCMTU3MjFCOUQ4NDNDNEY2NDM4MjFFNDJEMDJDRDM
1RDg3NTUwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM
DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA
wMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM
DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA
wMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM
DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA
wMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM
DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA
wMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM
DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA
wMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM

DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA
wMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM

wMDAwMDAwMDAw

DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA
wMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM

wMDAwMDAwMDAw

```
DAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDA
wMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwMDAwM
```

```
Content-Length: 9972

{
  "parameters" : {
    "signingCertificate" : {
      "encodedCertificate" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+XqO"
    },
    "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" : 1607960338417,
      "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" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48ZHM6U2lnbmF0dXJlIHhtbG5zOmRzPSJod
HRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIiBJZD0iaWQtYWZkZTc4MjQzNjQ2OGRkNzRlZWIxODF
mN2NlMTEwZTEiPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpemF0aW9uTWV0aG9kIEFsZ29yaXRobT0ia
HR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PGRzOlNpZ25hdHVyZU1ldGhvZCBBbGd
vcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiLz48ZHM6U
mVmZXJlbmNlIElkPSJyLWlkLTEiIFR5cGU9IiIgVVJJPSJzYW1wbGUueG1sIj48ZHM6RGlnZXN0TWV0aG9kIEF
sZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+PGRzOkRpZ2VzdFZhb
HVlPmtjREhPWmp3WmhWZnVEaHVoQ2VDRVJSbVlwVEg0Smo0Um1mVlZpMzFROWc9PC9kczpEaWdlc3RWYWx1ZT4
8L2RzOlJlZmVyZW5jZT48ZHM6UmVmZXJlbmNlIFR5cGU9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDE5MDMjU2lnb
mVkUHJvcGVydGllcyIgVVJJPSIjeGFkZXMtaWQtYWZkZTc4MjQzNjQ2OGRkNzRlZWIxODFmN2NlMTEwZTEiPjx
kczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzEwL
3htbC1leGMtYzE0biMiLz48L2RzOlRyYW5zZm9ybXM+PGRzOkRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA
6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPjxkczpEaWdlc3RWYWx1ZT5EenR3TlRtUm9PQ
W02L2xNSThSeW01eFpQekl2TFlEem4vZWJZWWtQc3I0PTwvZHM6RGlnZXN0VmFsdWU+PC9kczpSZWZlcmVuY2U
+PC9kczpTaWduZWRJbmZvPjxkczpTaWduYXR1cmVWYWx1ZSBJZD0idmFsdWUtaWQtYWZkZTc4MjQzNjQ2OGRkN
zRlZWIxODFmN2NlMTEwZTEiPllBN3NFTnQzTjh1ZkxGTW5LcjM2cjBQcXpNaVkzUTBzKytJR1RFVUMwc3BheFV
2MGRIWk0wZC95bjNrcExKTG9Va0k0TTNmbGo1V0duODNrZjA1QnFNMWtoc1g2MUdKemFGVFBHcG03YWtSUUtod
m9IMjV5eXFUWVhFU2xCY20wNGl6aUtoTE16WmpVZng0L0IxWkl5c3Y1cElCZ0oycjJvaTZqTG9wOXd3M2dlNGM
0WUpvYUsrU1hrNmh5VE5PY044UGpHZTYzV1lPVE5WUFFGdmphOEJud2crYTBiQnV3RCs4TjZmd2lnQ2RXNWEvN
ERKVWUvSjhNYjcwWkk4UG9Pem5HRGZpK1RQYmlJZVZtQ2JsNW1Vb1VnMlEveFlsdUpmTGgzdUdRQVhLQnZGNDV
vRElIUlZlZm5OL0QvV3l0QUNsVVZEb1FTeXdlbW5qUHBxRjhlZz09PC9kczpTaWduYXR1cmVWYWx1ZT48ZHM6S
2V5SW5mbz48ZHM6WDUwOURhdGE+PGRzOlg1MDlDZXJ0aWZpY2F0ZT5NSUlEMURDQ0FyeWdBd0lCQWdJQkNqQU5
CZ2txaGtpRzl3MEJBUXNGQURCTk1SQXdEZ1lEVlFRRERBZG5iMjlrTFdOaE1Sa3dGd1lEVlFRS0RCQk9iM2RwY
m1FZ1UyOXNkWFJwYjI1ek1SRXdEd1lEVlFRTERBaFFTMGt0VkVWVFZERUxNQWtHQTFVRUJoTUNURlV3SGhjTk1
UWXhNREkyTURjMU5ETXhXaGNOTVRnd09ESTJNRGMxTkRNeFdqQlBNUkl3RUFZRFZRUUREQWxuYjI5a0xYVnpaW
El4R1RBWEJnTlZCQW9NRUU1dmQybHVZU0JUYjJ4MWRHbHZibk14RVRBUEJnTlZCQXNNQ0ZCTFNTMVVSVk5VTVF
zd0NRWURWUVFHRXdKTVZUQ0NBU0l3RFFZSktvWklodmNOQVFFQkJRQURnZ0VQQURDQ0FRb0NnZ0VCQUxSQ1VJU
VpidzNuU2RMcCtCOWN6RUNncFpra1E1eFY0ZzlNLzd3bGc5N29DQ2Y3VUVoOUJBMWQrellqc3p2K0JKMWJKWlB
nYW4yMTQ0QXZnc29HSmZiNlVJeVZXNGdrbFVnSWwxYXJVdm9uK1drS25zZUZ1UU9mSnlqU0ZVREl3bnV2cDBoe
mNKWEhYUm1MZG15aCtuKzZOTUgwb201dFZvU2ZRcnRCVmlDTGVTTVZ6dUQ1RVBqMG1JUmN4OTFwTDM4ZTNGTlR
```

```
XN05hR1pMZWV6dUZ1Ui9xN3o5M2xMa3ZaNFZBTU5HR0x2SVhPWWVSQlpNeVBocEJaNEwzQThJM0VFbEtXSC8xT
HdpaVhUVFNHMXNNNld2TVRWYmYydmJkNDduWlJRQTJtU3BOR2pRb3VPQUVyZmVWVlVxeklDZ2hRQ0hSR09OdVN
MRy9IZnFGSGI0aldnMENBd0VBQWFPQnZEQ0J1VEFPQmdOVkhROEJBZjhFQkFNQ0JrQXdnWWNHQ0NzR0FRVUZCd
0VCQkhzd2VUQTVCZ2dyQmdFRkJRY3dBWVl0YUhSMGNEb3ZMMlJ6Y3k1dWIzZHBibUV1YkhVdmNHdHBMV1poWTN
SdmNua3ZiMk56Y0M5bmIyOWtMV05oTUR3R0NDc0dBUVVGQnpBQ2hqQm9kSFJ3T2k4dlpITnpMbTV2ZDJsdVlTN
XNkUzl3YTJrdFptRmpkRzl5ZVM5amNuUXZaMjl2WkMxallTNWpjblF3SFFZRFZSME9CQllFRk4ycEhELzdQZWZ
tQlQ4b1gyOVpoV3kvT0hKMU1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQkszVk9MaERJVldLb0ZycmhoV3phZ
GR0azZYUXRjd1JvTlBWU3NpL2dPcnpzZE03MEEzMXhJVHc3WWZMaHBvVkExeG83b3ZIbGRwTGxocXk5bzV3aDI
4MnlDcHFCVUF0Z3JTa0RHb2crSzdDTDZnVXBybFlpWnVHWnJ0ZzJYM2ZIUzJVc3g0WkozdElqNndWZWNERVVxS
VNGZkZUMkVzbTBRWFVuZ0lLRk1sOTVYZ210dzJ3eFhiT3pVZURkNERJUHJ2K21XNXBvQVdyNkl0c1YrSDJWUSt
aTC9rQm53V0hqU1RPYUdGaXNxWFkvYUgvMVB0QlhBKzE1K1lJV2VtSkJTdjNrRGFGek9YQUV0UjlaSThsWU9KY
XJuWTdBeS9hTjZiOXVHZmZyYm8vaFZBY0w0V0RkaGtiQk4zbTh3K2c3NkxvQVhOZUVldTA0QS8weExaelVCPC9
kczpYNTA5Q2VydGlmaWNhdGU+PGRzOlg1MDlDZXJ0aWZpY2F0ZT5NSUlENmpDQ0F0S2dBd0lCQWdJQkJEQU5CZ
2txaGtpRzl3MEJBUXNGQURCTk1SQXdEZ1lEVlFRRERBZHliMjkwTFdOaE1Sa3dGd1lEVlFRS0RCQk9iM2RwYm1
FZ1UyOXNkWFJwYjI1ek1SRXdEd1lEVlFRTERBaFFTMGt0VkVWVFZERUxNQWtHQTFVRUJoTUNURlV3SGhjTk1UW
XhNREkyTURjMU5ETXdXaGNOTVRnd09ESTJNRGMxTkRNd1dqQk5NUkF3RGdZRFZRUUREQWRuYjI5a0xXTmhNUmt
3RndZRFZRUUtEQkJPYjNkcGJtRWdVMjlzZFhScGIyNXpNUkV3RHdZRFZRUUxEQWhRUzBrdFZFVlRWREVMTUFrR
0ExVUVCaE1DVEZVd2dnRWlNQTBHQ1NxR1NJYjNEUUVCQVFVQUE0SUJEd0F3Z2dFS0FvSUJBUUNiYmw1c0tCQ2p
TQjhUTWRhY3lteC9XZk9qTVcxZ2lJalZKUlkyOEpiTldrQ1ZtdHpnbWl0Z2hmclBRUGx1ZXUwRFRhbGJEa3JTU
3loQ3Z6elBTR1B3Q0ZPYWhGL243aFFhMUYzVWFIU3hUS3JGQzVuT3dkTHp4S1JPM1dqVnRJR1JTWDJrdjFGZlV
wUXk1RXl5K3JzZlN6SjU5ZFU1WlpkV3BkYUR1RHhWVnlEZXIzRUl5Q2JHNy81SDlNRDRZdXpOcGVURldtTTZjV
VNUMDc5NlhEbGJFeFNUVEdRWEZKQTIrQ0NzeTlEWG5KYThuejBGRThmbWN2UUhlVTZrOVFicHpHak1kM0RXbEU
2bm83VWRDWUQxSDA0K3VzQnA1aGhDckFCNjcwTmRvVHJOVG1HTkFGdDRKVDB2aXRqS0hxOUtFSWQ2TGhkY20yV
Gc5M2REY1dGdEFnTUJBQUdqZ2RRd2dkRXdEZ1lEVlIwUEFRSC9CQVFEQWdlQU1FRUdBMVVkSHdRNk1EZ3dOcUE
wb0RLR01HaDBkSEE2THk5a2MzTXVibTkzYVc1aExteDFMM0JyYVMxbVlXTjBiM0o1TDJOeWJDOXliMjkwTFdOa
ExtTnliREJNQmdnckJnRUZCUWNCQVFSQU1ENHdQQVlJS3dZQkJRVUhNQUtHTUdoMGRIQTZMeTlrYzNNdWJtOTN
hVzVoTG14MUwzQnJhUzFtWVdOMGIzSjVMMk55ZEM5eWIyOTBMV05oTG1OeWREQWRCZ05WSFE0RUZnUVUrMnRGc
XBOZTNHMjNZUjh5cUJaSWlWV1MzdlV3RHdZRFZSMFRBUUgvQkFVd0F3RUIvekFOQmdrcWhraUc5dzBCQVFzRkF
BT0NBUUVBRStOdWQwNVhHT002RkVaSFdUYzgrYm16LzZCMFhRWE41NjRLV0JCaGNoOWk1R2FkanFwU3NldmtuK
3RlTHE1bTZDTG8zZTRsWDJkSjdoc1BBdnlhTHFPSXB6ZzQ5VEdkaWIxbk9CMk83NCt5QWhUOHY5Rlp0SDFFQ0h
YeFlzdXlTR0lLdmQrTDVJakpUaXMzbGw0dlU4Rkh6eVJsTTlJUW53WlI1MDZqRmNKZUdsT2d5WmgrVUxXblJOR
UV3cU44RFRGMkQwWG9nWUJzckN4Q0JqMFBwYUpGcnV2RVFxcFV1dVlnMTRSMURKRmFoTHdxVllTT0Q1Z1BobUE
wSFI0ejNHRjNqSFN6MGk5alhTVE9zVWNka3ZVSnkwdElPbnVqc1VFa2czSDZXZzNsejhUdzNJYzdWMU5IYitNQ
zVLNFp2WCs1U1l5dTArZXI3YkZzY0lyWVp3PT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT48L2RzOlg1MDlEYXRhPjw
vZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWRlcz0ia
HR0cDovL3VyaS5ldHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9IiNpZC1hZmRlNzgyNDM2NDY4ZGQ3NGV
lYjE4MWY3Y2UxMTBlMSI+PHhhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzLWlkLWFmZGU3ODI0MzY0N
jhkZDc0ZWViMTgxZjdjZTExMGUxIj48eGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZXM6U2l
nbmluZ1RpbWU+MjAxNy0wOS0yOFQxMTowOTowNFo8L3hhZGVzOlNpZ25pbmdUaW1lPjx4YWRlczpTaWduaW5nQ
2VydGlmaWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlnZXN0TWV0aG9kIEFsZ29
yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI3NoYTEiLz48ZHM6RGlnZXN0VmFsdWU+Y
ytWb2hnMGpJY1o0VVFTV2VnbENnMG9HTldzPTwvZHM6RGlnZXN0VmFsdWU+PC94YWRlczpDZXJ0RGlnZXN0Pjx
4YWRlczpJc3N1ZXJTZXJpYWxWMj5NRll3VWFSUE1FMHhFREFPQmdOVkJBTU1CMmR2YjJRdFkyRXhHVEFYQmdOV
kJBb01FRTV2ZDJsdVlTQlRiMngxZEdsdmJuTXhFVEFQQmdOVkJBc01DRkJMU1MxVVJWTlVNUXN3Q1FZRFZRUUd
Fd0pNVlFJQkNnPT08L3hhZGVzOklzc3VlclNlcmlhbFYyPjwveGFkZXM6Q2VydD48L3hhZGVzOlNpZ25pbmdDZ
XJ0aWZpY2F0ZVYyPjwveGFkZXM6U2lnbmF0dXJlUHJvcGVydGllcz48eGFkZXM6U2lnbmVkRGF0YU9
iamVjdFByb3BlcnRpZXM+PHhhZGVzOkRhdGFPYmplY3RGb3JtYXQgT2JqZWN0UmVmZXJlbmNlPSIjci1pZC0xI
j48eGFkZXM6TWltZVR5cGU+dGV4dC94bWw8L3hhZGVzOk1pbWVUeXBlPjwveGFkZXM6RGF0YU9iamVjdEZvcm1
hdD48L3hhZGVzOlNpZ25lZERhdGFPYmplY3RQcm9wZXJ0aWVzPjwveGFkZXM6U2lnbmVkUHJvcGVydGllcz48L
3hhZGVzOlF1YWxpZnlpbmdQcm9wZXJ0aWVzPjwvZHM6T2JqZWN0PjwvZHM6U2lnbmF0dXJlPg==",
    "digestAlgorithm" : null,
```

```
    "name" : "xades-detached.xml"
  }
}
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2020 15:38:58 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" :
"PGRzOlNpZ25lZEluZm8geG1sbnM6ZHM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyMiPjxkc
zpDYW5vbmljYWxpemF0aW9uTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWw
tZXhjLWMxNG4jIj48L2RzOkNhbm9uaWNhbGl6YXRpb25NZXRob2Q+PGRzOlNpZ25hdHVyZU1ldGhvZCBBbGdvc
ml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiPjwvZHM6U2l
nbmF0dXJlTWV0aG9kPjxkczpSZWZlcmVuY2UgVHlwZT0iaHR0cDovL3VyaS5ldHNpLm9yZy8wMTkwMyNDb3Vud
GVyc2lnbmVkU2lnbmF0dXJlIiBVUkk9IiN2YWx1ZS1pZC1hZmRlNzgyNDM2NDY4ZGQ3NGVlYjE4MWY3Y2UxMTB
lMSI+PGRzOlRyYW5zZm9ybXM+PGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwM
DEvMTAveG1sLWV4Yy1jMTRuIyI+PC9kczpUcmFuc2Zvcm0+PC9kczpUcmFuc2Zvcm1zPjxkczpEaWdlc3RNZXR
ob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGVuYyNzaGEyNTYiPjwvZHM6RGlnZ
XN0TWV0aG9kPjxkczpEaWdlc3RWYWx1ZT51bTBWdTF1MEgyS0lpWkVjM1I2czEzQ3I1ZDdjZTZTVFZXb1BHdnd
GaTdjPTwvZHM6RGlnZXN0VmFsdWU+PC9kczpSZWZlcmVuY2U+PGRzOlJlZmVyZW5jZSBUeXBlPSJodHRwOi8vd
XJpLmV0c2kub3JnLzAxOTAzI1NpZ25lZFByb3BlcnRpZXMiIFVSST0iI3hhZGVzLWlkLWQyMGMxMzM0YmVjOGY
2ZGU1ODFhMWE4YTdmMzc5ZGE3Ij48ZHM6VHJhbnNmb3Jtcz48ZHM6VHJhbnNmb3JtIEFsZ29yaXRobT0iaHR0c
DovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIj48L2RzOlRyYW5zZm9ybT48L2RzOlRyYW5zZm9
ybXM+PGRzOkRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI
3NoYTI1NiI+PC9kczpEaWdlc3RNZXRob2Q+PGRzOkRpZ2VzdFZhbHVlPmVPc3NocTZXOEtUc0kvajRYdGVCNTh
mY1FCV3JHQWVqdlpVS1NMNWlDdkE9PC9kczpEaWdlc3RWYWx1ZT48L2RzOlJlZmVyZW5jZT48L2RzOlNpZ25lZ
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" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+XqO"
    },
    "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" : 1607960338417,
      "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" :
"M5UXygOOjF120YLI+JJGqBo+iytC9uhMxdpScE9tMVd5C6LN8Az4+DvKwSS48zBpc5/jbVz0F9t6RvfPPF7Lq
1myl30+j+bIcfJAORvBks02gm7yh/Fi4lioDPCL0fLLVPXNmByIgcvh4J8bQvO6qAxLv/8XTaXjy73WJjZDZi5
fBXKaUeXG1svq779ALr86PNL5fJJeWvqCs3BoAZTYMqc+v5Jr8u+lxyRz79A7B7NINtiIJo9fLCP0w/ew4jD5z
jCKMOkipS1CIF3xsKifCRd0rHfjPJMW2I9KWcb7MUjA9RnQGoYUC037deta8nkvBF6KNikH1zfJ9Jb+VF9/tQ=
="
  },
  "signatureDocument" : {
    "bytes" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48ZHM6U2lnbmF0dXJlIHhtbG5zOmRzPSJod
HRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIiBJZD0iaWQtYWZkZTc4MjQzNjQ2OGRkNzRlZWIxODF
mN2NlMTEwZTEiPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpemF0aW9uTWV0aG9kIEFsZ29yaXRobT0ia
HR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PGRzOlNpZ25hdHVyZU1ldGhvZCBBbGd
vcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiLz48ZHM6U
mVmZXJlbmNlIElkPSJyLWlkLTEiIFR5cGU9IiIgVVJJPSJzYW1wbGUueG1sIj48ZHM6RGlnZXN0TWV0aG9kIEF
sZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+PGRzOkRpZ2VzdFZhb
HVlPmtjREhPWmp3WmhWZnVEaHVoQ2VDRVJSbVlwVEg0Smo0Um1mVlZpMzFROWc9PC9kczpEaWdlc3RWYWx1ZT4
8L2RzOlJlZmVyZW5jZT48ZHM6UmVmZXJlbmNlIFR5cGU9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDE5MDMjU2lnb
mVkUHJvcGVydGllcyIgVVJJPSIjeGFkZXMtaWQtYWZkZTc4MjQzNjQ2OGRkNzRlZWIxODFmN2NlMTEwZTEiPjx
kczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzEwL
```

```
3htbC1leGMtYzE0biMiLz48L2RzOlRyYW5zZm9ybXM+PGRzOkRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA
6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPjxkczpEaWdlc3RWYWx1ZT5EenR3TlRtUm9PQ
W02L2xNSThSeW01eFpQekl2TFlEem4vZWJZWWtQc3I0PTwvZHM6RGlnZXN0VmFsdWU+PC9kczpSZWZlcmVuY2U
+PC9kczpTaWduZWRJbmZvPjxkczpTaWduYXR1cmVWYWx1ZSBJZD0idmFsdWUtaWQtYWZkZTc4MjQzNjQ2OGRkN
zRlZWIxODFmN2NlMTEwZTEiPllBN3NFTnQzTjh1ZkxGTW5LcjM2cjBQcXpNaVkzUTBzKytJR1RFVUMwc3BheFV
2MGRIWk0wZC95bjNrcExKTG9Va0k0TTNmbGo1V0duODNrZjA1QnFNMWtoc1g2MUdkemFGVFBHcG03YWtSUUtod
m9IMjV5eXFUWVhFU2xCY20wNGl6aUtoTE16WmpVZng0L0IxWkl5c3Y1cElCZ0oycjJvaTZqTG9wOXd3M2dlNGM
0WUpvYUsrU1hrNmh5VE5PY044UGpHZTYzV1lPVE5WUFFGdmphOEJud2crYTBiQnV3RCs4TjZmd2lnQ2RXNWEvN
ERKVWUvSjhNYjcwWkk4UG9Pem5HRGZpK1RQYmlJZVZtQ2JsNW1Vb1VnMlEveFlsdUpmTGgzdUdRQVhLQnZGNDV
vRElIUlZlZm5OL0QvV3l0QUNsVVZEb1FTeXdlbW5qUHBxRjhlZz09PC9kczpTaWduYXR1cmVWYWx1ZT48ZHM6S
2V5SW5mbz48ZHM6WDUwOURhdGE+PGRzOlg1MDlDZXJ0aWZpY2F0ZT5NSUlEMURDQ0FyeWdBd0lCQWdJQkNqQU5
CZ2txaGtpRzl3MEJBUXNGQURCTk1SQXdEZ1lEVlFRRERBZG5iMjlrTFdOaE1Sa3dGd1lEVlFRS0RCQk9iM2RwY
m1FZ1UyOXNkWFJwYjI1ek1SRXdEd1lEVlFRTERBaFFTMGt0VkVWVFZERUxNQWtHQTFVRUJoTUNURlV3SGhjTk1
UWXhNREkyTURjMU5ETXhXaGNOTVRnd09ESTJNRGMxTkRNeFdqQlBNUkl3RUFZRFZRUUREQWxuYjI5a0xYVnpaW
El4R1RBWEJnTlZCQW9NRUU1dmQybHVZU0JUYjJ4MWRHbHZibk14RVRBUEJnTlZCQXNNQ0ZCTFNTMVVSVk5VTVF
zd0NRWURWUVFHRXdKTVZUQ0NBU0l3RFFZSktvWklodmNOQVFFQkJRQURnZ0VQQURDQ0FRb0NnZ0VCQUxSQ1VJU
VpidzNuU2RMcCtCOWN6RUNncFpra1E1eFY0ZzlNLzd3bGc5N29DQ2Y3VUVoOUJBMWQrellqc3p2K0JKMWJKWlB
nYW4yMTQ0QXZnc29HSmZiNlVJeVZXNGdrbFVnSWwxYXJVdm9uK1drS25zZUZ1UU9mSnlqU0ZVREl3bnV2cDBoe
mNKWEhYUm1MZG15aCtuKzZOTUgwb201dFZvU2ZRcnRCVmlDTGVTTVZ6dUQ1RVBqMG1JUmN4OTFwTDM4ZTNGTlR
XN05hR1pMZWV6dUZ1Ui9xN3o5M2xMa3ZaNFZBTU5HR0x2SVhPWWVSQlpNeVBocEJaNEwzQThJM0VFbEtXSC8xT
HdpaVhUVFNHMXNNNld2TVRWYmYydmJkNDduWlJRQTJtU3BOR2pRb3VPQUVyZmVWVlVxeklDZ2hRQ0hSR09OdVN
MRy9IZnFGSGI0aldnMENBd0VBQWFPQnZEQ0J1VEFPQmdOVkhROEJBZjhFQkFNQ0JrQXdnWWNHQ0NzR0FRVUZCd
0VCQkhzd2VUQTVCZ2dyQmdFRkJRY3dBWVl0YUhSMGNEb3ZMMlJ6Y3k1dWIzZHBibUV1YkhVdmNHdHBMV1poWTN
SdmNua3ZiMk56Y0M5bmIyOWtMV05oTUR3R0NDc0dBUVVGQnpBQ2hqQm9kSFJ3T2k4dlpITnpMbTV2ZDJsdVlTN
XNkUzl3YTJrdFptRmpkRzl5ZVM5amNuUXZaMjl2WkMxallTNWpjblF3SFFZRFZSME9CQllFRk4ycEhELzdQZWZ
tQlQ4b1gyOVpoV3kvT0hKMU1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQkszVk9MaERJVldLb0ZycmhoV3phZ
GR0azZYUXRjd1JvTlBWU3NpL2dPcnpzZE03MEEzMXhJVHc3WWZMaHBvVkExeG83b3ZIbGRwTGxocXk5bzV3aDI
4MnlDcHFCVUF0Z3JTa0RHb2crSzdDTDZnVXBybFlpWnVHWnJ0ZzJYM2ZIUzJVc3g0WkozdElqNndWZWNERVVxS
VNGZkZUMkVzbTBRWFVuZ0lLRk1sOTVYZ210dzJ3eFhiT3pVZURkNERJUHJ2K21XNXBvQVdyNkl0c1YrSDJWUSt
aTC9rQm53V0hqU1RPYUdGaXNxWFkvYUgvMVB0QlhBKzE1K1lJV2VtSkJTdjNrRGFGek9YQUV0UjlaSThsWU9KY
XJuWTdBeS9hTjZiOXVHZmZyYm8vaFZBY0w0V0RkaGtiQk4zbTh3K2c3NkxvQVhOZUVldTA0QS8weExaelVCPC9
kczpYNTA5Q2VydGlmaWNhdGU+PGRzOlg1MDlDZXJ0aWZpY2F0ZT5NSUlENmpDQ0F0S2dBd0lCQWdJQkJEQU5CZ
2txaGtpRzl3MEJBUXNGQURCTk1SQXdEZ1lEVlFRRERBZHliMjkwTFdOaE1Sa3dGd1lEVlFRS0RCQk9iM2RwYm1
FZ1UyOXNkWFJwYjI1ek1SRXdEd1lEVlFRTERBaFFTMGt0VkVWVFZERUxNQWtHQTFVRUJoTUNURlV3SGhjTk1UW
XhNREkyTURjMU5ETXdXaGNOTVRnd09ESTJNRGMxTkRNd1dqQk5NUkF3RGdZRFZRUUREQWRuYjI5a0xXTmhNUmt
3RndZRFZRUUtEQkJPYjNkcGJtRWdVMjlzZFhScGIyNXpNUkV3RHdZRFZRUUxEQWhRUzBrdFZFVlRWREVMTUFrR
0ExVUVCaE1DVEZVd2dnRWlNQTBHQ1NxR1NJYjNEUUVCQVFVQUE0SUJEd0F3Z2dFS0FvSUJBUUNiYmw1c0tCQ2p
TQjhUTWRhY3lteC9XZk9qTVcxZ2lJalZKUlkyOEpiTldrQ1ZtdHpnbWl0Z2hmclBRUGx1ZXUwRFRhbGJEa3JTU
3loQ3Z6elBTR1B3Q0ZPYWhGL243aFFhMUYzVWFIU3hUS3JGQzVuT3dkTHp4S1JPM1dqVnRJR1JTWDJrdjFGZlV
wUXk1RXl5K3JzZlN6SjU5ZFU1WlpkV3BkYUR1RHhWVnlEZXIzRUl5Q2JHNy81SDlNRDRZdXpOcGVURldtTTZjV
VNUMDc5NlhEbGJFeFNUVEdRWEZKQTIrQ0NzeTlEWG5KYThuejBGRThmbWN2UUhlVTZrOVFicHpHak1kM0RXbEU
2bm83VWRDWUQxSDA0K3VzQnA1aGhDckFCNjcwTmRvVHJOVG1HTkFGdDRKVDB2aXRqS0hxOUtFSWQ2TGhkY20yV
Gc5M2REY1dGdEFnTUJBQUdqZ2RRd2dkRXdEZ1lEVlIwUEFRSC9CQVFEQWdlQU1FRUdBMVVkSHdRNk1EZ3dOcUE
wb0RLR01HaDBkSEE2THk5a2MzTXVibTkzYVc1aExteDFMM0JyYVMxbVlXTjBiM0o1TDJOeWJDOXliMjkwTFdOa
ExtTnliREJNQmdnckJnRUZCUWNCQVFSQU1ENHdQQVlJS3dZQkJRVUhNQUtHTUdoMGRIQTZMeTlrYzNNdWJtOTN
hVzVoTG14MUwzQnJhUzFtWVdOMGIzSjVMMk55ZEM5eWIyOTBMV05oTG1OeWREQWRCZ05WSFE0RUZnUVUrMnRGc
XBOZTNHMjNZUjh5cUJaSWlWV1MzdlV3RHdZRFZSMFRBUUgvQkFVd0F3RUIvekFOQmdrcWhraUc5dzBCQVFzRkF
BT0NBUUVBRStOdWQwNVhHT002RkVaSFdUYzgrYm16LzZCMFhRWE41NjRLV0JCaGNoOWk1R2FkanFwU3NldmtuK
3RlTHE1bTZDTG8zZTRsWDJkSjdoc1BBdnlhTHFPSXB6ZzQ5VEdkaWIxbk9CMk83NCt5QWhUOHY5Rlp0SDFFQ0h
YeFlzdXlTR0lLdmQrTDVJakpUaXMzbGw0dlU4Rkh6eVJsTTlJUW53WlI1MDZqRmNKZUdsT2d5WmgrVUxXblJOR
UV3cU44RFRGMkQwWG9nWUJzckN4Q0JqMFBwYUpGcnV2RVFxcFV1dVlnMTRSMURKRmFoTHdxVllTT0Q1Z1BobUE
```

```
wSFI0ejNHRjNqSFN6MGk5alhTVE9zVWNka3ZVSnkwdElPbnVqc1VFa2czSDZXZzNsejhUdzNJYzdWMU5IYitNQ
zVLNFp2WCs1U1l5dTArZXI3YkZzY0lyWVp3PT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT48L2RzOlg1MDlEYXRhPjw
vZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWRlcz0ia
HR0cDovL3VyaS5ldHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9IiNpZC1hZmRlNzgyNDM2NDY4ZGQ3NGV
lYjE4MWY3Y2UxMTBlMSI+PHhhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzLWlkLWFmZGU3ODI0MzY0N
jhkZDc0ZWViMTgxZjdjZTExMGUxIj48eGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZXM6U2l
nbmluZ1RpbWU+MjAxNy0wOS0yOFQxMTowOTowNFo8L3hhZGVzOlNpZ25pbmdUaW1lPjx4YWRlczpTaWduaW5nQ
2VydGlmaWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlnZXN0TWV0aG9kIEFsZ29
yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI3NoYTEiLz48ZHM6RGlnZXN0VmFsdWU+Y
ytWb2hnMGpJY1o0VVFTV2VnbENnMG9HTldzPTwvZHM6RGlnZXN0VmFsdWU+PC94YWRlczpDZXJ0RGlnZXN0Pjx
4YWRlczpJc3N1ZXJTZXJpYWxWMj5NRll3VWFSUE1FMHhFREFPQmdOVkJBTU1CMmR2YjJRdFkyRXhHVEFYQmdOV
kJBb01FRTV2ZDJsdVlTQlRiMngxZEdsdmJuTXhFVEFQQmdOVkJBc01DRkJMU1MxVVJWTlVNUXN3Q1FZRFZRUUd
Fd0pNVlFJQkNnPT08L3hhZGVzOklzc3VlclNlcmlhbFYyPjwveGFkZXM6Q2VydD48L3hhZGVzOlNpZ25pbmdDZ
XJ0aWZpY2F0ZVYyPjwveGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZXM6U2lnbmVkRGF0YU9
iamVjdFByb3BlcnRpZXM+PHhhZGVzOkRhdGFPYmplY3RGb3JtYXQgT2JqZWN0UmVmZXJlbmNlPSIjci1pZC0xI
j48eGFkZXM6TWltZVR5cGU+dGV4dC94bWw8L3hhZGVzOk1pbWVUeXBlPjweGFkZXM6RGF0YU9iamVjdEZvcm1
hdD48L3hhZGVzOlNpZ25lZERhdGFPYmplY3RQcm9wZXJ0aWVzPjweGFkZXM6U2lnbmVkUHJvcGVydGllcz48L
3hhZGVzOlF1YWxpZnlpbmdQcm9wZXJ0aWVzPjwvZHM6T2JqZWN0PjwvZHM6U2lnbmF0dXJlPg==",
    "digestAlgorithm" : null,
    "name" : "xades-detached.xml"
  }
}
```


Response

```
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2020 15:38:58 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" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9Im5vIj8+PGRzOlNpZ25hd
HVyZSB4bWxuczpkcz0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnIyIgSWQ9ImlkLWFmZGU3ODI
0MzY0NjhkZDc0ZWViMTgxZjdjZTExMGUxIj48ZHM6U2lnbmVkSW5mbz48ZHM6Q2Fub25pY2FsaXphdGlvbk1ld
GhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxkczpTaWd
uYXR1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNyc
2Etc2hhMjU2Ii8+PGRzOlJlZmVyZW5jZSBJZD0ici1pZC0xIiBUeXBlPSIiIFVSST0ic2FtcGxlLnhtbCI+PGR
zOkRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1N
iIvPjxkczpzaWdlcaWdlc3RWYWx1ZT5rY0RIT1pqd1poVmZ1RGh1aENlQ0VSUm1ZcFRINEpqNFJtZlZWaTMxUTlnPTw
vZHM6RGlnZXN0VmFsdWU+PC9kczpSZWZlcmVuY2U+PGRzOlJlZmVyZW5jZSBUeXBlPSJodHRwOi8vdXJpLmV0c
```

```
2kub3JnLzAxOTAzI1NpZ25lZFByb3BlcnRpZXMiIFVSST0iI3hhZGVzLWlkLWFmZGU3ODI0MzY0NjhkZDc0ZWV
iMTgxZjdjZTExMGUxIj48ZHM6VHJhbnNmb3Jtcz48ZHM6VHJhbnNmb3JtIEFsZ29yaXRobT0iaHR0cDovL3d3d
y53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PC9kczpUcmFuc2Zvcm1zPjxkczpEaWdlc3RNZXRob2Q
gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGVuYyNzaGEyNTYiLz48ZHM6RGlnZXN0V
mFsdWU+RHp0d05UbVJvT0FtNi9sTUk4UnltNXhaUHpJdkxZRHpuL2ViWVlrUHNyND08L2RzOkRpZ2VzdFZhbHV
lPjwvZHM6UmVmZXJlbmNlPjwvZHM6U2lnbmVkSW5mbz48ZHM6U2lnbmF0dXJlVmFsdWUgSWQ9InZhbHVlLWlkL
WFmZGU3ODI0MzY0NjhkZDc0ZWViMTgxZjdjZTExMGUxIj5ZQTdzRU50M044dWZMRk1uS3IzNnIwUHF6TWlZM1E
wcysrSUdURVVDMHNwYXhVdjBkSFpNMGQveW4za3BMSkxvVWtJNE0zZmxqNVdHbjgza2YwNUJxTTFraHNYNjFHS
nphRlRQR3BtN2FrUlFLaHZvSDI1eXlxVFlYRVNsQmNtMDRpemllaExNelpqVWZ4NC9CMVpJeXN2NXBJQmdKMnI
yb2k2akxvcDl3dzNnZTRjNFlKb2FLK1NYazZoeVROT2NOOFBqR2U2M1dZT1ROVlBRRnZqYThCbndnK2EwYkJ1d
0QrOE42ZndpZ0NkVzVhLzRESlVlL0o4TWI3MFpJOFBvT3puR0RmaStUUGJpSWVWbUNibDVtVW9VZzJRL3hZbHV
KZkxoM3VHUUFYS0J2RjQ1b0RJSFJWZWZuTi9EL1d5dEFDbFVWRG9RU3l3ZW1ualBwcUY4ZWc9PTwvZHM6U2lnb
mF0dXJlVmFsdWU+PGRzOktleUluZm8+PGRzOlg1MDlEYXRhPjxkczpYNTA5Q2VydGlmaWNhdGU+TUlJRDFEQ0N
BcnlnQXdJQkFnSUJDakFOQmdrcWhraUc5dzBCQVFzRkFEQk5NUkF3RGdZRFZRUUREQWRuYjI5a0xXTmhNUmt3R
ndZRFZRUUtEQkJPYjNkcGJtRWdVMjlzZFhScGIyNXpNUkV3RHdZRFZRUUxEQWhRUzBrdFZFVlRWREVMTUFrR0E
xVUVCaE1DVEZVd0hoY05NVFl4TURJMk1EYzFORE14V2hjTk1UZ3dPREkyTURjMU5ETXhXakJQTVJJd0VBWURWU
VFEREFsbmIyOWtMWFZ6WlhJeEdUQVhCZ05WQkFvTUVFNXZkMmx1WVNCVGIyeDFkR2x2Ym5NeEVUQVBCZ05WQkF
zTUNGQkxTUzFVUlZOVU1Rc3dDUVlEVlFRR0V3Sk1WVENDQVNJd0RRWUpLb1pJaHZjTkFRRUJCUUFEZ2dFUEFEQ
0NBUW9DZ2dFQkFMUkNVSVFaYnczblNkTHArQjljekVDZ3Baa2tRNXhWNGc5TS83d2xnOTdvQ0NmN1VFaDlCQTF
kK3pZanN6ditCSjFiSlpQZ2FuMjE0NEF2Z3NvR0pmYjZVSXlWVzRna2xVZ0lsMWFyVXZvbitXa0tuc2VGdVFPZ
kp5alNGVURJd251dnAwaHpjSlhIWFJtTGRteWgrbis2Tk1IMG9tNXRWb1NmUXJ0QlZpQ0xlU01WenVENUVQajB
tSVJjeDkxcEwzOGUzRk5UVzdOYUdaTGVlenVGdVIvcTd6OTNsTGt2WjRWQU1OR0dMdklYT1llUkJaTXlQaHBCW
jRMM0E4STNFRWxLV0gvMUx3aWlYVFRTRzFzTTZXdk1UVmJmMnZiZDQ3blpSUEybVNwTkdqUW91T0FFcmZlVlZ
VcXpJQ2doUUNIUkdPTnVTTEcvSGZxRkhiNGpXZzBDQXdFQUFhT0J2RENCdVRBT0JnTlZIUThCQWY4RUJBTUNCa
0F3Z1ljR0NDc0dBUVVGQndFQkJIc3dlVEE1QmdnckJnRUZCUWN3QVlZdGFIUjBjRG92TDJSemN5NXViM2RwYm1
FdWJIVXZjR3RwTFdaaFkzUnZjbmt2YjJOemNDOW5iMjlrTFdOaE1Ed0dDQ3NHQVFVRkJ6QUNoakJvZEhSd09pO
HZaSE56TG01dmQybHVZUzVzZFM5d2Eya3RabUZqZEc5eWVTOWpjblF2WjI5dlpDMWpZUzVqY25Rd0hRWURWUjB
PQkJZRUZOMnBIRC83UGVmbUJUOG9YMjlaaFd5L09ISjFNQTBHQ1NxR1NJYjNEUUVCQ3dVQUE0SUJBUUJLM1ZPT
GhESVZXS29GcnJoaFd6YWRkdGs2WFF0Y3dSb05QVlNzaS9nT3J6c2RNNzBBMzF4SVR3N1lmTGhwb1ZBMXhvN29
2SGxkcExsaHF5OW81d2gyODJ5Q3BxQlVBdGdyU2tER29nK0s3Q0w2Z1VwcmxZaVp1R1pydGcyWDNmSFMyVXN4N
FpKM3RJajZ3VmVjREVVcUlTRmZGVDJFc20wUVhVbmdJS0ZNbDk1WGdtdHcyd3hYYk96VWVEZDRESVBydittVzV
wb0FXcjZJdHNWK0gyVlErWkwva0Jud1dIalNUT2FHRmlzcVhZL2FILzFQdEJYQSsxNStZSVdlbUpCU3Yza0RhR
npPWEFFdFI5Wkk4bFlPSmFyblk3QXkvYU42Yjl1R2ZmcmJvL2hWQWNMNFdEZGhrYkJOM204dytnNzZMb0FYTmV
FZXUwNEEvMHhMWnpVQjwvZHM6WDUwOUNlcnRpZmljYXRlPjxkczpYNTA5Q2VydGlmaWNhdGU+TUlJRDZqQ0NBd
EtnQXdJQkFnSUJCREFOQmdrcWhraUc5dzBCQVFzRkFEQk5NUkF3RGdZRFZRUUREQWR5YjI5MExXTmhNUmt3Rnd
ZRFZRUUtEQkJPYjNkcGJtRWdVMjlzZFhScGIyNXpNUkV3RHdZRFZRUUxEQWhRUzBrdFZFVlRWREVMTUFrR0ExV
UVCaE1DVEZVd0hoY05NVFl4TURJMk1EYzFORE13V2hjTk1UZ3dPREkyTURjMU5ETXdXakJOTVJBd0RnWURWUVF
EREFkbmIyOWtMV05oTVJrd0Z3WURWUVFLREJCT2IzZHBibUVnVTI5c2RYUnBiMjV6TVJFd0R3WURWUVFMREFoU
VMwa3RWRVZUVkRFTE1Ba0dBMVVFQmhNQ1RGVXdnZ0VpTUEwR0NTcUdTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUt
Bb0lCQVFDYmJsNXNLQkNqU0I4VE1kYWN5bXgvV2ZPak1XMWdpSWpWSlJZMjhKYk5Xa0NWbXR6Z21pdGdoZnJQU
VBsdWV1MERUYWxiRGtyU1N5aEN2enpQU0dQd0NGT2FoRi9uN2hRYTFGM1VhSFN4VEtyRkM1bk93ZEx6eEtSTzN
XalZ0SUdSU1gya3YxRmZVcFF5NUV5eStyc2ZTeko1OWRVNVpaZFdwZGFEdUR4VlZ5RGVyM0VJeUNiRzcvNUg5T
UQ0WXV6TnBlVEZXbU02Y1VTVDA3OTZYRGxiRXhTVFRHUVhGSkEyK0NDc3k5RFhuSmE4bnowRkU4Zm1jdlFIZVU
2azlRYnB6R2pNZDNEV2xFNm5vN1VkQ1lEMUgwNCt1c0JwNWhoQ3JBQjY3ME5kb1RyTlRtR05BRnQ0SlQwdml0a
ktIcTlLRUlkNkxoZGNtMlRnOTNkRGNXRnRBZ01CQUFHamdkUXdnZEV3RGdZRFZSMFBBUUgvQkFRREFnZUFNRUV
HQTFVZEh3UTZNRGd3TnFBMG9ES0dNR2gwZEhBNkx5OWtjM011Ym05M2FXNWhMbXgxTDNCcmFTMW1ZV04wYjNKN
UwyTnliQzl5YjI5MExXTmhMbU55YkRCTUJnZ3JCZ0VGQlFjQkFRUkFNRDR3UEFZSUt3WUJCUVVITUFLR01HaDB
kSEE2THk5a2MzTXVibTkzYVc1aExteDFMM0JyYVMxbVlXTjBiM0o1TDJOeWRDOXliMjkwTFdOaExtTnlkREFkQ
mdOVkhRNEVGZ1FVKzJ0RnFwTmUzRzIzWVI4eXFCWklpVldTM3ZVd0R3WURWUjBUQVFIL0JBVXdBd0VCL3pBTkJ
na3Foa2lHOXcwQkFRc0ZBQU9DQVFFQUUrTnVkMDVYR09NNkZFWkhXVGM4K2Jtei82QjBYUVhONTY0S1dCQmhja
DlpNUdhZGpxcFNzZXZrbit0ZUxxNW02Q0xvM2U0bFgyZEo3aHNQQXZ5YUxxT0lwemc0OVRHZGliMW5PQjJPNzQ
```

```
reUFoVDh2OUZadEgxRUNIWHhZc3V5U0dJS3ZkK0w1SWpKVGlzM2xsNHZVOEZIenlSbE05SVFud1pSNTA2akZjS
mVHbE9neVpoK1VMV25STkVFd3FOOERURjJEMFhvZ1lCc3JDeENCajBQcGFKRnJ1dkVRcXBVdXVZZzE0UjFESkZ
haEx3cVZZU09ENWdQaG1BMEhSNHozR0YzakhTejBpOWpYU1RPc1VjZGt2VUp5MHRJT251anNVRWtnM0g2V2czb
Ho4VHczSWM3VjFOSGIrTUM1SzRadlgrNVNZeXUwK2VyN2JGc2NJclladz09PC9kczpYNTA5Q2VydGlmaWNhdGU
+PC9kczpYNTA5RGF0YT48L2RzOktleUluZm8+PGRzOk9iamVjdD48eGFkZXM6UXVhbGlmeWluZ1Byb3BlcnRpZ
XMgeG1sbnM6eGFkZXM9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDE5MDMvdjEuMy4yIyIgVGFyZ2V0PSIjaWQtYWZ
kZTc4MjQzNjQ2OGRkNzRlZWIxODFmN2NlMTEwZTEiPjx4YWRlczpTaWduZWRQcm9wZXJ0aWVzIElkPSJ4YWRlc
y1pZC1hZmRlNzgyNDM2NDY4ZGQ3NGVlYjE4MWY3Y2UxMTBlMSI+PHhhZGVzOlNpZ25lZFNpZ25hdHVyZVByb3B
lcnRpZXM+PHhhZGVzOlNpZ25pbmdUaW1lPjIwMTctMDktMjhUMTE6MDk6MDRaPC94YWRlczpTaWduaW5nVGltZ
T48eGFkZXM6U2lnbmluZ0NlcnRpZmljYXRlVjI+PHhhZGVzOkNlcnQ+PHhhZGVzOkNlcnREaWdlc3Q+PGRzOkR
pZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyNzaGExIi8+PGR
pZ2VzdFZhbHVlPmMrVm9oZzBqSWNaNFVRU1dlZ2xDZzBvR05Xcz08L2RzOkRpZ2VzdFZhbHVlPjwveGFkZXM6Q
2VydERpZ2VzdD48eGFkZXM6SXNzdWVyU2VyaWFsVjI+TUZZd1VhUlBNRTB4RURBT0JnTlZCQU1NQjJkdmIyUXRZMkV4R1RBWEJnTlZCQW9NRUU1dmQybHVZU0JUYjJ4MWRHbHZibk14RVRBUEJnTlZCQXNNQ0ZCTFNTMVVSVk5VTVFzd0NRWURWUVFHRXdKTVZRSUJDZz09PC94YWRlczpJc3N1ZXJTZXJpYWxWMj48L3hhZGVzOkNlcnQ+PC94YWRlczpTaWduaW5nQ2VydGlmaWNhdGVWMj48L3hhZGVzOlNpZ25lZFNpZ25hdHVyZVByb3BlcnRpZXM+PHhhZGVzOlNpZ25lZERhdGFPYmplY3RQcm9wZXJ0aWVzPjx4YWRlczpEYXRhT2JqZWN0Rm9ybWF0IE9iamVjdFJlZmVyZW5jZT0iI3ItaWQtMSI+PHhhZGVzOk1pbWVUeXBlPnRleHQveG1sPC94YWRlczpNaW1lVHlwZT48L3hhZGVzOkRhdGFPYmplY3RQcm9wZXJ0aWVzPjx4YWRlczpTaWduYXR1cmVQcm9wZXJ0aWVzIHhtbG5zOmRzPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIiBUYXJnZXQ9IiNpZCIgVHlwZT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PGRzOlNpZ25hdHVyZSB1ZD0iaWQtZDIwYzEzMzRiZWM4ZjZkZTU4MWExYThhN2YzNzlkYTciPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpemF0aW9uTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PGRzOlNpZ25hdHVyZSB1ZD0iaWQtZDIwYzEzMzRiZWM4ZjZkZTU4MWExYThhN2YzNzlkYTciPjxkczpDYW5vbmljYWxpemF0aW9uTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PGRzOlNpZ25hdHVyZSB1ZD0iaWQtZDIwYzEzMzRiZWM4ZjZkZTU4MWExYThhN2YzNzlkYTciPjxkczpTaWduZWRJbmZv
```

```
VhXOXZ6NjRlVGJkZTR2UUo2cGpIR2FySmYxZ1FORWMyWHpobUkvcHJYTHlzV05xQzdsWmc3UFVaVVRyZGVnQUJ
UVXpZQ1JKMWtXQlJQbTRxbzBMTjQwNWM5NFFRZDQ1YTVrVGdvd0h6RWdMbkFRSTI4eDBNM0E1OVRLQytpZU5jN
lZGMVBzVExwVXc3UE5JMlZzdFg1akF1YXNDQXdFQUFhTVNNQkF3RGdZRFZSMFBBUUgvQkFRREFnRUdNQTBHQ1N
xR1NJYjNEUUVCQ3dVQUE0SUJBUUNLNkxHQTAxVFIrcm1VOHA2eWhBaTRPa0ROMmIxZGJJTDhsOGlDTVlvcExDe
Hg4eHFxM3ViWkNPeHFoMVgyajZwZ1d6YXJiMGIvTVVpeDAwSW9Vdk5iRk94QVc3UEJaSUtETG5tNkxzY2tSeHM
xVTMyc0M5ZDFMT0hlM1dLQk5CNkdaQUxUMWV3amg3aFNiV2pmdGxtY292cSs2ZVZHQTVjdmYydS8yK1RrS2t5S
FYvTlIzOTRuWHJkc2Rwdnlnd3lwRXRYamV0ekQ3VVQ5M051dzN4Y1Y4VklmdEl2SGY5TGpVN2grVWpHbUtYRzl
jMTVlWXIzU3pVbXY2a3lPSTBCdncxNFBXdHNXR2wwUWRPU1J2SUJCclA0YWRDbkdUZ2pnams5TFRjTzhCOEZLc
nIrOGxIR3VjMGJwNGxJVVRvaVVrR0lMWHNpRWVFZzlXQXFtK1hxTzwvZHM6WDUwOUNlcnRpZmljYXRlPjwvZHM
6WDUwOURhdGE+PC9kczpLZXlJbmZvPjxkczpPYmplY3Q+PHhhZGVzOlF1YWxpZnlpbmdQcm9wZXJ0aWVzIFRhc
mdldD0iI2lkLWQyMGMxMzM0YmVjOGY2ZGU1ODFhMWE4YTdmMzc5ZGE3Ij48eGFkZXM6U2lnbmVkUHJvcGVydGl
lcyBJZD0ieGFkZXMtaWQtZDIwYzEzMzRiZWM4ZjZkZTU4MWExYThhN2YzNzlkYTciPjx4YWRlczpTaWduZWRTa
WduYXR1cmVQcm9wZXJ0aWVzPjx4YWRlczpTaWduaW5nVGltZT4yMDIwLTEyLTE0VDE1OjM4OjU4WjwveGFkZXM
6U2lnbmluZ1RpbWU+PHhhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZVYyPjx4YWRlczpDZXJ0Pjx4YWRlczpDZXJ0R
GlnZXN0PjxkczpEaWdlc3RNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGV
uYyNzaGE1MTIiLz48ZHM6RGlnZXN0VmFsdWU+MTR3TWpERnpmc3FrZFpVc25QSDAvaCtabzhyejhERXdpTXEyW
U8xd05UZnBsTDN3a1E3RTBsMnlaUFlkZXFHSzlTeENUbHpwMTFSTkVVZUxLTXNDZVE9PTwvZHM6RGlnZXN0VmF
sdWU+PC94YWRlczpDZXJ0RGlnZXN0Pjx4YWRlczpJc3N1ZXJTZXJpYWxWMj5NRDR3TktReU1EQXhHekFaQmdOV
kJBTU1FbEp2YjNSVFpXeG1VMmxuYm1Wa1JtRnJaVEVSTUE4R0ExVUVDZ3dJUkZOVExYUmxjM1FDQmk3V0ZOZTd
Wdz09PC94YWRlczpJc3N1ZXJTZXJpYWxWMj48L3hhZGVzOkNlcnQ+PC94YWRlczpTaWduaW5nQ2VydGlmaWNhd
GVWMj48L3hhZGVzOlNpZ25lZFNpZ25hdHVyZVByb3BlcnRpZXM+PC94YWRlczpTaWduZWRQcm9wZXJ0aWVzPjw
veGFkZXM6UXVhbGlmeWluZ1Byb3BlcnRpZXM+PC9kczpPYmplY3Q+PC9kczpTaWduYXR1cmU+PC94YWRlczpDb
3VudGVyU2lnbmF0dXJlPjwveGFkZXM6VW5zaWduZWRTaWduYXR1cmVQcm9wZXJ0aWVzPjwveGFkZXM6VW5zaWd
uZWRQcm9wZXJ0aWVzPjwveGFkZXM6UXVhbGlmeWluZ1Byb3BlcnRpZXM+PC9kczpPYmplY3Q+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: Mon, 14 Dec 2020 15:38:59 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" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+XqO"
  },
  "certificateChain" : [ {
    "encodedCertificate" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+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: Mon, 14 Dec 2020 15:38:58 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" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+XqO"
  },
  "certificateChain" : [ {
    "encodedCertificate" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+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: 392

{
  "bytes" :
"MYIBETAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb3DQEJBTEPFw0yMDEyMTQxNTM4NTlaMC0GC
SqGSIb3DQEJNDEgMB4wDQYJYIZIAWUDBAIBBQChDQYJKoZIhvcNAQELBQAwLwYJKoZIhvcNAQkEMSIEIJHAxzm
Y8GYVX7g4boQnghEUZmKUx+CY+EZn1VYt9UPYMHcGCyqGSIb3DQEJEAIvMWgwZjBkMGIEIALz68oBYydCU7yAn
SdJjdQbsDFtfmsGaWARXeFVWJ2cMD4wNKQyMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkRmFrZTERMA8GA1U
ECgwIRFNTLXRlc3QCBi7WFNe7Vw=="
}
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2020 15:38:59 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" :
"Fw/ulQqfxRgcEmZI6yTNDSOe7hebRuUPoJetZLmjcD+YZOcbYwsbqGtr9p2jnteVnZssNMLz4IlNkqaE4Y1B0
hpgDYeST5dylxxlgntRYuADvuZroUmPmkMDxWwJusqxYVa/o9LRHJup8VcSVeqUCZZ8JuFtkTLYUiYvGsj5Bqy
PijOBS8PGuxycmLq23ijB7mR9kA8wGuP4F4P69T8sgYwk00mudn+zFRVwTAm0Ao8ChPCKJrp7r0zFsCJ2EI14q
QYk1xzIbTZq5Z/QS4bHB4YKX0hVieZO/Mw4hi4dw6wguf1BneKiJwwMe4Y0wbc2MNL63NRQ7cYcWqcjOki32g=
="
}
```

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" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48ZHM6U2lnbmF0dXJlIHhtbG5zOmRzPSJod
HRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIiBJZD0iaWQtYWZkZTc4MjQzNjQ2OGRkNzRlZWIxODF
mN2NlMTEwZTEiPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpemF0aW9uTWV0aG9kIEFsZ29yaXRobT0ia
HR0cDovL3d3dy53My5vcmcvMjAwMS8xMC94bWwtZXhjLWMxNG4jIi8+PGRzOlNpZ25hdHVyZU1ldGhvZCBBbGd
vcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZHNpZy1tb3JlI3JzYS1zaGEyNTYiLz48Z6HM6U
mVmZXJlbmNlIElkPSJyLWlkLTEiIFR5cGU9IiIgVVJJPSJzYW1wbGUueG1sIj48ZHM6RGlnZXN0TWV0aG9kIEF
sZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+PGRzOkRpZ2VzdFZhb
HVlPmtjREhPWmp3WmhWZnVEaHVoQ2VDRVJSbVlwVEg0Smo0Um1mVlZpMzFROWc9PC9kczpEaWdlc3RWYWx1ZT4
8L2RzOlJlZmVyZW5jZT48ZHM6UmVmZXJlbmNlIFR5cGU9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDE5MDMjU2lnb
mVkUHJvcGVydGllcyIgVVJJPSIjeGFkZXMtaWQtYWZkZTc4MjQzNjQ2OGRkNzRlZWIxODFmN2NlMTEwZTEiPjx
kczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzEwL
3htbC1leGMtYzE0biMiLz48L2RzOlRyYW5zZm9ybXM+PGRzOkRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA
6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPjxkczpEaWdlc3RWYWx1ZT5EenR3TlRtUm9PQ
W02L2xNSThSeW01eFpQekl2TFlEem4vZWJZWWtQc3I0PTwvZHM6RGlnZXN0VmFsdWU+PC9kczpSZWZlcmVuY2U
+PC9kczpTaWduZWRJbmZvPjxkczpTaWduYXR1cmVWYWx1ZSBJZD0idmFsdWUtaWQtYWZkZTc4MjQzNjQ2OGRkN
zRlZWIxODFmN2NlMTEwZTEiPllBN3NFTnQzTjh1ZkxGTW5LcjM2cjBQcXpNaVkzUTBzKytJR1RFVUMwc3BheFV
2MGRIWk0wZC95bjNrcExxKTG9Va0k0TTNmbGo1V0duODNrZjA1QnFNMWtoc1g2MUdKemFGVFBHcG03YWtSUUtod
m9IMjV5eXFUWVhFU2xCY20wNGl6aUtoTE16WmpVZng0L0IxWkl5c3Y1cElCZ0oycjJvaTZqTG9wOXd3M2dlNGM
0WUpvYUsrU1hrNmh5VE5PY044UGpHZTYzV1lPVE5WUFFGdmphOEJud2crYTBiQnV3RCs4TjZmd2lnQ2RXNWEvN
ERKVWUvSjhNYjcwWkk4UG9Pem5HRGZpK1RQYmlJZVZtQ2JsNW1Vb1VnMlEveFlsdUpmTGgzdUdRQVhLQnZGNDV
vRElUlZlZm5OL0QvV3l0QUNsVVZEb1FTeXdlbW5qUHBxRjhlZz09PC9kczpTaWduYXR1cmVWYWx1ZT48ZHM6S
2V5SW5mbz48ZHM6WDUwOURhdGE+PGRzOlg1MDlDZXJ0aWZpY2F0ZT5NSUlEMURDQ0FyeWdBd0lCQWdJQkNqQU5
CZ2txaGtpRzl3MEJBUXNGQURCTk1SQXdEZ1lEVlFRRERBZG5iMjlrTFdOaE1Sa3dGd1lEVlFRS0RCQk9iM2RwY
m1FZ1UyOXNkWFJwYjI1ek1SRXdEd1lEVlFRTERBaFFTMGt0VkVWVFZERUxNQWtHQTFVRUJoTUNURlV3SGhjTk1
UWXhNREkyTURjMU5ETXhXaGNOTVRnd09ESTJNRGMxTkRNeFdqQlBNUkl3RUFZRFZRUUREQWxuYjI5a0xYVnpaW
El4R1RBWEJnTlZCQW9NRUU1dmQybHVZU0JUYjJ4MWRHbHZibk14RVRBUEJnTlZCQXNNQ0ZCTFNTMVVSVk5VTVF
zd0NRWURWUVFHRXdKTVZUQ0NBU0l3RFFZSktvWklodmNOQVFFQkJRQURnZ0VQQURDQ0FRb0NnZ0VCQUxSQ1VJU
VpidzNuU2RMcCtCOWN6RUNncFpra1E1eFY0ZzlNLzd3bGc5N29DQ2Y3VUVoOUJBMWQrellqc3p2K0JKMWJKWlB
nYW4yMTQ0QXZnc29HSmZiNlVJeVZXNGdrbFVnSWwxYXJVdm9uK1drS25zZUZ1UU9mSnlqU0ZVREl3bnV2cDBoe
mNKWEhYUm1MZG15aCtuKzZOTUgwb201dFZvU2ZRcnRCVmlDTGVTTVZ6dUQ1RVBqMG1JUmN4OTFwTDM4ZTNGTlR
XN05hR1pMZWV6dUZ1Ui9xN3o5M2xMa3ZaNFZBTU5HR0x2SVhPWWVSQlpNeVBocEJaNEwzQThJM0VFbEtXSC8xT
HdpaVhUVFNHMXNNNld2TVRWYmYydmJkNDduWlJRQTJtU3BOR2pRb3VPQUVyZmVWVlVxeklDZ2hRQ0hSR09OdVN
```

```
MRy9IZnFGSGI0aldnMENBd0VBQWFPQnZEQ0J1VEFPQmdOVkhROEJBZjhFQkFNQ0JrQXdnWWNHQ0NzR0FRVUZCd
0VCQkhzd2VUQTVCZ2dyQmdFRkJRY3dBWVl0YUhSMGNEb3ZMMlJ6Y3k1dWIzZHBibUV1YkhVdmNHdHBMV1poWTN
SdmNua3ZiMk56Y0M5bmIyOWtMV05oTUR3R0NDc0dBUVVGQnpBQ2hqQm9kSFJ3T2k4dlpITnpMbTV2ZDJsdVlTN
XNkUzl3YTJrdFptRmpkRzl5ZVM5amNuUXZaMjl2WkMxallTNWpjblF3SFFZRFZSME9CQllFRk4ycEhELzdQZWZ
tQlQ4b1gyOVpoV3kvT0hKMU1BMEdDU3FHU0liM0RRRUJDd1VBQTRJQkFRQkszVk9MaERJVldLb0ZycmhoV3phZ
GR0azZYUXRjd1JvTlBWU3NpL2dPcnpzZE03MEEzMXhJVHc3WWZMaHBvVkExeG83b3ZIbGRwTGxocXk5bzV3aDI
4MnlDcHFCVUF0Z3JTa0RHb2crSzdDTDZnVXBybFlpWnVHWnJ0ZzJYM2ZIUzJVc3g0WkozdElqNndWZWNERVVxS
VNGZkZUMkVzbTBRWFVuZ0lLRk1sOTVYZ210dzJ3eFhiT3pVZURkNERJUHJ2K21XNXBvQVdyNkl0c1YrSDJWUSt
aTC9rQm53V0hqU1RPYUdGaXNxWFkvYUgvMVB0QlhBKzE1K1lJV2VtSkJTdjNrRGFGek9YQUV0UjlaSThsWU9KY
XJuWTdBeS9hTjZiOXVHZmZyYm8vaFZBY0w0V0RkaGtiQk4zbTh3K2c3NkxvQVhOZUVldTA0QS8weExaelVCPC9
kczpYNTA5Q2VydGlmaWNhdGU+PGRzOlg1MDlDZXJ0aWZpY2F0ZT5NSUlENmpDQ0F0S2dBd0lCQWdJQkJEQU5CZ
2txaGtpRzl3MEJBUXNGQURCTk1SQXdEZ1lEVlFRRERBZHliMjkwTFdOaE1Sa3dGd1lEVlFRS0RCQk9iM2RwYm1
FZ1UyOXNkWFJwYjI1ek1SRXdEd1lEVlFRTERBaFFTMGt0VkVWVFZERUxNQWtHQTFVRUJoTUNURlV3SGhjTk1UW
XhNREkyTURjMU5ETXdXaGNOTVRnd09ESTJNRGMxTkRNd1dqQk5NUkF3RGdZRFZRUUREQWRuYjI5a0xXTmhNUmt
3RndZRFZRUUtEQkJPYjNkcGJtRWdVMjlzZFhScGIyNXpNUkV3RHdZRFZRUUxEQWhRUzBrdFZFVlRWREVMTUFrR
0ExVUVCaE1DVEZVd2dnRWlNQTBHQ1NxR1NJYjNEUUVCQVFVQUE0SUJEd0F3Z2dFS0FvSUJBUUNiYmw1c0tCQ2p
TQjhUTWRhY3lteC9XZk9qTVcxZ2lJalZKUlkyOEpiTldrQ1ZtdHpnbWl0Z2hmclBRUGx1ZXUwRFRhbGJEa3JTU
3loQ3Z6elBTR1B3Q0ZPYWhGL243aFFhMUYzVWFIU3hUS3JGQzVuT3dkTHp4S1JPM1dqVnRJR1JTWDJrdjFGZlV
wUXk1RXl5K3JzZlN6SjU5ZFU1WlpkV3BkYUR1RHhWVnlEZXIzRUl5Q2JHNy81SDlNRDRZdXpOcGVURldtTTZjV
VNUMDc5NlhEbGJFeFNUVEdRWEZKQTIrQ0NzeTlEWG5KYThuejBGRThmbWN2UUhlVTZrOVFicHpHak1kM0RXbEU
2bm83VWRDWUQxSDA0K3VzQnA1aGhDckFCNjcwTmRvVHJOVG1HTkFGdDRKVDB2aXRqS0hxOUtFSWQ2TGhkY20yV
Gc5M2REY1dGdEFnTUJBQUdqZ2RRd2dkRXdEZ1lEVlIwUEFRSC9CQVFEQWdlQU1FRUdBMVVkSHdRNk1EZ3dOcUE
wb0RLR01HaDBkSEE2THk5a2MzTXVibTkzYVc1aExteDFMM0JyYVMxbVlXTjBiM0o1TDJOeWJDOXliMjkwTFdOa
ExtTnliREJNQmdnckJnRUZCUWNCQVFSQU1ENHdQQVlJS3dZQkJRVUhNQUtHTUdoMGRIQTZMeTlrYzNNdWJtOTN
hVzVoTG14MUwzQnJhUzFtWVdOMGIzSjVMMk55ZEM5eWIyOTBMV05oTG1OeWREQWRCZ05WSFE0RUZnUVUrMnRGc
XBOZTNHMjNZUjh5cUJaSWlWV1MzdlV3RHdZRFZSMFRBUUgvQkFVd0F3RUIvekFOQmdrcWhraUc5dzBCQVFzRkF
BT0NBUUVBRStOdWQwNVhHT002RkVaSFdUYzgrYm16LzZCMFhRWE41NjRLV0JCaGNoOWk1R2FkanFwU3NldmtuK
3RlTHE1bTZDTG8zZTRsWDJkSjdoc1BBdnlhTHFPSXB6ZzQ5VEdkaWIxbk9CMk83NCt5QWhUOHY5Rlp0SDFFQ0h
YeFlzdXlTR0lLdmQrTDVJakpUaXMzbGw0dlU4Rkh6eVJsTTlJUW53WlI1MDZqRmNKZUdsT2d5WmgrVUxXblJOR
UV3cU44RFRGMkQwWG9nWUJzckN4Q0JqMFBwYUpGcnV2RVFxcFV1dVlnMTRSMURKRmFoTHdxVllTT0Q1Z1BobUE
wSFI0ejNHRjNqSFN6MGk5alhTVE9zVWNka3ZVSnkwdElPbnVqc1VFa2czSDZXZzNsejhUdzNJYzdWMU5IYitNQ
zVLNFp2WCs1U1l5dTArZXI3YkZzY0lyWVp3PT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT48L2RzOlg1MDlEYXRhPjw
vZHM6S2V5SW5mbz48ZHM6T2JqZWN0Pjx4YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWRlcz0ia
HR0cDovL3VyaS5ldHNpLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9IiNpZC1hZmRlNzgyNDM2NDY4ZGQ3NGV
lYjE4MWY3Y2UxMTBlMSI+PHhhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzLWlkLWFmZGU3ODI0MzY0N
jhkZDc0ZWViMTgxZjdjZTExMGUxIj48eGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZXM6U2l
nbmluZ1RpbWU+MjAxNy0wOS0yOFQxMTowOTowNFo8L3hhZGVzOlNpZ25pbmdUaW1lPjx4YWRlczpTaWduaW5nQ
2VydGlmaWNhdGVWMj48eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlnZXN0TWV0aG9kIEFsZ29
yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI3NoYTEiLz48ZHM6RGlnZXN0VmFsdWU+Y
ytWb2hnMGpJY1o0VVFTV2VnbENnMG9HTldzPTwvZHM6RGlnZXN0VmFsdWU+PC94YWRlczpDZXJ0RGlnZXN0Pjx
4YWRlczpJc3N1ZXJTZXJpYWxWMj5NRll3VWFSUE1FMHhFREFPQmdOVkJBTU1CMmR2YjJRdFkyRXhHVEFYQmdOV
kJBb01FRTV2ZDJsdVlTQlRiMngxZEdsdmJuTXhFVEFQQmdOVkJBc01DRkJMU1MxVVJWTlVNUXN3Q1FZRFZRUUd
Fd0pNVlFJQkNnPT08L3hhZGVzOklzc3VlclNlcmlhbFYyPjwveGFkZXM6Q2VydD48L3hhZGVzOlNpZ25pbmdDZ
XJ0aWZpY2F0ZVYyPjwveGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZXM6U2lnbmVkRGF0YU9
iamVjdFByb3BlcnRpZXM+PHhhZGVzOkRhdGFPYmplY3RGb3JtYXQgT2JqZWN0UmVmZXJlbmNlPSIjci1pZC0xI
j48eGFkZXM6TWltZVR5cGU+dGV4dC94bWw8L3hhZGVzOk1pbWVUeXBlPjwveGFkZXM6RGF0YU9iamVjdEZvcm1
hdD48L3hhZGVzOlNpZ25lZERhdGFPYmplY3RQcm9wZXJ0aWVzPjwveGFkZXM6U2lnbmVkUHJvcGVydGllcz48L
3hhZGVzOlF1YWxpZnlpbmdQcm9wZXJ0aWVzPjwvZHM6T2JqZWN0PjwvZHM6U2lnbmF0dXJlPg==",
    "digestAlgorithm" : null,
    "name" : "xades-detached.xml"
  },
```

```
  "originalDocuments" : [ {
    "bytes" :
"77u/PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4NCjxoOnRhYmxlIHhtbG5zOmg9Imh0d
HA6Ly93d3cudzMub3JnL1RSL2h0bWw0LyI+DQoJPGg6dHI+DQoJCTxoOnRkPkhlbGxvPC9oOnRkPg0KCQk8aDp
0ZD5Xb3JsZDwvaDp0ZD4NCgk8L2g6dHI+DQo8L2g6dGFibGU+",
    "digestAlgorithm" : null,
    "name" : "sample.xml"
  } ],
  "policy" : null,
  "tokenExtractionStrategy" : "NONE",
  "signatureId" : null
}
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2020 15:38:58 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: 45915

{
  "validationReportDataHandler" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiIHN0YW5kYWxvbmU9InllcyI/Pgo8VmFsaWRhd
GlvblJlcG9ydCB4bWxucz0iaHR0cDovL3VyaS5ldHNpLm9yZy8xOTEwMi92MS4yLjEjIiB4bWxuczpuczI9Imh
0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyMiIHhtbG5zOm5zMz0iaHR0cDovL3VyaS5ldHNpLm9yZ
y8wMTkwMy92MS4zLjIjIiB4bWxuczpuczQ9Imh0dHA6Ly91cmkuZXRzaS5vcmcvMDIyMzEvdjIjIj4KICAgIDx
TaWduYXR1cmVWYWxpZGF0aW9uUmVwb3J0PgogICAgICAgIDxTaWduYXR1cmVJZGVudGlmaWVyIGlkPSJTLUEyN
DFCQ0ExNzI3MUQ2MDNCMzI3MTBFNTAzNTM2OEZBRkJDNThGNjlDQ0M2NkE0MzJFQzczOEM5QjRDMjc1M0EiPgo
gICAgICAgICAgICA8RGlnZXN0QWxnQW5kVmFsdWU+CiAgICAgICAgICAgICAgICA8bnMyOkRpZ2VzdE1ldGhvZ
CBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPgogICAgICAgICA
gICAgICAgPG5zMjpEaWdlc3RWYWx1ZT4rakxLZXZ2MDhaTWdZL2ZSbnZjN0RneTlaSnBGUktmM01ZTWUrK2t6O
FBNPTwvbnMyOkRpZ2VzdFZhbHVlPgogICAgICAgICAgICA8L0RpZ2VzdEFsZ0FuZFZhbHVlPgogICAgICAgICA
gICA8bnMyOlNpZ25hdHVyZVZhbHVlPllBN3NFTnQzTjh1ZkxGTW5LcjM2cjBQcXpNaVkzUTBzKytJR1RFVUMwc
3BheFV2MGRIWk0wZC95bjNrcExKTG9Va0k0TTNmbGo1V0duODNrZjA1QnFNMWtoc1g2MUdKemFGVFBHcG03YWt
SUUtodm9IMjV5eXFUWVhFU2xCY20wNGl6aUtoTE16WmpVZng0L0IxWkl5c3Y1cElCZ0oycjJvaTZqTG9wOXd3M
2dlNGM0WUpvYUsrU1hrNmh5VE5PY044UGpHZTYzV1lPVE5WUFFGdmphOEJud2crYTBiQnV3RCs4TjZmd2lnQ2R
XNWEvNERKVWUvSjhNYjcwWkk4UG9Pem5HRGZpK1RQYmlJZVZtQ2JsNW1Vb1VnMlEveFlsdUpmTGgzdUdRQVhLQ
nZGNDVvRElIUlZlZm5OL0QvV3l0QUNsVVZEb1FTeXdlbW5qUHBxRjhlZz09PC9uczI6U2lnbmF0dXJlVmFsdWU
+CiAgICAgICAgICAgIDxIYXNoT25seT5mYWxzZTwvSGFzaE9ubHk+CiAgICAgICAgICAgIDxEb2NIYXNoT25se
T5mYWxzZTwvRG9jSGFzaE9ubHk+CiAgICAgICAgICAgIDxEQUlkZW50aWZpZXI+aWQtYWZkZTc4MjQzNjQ2OGR
kNzRlZWIxODFmN2NlMTEwZTE8L0RBSWRlbnRpZmllcj4KICAgICAgICAgICA8L1NpZ25hdHVyZUlkZW50aWZpZXI+C
```

```
iAgICAgICAgPFZhbGlkYXRpb25Db25zdHJhaW50c0V2YWx1YXRpb25SZXBvcnQ+CiAgICAgICAgICAgIDxWYWx
pZGF0aW9uQ29uc3RyYWludD4KICAgICAgICAgICAgICAgIDxWYWxpZGF0aW9uQ29uc3RyYWludElkZW50aWZpZ
XI+dXJuOmNlZjpkc3M6YmJiOmZvcm1hdENoZWNraW5nPC9WYWxpZGF0aW9uQ29uc3RyYWludElkZW50aWZpZXI
+CiAgICAgICAgICAgICAgICA8Q29uc3RyYWludFN0YXR1cz4KICAgICAgICAgICAgICAgICA8U3RhdHVzP
nVybjpldHNpOjAxOTEwMjpjb25zdHJhaW50U3RhdHVzOmFwcGxpZWQ8L1N0YXR1cz4KICAgICAgICAgICAgICA
gIDwvQ29uc3RyYWludFN0YXR1cz4KICAgICAgICAgICAgICAgIDxWYWxpZGF0aW9uU3RhdHVzPgogICAgICAgI
CAgICAgICAgICAgIDxNYWluSW5kaWNhdGlvbj51cm46ZXRzaTowMTkxMDI6bWFpbmluZGljYXRpb246cGFzc2V
kPC9NYWluSW5kaWNhdGlvbj4KICAgICAgICAgICAgICAgICAgIDwvVmFsaWRhdGlvblN0YXR1cz4KICAgICAgICA
gICAgPC9WYWxpZGF0aW9uQ29uc3RyYWludD4KICAgICAgICAgICAgPFZhbGlkYXRpb25Db25zdHJhaW50PgogICA
gICAgICAgICAgICA8VFpbkluZGljYXRpb24+dXJuOmV0c2k6MDE5MTAyOm1haW5pbmRpY2F0aW9uOnBhc3N
lZDwvTWFpbkluZGljYXRpb24+CiAgICAgICAgICAgICAgICA8L1ZhbGlkYXRpb25TdGF0dXM+CiAgICAgICAgICAgI
CAgIDwvVmFsaWRhdGlvbkNvbnN0cmFpbnQ+CiAgICAgICAgICAgIDxWYWxpZGF0aW9uQ29uc3RyYWludD4KICA
gICAgICAgICAgICAgICAgICAgIDxWYWxpZGF0aW9uQ29uc3RyYWludElkZW50aWZpZXI+dXJuOmNlZjpkc3M6YmJiOnZhb
GlkYXRpb25Db250ZXh0SW5pdGlhbGl6YXRpb248L1ZhbGlkYXRpb25Db25zdHJhaW50SWRlbnRpZmllcj4KICA
gICAgICAgICAgICAgICAgICAgIDxDb25zdHJhaW50U3RhdHVzPgogICAgICAgICAgICAgICAgICAgICAgICA8TWF
pbkluZGljYXRpb24+dXJuOmV0c2k6MDE5MTAyOm1haW5pbmRpY2F0aW9uOnBhc3N
lZDwvTWFpbkluZGljYXRpb24+CiAgICAgICAgICAgICAgICA8L1ZhbGlkYXRpb25TdGF0dXM+CiAgICAgICAgICAgI
CAgIDwvVmFsaWRhdGlvbkNvbnN0cmFpbnQ+CiAgICAgICAgICAgIDxWYWxpZGF0aW9uQ29uc3RyYWludD4KICA
gICAgICAgICAgICAgICAgICAgIDxWYWxpZGF0aW9uQ29uc3RyYWludElkZW50aWZpZXI+dXJuOmNlZjpkc3M6YmJiOnZhb
GlkYXRpb25Db250ZXh0SW5pdGlhbGl6YXRpb248L1ZhbGlkYXRpb25Db25zdHJhaW50SWRlbnRpZmllcj4KICA
gICAgICAgICAgICAgICAgICAgIDxDb25zdHJhaW50U3RhdHVzPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxNYW
5JbmRpY2F0aW9uPnVybjpldHNpOjAxOTEwMjptYWluaW5kaWNhdGlvbjpwYXNzZWQ8L01haW5JbmRpY2F0aW9uPgogICA
gICAgICAgICAgICAgICAgICAgICAgICA8L1ZhbGlkYXRpb25Db25zdHJhaW50PgogICAgICAgICAgICAgICAgICAgICAgICA8VmFsaWRhdGlvbkNvbnN0cmFpbnQ+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8VmFsaWRhdGlvbkNvbnN0cmFpbnRJZGVudGlmaWVyPnVybjpjZWY6ZHNzOmJiYjpjcnlwdG9nc
mFwaGljVmVyaWZpY2F0aW9uPC9WYWxpZGF0aW9uQ29uc3RyYWludElkZW50aWZpZXI+CiAgICAgICAgICAgICAgICAgICAgICAgICAgICA8Q29uc3RyYWludFN0YXR1cz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAg
```

uOmNlZjpkc3M6YmJiOnBhc3RTaWduYXR1cmVWYWxpZGF0aW9uPC9WYWxpZGF0aW9uQ29uc3RyYWludElkZW50aWZpZXI+CiAgICAgICAgICAgICAgICA8Q29uc3RyYWludFN0YXR1cz4KICAgICAgICAgICAgICAgICAgICA8U3RhdHVzPnVybjpldHNpOjAxOTEwMjpjb25zdHJhaW50U3RhdHVzOmRpc2FibGVkPC9TdGF0dXM+CiAgICAgICAgICAgICAgICA8L0NvbnN0cmFpbnRTdGF0dXM+CiAgICAgICAgICAgICAgICA8L0NvbnN0cmFpbnRTdGF0dXM+CiAgICAgICAgICAgICAgICA8L0NvbnN0cmFpbnQ+CiAgICAgICAgICAgICAgICA8Q29uc3RyYWludD4KICAgICAgICAgICAgICAgICAgICA8SWRlbnRpZmllcj4KICAgICAgICAgICAgICAgICAgICAgICAgPFZhbGlkYXRpb25Db25zdHJhaW50PgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPFZhbGlkYXRpb25Db25zdHJhaW50SWRlbnRpZmllcj51cm46Y2VmOmRzczpiYmI6dmFsaWRhdGlvblRpbWVTbGkaW5nPC9WYWxpZGF0aW9uQ29uc3RyYWludElkZW50aWZpZXI+CiAgICAgICAgICAgICAgICA8Q29uc3RyYWludFN0YXR1cz4KICAgICAgICAgICAgICAgICAgICA8U3RhdHVzPnVybjpldHNpOjAxOTEwMjpjb25zdHJhaW50PgogICAgICAgICAgICAgICAgICAgICAgICA8TWVzc2FibGVkPC9NZXNzYWJsZWQ+CiAgICAgICAgICAgICAgICAgICAgICAgIDxDb25zdHJhaW50PgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPFZhbGlkYXRpb25Db25zdHJhaW50SWRlbnRpZmllcj51cm46Y2VmOmRzczpiYmI6dmFsaWRhdGlvblRpbWVTbGkaW5nPC9WYWxpZGF0aW9uQ29uc3RyYWludElkZW50aWZpZXI+CiAgICAgICAgICAgICAgICA8Q29uc3RyYWludFN0YXR1cz4KICAgICAgICAgICAgICAgICAgICA8U3RhdHVzPnVybjpldHNpOjAxOTEwMjpjb25zdHJhaW50SWRlbnRpZmllcj51cm46Y2VmOmRzczpiYmI6dmFsaWRhdGlvblRpbWVTbGkaW5nPC9WYWxpZGF0aW9uQ29uc3RyYWludFN0YXR1cz4KICAgICAgICAgICAgICAgIDwvQ29uc3RyYWludD4KICAgICAgICAgICAgICAgIDxDb25zdHJhaW50PgogICAgICAgICAgICAgICAgICAgIDxJZGVudGlmaWVyPnVybjpldHNpOjAxOTEwMjpjb25zdHJhaW50SWRlbnRpZmllcjpkZWxlZ2F0aW9uPC9JZGVudGlmaWVyPgogICAgICAgICAgICAgICAgICAgIDxEZXNjcmlwdG9yPgogICAgICAgICAgICAgICAgICAgICAgICA8VmFsaWRhdGlvblRpbWVTbGkgeG1sbnM6dHM9InVybjpldHNpOjAxOTEwMjpjb25zdHJhaW50SWRlbnRpZmllcjpkZWxlZ2F0aW9uIiB0czpkZWxlZ2F0aW9uPSJ0cnVlIi8+CiAgICAgICAgICAgICAgICAgICAgPC9EZXNjcmlwdG9yPgogICAgICAgICAgICAgICAgPC9Db25zdHJhaW50PgogICAgICAgICAgICA8L0NvbnN0cmFpbnRzPgogICAgICAgICAgICA8U2lnbmF0dXJlIHhtbG5zPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWciIElkPSJpZC1kZWxlZ2F0aW9uLTIwMjAtMTItMTRUMTU6Mzg6NThaPC9QT0VUaW1lPgogICAgICAgICAgICAgICAgPFBFR5cGVPZlByb29mPnVybjpldHNpOjAxOTEwMjpwb2V0eXBlOnJlcGx5PC9QT0VUaW1lPgogICAgICAgICAgICAgICAgPFR5cGVPZlByb29mPnVybjpldHNpOjAxOTEwMjpwb2V0eXBlOnJlcGx5PC9UeXBlT2ZQcm9vZj4KICAgICAgICAgICAgICAgIDxTaWduYXR1cmVWYWxpZGF0aW9uRGF0YT4KICAgICAgICAgICAgICAgICAgICA8VHJhbnNmb3JtcyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxkc2lnIyI+CiAgICAgICAgICAgICAgICAgICAgICAgIDxUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzEwL3htbC1leGMtYzE0biMiLz4KICAgICAgICAgICAgICAgICAgICAgICAgPFRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPgogICAgICAgICAgICAgICAgICAgIDwvVHJhbnNmb3JtcyB4bWxucz0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxkc2lnIyI+CiAgICAgICAgICAgICAgICAgICAgICAgIDxEaWdlc3RBbGdBbmRWYWx1ZT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxEaWdlc3RBbGdvcml0aG0gQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGVuYyNzaGEyNTYiLz4KICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxEaWdlc3RWYWx1ZT5jREhPWmp3WmhWZnVEaHVoQ2VDRVJSbVlwVEg0Smo0Um1mVlZpMzFROWc9PC9EaWdlc3RWYWx1ZT4KICAgICAgICAgICAgICAgICAgICAgICAgPC9EaWdlc3RBbGdBbmRWYWx1ZT4KICAgICAgICAgICAgICAgICAgICA8L1NpZ25hdHVyZVZhbHVlPgogICAgICAgICAgICAgICAgICAgIDxLZXlJbmZvIHhtbG5zPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWciIElkPSJpZC1kZWxlZ2F0aW9uLTIwMjAtMTItMTRUMTU6Mzg6NThaIj4KICAgICAgICAgICAgICAgICAgICAgICAgPFg1MDlEYXRhPgogICAgICAgICAgICAgICAgICAgICAgICAgICAgPFg1MDlDZXJ0aWZpY2F0ZT5NSUlGTVRDQ0JEaWdBd0lCQWdJVEh3QUFBQUFBQUFBQ

```
0NlcnRpZmljYXRlIFZPUmVmZXJlbmNlPSJDLUYwRkYwQjQ1MTREMzE2MzA0RjI4MTdEQkEwQkZCMDVERURCOTg
1MjdDMEU0N0M3M0U4RDhGREZFMTZERjI2N0UiLz4KICAgICAgICAgICAgICAgICA8SW50ZXJtZWRpYXRlQ
2VydGlmaWNhdGUgVk9SZWZlcmVuY2U9IkMtNkYzNURFMzk2NUI5QTY5QkMzNjYxRDFBMzU1QjBBRTYwOTA3QUR
CNzQxQ0MxOTExRUZEMEYzQkU3MkQ2QTZFOSIvPgogICAgICAgICAgICAgICAgPC9DZXJ0aWZpY2F0ZUNoYWluP
gogICAgICAgICAgICAgICAgPENyeXB0b0luZm9ybWF0aW9uPgogICAgICAgICAgICAgICAgICAgIDxWYWxpZGF
0aW9uT2JqZWN0SWQgVk9SZWZlcmVuY2U9IlMtQTI0MUJDQTE3MjcxRDYwM0IzMjcxMEU1MDM1MzY4RkFGQkM1O
EY2OUNDQzY2QTQzMkVDNzM4QzlCNEMyNzUzQSIvPgogICAgICAgICAgICAgICAgICAgIDxBbGdvcml0aG0+aHR
0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxkc2lnLW1vcmUjcnNhLXNoYTI1NjwvQWxnb3JpdGhtPgogICAgI
CAgICAgICAgICAgICAgICAgIDxTZWN1cmVBbGdvcml0aG0+dHJ1ZTwvU2VjdXJlQWxnb3JpdGhtPgogICAgICAgICA
gICAgICAgICAgICAgICAgICAgIDxOb3RBZnRlcj4yMDIyLTEyLTMxVDIzOjAwOjAwWjwvTm90QWZ0ZXI+CiAgICAgICAgICAgI
CAgICA8L0NyeXB0b0luZm9ybWF0aW9uPgogICAgICAgICAgICA8L0Fzc29jaWF0ZWRWYWxpZGF0aW9uUmVwb3J
0RGF0YT4KICAgICAgICA8L1NpZ25hdHVyZVZhbGlkYXRpb25TdGF0dXM+CiAgICA8L1NpZ25hdHVyZVZhbGlkY
XRpb25SZXBvcnQ+CiAgICA8U2lnbmF0dXJlVmFsaWRhdGlvbk9iamVjdHM+CiAgICAgICAgICAgPFZhbGlkYXRpb25
PYmplY3QgaWQ9IkMtNkYzNURFMzk2NUI5QTY5QkMzNjYxRDFBMzU1QjBBRTYwOTA3QURCNzQxQ0MxOTExRUZEM
EYzQkU3MkQ2QTZFOSI+CiAgICAgICAgICAgICAgICAgICAgIDxPYmplY3RUeXBlPnVybjpldHNpOjAxOTEwMjp2YWxpZGF0aW9
uT2JqZWN0OmNlcnRpZmljYXRlPC9PYmplY3RUeXBlPgogICAgICAgICAgICA8VmFsaWRhdGlvbk9iamVjdFJlc
HJlc2VudGF0aW9uPgogICAgICAgICAgICAgICAgICAgIDxERpZ2VzdEFsZ0FuZFZhbHVlPgogICAgICAgICAgICAgICAgICAgICA
gICAgIDxuczI6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlb
mMjc2hhMjU2Ii8+CiAgICAgICAgICAgICAgICAgICAgICAgPG5zMjpEaWdlc3RWYWx1ZT5ielhlT1dXNXBwdkRaaDB
hTlZzSzVna0hyYmRCekJrUjc5RHp2bkxXcHVrPTwvbnMyOkRpZ2VzdFZhbHVlPgogICAgICAgICAgICAgICAgICAgICAgP
C9EaWdlc3RBbGdBbmRWYWx1ZT4KICAgICAgICAgICAgICAgICAgICAgPC9WYWxpZGF0aW9uT2JqZWN0UmVwcmVzZW50YXRpb24
+CiAgICAgICAgICAgICAgICAgICAgIDxQT0U+CiAgICAgICAgICAgICAgICA8UE9FVGltZT4yMDIwLTEyLTE0VDE1OjM4OjU4W
jwvUE9FVGltZT4KICAgICAgICAgICAgICAgICAgICAgIDxUeXBlT2ZQcm9vZj51cm46ZXRzaTowMTkxMDI6cG9ldHlwZTp
2YWxpZGF0aW9uPC9UeXBlT2ZQcm9vZj4KICAgICAgICAgICAgICAgICAgICAgPC9WYWxpZGF0aW9uT
2JqZWN0PgogICAgICAgICAgICAgICAgICAgIDxWYWxpZGF0aW9uT2JqZWN0IGlkPSJDLUYwRkYwQjQ1MTREMzE2MzA0RjI4MTdEQkE
wQkZCMDVERURCOTg1MjdDMEU0N0M3M0U4RDhGREZFMTZERjI2N0UiPgogICAgICAgICAgICAgICAgICAgICA8T2JqZWN0VHlwZ
T51cm46ZXRzaTowMTkxMDI6dmFsaWRhdGlvbk9iamVjdDpjZXJ0aWZpY2F0ZTwvT2JqZWN0VHlwZT4KICAgICAgICA
gICAgICAgICAgICAgICAgICAgICAgPFZhbGlkYXRpb25PYmplY3RSZXByZXNlbnRhdGlvbj4KICAgICAgICAgICAgICAgICAgICAgICAgIDxEaWdlc3RBb
GdBbmRWYWx1ZT4KICAgICAgICAgICAgICAgICAgICAgICA8bnMyOkRpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA
6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPgogICAgICAgICAgICAgICAgICAgICAgICAgIDxuczI6R
GlnZXN0VmFsdWU+OFA4TFJSVFRGakJQS0JmYm9MK3dYZTI1aFNmQTVIeHo2Tmo5L2hiZkpuND08L25zMjpEaWd
lc3RWYWx1ZT4KICAgICAgICAgICAgICAgICAgICAgICAgIDwvRGlnZXN0QWxnQW5kVmFsdWU+CiAgICAgICAgICAgICAgICAgICAgICAgIDwvVmFsa
WRhdGlvbk9iamVjdFJlcHJlc2VudGF0aW9uPgogICAgICAgICAgICA8UE9FPgogICAgICAgICAgICAgICAgICAgICA8VHlwZU9mUHJvb
2Y+dXJuOmV0c2k6MDE5MTAyOnBvZXR5cGU6dmFsaWRhdGlvbjwvVHlwZU9mUHJvb2Y+CiAgICAgICAgICAgICAgICAgICAgICAgIDw
vUE9FPgogICAgICAgICAgICAgICAgICAgIDwvVmFsaWRhdGlvbk9iamVjdD4KICAgICAgICAgICAgICAgICA8VmFsaWRhdGlvbk9iamVjdCBpZD0iR
C1DNThDODBBODA1MzBFMEYzNDlCQzMyREVGNTAyODBENzRDNEI3RUJGMjUyODA0NDAxODExNjdBMkYxQTBCMzF
EIj4KICAgICAgICAgICAgICAgICAgICAgPE9iamVjdFR5cGU+dXJuOmV0c2k6MDE5MTAyOnZhbGlkYXRpb25PYmplY3Q6c2lnb
mVkRGF0YTwvT2JqZWN0VHlwZT4KICAgICAgICAgICAgICAgICAgICAgPFZhbGlkYXRpb25PYmplY3RSZXByZXNlbnRhdGlvbj4
KICAgICAgICAgICAgICAgICAgICAgICAgIDxEaWdlc3RBbGdBbmRWYWx1ZT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8bnMyOkRpZ
2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMDQveG1sZW5jI3NoYTI1NiIvPgo
gICAgICAgICAgICAgICAgICAgICAgICAgICAgIDxuczI6RGlnZXN0VmFsdWU+a2NESE9aandaaFZmdURodWhDZUNFUlJtWXBUS
DRKajRSbWZWVmkzMVE5Zz08L25zMjpEaWdlc3RWYWx1ZT4KICAgICAgICAgICAgICAgICAgICAgICAgICAgICA8L0RpZ2VzdEFsZ
kVmFsdWU+CiAgICAgICAgICAgICAgICAgICAgIDwvVmFsaWRhdGlvbk9iamVjdFJlcHJlc2VudGF0aW9uPgogICAgICAgICAgICAgICAgICAgICAgI
CA8UE9FPgogICAgICAgICAgICAgICAgICAgICAgPFBBPRVRpbWU+MjAyMC0xMi0xNFQxNTozODo1OFo8L1BPRVRpbWU+CiA
gICAgICAgICAgICAgICAgICAgICAgICA8VHlwZU9mUHJvb2Y+dXJuOmV0c2k6MDE5MTAyOnBvZXR5cGU6dmFsaWRhdGlvbjwvV
HlwZU9mUHJvb2Y+CiAgICAgICAgICAgICAgICAgICAgIDwvVmFsaWRhdGlvbk9iamVjdD4KICAgICAgICAgICAgICAgICA8VmFsaWRhdGlvbk9iamVjdD4KICAgICAgICAgICAgICAgICAgICAgIDw
vU2lnbmF0dXJlVmFsaWRhdGlvbk9iamVjdHM+CjwvVmFsaWRhdGlvbk9iamVjdHM+CjwvRGlnaWRlbFJlcG9ydD4K",
  "DiagnosticData" : {
    "DocumentName" : "xades-detached.xml",
    "ValidationDate" : "2020-12-14T15:38:58",
```

```
    "ContainerInfo" : null,
    "Signature" : [ {
      "Id" : "S-A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
      "DAIdentifier" : "id-afde782436468dd74eeb181f7ce110e1",
      "SignatureFilename" : "xades-detached.xml",
      "ErrorMessage" : null,
      "ClaimedSigningTime" : "2017-09-28T11:09:04",
      "SignatureFormat" : "XAdES-BASELINE-B",
      "StructuralValidation" : {
        "Message" : [ ],
        "valid" : true
      },
      "DigestMatcher" : [ {
        "DataFound" : true,
        "DataIntact" : true,
        "DigestMethod" : "SHA256",
        "DigestValue" : "kcDHOZjwZhVfuDhuhCeCERRmYpTH4Jj4RmfVVi31Q9g=",
        "match" : null,
        "type" : "REFERENCE",
        "name" : "r-id-1",
        "duplicated" : null
      }, {
        "DataFound" : true,
        "DataIntact" : true,
        "DigestMethod" : "SHA256",
        "DigestValue" : "DztwNTmRoOAm6/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-
F0FF0B4514D316304F2817DBA0BFB05DEDB98527C0E47C73E8D8FDFE16DF267E"
      },
      "ChainItem" : [ {
        "Certificate" : "C-
F0FF0B4514D316304F2817DBA0BFB05DEDB98527C0E47C73E8D8FDFE16DF267E"
      }, {
        "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" :
"MFYwUaRPME0xEDAOBgNVBAMMB2dvb2QtY2ExGTAXBgNVBAoMEE5vd2luYSBTb2x1dGlvbnMxETAPBgNVBAsMC
FBLSS1URVNUMQswCQYDVQQGEwJMVQIBCg==",
              "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/yn3kpLJLoUkI4M3flj5WGn83kf05Bq
M1khsX61GJzaFTPGpm7akRQKhvoH25yyqTYXESlBcm04iziKhLMzZjUfx4/B1ZIysv5pIBgJ2r2oi6jLop9ww3
ge4c4YJoaK+SXk6hyTNOcN8PjGe63WYOTNVPQFvja8Bnwg+a0bBuwD+8N6fwigCdW5a/4DJUe/J8Mb70ZI8PoO
znGDfi+TPbiIeVmCbl5mUoUg2Q/xYluJfLh3uGQAXKBvF45oDIHRVefnN/D/WytAClUVDoQSywemnjPpqF8eg=
=",
      "CounterSignature" : null,
      "Parent" : null,
      "Duplicated" : null
    } ],
    "Certificate" : [ {
      "Id" : "C-6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9",
      "SubjectDistinguishedName" : [ {
        "value" : "c=lu,ou=pki-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=pki-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" : "2018-08-26T07:54:30",
      "NotBefore" : "2016-10-26T07:54:30",
      "PublicKeySize" : 2048,
      "PublicKeyEncryptionAlgo" : "RSA",
      "EntityKey" : "PK-
4B0F8D40371579A6397E2DE77E4F6E9F51E676B1B1C9A2C5FB043DAF56F42C50",
      "KeyUsage" : [ "digitalSignature" ],
      "extendedKeyUsagesOid" : [ ],
      "PSD2Info" : null,
      "IdPkixOcspNoCheck" : false,
      "BasicSignature" : {
        "EncryptionAlgoUsedToSignThisToken" : "RSA",
        "KeyLengthUsedToSignThisToken" : "?",
        "DigestAlgoUsedToSignThisToken" : "SHA256",
        "MaskGenerationFunctionUsedToSignThisToken" : null,
        "SignatureIntact" : false,
        "SignatureValid" : false
      },
      "SigningCertificate" : null,
      "ChainItem" : [ ],
      "Trusted" : false,
      "SelfSigned" : false,
      "certificatePolicy" : [ ],
      "qcStatementOid" : [ ],
      "qcTypeOid" : [ ],
      "QCLimitValue" : null,
      "SemanticsIdentifier" : null,
      "TrustedServiceProvider" : [ ],
      "CertificateRevocation" : [ ],
      "Base64Encoded" : null,
      "DigestAlgoAndValue" : {
        "DigestMethod" : "SHA256",
        "DigestValue" : "bzXeOWW5ppvDZh0aNVsK5gkHrbdBzBkR79DzvnLWpuk=",
        "match" : null
      }
    }, {
      "Id" : "C-F0FF0B4514D316304F2817DBA0BFB05DEDB98527C0E47C73E8D8FDFE16DF267E",
      "SubjectDistinguishedName" : [ {
        "value" : "c=lu,ou=pki-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=pki-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" : "2018-08-26T07:54:31",
      "NotBefore" : "2016-10-26T07:54:31",
      "PublicKeySize" : 2048,
      "PublicKeyEncryptionAlgo" : "RSA",
      "EntityKey" : "PK-
DF0FFC7C7CFE69DA1465D07B837F6A241D76761DBE89FF53E5C8B8EF26FDB8D4",
      "KeyUsage" : [ "nonRepudiation" ],
      "extendedKeyUsagesOid" : [ ],
      "PSD2Info" : null,
      "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" : [ ],
      "qcStatementOid" : [ ],
      "qcTypeOid" : [ ],
      "QCLimitValue" : null,
      "SemanticsIdentifier" : null,
      "TrustedServiceProvider" : [ ],
      "CertificateRevocation" : [ ],
      "Base64Encoded" : null,
      "DigestAlgoAndValue" : {
        "DigestMethod" : "SHA256",
        "DigestValue" : "8P8LRRTTFjBPKBfboL+wXe25hSfA5Hxz6Nj9/hbfJn4=",
        "match" : null
      }
    } ],
    "Revocation" : [ ],
    "Timestamp" : [ ],
    "OrphanTokens" : null,
    "SignerData" : [ {
      "Id" : "D-C58C80A80530E0F349BC32DEF50280D74C4B7EBF25280440181167A2F1A0B31D",
      "ReferencedName" : "sample.xml",
      "DigestAlgoAndValue" : {
        "DigestMethod" : "SHA256",
        "DigestValue" : "kcDHOZjwZhVfuDhuhCeCERRmYpTH4Jj4RmfVVi31Q9g=",
        "match" : 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\tsigner'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" : "2017-09-28T11:09:04",
```

```
        "BestSignatureTime" : "2020-12-14T15:38:58",
        "SignedBy" : "good-user",
        "SignatureLevel" : {
          "value" : "N/A",
          "description" : "Not applicable"
        },
        "SignatureScope" : [ {
          "value" : "Full document",
          "name" : "sample.xml",
          "scope" : "FULL"
        } ],
        "Filename" : null,
        "CertificateChain" : {
          "Certificate" : [ {
            "id" : "C-
F0FF0B4514D316304F2817DBA0BFB05DEDB98527C0E47C73E8D8FDFE16DF267E",
            "qualifiedName" : "good-user"
          }, {
            "id" : "C-
6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9",
            "qualifiedName" : "good-ca"
          } ]
        },
        "Indication" : "INDETERMINATE",
        "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
        "Errors" : [ "Unable to build a certificate chain until a trusted list!", "The
result of the LTV validation process is not acceptable to continue the process!", "The
certificate chain for signature is not trusted, it does not contain a trust anchor."
],
        "Warnings" : [ "The signature/seal is an INDETERMINATE AdES digital
signature!" ],
        "Infos" : [ ],
        "Id" : "S-A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
        "CounterSignature" : null,
        "ParentId" : null,
        "SignatureFormat" : "XAdES-BASELINE-B",
        "ExtensionPeriodMin" : null,
        "ExtensionPeriodMax" : null
      }
    } ],
    "Semantic" : null,
    "ValidationTime" : "2020-12-14T15:38:58"
  },
  "DetailedReport" : {
    "signatureOrTimestampOrCertificate" : [ {
      "Signature" : {
        "ValidationProcessBasicSignature" : {
          "Constraint" : [ {
            "Name" : {
              "value" : "Is the result of the Basic Validation Process conclusive?",
              "NameId" : "ADEST_ROBVPIIC"
```

```
            },
            "Status" : "NOT OK",
            "Error" : {
              "value" : "The result of the Basic validation process is not
conclusive!",
              "NameId" : "ADEST_ROBVPIIC_ANS"
            },
            "Warning" : null,
            "Info" : null,
            "AdditionalInfo" : null,
            "Id" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A"
          } ],
          "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.",
              "NameId" : "BBB_XCV_CCCBB_SIG_ANS"
            } ],
            "Warnings" : [ ],
            "Infos" : [ ]
          },
          "Title" : "Validation Process for Basic Signatures",
          "ProofOfExistence" : {
            "Time" : "2020-12-14T15:38:58",
            "TimestampId" : null
          }
        },
        "Timestamp" : [ ],
        "ValidationProcessLongTermData" : {
          "Constraint" : [ {
            "Name" : {
              "value" : "Is the result of the Basic Validation Process acceptable?",
              "NameId" : "LTV_ABSV"
            },
            "Status" : "NOT OK",
            "Error" : {
              "value" : "The result of the Basic validation process is not acceptable
to continue the process!",
              "NameId" : "LTV_ABSV_ANS"
            },
            "Warning" : null,
            "Info" : null,
            "AdditionalInfo" : null,
            "Id" : 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.",
              "NameId" : "BBB_XCV_CCCBB_SIG_ANS"
            } ],
            "Warnings" : [ ],
            "Infos" : [ ]
          },
          "Title" : "Validation Process for Signatures with Time and Signatures with
Long-Term Validation Data",
          "ProofOfExistence" : {
            "Time" : "2020-12-14T15:38:58",
            "TimestampId" : null
          }
        },
        "ValidationProcessArchivalData" : {
          "Constraint" : [ {
            "Name" : {
              "value" : "Is the result of the LTV validation process acceptable?",
              "NameId" : "ARCH_LTVV"
            },
            "Status" : "NOT OK",
            "Error" : {
              "value" : "The result of the LTV validation process is not acceptable to
continue the process!",
              "NameId" : "ARCH_LTVV_ANS"
            },
            "Warning" : null,
            "Info" : null,
            "AdditionalInfo" : null,
            "Id" : 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.",
              "NameId" : "BBB_XCV_CCCBB_SIG_ANS"
            } ],
            "Warnings" : [ ],
            "Infos" : [ ]
          },
          "Title" : "Validation Process for Signatures with Archival Data",
          "ProofOfExistence" : {
            "Time" : "2020-12-14T15:38:58",
            "TimestampId" : null
          }
        },
        "ValidationSignatureQualification" : {
          "ValidationCertificateQualification" : [ ],
```

```
          "Constraint" : [ {
            "Name" : {
              "value" : "Is the signature/seal an acceptable AdES digital signature
(ETSI EN 319 102-1)?",
              "NameId" : "QUAL_IS_ADES"
            },
            "Status" : "WARNING",
            "Error" : null,
            "Warning" : {
              "value" : "The signature/seal is an INDETERMINATE AdES digital
signature!",
              "NameId" : "QUAL_IS_ADES_IND"
            },
            "Info" : null,
            "AdditionalInfo" : null,
            "Id" : null
          }, {
            "Name" : {
              "value" : "Has a trusted list been reached for the certificate chain?",
              "NameId" : "QUAL_CERT_TRUSTED_LIST_REACHED"
            },
            "Status" : "NOT OK",
            "Error" : {
              "value" : "Unable to build a certificate chain until a trusted list!",
              "NameId" : "QUAL_CERT_TRUSTED_LIST_REACHED_ANS"
            },
            "Warning" : null,
            "Info" : null,
            "AdditionalInfo" : null,
            "Id" : null
          } ],
          "Conclusion" : {
            "Indication" : "FAILED",
            "SubIndication" : null,
            "Errors" : [ {
              "value" : "Unable to build a certificate chain until a trusted list!",
              "NameId" : "QUAL_CERT_TRUSTED_LIST_REACHED_ANS"
            } ],
            "Warnings" : [ {
              "value" : "The signature/seal is an INDETERMINATE AdES digital
signature!",
              "NameId" : "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?",
          "NameId" : "BBB_FC_IEFF"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null
      }, {
        "Name" : {
          "value" : "Is the signature identification not ambiguous?",
          "NameId" : "BBB_FC_ISD"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null
      }, {
        "Name" : {
          "value" : "Is the signed references identification not ambiguous?",
          "NameId" : "BBB_FC_ISRIA"
        },
        "Status" : "OK",
        "Error" : null,
        "Warning" : null,
        "Info" : null,
        "AdditionalInfo" : null,
        "Id" : null
      } ],
      "Conclusion" : {
        "Indication" : "PASSED",
        "SubIndication" : null,
        "Errors" : null,
        "Warnings" : [ ],
        "Infos" : [ ]
      },
```

```
        "Title" : "Format Checking"
      },
      "ISC" : {
        "CertificateChain" : {
          "ChainItem" : [ {
            "Source" : "SIGNATURE",
            "Id" : "C-
F0FF0B4514D316304F2817DBA0BFB05DEDB98527C0E47C73E8D8FDFE16DF267E"
          }, {
            "Source" : "SIGNATURE",
            "Id" : "C-
6F35DE3965B9A69BC3661D1A355B0AE60907ADB741CC1911EFD0F3BE72D6A6E9"
          } ]
        },
        "Constraint" : [ {
          "Name" : {
            "value" : "Is there an identified candidate for the signing certificate?",
            "NameId" : "BBB_ICS_ISCI"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        }, {
          "Name" : {
            "value" : "Is the signed attribute: 'signing-certificate' present?",
            "NameId" : "BBB_ICS_ISASCP"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        }, {
          "Name" : {
            "value" : "Is the signed attribute: 'signing-certificate' present only
once?",
            "NameId" : "BBB_ICS_ISASCPU"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        }, {
          "Name" : {
            "value" : "Is the signed attribute: 'cert-digest' of the certificate
```

```
present?",
            "NameId" : "BBB_ICS_ISACDP"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        }, {
          "Name" : {
            "value" : "Does the certificate digest value match a digest value found in
the certificate reference(s)?",
            "NameId" : "BBB_ICS_ICDVV"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        }, {
          "Name" : {
            "value" : "Are the issuer distinguished name and the serial number
equal?",
            "NameId" : "BBB_ICS_AIDNASNE"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : 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?",
            "NameId" : "BBB_VCI_ISPK"
          },
          "Status" : "OK",
          "Error" : null,
```

```
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : 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?",
            "NameId" : "BBB_XCV_CCCBB"
          },
          "Status" : "NOT OK",
          "Error" : {
            "value" : "The certificate chain for signature is not trusted, it does not
contain a trust anchor.",
            "NameId" : "BBB_XCV_CCCBB_SIG_ANS"
          },
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : 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.",
            "NameId" : "BBB_XCV_CCCBB_SIG_ANS"
          } ],
          "Warnings" : [ ],
          "Infos" : [ ]
        },
        "Title" : "X509 Certificate Validation"
      },
      "CV" : {
        "Constraint" : [ {
          "Name" : {
            "value" : "Has the reference data object been found?",
            "NameId" : "BBB_CV_IRDOF"
          },
```

```
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : "Reference : r-id-1",
          "Id" : null
        }, {
          "Name" : {
            "value" : "Is the reference data object intact?",
            "NameId" : "BBB_CV_IRDOI"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : "Reference : r-id-1",
          "Id" : null
        }, {
          "Name" : {
            "value" : "Has the reference data object been found?",
            "NameId" : "BBB_CV_IRDOF"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : "Reference : #xades-id-afde782436468dd74eeb181f7ce110e1",
          "Id" : null
        }, {
          "Name" : {
            "value" : "Is the reference data object intact?",
            "NameId" : "BBB_CV_IRDOI"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : "Reference : #xades-id-afde782436468dd74eeb181f7ce110e1",
          "Id" : null
        }, {
          "Name" : {
            "value" : "Is the signature intact?",
            "NameId" : "BBB_CV_ISI"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : "Id = S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
          "Id" : null
```

```
        } ],
        "Conclusion" : {
          "Indication" : "PASSED",
          "SubIndication" : null,
          "Errors" : null,
          "Warnings" : [ ],
          "Infos" : [ ]
        },
        "Title" : "Cryptographic Verification"
      },
      "SAV" : {
        "CryptographicInfo" : {
          "Algorithm" : "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256",
          "KeyLength" : "2048",
          "Secure" : true,
          "NotAfter" : "2022-12-31T23:00:00",
          "ConcernedMaterial" : "S-
A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A"
        },
        "Constraint" : [ {
          "Name" : {
            "value" : "Is the structure of the signature valid?",
            "NameId" : "BBB_SAV_ISSV"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        }, {
          "Name" : {
            "value" : "Does Signing Certificate attribute contains references only to
the certificate chain?",
            "NameId" : "BBB_SAV_DSCACRCC"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        }, {
          "Name" : {
            "value" : "Is the signed qualifying property: 'signing-time' present?",
            "NameId" : "BBB_SAV_ISQPSTP"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
```

```
          "AdditionalInfo" : null,
          "Id" : null
        }, {
          "Name" : {
            "value" : "Is the signed qualifying property: 'message-digest' or
'SignedProperties' present?",
            "NameId" : "BBB_SAV_ISQPMDOSPP"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        }, {
          "Name" : {
            "value" : "Are cryptographic constraints met for the signature creation?",
            "NameId" : "ACCM"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : "Validation time : 2020-12-14 15:38",
          "Id" : null
        }, {
          "Name" : {
            "value" : "Are cryptographic constraints met for the object reference?",
            "NameId" : "ACCM"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : "Validation time : 2020-12-14 15:38 for object reference
with name : [r-id-1]",
          "Id" : null
        }, {
          "Name" : {
            "value" : "Are cryptographic constraints met for the signed properties?",
            "NameId" : "ACCM"
          },
          "Status" : "OK",
          "Error" : null,
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : "Validation time : 2020-12-14 15:38 for signed properties
with name : [#xades-id-afde782436468dd74eeb181f7ce110e1]",
          "Id" : null
        } ],
        "Conclusion" : {
```

```
          "Indication" : "PASSED",
          "SubIndication" : null,
          "Errors" : null,
          "Warnings" : [ ],
          "Infos" : [ ]
        },
        "Title" : "Signature Acceptance Validation",
        "ValidationTime" : "2020-12-14T15:38:58"
      },
      "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.",
          "NameId" : "BBB_XCV_CCCBB_SIG_ANS"
        } ],
        "Warnings" : null,
        "Infos" : null
      },
      "Id" : "S-A241BCA17271D603B32710E5035368FAFBC58F69CCC66A432EC738C9B4C2753A",
      "Type" : "SIGNATURE"
    } ],
    "TLAnalysis" : [ ],
    "Semantic" : null,
    "ValidationTime" : "2020-12-14T15:38:58"
  }
}
```


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" :
"PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48ZHM6U2lnbmF0dXJlIHhtbG5zOmRzPSJod
HRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIiBJZD0iaWQtZWExMGEwNTE3Y2JjN2Y1NDllYTNlNjg
1ODY3YWM5NWUiPjxkczpTaWduZWRJbmZvPjxkczpDYW5vbmljYWxpemF0aW9uTWV0aG9kIEFsZ29yaXRobT0ia
HR0cDovL3d3dy53My5vcmcvVFIvMjAwMS9SRUMteG1sLWMxNG4tMjAwMTAzMTUiLz48ZHM6U2lnbmF0dXJlTWV
0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxkc2lnLW1vcmUjcnNhLXNoYTI1N
iIvPjxkczpSZWZlcmVuY2UgSWQ9InItaWQtMSIgVHlwZT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWx
kc2lnI09iamVjdCIgVVJJPSIjby1pZC0xIj48ZHM6VHJhbnNmb3Jtcz48ZHM6VHJhbnNmb3JtIEFsZ29yaXRob
T0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI2Jhc2U2NCIvPjwvZHM6VHJhbnNmb3Jtcz48ZHM
6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii
i8+PGRzOkRpZ2VzdFZhbHVlPkxQSk51bCt3b3c0bTZEc3F4Ym5pbmhzV0hsd2ZwMEplY3dRellwT0xtQ1E9PC9
kczpEaWdlc3RWYWx1ZT48L2RzOlJlZmVyZW5jZT48ZHM6UmVmZXJlbmNlIFR5cGU9Imh0dHA6Ly91cmkuZXRza
S5vcmcvMDE5MDMjU2lnbmVkUHJvcGVydGllcyIgVVJJPSIjeGFkZXMtaWQtZWExMGEwNTE3Y2JjN2Y1NDllYTN
lNjg1ODY3YWM5NWUiPjxkczpUcmFuc2Zvcm1zPjxkczpUcmFuc2Zvcm0gQWxnb3JpdGhtPSJodHRwOi8vd3d3L
nczLm9yZy9UUi8yMDAxL1JFQy14bWwtYzE0bi0yMDAxMDMxNSIvPjwvZHM6VHJhbnNmb3Jtcz48ZHM6RGlnZXN
0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS8wNC94bWxlbmMjc2hhMjU2Ii8+PGRzO
kRpZ2VzdFZhbHVlPnpUZlc5bjFUOWwwTHg2TlhGNUFUM1Btb3FLOWFHbUpIZlBDaXl0Z1JNeVU9PC9kczpEaWd
lc3RWYWx1ZT48L2RzOlJlZmVyZW5jZT48L2RzOlNpZ25lZEluZm8+PGRzOlNpZ25hdHVyZVZhbHVlIElkPSJ2Y
Wx1ZS1pZC1lYTEwYTA1MTdjYmM3ZjU0OWVhM2U2ODU4NjdhYzk1ZSI+Qy9FZnJvWmdGTkJvak40ZnpJd2UzVTR
ibDQ5S2xBbmhKSmI3elc0T2MxNWsweWpPSkZDcm9jY2JGV0U1eU52R3cyVHpxYVo0SVFYRjBKRlllGM2IrNW5sa
G1EcTJacHBJYnNlOWY5M005cGU4cTVHaFBjWkRDV1FmNnp2TnNvUHRPYktzL04vWjlzODVVcmY3UGd1SWtVZlI
3eUJUaW5waGhJNVpuRHZuSnZsQ1RNRE5tYm4yMlBYSlYyc1IxWFpsa1BLWUsvVGhyOFdnSlcrUU9VREdTeGVST
mgyUWJPTllhR1FsUVJ6Vkhqb0c3emppVHM3UkdrR1ZVNGh3Q0pieW9TZThkd20zbkUxenVuQmp2TkRWenVqZVF
yZkhTSjNrQUsxbS9odkt4TnhlNXR4bkNmd0ZkNldDanpDWXBBaE9sSmI3MDJvV3RYam56azBHbkFZQndnPT08L
2RzOlNpZ25hdHVyZVZhbHVlPjxkczpLZXlJbmZvPjxkczpYNTA5RGF0YT48ZHM6WDUwOUNlcnRpZmljYXRlPk1
JSUQxRENDQXJ5Z0F3SUJBZ0lCQ2pBTkJna3Foa2lHOXcwQkFRc0ZBREJOTVJBd0RnWURWUVFEREFkbmIyOWtMV
05oTVJrd0Z3WURWUVFLREJCT2IzZHBibUVnVTI5c2RYUnBiMjV6TVJFd0R3WURWUVFMREFoUVMwa3RWRVZUVkR
FTE1Ba0dBMVVFQmhNQ1RGVXdIaGNOTVRjeE1ERTVNRGN5TXpFd1doY05NVGt3T0RFNU1EY3lNekV3V2pCUE1SS
XdFQVlEVlFRRERBbG5iMjlrTFhWelpYSXhHVEFYQmdOVkJBb01FRTV2ZDJsdVlTQlRiMngxZEdsdmJuTXhFVEF
QQmdOVkJBc01DRkJMU1MxVVJWTlVNUXN3Q1FZRFZRUUdFd0pNVlRDQ0FTSXdEUVlKS29aSWh2Y05BUUVCQlFBR
GdnRVBBRENDQVFvQ2dnRUJBSi9SVHV5WVRvR0RpbUZGR21STDhsN3lyRDZ1WDh1bkYzZkFmMGVTclpSRGZpS1R
3RjlPT3RVclc1OU5EMnNyQyt5aXBWSllHRlNBcTJxS0NuRlBMcXZOV0ZTbCtnZnJtWnNuN2tjUUJWdlFqQjljS
1pHMmc0VW55VUxCa3JQMFIvcW1pTmRuM0kzNHE2a0lBV3hXUnprUC9CaFAxdWVVNjBnUnhlV01HUGEyeVZabnh
KbDFUOEplSGkvSmpoN2tQSTgwR3V4UXJDQkg0eGJRWngvU1FpV2pJdDdwcm1WZ0crR0hoNHFpb0JGWWp2OGdQM
2ZMdExrM3ZGRHptRWFPeHRMMWRHQXdwcG83R0x1VXRVNkp4QXYwZm52eTZVT055QW4rK3V0OHFrOFU4WlM0Sk9
nRnpKdlZLT2N0UXBUVGptbitMbG9uMWlncmtiSEptN2lxYWJrQ0F3RUFBYU9CdkRDQnVUQU9CZ05WSFE4QkFmO
EVCQU1DQmtBd2dZY0dDQ3NHQVFVRkJ3RUJCSHN3ZVRBNUJnZ3JCZ0VGQlFjd0FZWXRhSFIwY0RvdkwyUnpjeTV
1YjNkcGJtRXViSFV2Y0d0cExxXWmhZM1J2Y25rdmIyTnpjQzluYjI5a0xXTmhNRHdHQ0NzR0FRVUZCekFDaGpCb
2RIUndPaTh2WkhOekxtNXZkMmx1WVM1c2RTOXdhMmt0Wm1GamRHOXllUzlqY25RdloyOXZaQzFqWVM1amNuUXd
IUVlEVlIwT0JCWUVGTy9QL2lxRW92UW4zYXByN1VTT0tTUzdSTUU5TUEwR0NTcUdTSWIzRFFFQkN3VUFBNElCQ
VFBRTJMSjdKa1RlQ2lRT1plSTBTalcySnAwTFA2S1lpNWNOYzR6SW4wTlkwQ05UUkZzejdrRlc2QW9FVGhoSjk
zMmd5NmxSK0ZMR3BwS1NRNVZtUDBLd2JZV3g2MGFUSFJTbmtramRvZnlrYStoNitSbk1mRnl3NG9pZGVxdTBFW
HBMNFhtVFNQN3hPNi9PN2EzZk9kM01DUy9Udm4wQllmNVlTOFJuZXd4MHFBZk5hb3czYUhDMTEyQkFTMUFNZVV
Tc2x5QVBYMUNGZ2dtK25aUEgwcnVUL0NQVko5WlR6VFcyMlhLa1liaytHTVFEOGxRR1RwYTBzVnU1K2Z3Zm1JZ
28xZ1NqY20raXhKN04raDVtVXFZcE1Ydkp1TnJLUWwvSjA3RURWWmlrRWVnL2NQTkV2TE1XOXU5ckxqdU1rZWp
hQytETFUxRkpCZEpvd3FJS2NBakE8L2RzOlg1MDlDZXJ0aWZpY2F0ZT48ZHM6WDUwOUNlcnRpZmljYXRlPk1JS
```

```
UQ2akNDQXRLZ0F3SUJBZ0lCQkRBTkJna3Foa2lHOXcwQkFRc0ZBREJOTVJBd0RnWURWUVFEREFkeWIyOTBMV05
oTVJrd0Z3WURWUVFLREJCT2IzZHBibUVnVTI5c2RYUnBiMjV6TVJFd0R3WURWUVFMREFoUVMwa3RWRVZUVkRFT
E1Ba0dBMVVFQmhNQ1RGVXdIaGNOTVRjeE1ERTVNRGN5TWpVeFdoY05NVGt3T0RFNU1EY3lNalV4V2pCTk1SQXd
EZ1lEVlFRRERBZG5iMjlrTFdOaE1Sa3dGd1lEVlFRS0RCQk9iM2RwYm1FZ1UyOXNkWFJwYjI1ek1SRXdEd1lEV
lFRTERBaFFTMGt0VkVWVFZERUxNQWtHQTFVRUJoTUNURlV3Z2dFaU1BMEdDU3FHU0liM0RRRUJBUVVBQTRJQkR
3QXdnZ0VLQW9JQkFRQ2U4bjJoTDJrKzRRcklXUDJ6UmxMQkhBK1RGRDVtZlFrNWlna3plRzI4UDZSSXAxZlQwM
HdDQzk3MVRndktlZ0xyTmx5REducEFzQ2k1UDZndXd3dDk3NFhKSGJoTitZc0xJa2g3djRYbVVQSFpDcEpLS1h
ScCs1bThpS002cGJGSS8rOE9KQ0JYaDMxY3pHTFlnRUFnQ0ZkVTg5WXY5YTl2Z1FJVkQ3bko3aUFRV0xoSHJ6S
1lwSkQ0OEt2WklHMVJDNDhZNjhtNjFDZEdzenRVTHVHV1I1OGo5Zm5qanVRSTRITWNmY1ljK1pWRWR1dUp0bWp
1M3h4UkE1aGhIYkczaHN1NHpjSVJLd1pBT0hGcGJNVnZWVDVSZk9GTE9rNkt6WlR0NzFUSzVMbk5WN1lvSHc3O
XJXU29yRkxrRzRMVUxTR2d5bHllTVVUdHd5R25GeVpuQWdNQkFBR2pnZFF3Z2RFd0RnWURWUjBQQVFIL0JBUUR
BZ2VBTUVFR0ExVWRId1E2TURnd05xQTBvREtHTUdoMGRIQTZMeTlrYzNNdWJtOTNhVzVoTG14MUwzQnJhUzFtW
VdOMGIzSjVMMk55YkM5eWIyOTBMV05oTG1OeWJEQk1CZ2dyQmdFRkJRY0JBUVJBTUQ0d1BBWUlLd1lCQlFVSE1
BS0dNR2gwZEhBNkx5OWtjM011Ym05M2FXNWhMbXgxTDNCcmFTMW1ZV04wYjNKNUwyTnlkQzl5YjI5MExXTmhMb
U55ZERBZEJnTlZIUTRFRmdRVUhGUXMweWRjUDFSUHlvWXJ2bExHUjFaYksxZ3dEd1lEVlIwVEFRSC9CQVV3QXd
FQi96QU5CZ2txaGtpRzl3MEJBUXNGQUFPQ0FRRUFIM0hkZkpQYkhPQ3BjRXBteHZaRi9VMjcreTB3VFd6aUo0a
3Z1Rnp5YmNMcjJyRWt3UkpldDBPaEZBMlBTSXFZZXc5S1lpb3BEd0VsOGQxSXA4L3k5Tk1kYU9VWUVpK2RTZzk
wMWNnVnhxRlFFRHJadUpWdEljQnh3MzBiNWFPMUE1V0FRRzhCMVhaNjI1K0NielRNQllOK0xoRHFZRWJhK1FXW
mdBR3BzWDFOS281TmxtK0wySmlVdng5QjlXcU95YkxZSWxWbmxuSGk3bFRJNDBjMjNTM2hTYVp6Z3lBdUFWR2N
TKzZFSldSc0dYNXJtaUE1MUNlTUhoMEtCdXRlL0FkczVOb0RteW93bHlhYU5vZHBTc2NiVWxIK0hneGlMVWRYN
0tJNDlabWRGSWtzUDB2Q1VvWFliWFlTekdmYmt2VGZ5SjQ5NXJzcktkaTcreWg0NlE9PTwvZHM6WDUwOUNlcnR
pZmljYXRlPjxkczpYNTA5Q2VydGlmaWNhdGU+TUlJRFZ6Q0NBaitnQXdJQkFnSUJBVEFOQmdrcWhraUc5dzBCQ
VEwRkFEQk5NUkF3RGdZRFZRUUREQWR5YjI5MExXTmhNUmt3RndZRFZRUUtEQkJPYjNkcGJtRWdVMjlzZFhScGI
yNXpNUkV3RHdZRFZRUUxEQWhRUzBrdFZFVlRWREVMTUFrR0ExVUVCaE1DVEZVd0hoY05NVGN3T1RFNU1EY3lNa
lF4V2hjTk1Ua3dPVEU1TURjeU1qUXhXakJOTVJBd0RnWURWUVFEREFkeWIyOTBMV05oTVJrd0Z3WURWUVFLREJ
CT2IzZHBibUVnVTI5c2RYUnBiMjV6TVJFd0R3WURWUVFMREFoUVMwa3RWRVZUVkRFTE1Ba0dBMVVFQmhNQ1RGV
XdnZ0VpTUEwR0NTcUdTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUtBb0lCQVFEUmc0SDRvbEhveDFlNzljVUprSTV
1SitETGtYVnl2ZkExNWlWZGVJY3ZhS0xrUmdoYWlheTRsbmRJWTVGRjR0TVkwRWI2aW45Z1B2VzlnZytPMy9BM
HFUcHc0OXA5Z0FSdXE0SzJmNGFUZC8zUmdVem8wNHRXblJkbUg3Tm5Nc3ZKcmhHcGRvclpnejd5SmlHUVVWRjQ
4bFkzT0VLd3dCWUQzOGJER01UZG9jdGdrY2F6bThFVGF6M0hwQm9yRi9GM09nZ3JPNUc0SldtNGFuTlBvYUdZM
WZaR3ZJQ0RTNCtlejNlaElkNytobS80Sjkyc2hwUkRuMjl4djdra3g5VVBCQWVSYjZ3YzVhTkxmdGx2aEF4S0U
2bk5Dbk0wYXBvQmRCRGVuY3IzZk9SWlU0cmxxd1NsNkg2T3pseHFDdW9QQkp5R2tra3hvRzRabHVmWGhQV2JBZ
01CQUFHalFqQkFNQTRHQTFVZER3RUIvd1FFQXdJQkJqQWRCZ05WSFE0RUZnUVVZVVB6YmV6NXNiY0pIcys5OE1
HU2haEw3UEF3RHdZRFZSMFRBUUgvQkFVd0F3RUIvekFOQmdrcWhraUc5dzBCQVEwRkFBT0NBUUVBVG1QYVpTb
0dMNFg5UTFmOXhOa0NCYjZUQjlTUmEwZVVCKy9wUUVReXR5Rys5c3lFRkY4aGVmVjB6bGdGOUZqM1VwbWwyM0h
1dnZRQXk1YmE4dGxxWStMdE52THBRb1pHcXZEUDN0NkFlRDNONTQwNFNzd2FpT2tPL1gySmVZZzl3RDN4RU9na
kNSTVdyTU1FSWhxb1pOZXFIN2dLS1pKL3RHT01vSExjSHFXYVZmbGpqVmNUNnA0enI4bzB0MXl5T3AzNlNqVS9
LOHBNdFg0YU1PU05uUlpTdnl2a3F5Ly9pNHlFbmFRNnMvVks1eUgzYStXcENiTnpLQ0xmbTEzMS8rVUdZV1FOV
GIzWURYUUtGWmkwcnZoOGtodFFDeVEzYXVzQUZMdWsyc0FmSUszVGtIZml1ZnFZSXZsbEMzclZZQUo2TU9lWWl
XWGpTd1RrK25pVWFBPT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT48L2RzOlg1MDlEYXRhPjwvZHM6S2V5SW5mbz48Z
HM6T2JqZWN0Pjx4YWRlczpRdWFsaWZ5aW5nUHJvcGVydGllcyB4bWxuczp4YWRlcz0iaHR0cDovL3VyaS5ldHN
pLm9yZy8wMTkwMy92MS4zLjIjIiBUYXJnZXQ9IiNpZC1lYTEwYTA1MTdjYmM3ZjU0OWVhM2U2ODU4NjdhYzk1Z
SI+PHhhZGVzOlNpZ25lZFByb3BlcnRpZXMgSWQ9InhhZGVzLWlkLWVhMTBhMDUxN2NiYzdmNTQ5ZWEzZTY4NTg
2N2FjOTVlIj48eGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZXM6U2lnbmluZ1RpbWU+MjAxO
C0wOS0yN1QxMTo1ODo0M1o8L3hhZGVzOlNpZ25pbmdUaW1lPjx4YWRlczpTaWduaW5nQ2VydGlmaWNhdGVWMj4
8eGFkZXM6Q2VydD48eGFkZXM6Q2VydERpZ2VzdD48ZHM6RGlnZXN0TWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL
3d3dy53My5vcmcvMjAwMC8wOS94bWxkc2lnI3NoYTEiLz48ZHM6RGlnZXN0VmFsdWU+aE5yb1k4cjFDQjU5Nmp
HQlBnUmdaRnZSRGJjPTwvZHM6RGlnZXN0VmFsdWU+PC94YWRlczpDZXJ0RGlnZXN0Pjx4YWRlczpJc3N1ZXJTZ
XJpYWxWMj5NRll3VWFSUE1FMHhFREFPQmdOVkJBTU1CMmR2YjJRdFkyRXhHVEFYQmdOVkJBb01FRTV2ZDJsdVl
TQlRiMngxZEdsdmJuTXhFVEFQQmdOVkJBc01DRkJMU1MxVVJWTlVNUXN3Q1FZRFZRUUdFd0pNVlFJQkNnPT08L
3hhZGVzOklzc3VlclNlcmlhbFYyPjwveGFkZXM6Q2VydD48L3hhZGVzOlNpZ25pbmdDZXJ0aWZpY2F0ZVYyPjw
```

```
veGFkZXM6U2lnbmVkU2lnbmF0dXJlUHJvcGVydGllcz48eGFkZXM6U2lnbmVkRGF0YU9iamVjdFByb3BlcnRpZ
XM+PHhhZGVzOkRhdGFPYmplY3RGb3JtYXQgT2JqZWN0UmVmZXJlbmNlPSIjci1pZC0xIj48eGFkZXM6TWltZVR
5cGU+dGV4dC9wbGFpbjwveGFkZXM6TWltZVR5cGU+PC94YWRlczpEYXRhT2JqZWN0Rm9ybWF0PjwveGFkZXM6U
2lnbmVkRGF0YU9iamVjdFByb3BlcnRpZXM+PC94YWRlczpTaWduZWRQcm9wZXJ0aWVzPjwveGFkZXM6UXVhbGl
meWluZ1Byb3BlcnRpZXM+PC9kczpPYmplY3Q+PGRzOk9iamVjdCBJZD0iby1pZC0xIj5hR1ZzYkc4PTwvZHM6T
2JqZWN0PjwvZHM6U2lnbmF0dXJlPg==",
    "digestAlgorithm" : null,
    "name" : "hello-signed-xades.xml"
  },
  "originalDocuments" : null,
  "policy" : null,
  "tokenExtractionStrategy" : "NONE",
  "signatureId" : null
}
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2020 15:38:59 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" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+XqO"
  },
  "certificateChain" : [ {
    "encodedCertificate" :
"MIIC6jCCAdKgAwIBAgIGLtYU17tXMA0GCSqGSIb3DQEBCwUAMDAxGzAZBgNVBAMMElJvb3RTZWxmU2lnbmVkR
mFrZTERMA8GA1UECgwIRFNTLXRlc3QwHhcNMTcwNjA4MTEyNjAxWhcNNDcwNzA0MDc1NzI0WjAoMRMwEQYDVQQ
DDApTaWduZXJGYWtlMREwDwYDVQQKDAhEU1MtdGVzdDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBA
MI3kZhtnipn+iiZHZ9ax8FlfE5Ow/cFwBTfAEb3R1ZQUp6/BQnBt7Oo0JWBtc9qkv7JUDdcBJXPV5QWS5AyMPH
pqQ75Hitjsq/Fzu8eHtkKpFizcxGa9BZdkQjh4rSrtO1Kjs0Rd5DQtWSgkeVCCN09kN0ZsZ0ENY+Ip8QxSmyzt
sStkYXdULqpwz4JEXW9vz64eTbde4vQJ6pjHGarJf1gQNEc2XzhmI/prXLysWNqC7lZg7PUZUTrdegABTUzYCR
J1kWBRPm4qo0LN405c94QQd45a5kTgowHzEgLnAQI28x0M3A59TKC+ieNc6VF1PsTLpUw7PNI2VstX5jAuasCA
wEAAaMSMBAwDgYDVR0PAQH/BAQDAgEGMA0GCSqGSIb3DQEBCwUAA4IBAQCK6LGA01TR+rmU8p6yhAi4OkDN2b1
dbIL8l8iCMYopLCxx8xqq3ubZCOxqh1X2j6pgWzarb0b/MUix00IoUvNbFOxAW7PBZIKDLnm6LsckRxs1U32sC
9d1LOHe3WKBNB6GZALT1ewjh7hSbWjftlmcovq+6eVGA5cvf2u/2+TkKkyHV/NR394nXrdsdpvygwypEtXjetz
D7UT93Nuw3xcV8VIftIvHf9LjU7h+UjGmKXG9c15eYr3SzUmv6kyOI0Bvw14PWtsWGl0QdOSRvIBBrP4adCnGT
gjgjk9LTcO8B8FKrr+8lHGuc0bp4lIUToiUkGILXsiEeEg9WAqm+XqO"
  } ],
  "validationTime" : null,
  "tokenExtractionStrategy" : "NONE"
}
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2020 15:38:59 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: 6880

{
  "diagnosticData" : {
    "DocumentName" : null,
    "ValidationDate" : "2020-12-14T15:38:59",
    "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" ],
```

```
      "NotAfter" : "2047-07-04T07:57:24",
      "NotBefore" : "2017-06-08T11:26:01",
      "PublicKeySize" : 2048,
      "PublicKeyEncryptionAlgo" : "RSA",
      "EntityKey" : "PK-
3CFCA257859E202BCC83864D02B267B08A997C357AB98D923BBC63F00607C7B6",
      "KeyUsage" : [ "keyCertSign", "crlSign" ],
      "extendedKeyUsagesOid" : [ ],
      "PSD2Info" : null,
      "IdPkixOcspNoCheck" : false,
      "BasicSignature" : {
        "EncryptionAlgoUsedToSignThisToken" : "RSA",
        "KeyLengthUsedToSignThisToken" : "?",
        "DigestAlgoUsedToSignThisToken" : "SHA256",
        "MaskGenerationFunctionUsedToSignThisToken" : null,
        "SignatureIntact" : null,
        "SignatureValid" : null
      },
      "SigningCertificate" : null,
      "ChainItem" : [ ],
      "Trusted" : false,
      "SelfSigned" : false,
      "certificatePolicy" : [ ],
      "qcStatementOid" : [ ],
      "qcTypeOid" : [ ],
      "QCLimitValue" : null,
      "SemanticsIdentifier" : 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" : "2017-06-08T11:26:01",
      "notAfter" : "2047-07-04T07:57:24",
      "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" : "2020-12-14T15:38:59"
  },
  "detailedReport" : {
    "signatureOrTimestampOrCertificate" : [ {
      "Certificate" : {
        "ValidationCertificateQualification" : [ ],
        "Constraint" : [ {
          "Name" : {
            "value" : "Is the result of the Basic Building Block conclusive?",
            "NameId" : "BBB_ACCEPT"
          },
          "Status" : "WARNING",
          "Error" : null,
          "Warning" : {
            "value" : "The result of the Basic Building Block is not conclusive!",
            "NameId" : "BBB_ACCEPT_ANS"
          },
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        } ],
        "Conclusion" : {
          "Indication" : "INDETERMINATE",
          "SubIndication" : null,
```

```
          "Errors" : [ ],
          "Warnings" : [ {
            "value" : "The result of the Basic Building Block is not conclusive!",
            "NameId" : "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?",
            "NameId" : "BBB_XCV_CCCBB"
          },
          "Status" : "NOT OK",
          "Error" : {
            "value" : "The certificate chain is not trusted, it does not contain a
trust anchor.",
            "NameId" : "BBB_XCV_CCCBB_ANS"
          },
          "Warning" : null,
          "Info" : null,
          "AdditionalInfo" : null,
          "Id" : null
        } ],
        "Conclusion" : {
          "Indication" : "INDETERMINATE",
          "SubIndication" : "NO_CERTIFICATE_CHAIN_FOUND",
          "Errors" : [ {
            "value" : "The certificate chain is not trusted, it does not contain a
trust anchor.",
            "NameId" : "BBB_XCV_CCCBB_ANS"
          } ],
          "Warnings" : null,
          "Infos" : null
        },
        "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.",
          "NameId" : "BBB_XCV_CCCBB_ANS"
        } ],
        "Warnings" : null,
        "Infos" : null
      },
      "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" : "ZOyIygCyaOW6GjVnihtTFtIS9PNmskdyMlNKiuyjfzw="
}
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2020 15:38:58 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: 2227

{
  "binaries" :
"MIIGcAYJKoZIhvcNAQcCoIIGYTCCBl0CAQMxDzANBglghkgBZQMEAgEFADByBgsqhkiG9w0BCRABBKBjBGEwX
wIBAQYDKgMEMDEwDQYJYIZIAWUDBAIBBQAEIGTsiMoAsmjluho1Z4obUxbSEvTzZrJHcjJTSorso388AhEArc/
NiyneM2PlqM3DQroKCBgPMjAyMDEyMTQxNTM4NThaoIIDcjCCA24wggJWoAMCAQICAWQwDQYJKoZIhvcNAQELB
QAwVTEYMBYGA1UEAwwPc2VsZi1zaWduZWQtdHNhMRkwFwYDVQQKDBBOb3dpbmEgU29sdXRpb25zMREwDwYDVQQ
LDAhQS0ktVEVTVDELMAkGA1UEBhMCTFUwHhcNMTkwNzExMDQ1MjE5WhcNMjEwNTExMDQ1MjE5WjBVMRgwFgYDV
QQDDA9zZWxmLXNpZ25lZC10c2ExGTAXBgNVBAoMEE5vd2luYSBTb2x1dGlvbnMxETAPBgNVBAsMCFBLSS1URVN
UMQswCQYDVQQGEwJMVTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALD2WoKIuG0HIdyWcj8iynQur
ThwcZBr1NCJndhS8bSJaFORSCYwfUyiOvUEdFeAUOGHBpJLG8PmtB5UAnn4o+sicgHDO+BaA96F9EMPaZk5MjI
49R5aVtixhadNd41kWxeysBf/KZGuThrOXiV9Vk7O7mg9MEGj38KR46G1nwZr/9A/B47jTxLerBXS95P+mbciu
cj7WWnYOrPL4EcJX3riiNbJyoxYFhlLAeLUbqSkLiqJh/FkEZ6pgtOsPuYfd3R60aFmuLeMDpGSUoGblipqb8t
4qGlDbI3SqhI0DDVoiQY5+orJY9b2kiGFgj3LrJ4VHKHiCnlMbJPObxy6K9ECAwEAAaNJMEcwDgYDVR0PAQH/B
AQDAgeAMBYGA1UdJQEB/wQMMAoGCCsGAQUFBwMIMB0GA1UdDgQWBBQgGNMhuy8wru7TDYK9/tiKfETI+TANBgk
qhkiG9w0BAQsFAAOCAQEAl0yJSzLz34xorwqge4HYGpaGW0QxnF87yOEBA0aX8++ENQuCC5OVMyGyL7Xmy3Kgg
gKlfs25E1pYvrwxz31jTLv9MPdVxnbW3wbvUNa4oS2YBPKtS/34OsUzKkE0JAWg6S3lhmkUoM59UZVGKMUDhsx
Ybdb1/fF+EOq4zpyuz4YyIq7utE4MxI6cbSEQ2iysloK0AMhACcvIH0kpIpinnLyF7JstNQEA0rtFwxTRzb3Ev
2JR/AhqgUsgFTHKJGidVy9BGWyoMONIAD3MwAI/LzIYeFzZQRMRGdF/WaEgHrwclMkcU3yb1BOzifzgV3a46W6
2n5aLK5rMGVJt5E48PDGCAlswggJXAgEBMFowVTEYMBYGA1UEAwwPc2VsZi1zaWduZWQtdHNhMRkwFwYDVQQKD
BBOb3dpbmEgU29sdXRpb25zMREwDwYDVQQLDAhQS0ktVEVTVDELMAkGA1UEBhMCTFUCAWQwDQYJYIZIAWUDBAI
BBQCggdMwGgYJKoZIhvcNAQkDMQ0GCyqGSIb3DQEJEAEEMBwGCSqGSIb3DQEJBTEPFw0yMDEyMTQxNTM4NThaM
C0GCSqGSIb3DQEJNDEgMB4wDQYJYIZIAWUDBAIBBQChDQYJKoZIhvcNAQELBQAwLwYJKoZIhvcNAQkEMSIEIGW
HTa2zuXPyN2C6wGNKuC3MVUzxsOvzcl7dvz+ekyTsMDcGCyqGSIb3DQEJEAIvMSgwJjAkMCIEIOF7h4gurvatQ
IShO3Lidh12sB11RHlhyg8/6hqSMn7TMA0GCSqGSIb3DQEBCwUABIIBAAOwPm+hj2qKxwJTKZ/nTiScfRMc1cY
KEBtvJ3x6tVih/ulcG/+wvjft1ZMahoSaxL9BOyMG15Tx7mFWt0ji09NEjqM/px4cSbginZN138t+Pov+lEccb
sB7DRaUFHTll1kaV15sv9rrPoZxYRIzp1VjwzlGVGXcAVJAN6bLoqApQdwWyaszEYxYVe7we6E3oaU0djo5sNQ
DufEojjkH/D2Di2PWWepBZpq2+91sqKeOgpajkzxPll5qXqk2CUpwALdKMgeIKuvhVqCNr2kDBmxKjGizsdx6H
iu4VVVMEursaWgHvWbCw67vky00X1bWiyy9GSd2ytKGiEf6OGCCg0A="
}
```