The European Commission's science and knowledge service

Joint Research Centre

VECTO Data Integrity Measures

2018 VECTO Workshop

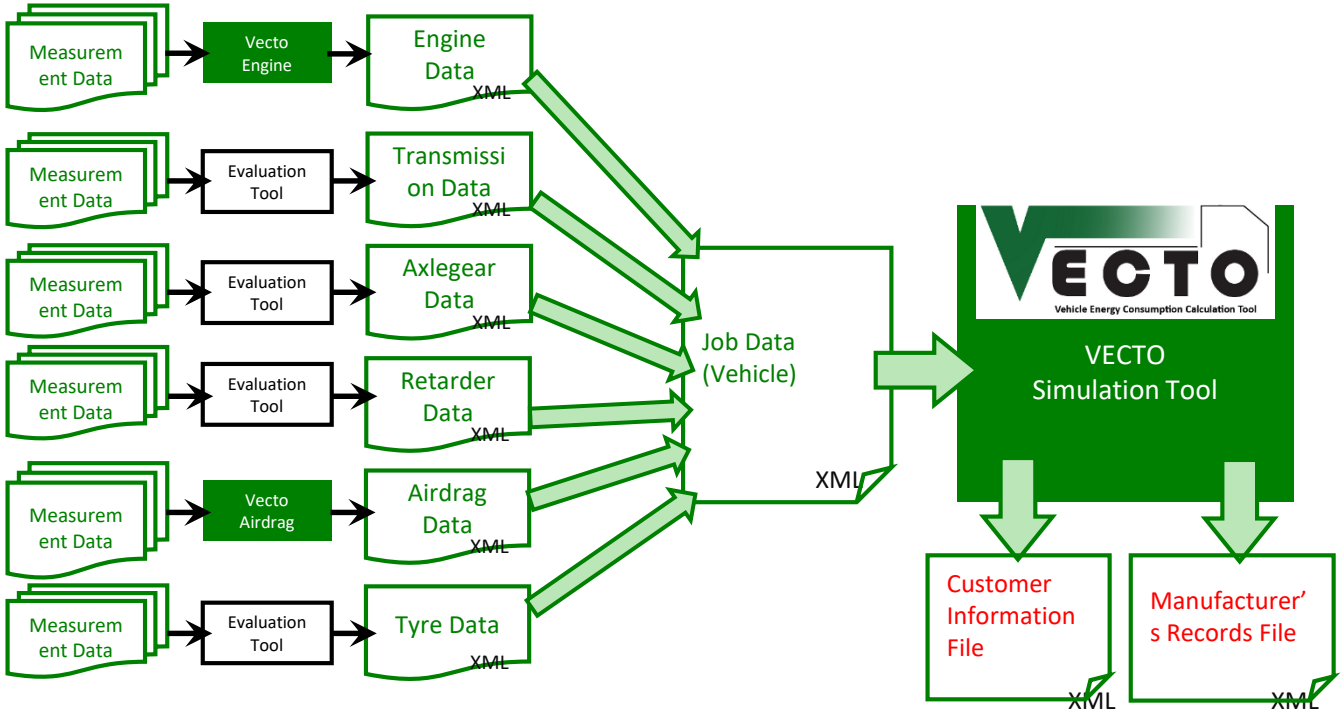
Ispra, November, 2018

European Commission

Contents

- Handling VECTO-related data
- Data integrity measures
- Method of hash computation
- VECTO Hashing Tool

VECTO Data Flow



European Commission

Requirements on Data Handling

- Structured data format
- Human readable format
- Traceability of components without actual component data
- Detect modifications during transmission
- Use data integrity measures

Implemented Solution for Component Data

- Hashing of component data (digest value) at component certification
- Digest value is part of certificate
- Digest value is stored at TAA/TS

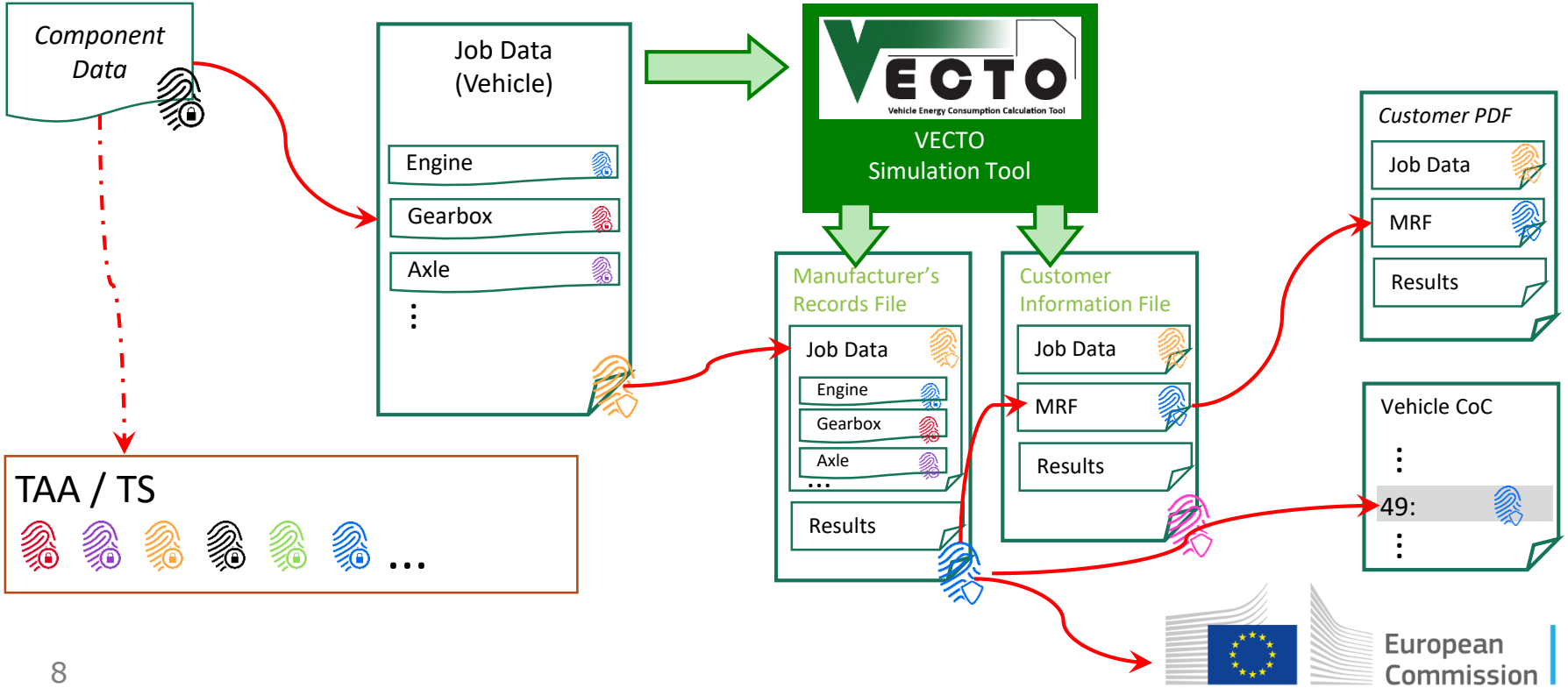
What is Hashing?

- Hash function
  - A hash function is any function that can be used to map data of arbitrary size to data of fixed size
- Cryptographic hash function
  - Deterministic: same message results in the same hash
  - Quick to compute for any given message size
  - Small change to the message should change the hash value significantly (new hash value should appear uncorrelated with old hash value)
  - Infeasible to generate message from hash value – one-way function
  - Infeasible to find two different messages with the same hash value

Hashing as thoroughgoing Concept

- All component data is hashed
- Hashing is applied to standard values
- Job data is hashed
- Reports are hashed

Chain of Digest Values



European Commission

Benefits of Hashing Component Data

- Component manufacturer can be sure that the correct component data was used for simulation as long as the digest value is not modified
- Vehicle manufacturer can be sure they received and used the correct component data as long as the digest value is not modified
- Clearly defined responsibilities
- Verification test: ensure the correct component data is used for simulation

Component Evaluation Tools & Hashing

- The provided pre-processing tools (VECTO Engine, VECTO Airdrag) already contain the hashing functionality
- In case no common component evaluation tool is available, the VECTO hashing tool (or the VECTO Hashing Libary) has to be used for computing the digest value

Hash Computation Method

Requirements Data Integrity

- Data integrity for
  - Component data, VECTO job data, VECTO results
- Use state-of-the-art algorithms for cryptographic operations
- Robust method for computing hash of XML documents
- Rely on existing standards
- Use existing implementations (programming libraries)

Implemented Method

- Based on XML-DSig (also used for eIDAS, XAdES)
  - Intended for signing XML documents
  - Provides means to use only hashing functionality
- Similar to ‘Detached Signatures’
  - Hash is not part of the data
- SHA256 hashing algorithm

Structure of Hashed XML Files

```
<?xml version="1.0" encoding="utf-8"?>
<tns:VectoInputDeclaration xmlns="urn:tugraz:ivt:VectoAPI:DeclarationDefinitions:v0.8"
    xmlns:tns="urn:tugraz:ivt:VectoAPI:DeclarationComponent:v0.8"
    xmlns:di="http://www.w3.org/2000/09/xmldsig#" schemaVersion="0.6"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:tugraz:ivt:VectoAPI:DeclarationComponent:v0.8 ../XSD/VectoComponent.xsd">
    <tns:Engine>
        <Data id="ENG-c481b13b8dba4d3682c4">

            [...]

        </Data>
        <Signature>
            <di:Reference URI="#ENG-c481b13b8dba4d3682c4">
                <di:Transforms>
                    <di:Transform Algorithm="urn:vecto:xml:2017:canonicalization" />
                    <di:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                </di:Transforms>
                <di:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
                <di:DigestValue>BWAxs/2pGjQJdvn2GJE7X21bNOBYSN3Xrrb+a+BfVUg=</di:DigestValue>
            </di:Reference>
        </Signature>
    </tns:Engine>
</tns:VectoInputDeclaration>
```

XML Declaration

Root Element
Namespace definitions,
Component, Job, or Report

Wrapping element
Identifies content type

Model Data

Digest Value
Reference to data,
C14N Methods,
Hashing Algorithm

Detached Hash (Signature)

```
<Engine>
    <Data id="ENG-gooZah3D">
        <Manufacturer>Generic Engine Manufacturer</Manufacturer>
        <Make>Generic 40t Long Haul Truck Engine</Make>
        <TypeId>ENG-gooZah3D</TypeId>
        <Date>2016-02-15T11:00:00Z</Date>
        <AppVersion>VectoEngine x.y</AppVersion>
        <Displacement>12730</Displacement>
        <IdlingSpeed>560</IdlingSpeed>
        <WHTCUrban>0.9700</WHTCUrban>
        <WHTCRural>0.9900</WHTCRural>
        <WHTCMotorway>1.0200</WHTCMotorway>
        <BFColdHot>1.0000</BFColdHot>
        <CFRegPer>1.0000</CFRegPer>
        <FuelType>Diesel CI</FuelType>
        <FuelConsumptionMap>...</FuelConsumptionMap>
        <FullLoadAndDragCurve>...</FullLoadAndDragCurve>
    </Data>
    <Signature>
        <di:Reference URI="#ENG-gooZah3D">
            <di:Transforms>
                <di:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithoutComments"/>
                <di:Transform Algorithm="urn:vecto:xml:2017:canonicalization"/>
            </di:Transforms>
            <di:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <di:DigestValue>e0c253b643f7f8f09b963aca4a264d06fbfa599f</di:DigestValue>
        </di:Reference>
    </Signature>
</Engine>
```

ID Attribute

```
<Engine>
    <Data id="ENG-gooZah3D">
        <Manufacturer>Generic Engine Manufacturer</Manufacturer>
        <Make>Generic 40t Long Haul Truck Engine</Make>
        <TypeId>ENG-gooZah3D</TypeId>
        <Date>2016-02-15T11:00:00Z</Date>
        <AppVersion>VectoEngine x.y</AppVersion>
        <Displacement>12730</Displacement>
        <IdlingSpeed>560</IdlingSpeed>
        <WHTCUrban>0.9700</WHTCUrban>
        <WHTCRural>0.9900</WHTCRural>
        <WHTCMotorway>1.0200</WHTCMotorway>
        <BFColdHot>1.0000</BFColdHot>
        <CFRegPer>1.0000</CFRegPer>
        <FuelType>Diesel CI</FuelType>
        <FuelConsumptionMap>...</FuelConsumptionMap>
        <FullLoadAndDragCurve>...</FullLoadAndDragCurve>
    </Data>
    <Signature>
        <di:Reference URI="#ENG-gooZah3D">
            <di:Transforms>
                <di:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithoutComments"/>
                <di:Transform Algorithm="urn:vecto:xml:2017:canonicalization"/>
            </di:Transforms>
            <di:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <di:DigestValue>e0c253b643f7f8f09b963aca4a264d06fbfa599f</di:DigestValue>
        </di:Reference>
    </Signature>
</Engine>
```

- Has to be (sufficiently) unique
- Auto-generated unless provided in input file
  - At least 5 characters long

Timestamp

```
<Engine>
    <Data id="ENG-gooZah3D">
        <Manufacturer>Generic Engine Manufacturer</Manufacturer>
        <Make>Generic 40t Long Haul Truck Engine</Make>
        <TypeId>ENG-gooZah3D</TypeId>
        <Date>2016-02-15T11:00:00Z</Date>
        <AppVersion>VectoEngine x.y</AppVersion>
        <Displacement>12730</Displacement>
        <IdlingSpeed>560</IdlingSpeed>
        <WHTCUrban>0.9700</WHTCUrban>
        <WHTCRural>0.9900</WHTCRural>
        <WHTCMotorway>1.0200</WHTCMotorway>
        <BFColdHot>1.0000</BFColdHot>
        <CFRegPer>1.0000</CFRegPer>
        <FuelType>Diesel CI</FuelType>
        <FuelConsumptionMap>...</FuelConsumptionMap>
        <FullLoadAndDragCurve>...</FullLoadAndDragCurve>
    </Data>
    <Signature>
        <di:Reference URI="#ENG-gooZah3D">
            <di:Transforms>
                <di:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithoutComments"/>
                <di:Transform Algorithm="urn:vecto:xml:2017:canonicalization"/>
            </di:Transforms>
            <di:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <di:DigestValue>e0c253b643f7f8f09b963aca4a264d06fbfa599f</di:DigestValue>
        </di:Reference>
    </Signature>
</Engine>
```

- Overwritten by hashing Tool
- Always in UTC (Z at the end)

Hashing XML Data

- XML is a plain-test data format
- Same XML data can have different representations
  - Line breaks
  - Whitespaces
  - Comments
  - …
- Issue with hashing
  - Semantically equivalent XML data may get different hash values

Basic Hashing Method Description

- Transform XML **`Data`** element (normalize, canonicalize)
  - Standardized methods available
- Compute digest over resulting **`Data`** element
- Create **`Reference`** element

$\Rightarrow$ Crucial to have a strict XML structure and appropriate canonicalization method

Strict XML Structure

- Strict sequence of elements
  - Sequence instead of choice elements
  - Enumerations where possible
- Definition of datatypes
  - Timestamps in UTC
  - Precision of floating-point values
    (2, 3, 4 digits after the decimal sign, no leading zeros)
  - Allowed values for enumerations

VECTO Canonicalization

- Step 1: use standardized method as basis: XML-C14N
  - Whitespace outside of elements, remove comments, ordering of attributes, line breaks, namespaces, contents of text nodes, …
  - Provided by libraries

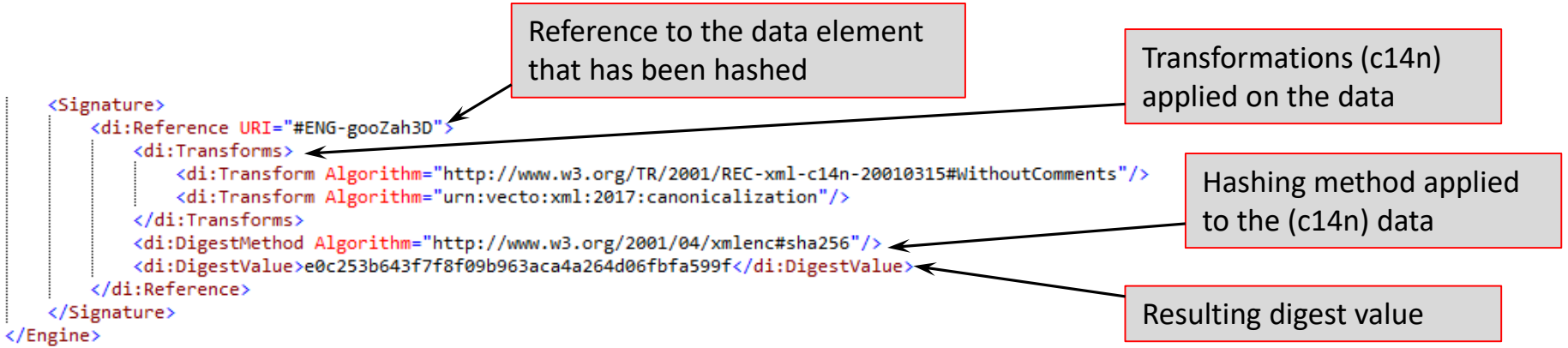
  http://www.w3.org/2001/10/xml-exc-c14n#

- Step 2: VECTO –specific canonicalization:
  - Sort entries in gear list, loss-map entries, fuel consumption map, …
  - Implemented as XSL transformation

  https://webgate.ec.europa.eu/CITnet/svn/VECTO/trunk/Share/XML/HashingXSLT/SortInputData.xslt

The `Signature` Element

- Contains information *how* the digest value has been computed



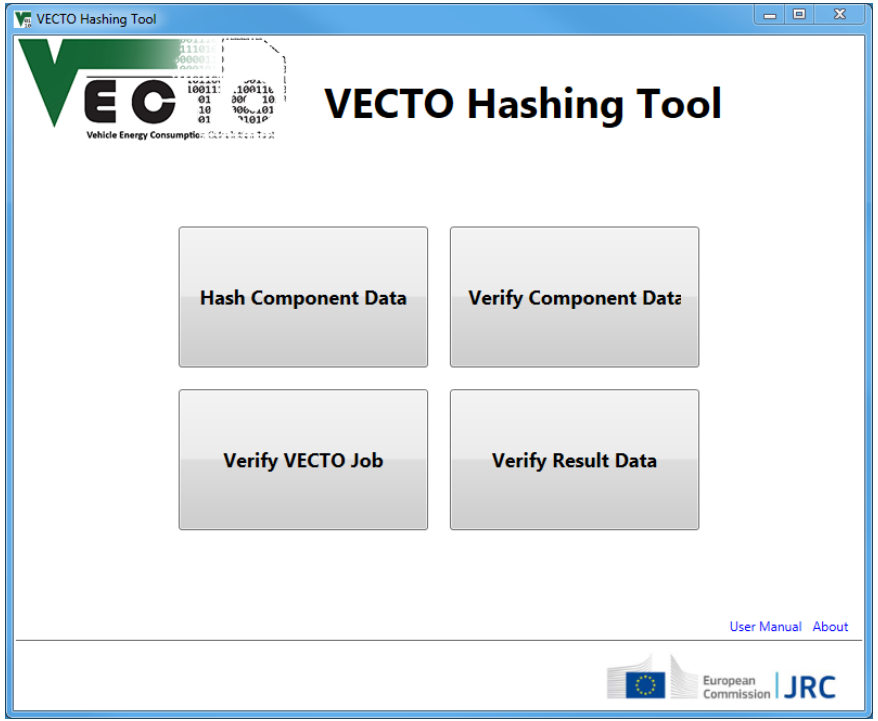
VECTO Hashing Tool

Online Demonstration

VECTO Hashing Tool

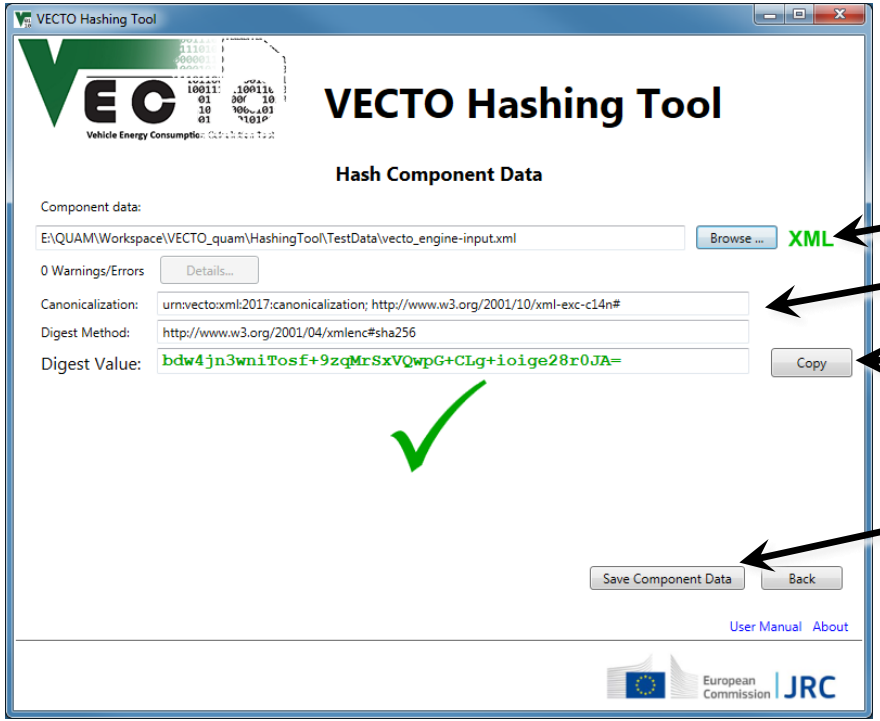
Offline Demonstration

Start Screen



European Commission

Hash Component Data



Load file to hash

C14N method, hashing algorithm

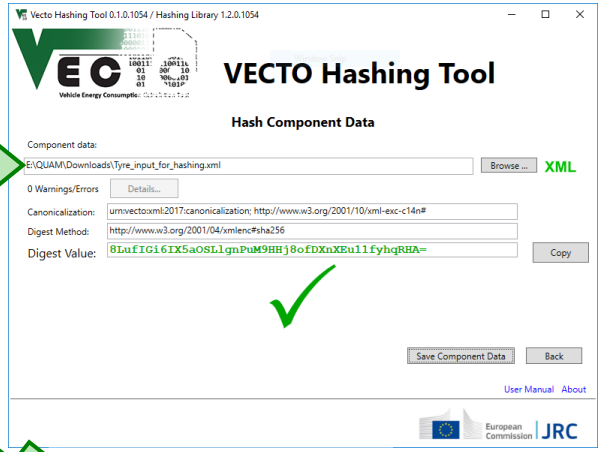
Digest value (as reference)

Save hashed component file

European Commission

Workflow: Hash Component Data

```xml
<?xml version="1.0" encoding="UTF-8"?>
<tns:VectoInputDeclaration xmlns="urn:tugraz:ivt:VectoAPI:DeclarationDefinitions:v1.0"
    <tns:Tyre>
        <Data>
            <Manufacturer>Generic Wheels Manufacturer</Manufacturer>
            <Model>Generic Wheel</Model>
            <CertificationNumber>e12*0815/8051*2017/05E0000*00</CertificationNumber>
            <Date>2017-01-11T14:00:00Z</Date>
            <AppVersion>Tyre Generation App 1.0</AppVersion>
            <Dimension>315/70 R22.5</Dimension>
            <RRCDeclared>0.0055</RRCDeclared>
            <FzISO>31300</FzISO>
        </Data>
    </tns:Tyre>
</tns:VectoInputDeclaration>
```




```xml
<?xml version="1.0" encoding="utf-8"?>
<tns:VectoInputDeclaration xmlns="urn:tugraz:ivt:VectoAPI:DeclarationDefinitions:v1.0"
    <tns:Tyre>
        <Data id="TYRE-61714fbfcbf34d57a5ef">
            <Manufacturer>Generic Wheels Manufacturer</Manufacturer>
            <Model>Generic Wheel</Model>
            <CertificationNumber>e12*0815/8051*2017/05E0000*00</CertificationNumber>
            <Date>2017-11-28T10:39:25.9732879Z</Date>
            <AppVersion>Tyre Generation App 1.0</AppVersion>
            <Dimension>315/70 R22.5</Dimension>
            <RRCDeclared>0.0055</RRCDeclared>
            <FzISO>31300</FzISO>
        </Data>
        <Signature>
            <di:Reference URI="#TYRE-61714fbfcbf34d57a5ef">
                <di:Transforms>
                    <di:Transform Algorithm="urn:vecto:xml:2017:canonicalization" />
                    <di:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
                </di:Transforms>
                <di:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
                <di:DigestValue>8LufIGi6IX5aOSLlgnPuM9HHj8ofDXnXEu11fyhqRHA=</di:DigestValue>
            </di:Reference>
        </Signature>
    </tns:Tyre>
</tns:VectoInputDeclaration>
```


European Commission

The XML File Status Indicator

XML • No XML file has been selected

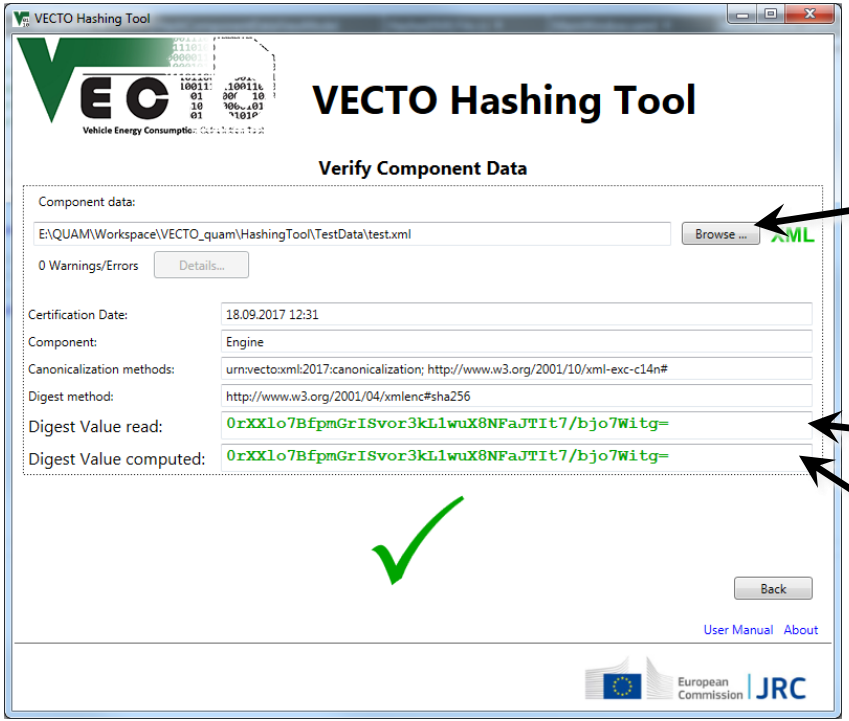
XML • The file could not be read. Ensure the file exists and is a valid XML file

XML • XML validation against a known VECTO XML schema failed

XML • The selected XML file is a valid VECTO XML file, but it has the wrong contents

XML • green the selected file is a valid VECTO XML file and has the correct contents

Verify Component Data



Load hashed component data

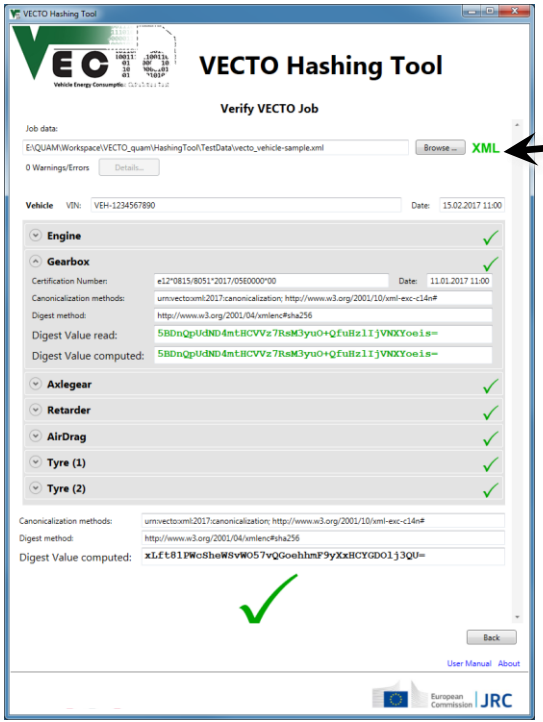
Information on component data, C14N method, digest method

Digest value & method from file

Re-computed digest value

European Commission

Verify Job Data

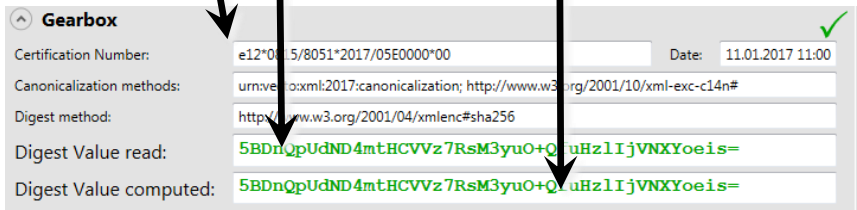


Load job data

Information on components

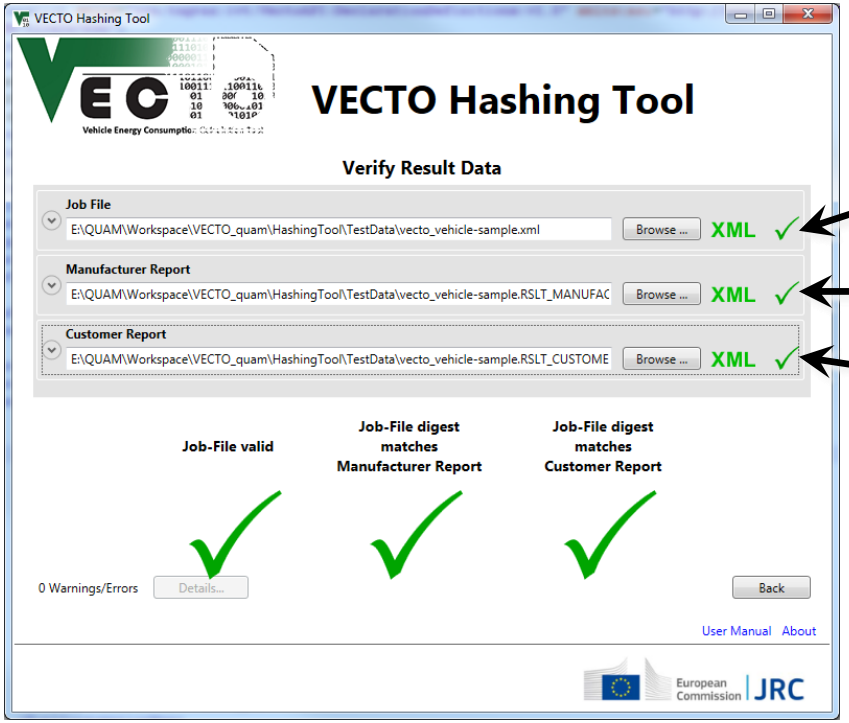
Digest value & method

Re-computed digest value



European Commission

Verify Reports



Load job data

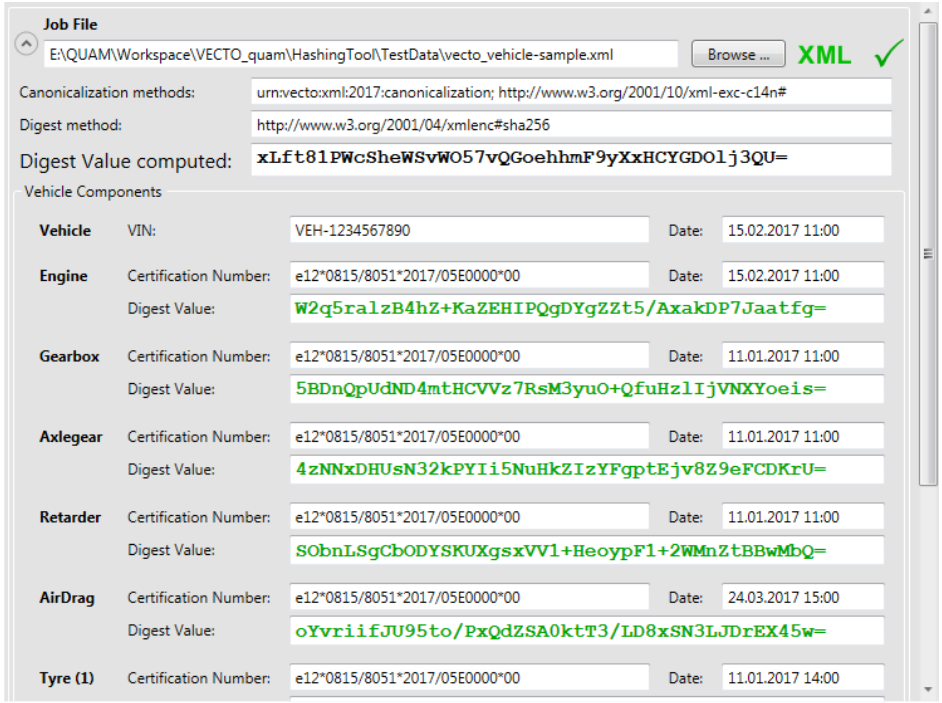
Load manufacturer report

Load customer report

Integrity status

European Commission

Verify Report: Details Job Data

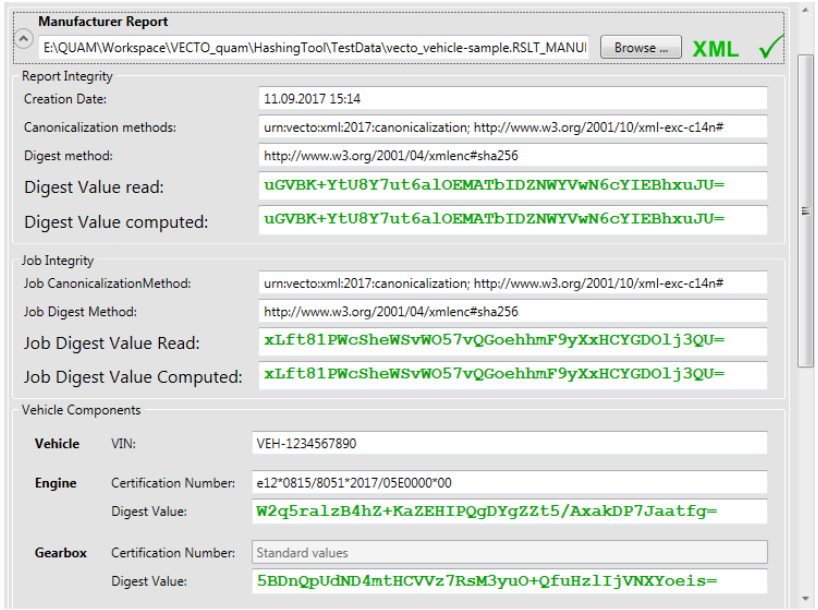


Information on components

Re-computed digest value (red/green)

European Commission

Verify Report: Details Manufacturer's Records File



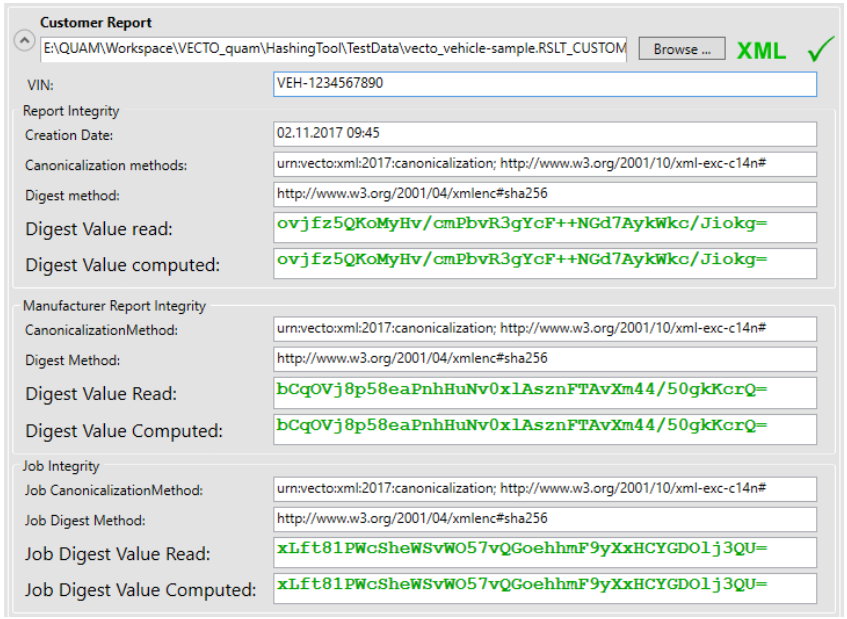
Report integrity

Job Integrity vs. manufacturer report

Information on vehicle and all components

European Commission

Verify Reports: Customer Information File



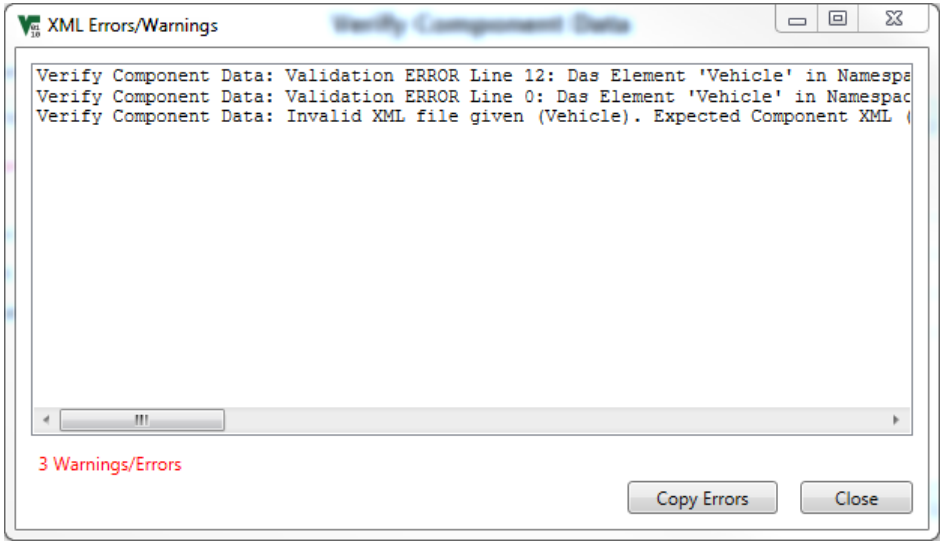
Report integrity

Manufacturer report vs. customer report

Job Integrity vs. customer report

European Commission

Error Dialog



European Commission

Using the VectoHashing Library

- Vecto Hashing Tool
  - GUI or as .Net Library
  - Supports all hash-related functionality

- Example

```
var h = VectoHash.Load(xml);
var hashed = h.AddHash();
hashed.WriteTo(writer);
```