tcpdf 9.0.0

component: interface, class

Tibor Roskó

Content

XMP metadata manipulator

namespace: tcpdf\metadata

use \tcpdf\parser\Parser
use \tcpdf\basic\Statics

final class XMP

    private $xref: array for xref information of pdf
    private $objects: array() for objects of pdf
    private $n: actual object number, starts with 0
    private $offsets: array for start offsets of objects
        example: "obj id first section"=>"offset" like "21"=>"12450"
    private $output: appended content, return value of manipulateMetadata method

    public function __construct(Parser $obj)
        Set the value of xref and objects.
        @incoming parameter
            A TCPDF Parser object.
        @return parameter

        Working process
            Set the xref and objects with the return array of getParsedData of Parser object.

    public function serializeObjBody()
        Preprocesses a given object body for easier rewrite into a file.
        @incoming parameter
            array: one of $this→objects value. Example $this→objects['2_0'].
        @return parameter
            array
        Working process
            Change numeric keyword to a space; objref reference to ' '.x x.' '.R instead of x_x;

    public function getXrefArgument($argument)
        Returns an attribute of trailer of $this→xref.
        @incoming parameter
            string: name of attribute
        @return parameter
            single value
        Working process
   1. Condition: trailer of xref is not empty;

public function getRootId()

Returns the root/Catalog object ID
@incoming parameter

@return parameter
    string: int_int
Working process
    Calls getXrefArgument method.

public function getMetadataObjId()

Returns the Catalog /Metadata object ID.
@incoming parameter
    array: root/Catalog object by serializeObjBody method
@return parameter
    string: int_int
Working process

public function getLastSize()

Returns the size value of $this→xref.
@incoming parameter

@return parameter
    int
Working process
    Calls getXrefArgument method.

public function getLastStartxref()

Returns startxref value of $this→xref.
@incoming parameter

@return parameter
    int
Working process

public function getLength()

Returns start length of pdf. This is before all modification. Length of $this→xref.
@incoming parameter

@return parameter
    int
Working process

public function getActualLength()
    Returns the amount of start length and length of $this→output.
    @incoming parameter

    @return parameter
      int
    Working process
      getLength()+length of $this→output

public function getNewObjId()
    Returns the next value of object ID.
    @incoming parameter

    @return parameter
      int
    Working process
      getLastSize()+$this→n

public function newMetadataObj($sign_array)
    Creates a metadata object with an RDF signatures block and append it to $this→output.
    @incoming parameter
      array: key-value pair array in numeric indexed array
    @return parameter

    Working process
1. Creates an RDF signatures block with Statics::getXMPSignature method.
2. Builds up RDF XMP structure with previous signatures block with xmpSignatureValidate.
3. Validates previous XMP against signatures namespace.
4. Creates a new metadata object with previous RDF XMP.
5. Sets $this→offsets, $this→n, $this→output.

public function createXref()
    Creates an xref table for new objects and append it to $this→output.
    @incoming parameter

    @return parameter

    Working process

public function xmpSignatureValidate(\DOMDocument $dom)
    Validates the signatures block of given document object.
    @incoming parameter
      \DOMDocument object
    @return parameter
      bool
    Working process
      Validates the signatures block against Statics::signNS() XSD.

public function manipulateMetadata($sign_array)

Adds signatures block to new or existing metadata object and returns the appendable string.

@incoming parameter

key-value array

1. array of signatures
    1. array of one signature
        1. email
        2. level
        3. vices: array of vices
            1. array of one vice
                1. email

An example: array(array("email"=>"info@example.com", "level"=>1, "vices"=>array(array("email"=>"info@example.com"))))

@return parameter

string

Working process

1. Condition: incoming parameter and Catalog body are not empty.
2. If metadata object exists then redefine it, add signatures block to existing RDF structure.
    1. Condition: metadata body is not empty; RDF block exists; incoming XMP is valid by xmpSignatureValidate;
3. If not then create a new metadata object with signatures block with newMetadataObj and add this object reference to Catalog object.
    1. Condition: size of xref is not empty; newMetadataObj XMP is valid by xmpSignatureValidate.
4. Calls createXref method.
5. return $this→output

public function readMetadataSignature()

Returns email=>level key-value pairs of a pdf XMP.

@incoming parameter

@return parameter

key-value array

1. array of signatures
    1. array of one signature
        1. email
        2. level
        3. vices: array of vices
            1. array of one vice
                1. email

An example: array(array("email"=>"info@example.com", "level"=>1, "vices"=>array(array("email"=>"info@example.com"))))

If vices array is empty, it is an empty array in result, like array().

Working process

1. Conditions: Catalog object exists; metadata object reference exists; metadata body exists;
2. Validates signatures block with xmpSignatureValidate method.
3. Returns the content of signatures block.

Statics

namespace: tcpdf\basic

use \tcpdf\webbasedcolor\Colors

class Statics

    public static function signNS()
        Returns the signatures XSD url.
        @incoming parameter

        @return parameter

        Working process

    public static function getXMPSignature($sign_array)
        Returns an RDF signatures block which contains given data.
        @incoming parameter
           array: key-value pair array in numeric indexed array

1. array of signatures
    1. array of one signature
        1. email
        2. level
        3. vices: array of vices
            1. array of one vice
                1. email

An example: array(array("email"=>"info@example.com", "level"=>1, "vices"=>array(array("email"=>"info@example.com"))))
        @return parameter
           string
        Working process