

# การพัฒนาใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3

## DEVELOPING ELECTRONIC TAX INVOICE BASES ON PDF/A-3

เวอร์ชัน 0.1.1

สำนักพัฒนาดิจิทัลเพื่อธุรกิจ

สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)

กระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม

ICS [xxx.xxx.xxx]

กรุณาเลือกวันที่ประกาศ

## คณะผู้จัดทำ

### ผู้จัดทำ

ดร.สุจิตรา ตัวริง	บริษัท อินบาเทค จำกัด
นายโยฮานเนส ตัวริง	บริษัท อินบาเทค จำกัด
นายพรีตตริค ฮอลล์	บริษัท อินบาเทค จำกัด
นายณธรรม ธรรมานิชาพันธ์	สำนักพัฒนาดิจิทัลเพื่อธุรกิจ สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)
นายประกาศิต เจริญรักษา	สำนักพัฒนาดิจิทัลเพื่อธุรกิจ สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)

### ให้ความคิดเห็นและตรวจทาน

นายสรณันท์ จิระสุรัตน์	รองผู้อำนวยการสำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ และ รักษาการผู้อำนวยการสำนักพัฒนาดิจิทัลเพื่อธุรกิจ สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)
นายศุภโชค จันทระประทีน	ผู้อำนวยการสำนักมาตรฐาน สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)
นายธิตกร ตระกูลศิริศักดิ์	รองผู้อำนวยการ และรักษาการผู้อำนวยการสำนักบริการโครงสร้างพื้นฐาน สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)
นางสาวนิษฐ์ ผาทอง	รองผู้อำนวยการสำนักพัฒนาดิจิทัลเพื่อธุรกิจ สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)
นางสาวจริยา สุขถนอมวงศ์	ผู้จัดการส่วนงาน Digital Transformation สำนักพัฒนาดิจิทัลเพื่อธุรกิจ สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)
นายชาณิน เลหาพันธุ์	สำนักวิจัยและพัฒนา สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)
นายพุทธิพร หงษ์สุรกุล	สำนักวิจัยและพัฒนา สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)

นายเฉลิมชัย บวรนนท์

สำนักมาตรฐาน

สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)

นายนพดล วรรณประภา

สำนักมาตรฐาน

สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)

เอกสารฉบับร่าง

เอกสารเผยแพร่ เรื่อง การพัฒนาใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 ฉบับนี้ จัดทำขึ้นเพื่อ  
เสนอแนะการพัฒนาใบกำกับภาษีในรูปแบบอิเล็กทรอนิกส์ (e-Tax Invoice Implementation Guideline) ซึ่ง  
เป็นเอกสารสำหรับผู้พัฒนาระบบไว้ใช้เป็นแนวทางในการสร้างใบกำกับภาษีอิเล็กทรอนิกส์ ให้มีความน่าเชื่อถือ  
และสอดคล้องกับข้อเสนอแนะมาตรฐาน ชมธอ. 3-2560 โดยพัฒนาตามแนวมาตรฐานของ

- ZUGFeRD (Central User Guidelines of the Forum for Electronic Invoicing in Germany)
- ISO 19005-3 Document management – Electronic document file format for long-term preservation – Part 3: Use of ISO 32000-1 with support for embedded files (PDF/A-3)
- ข้อเสนอแนะมาตรฐานด้านเทคโนโลยีสารสนเทศและการสื่อสารที่จำเป็นต่อธุรกรรมทางอิเล็กทรอนิกส์ ว่าด้วย  
ข้อความอิเล็กทรอนิกส์สำหรับการซื้อขายสินค้าและบริการ เวอร์ชัน 2.0 (ชมธอ. 3-2560)

สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน)

อาคารเดอะ ไนน์ ทาวเวอร์ แกรนด์ พระรามเก้า (อาคารบี) ชั้น 21

เลขที่ 33/4 ถนนพระราม 9 แขวงห้วยขวาง เขตห้วยขวาง กรุงเทพมหานคร 10210

โทรศัพท์: 0 2123 1234 โทรสาร: 0 2123 1200

## คำนำ

สืบเนื่องจากระเบียบกรมสรรพากร ว่าด้วยการจัดทำ ส่งมอบ และเก็บรักษาใบกำกับภาษีอิเล็กทรอนิกส์ผ่านระบบ e - Tax Invoice by Email พ.ศ. 2560 ข้อ 12 วรรค 4 ได้กำหนดให้การจัดเตรียมข้อมูลอิเล็กทรอนิกส์เพื่อส่งผ่านระบบ e - Tax Invoice by Email ที่ได้กระทำตั้งแต่ปี พ.ศ. 2561 เป็นต้นไป จะต้องอยู่ในรูปแบบ PDF/A-3 เท่านั้น สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน) หรือ สพธอ. จึงได้จัดทำเอกสารเผยแพร่ฉบับนี้ขึ้น เพื่ออธิบายให้ผู้พัฒนาซอฟต์แวร์ได้เข้าใจถึงที่มา การสร้าง และการใช้งานใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 อันประกอบด้วยข้อมูล XML ตามรูปแบบที่กำหนดในข้อเสนอแนะมาตรฐานด้านเทคโนโลยีสารสนเทศและการสื่อสารที่จำเป็นต่อธุรกรรมทางอิเล็กทรอนิกส์ ว่าด้วยข้อความอิเล็กทรอนิกส์สำหรับการซื้อขายสินค้าและบริการ เวอร์ชัน 2.0 (ชมธอ. 3-2560)

ทั้งนี้ ใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 (ISO 19005-3) เป็นมาตรฐานสากลของเอกสารอิเล็กทรอนิกส์ที่เหมาะสมกับการเปิดอ่านและพิมพ์ออก โดยเนื้อหาและการแสดงผลของเอกสารจะคงที่เสมอเพื่อรองรับการแสดงผลในระยะยาว รวมถึงรองรับการแนบเอกสาร XML ที่จะใช้ในการประมวลผล อีกทั้งจะช่วยลดต้นทุนและเวลาในการจัดส่งใบกำกับภาษีในรูปแบบกระดาษ ให้กับทุกภาคส่วนที่เกี่ยวข้องได้

# สารบัญ

	หน้า
1. ขอบข่าย .....	1
1.1 ข้อความสำคัญในใบกำกับภาษี	1
1.2 ข้อความสำคัญในใบเพิ่มหนี้	2
1.3 ข้อความสำคัญในใบลดหนี้	2
2. บทนิยาม.....	3
3. ข้อกำหนดทางเทคนิคในการจัดทำใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3.....	5
3.1 มาตรฐานเอกสารอิเล็กทรอนิกส์ PDF/A	5
3.2 ข้อกำหนดทางเทคนิคของใบกำกับภาษีอิเล็กทรอนิกส์	7
4. กระบวนการสร้างใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 .....	11
4.1. การสร้าง PDF/A-3 จาก XML	12
4.2 การสร้าง PDF/A-3 จากไฟล์ PDF	19
5. การตรวจสอบความถูกต้องของใบกำกับภาษีอิเล็กทรอนิกส์.....	23
5.1 การตรวจสอบรายการข้อมูลของใบกำกับภาษีตามกฎหมาย	23
5.2 การตรวจสอบโครงสร้างของข้อมูลในใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XML (STRUCTURAL VALIDATION)	23
5.3 การตรวจสอบข้อกำหนดของใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XML (RULES-BASED VALIDATION)	26
5.4 การตรวจสอบ METADATA ของใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XMP	28
5.5 การตรวจสอบข้อมูล CATALOG DICTIONARY ของใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF	30
5.6 การเปรียบเทียบ VISUAL REPRESENTATION และข้อมูลใน XML	38
6. APPENDIX .....	42
APPENDIX A	42
APPENDIX B	44
APPENDIX C	49
APPENDIX D	50
APPENDIX E	50

APPENDIX F	51
APPENDIX G	53
APPENDIX H	54
APPENDIX I	55
APPENDIX J	56
APPENDIX K	57
APPENDIX L	62
APPENDIX M	62
7. เอกสารอ้างอิง.....	63

## สารบัญรูป

หน้า

รูปที่ 1 ตัวอย่างข้อความของ XMP.....	8
รูปที่ 2 โครงสร้างของไฟล์ PDF สำหรับใบกำกับภาษีอิเล็กทรอนิกส์.....	9
รูปที่ 3 การสร้างใบกำกับภาษีอิเล็กทรอนิกส์รูปแบบ PDF/A-3 ด้วย XSLT .....	12
รูปที่ 4 ตัวอย่างโครงสร้างไฟล์ XML.....	13
รูปที่ 5 ตัวอย่างไฟล์ XSLT ส่วนแรกสำหรับกำหนดโครงสร้างของไฟล์ HTML.....	14
รูปที่ 6 ตัวอย่างไฟล์ XSLT ส่วนที่สองสำหรับรายละเอียดเบื้องต้นของใบกำกับภาษี .....	15
รูปที่ 7 ตัวอย่างข้อมูลจาก XML ภายใต้ TAG RSM:EXCHANGEDOCUMENT .....	16
รูปที่ 8 ตัวอย่าง HTML ที่ได้จากการแปลงไฟล์ XML .....	16
รูปที่ 9 ตัวอย่าง CODE การสร้าง XMP ด้วย iTEXTSHARP.....	16
รูปที่ 10 ตัวอย่าง MAIN FUNCTION สำหรับแปลงไฟล์ XML ไปเป็น HTML.....	17
รูปที่ 11 แสดง SOURCE CODE สร้าง PDF จาก HTML .....	18
รูปที่ 12 ตัวอย่างใบกำกับภาษีอิเล็กทรอนิกส์ ที่ถูกสร้างจาก HTML.....	18
รูปที่ 13 การสร้างใบกำกับภาษีอิเล็กทรอนิกส์รูปแบบ PDF/A-3 จากไฟล์ PDF.....	19
รูปที่ 14 ตัวอย่างไฟล์ TEMPLATE XMP สำหรับแนบไปกับเอกสาร .....	20
รูปที่ 15 ตัวอย่าง SOURCE CODE การสร้าง INVOICE จาก PDF และ XML .....	21
รูปที่ 16 ตัวอย่างใบกำกับภาษีอิเล็กทรอนิกส์ ที่ถูกสร้างจากไฟล์ PDF.....	22
รูปที่ 17 ตัวอย่างโครงสร้างข้อมูลของไฟล์ใบกำกับภาษีอิเล็กทรอนิกส์ที่ระบุใน XML.....	24
รูปที่ 18 กระบวนการตรวจสอบด้วย SCHEMA โดยใช้ XMLREADER และ XMLSCHEMASET.....	25
รูปที่ 19 ตัวอย่างการระบุไฟล์ SCHEMA ของ XML ที่ใช้ตรวจสอบ.....	25
รูปที่ 20 ตัวอย่างการตั้งค่าการตรวจสอบเป็นการตรวจสอบด้วย SCHEMA.....	25
รูปที่ 21 ตัวอย่างการตั้งค่าให้แสดงข้อความ เมื่อตรวจสอบพบสิ่งที่ไม่ตรงตาม SCHEMA.....	25
รูปที่ 22 ตัวอย่างการสร้าง XMLREADER สำหรับไฟล์ XML ที่ต้องการตรวจสอบ เลือกการตั้งค่าที่ระบุไว้ และการสั่งอ่านไฟล์.....	26
รูปที่ 23 วิธีการใช้งาน SCHEMATRON FOR ANT .....	27
รูปที่ 24 ตัวอย่าง ANT TASK ในไฟล์ BUILD .....	27
รูปที่ 25 ตัวอย่างผลการตรวจสอบไฟล์ XML ที่ไม่ถูกต้องตามข้อกำหนดของ SCHEMATRON .....	28



รูปที่ 26 ตัวอย่างผลการตรวจสอบไฟล์ XML ที่ถูกต้องตามข้อกำหนดของ SCHEMATRON .....	28
รูปที่ 27 ตัวอย่างหน้าจอ iTEXT RUPS และเมนูสำหรับเลือกไฟล์ PDF .....	29
รูปที่ 28 ตัวอย่างหน้าจอเมื่อเลือก BROWSE ไฟล์ PDF สำเร็จ.....	29
รูปที่ 29 ตัวอย่างการเลือก OBJECT เพื่อแสดงข้อมูล METADATA ในรูปแบบ XMP บน iTEXT RUPS .....	30
รูปที่ 30 ตัวอย่างหน้าจอแสดงข้อมูล METADATA ในรูปแบบ XMP บน iTEXT RUPS.....	30
รูปที่ 31 ตัวอย่างการเลือก OBJECT เพื่อค้นหา INDIRECT REFERENCE ของ ไฟล์ใบกำกับภาษีในรูปแบบ XML บน iTEXT RUPS .....	32
รูปที่ 32 ตัวอย่างการเลือก OBJECT เพื่อแสดงข้อมูลชื่อไฟล์ใบกำกับภาษีในรูปแบบ XML (/F) บน iTEXT RUPS .....	33
รูปที่ 33 ตัวอย่างการเลือก OBJECT เพื่อแสดงข้อมูลชื่อไฟล์ที่เป็น UNICODE ENCODING (/UF) บน iTEXT RUPS .....	34
รูปที่ 34 ตัวอย่างการเลือก OBJECT เพื่อแสดงความสัมพันธ์ระหว่างไฟล์ PDF และไฟล์ใบกำกับภาษีบน iTEXT RUPS.....	35
รูปที่ 35 ตัวอย่างการเลือก OBJECT เพื่อแสดงแสดงคำอธิบาย (DESCRIPTION) ของไฟล์ใบกำกับภาษีบน iTEXT RUPS .....	36
รูปที่ 36 ตัวอย่างการเลือก OBJECT เพื่อแสดงข้อมูลประเภทของไฟล์ (/SUBTYPE) บน iTEXT RUPS .....	37
รูปที่ 37 ตัวอย่างการเลือก OBJECT เพื่อแสดงข้อมูลวันที่ที่แก้ไขไฟล์ (/MODDATE) บน iTEXT RUPS.....	38

## สารบัญตาราง

หน้า

ตารางที่ 1 แสดงข้อมูล METADATA ที่เกี่ยวกับไฟล์ประเภท PDF/A-3 และไฟล์ข้อมูล XML .....	7
ตารางที่ 2 รายละเอียดของไฟล์ XML ที่กำหนดใน PDF CATALOG DIRECTORY .....	10
ตารางที่ 3 ค่าสำหรับตรวจสอบโครงสร้างของข้อมูลที่ระบุในใบกำกับภาษี .....	24
ตารางที่ 4 ค่า ที่ระบุใน ANT TASK สำหรับการตรวจสอบ .....	26
ตารางที่ 5 ข้อมูล CATALOG DICTIONARY ของไฟล์ใบกำกับภาษี และค่าที่แสดงในการตรวจสอบด้วยเครื่องมือ ITEXT RUPS .....	31
ตารางที่ 6 เปรียบเทียบระหว่างข้อมูลในรูปแบบ PDF และในรูปแบบ XML .....	39

## 1. ขอบข่าย

ปัจจุบันกรมสรรพากรได้กำหนดให้ผู้ประกอบการที่จดทะเบียนภาษีมูลค่าเพิ่ม เริ่มจัดทำใบกำกับภาษีให้อยู่ในรูปแบบข้อมูลอิเล็กทรอนิกส์ เพื่อนำส่งให้คู่ค้าภายใต้โครงการ e-Tax Invoice by Email โดยภายในปี 2561 ข้อมูลใบกำกับภาษีอิเล็กทรอนิกส์ดังกล่าวจะต้องอยู่ในรูปแบบ PDF/A-3

เอกสารฉบับนี้ จึงจะกล่าวถึงแนวทางในการพัฒนาใบกำกับภาษีอิเล็กทรอนิกส์ ให้เป็นรูปแบบ PDF/A-3 เพื่อสนับสนุนโครงการจัดทำ ส่งมอบ และเก็บรักษาใบกำกับภาษีอิเล็กทรอนิกส์สำหรับผู้ประกอบการที่มีความสนใจ เพื่อใช้งานภายใต้โครงการ e-Tax Invoice by Email ซึ่งเป็นโครงการความร่วมมือของสำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์ (องค์การมหาชน) (สพธอ.) สำนักงานรัฐบาลอิเล็กทรอนิกส์ (องค์การมหาชน) (สรอ.) และกรมสรรพากร และเป็นส่วนหนึ่งของโครงการ 3 ระบบภาษีและเอกสารธุรกรรมอิเล็กทรอนิกส์ ของการพัฒนาโครงสร้างพื้นฐานระบบการชำระเงินแบบอิเล็กทรอนิกส์แห่งชาติ (National e-Payment)

ในการจัดทำใบกำกับภาษี แม้จะอยู่ในรูปแบบข้อมูลอิเล็กทรอนิกส์ แต่ก็ยังต้องมีข้อมูลสำคัญตามที่กำหนดไว้ในประมวลรัษฎากร ซึ่งคำจำกัดความของ “ใบกำกับภาษีอิเล็กทรอนิกส์” ของระบบ e-Tax Invoice by Email หมายความว่า ใบกำกับภาษีแบบเต็มรูป ตามมาตรา 86/4 แห่งประมวลรัษฎากร ที่ได้มีการจัดทำข้อความขึ้นเป็นข้อมูลอิเล็กทรอนิกส์ซึ่งได้มีการประทับรับรองเวลาโดยระบบ e-Tax Invoice by Email ของสพธอ. และหมายความรวมถึง ใบเพิ่มเติมตามมาตรา 86/9 และใบลดหนี้ตามมาตรา 86/10 แห่งประมวลรัษฎากร ซึ่งเอกสารทั้ง 3 ประเภท มีข้อความสำคัญที่ต้องแสดงไว้ ดังนี้ (กรมสรรพากร, 2014)

### 1.1 ข้อความสำคัญในใบกำกับภาษี

- 1) คำว่า “ใบกำกับภาษี” ในที่ที่เห็นได้เด่นชัด
- 2) ชื่อ ที่อยู่ และเลขประจำตัวผู้เสียภาษีอากรของผู้ประกอบการจดทะเบียนที่ออกใบกำกับภาษี
- 3) ชื่อ ที่อยู่ และเลขประจำตัวผู้เสียภาษีอากรของผู้ซื้อสินค้าหรือผู้รับบริการ
- 4) หมายเลขลำดับของใบกำกับภาษี และหมายเลขของเล่ม (ถ้ามี)
- 5) ชื่อ ชนิด ประเภท ปริมาณ และมูลค่าของสินค้าหรือของบริการ
- 6) จำนวนภาษีมูลค่าเพิ่มที่คำนวณจากมูลค่าของสินค้าหรือของบริการ โดยให้แยกออกจากมูลค่าของสินค้าหรือของบริการให้ชัดเจน
- 7) วัน เดือน ปี ที่ออกใบกำกับภาษี

## 1.2 ข้อความสำคัญในใบเพิ่มหนี้

- 1) คำว่า "ใบเพิ่มหนี้" ในที่ที่เห็นได้เด่นชัด
- 2) ชื่อ ที่อยู่ และเลขประจำตัวผู้เสียภาษีอากรของผู้ประกอบการจดทะเบียนที่ออกใบเพิ่มหนี้
- 3) ชื่อ ที่อยู่ และเลขประจำตัวผู้เสียภาษีอากรของผู้ซื้อสินค้าหรือผู้รับบริการ
- 4) วัน เดือน ปี ที่ออกใบเพิ่มหนี้
- 5) หมายเลขลำดับของใบกำกับภาษีเดิม รวมทั้งหมายเลขลำดับของเล่ม (ถ้ามี) มูลค่าของสินค้าหรือบริการที่แสดงไว้ในใบกำกับภาษีดังกล่าว มูลค่าที่ถูกต้องของสินค้าหรือบริการ ผลต่างของจำนวนมูลค่าทั้งสองและจำนวนภาษีที่เรียกเก็บเพิ่มสำหรับส่วนต่างนั้น
- 6) คำอธิบายสั้น ๆ ถึงสาเหตุในการออกใบเพิ่มหนี้

## 1.3 ข้อความสำคัญในใบลดหนี้

- 1) คำว่า "ใบลดหนี้" ในที่ที่เห็นได้เด่นชัด
- 2) ชื่อ ที่อยู่ และเลขประจำตัวผู้เสียภาษีอากรของผู้ประกอบการจดทะเบียนที่ออกใบลดหนี้
- 3) ชื่อ ที่อยู่ และเลขประจำตัวผู้เสียภาษีอากรของผู้ซื้อสินค้าหรือผู้รับบริการ
- 4) วัน เดือน ปี ที่ออกใบลดหนี้
- 5) หมายเลขลำดับของใบกำกับภาษีเดิม รวมทั้งหมายเลขลำดับของเล่ม (ถ้ามี) มูลค่าของสินค้าหรือบริการที่แสดงไว้ในใบกำกับภาษีดังกล่าว มูลค่าที่ถูกต้องของสินค้าหรือบริการ ผลต่างของจำนวนมูลค่าทั้งสองและจำนวนภาษีที่ใช้คืนสำหรับส่วนต่างนั้น
- 6) คำอธิบายสั้น ๆ ถึงสาเหตุในการออกใบลดหนี้

ใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 จะประกอบด้วยข้อมูลสำหรับแสดงผลเพื่อการอ่านของบุคคลในรูปแบบของ PDF และข้อมูลสำหรับการประมวลผลด้วยคอมพิวเตอร์ในรูปแบบของไฟล์ XML ที่มีโครงสร้างตามข้อเสนอแนะมาตรฐานฯ ชมธอ. 3-2560 ซึ่งในบทที่ 3 จะกล่าวถึงข้อกำหนดทางเทคนิคในการจัดทำใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3

## 2. บทนิยาม

ความหมายของคำที่ใช้ในเอกสารเผยแพร่ ฉบับนี้ มีดังต่อไปนี้

- 1) ข้อความ หมายถึง เรื่องราวหรือข้อเท็จจริง ไม่ว่าจะปรากฏในรูปแบบของตัวอักษร ตัวเลข เสียง ภาพ หรือรูปแบบอื่นใดที่สื่อความหมายได้โดยสภาพของสิ่งนั้นเองหรือโดยผ่านวิธีการใดๆ
- 2) ข้อมูลอิเล็กทรอนิกส์ หมายถึง ข้อความที่ได้สร้าง ส่ง รับ เก็บรักษา หรือประมวลผลด้วยวิธีการทางอิเล็กทรอนิกส์
- 3) ใบกำกับภาษีอิเล็กทรอนิกส์ (e-Tax Invoice) หมายถึง ใบกำกับภาษีตามมาตรา 86/4 แห่งประมวลรัษฎากร ที่ได้มีการจัดทำข้อความขึ้นเป็นข้อมูลอิเล็กทรอนิกส์ ให้หมายรวมถึง ใบเพิ่มหนี้ตามมาตรา 86/9 และใบลดหนี้ตามมาตรา 86/10 แห่งประมวลรัษฎากร ที่ได้มีการจัดทำข้อความขึ้นเป็นข้อมูลอิเล็กทรอนิกส์
- 4) ธุรกิจทางอิเล็กทรอนิกส์ หมายถึง ธุรกิจที่กระทำขึ้นโดยใช้วิธีการทางอิเล็กทรอนิกส์ทั้งหมดหรือบางส่วน
- 5) อิเล็กทรอนิกส์ หมายถึง การประยุกต์ใช้วิธีการทางอิเล็กทรอนิกส์ โทรน ไฟฟ้า คลื่นแม่เหล็กไฟฟ้า หรือวิธีอื่นใดในลักษณะคล้ายกัน และให้หมายความรวมถึงการประยุกต์ใช้วิธีการทางแสง วิธีการทางแม่เหล็ก หรืออุปกรณ์ที่เกี่ยวข้องกับการประยุกต์ใช้วิธีต่างๆ เช่นว่านั้น
- 6) Adobe's Extensible Metadata Platform (XMP) คือ ไฟล์ประเภทหนึ่งที่สามารถฝังค่า metadata ไปยังไฟล์ในตอนที่สร้างไฟล์
- 7) Cascading Style Sheet (CSS) เป็นภาษาที่ใช้ในการจัดรูปแบบแสดงผลเอกสาร HTML ให้มีรูปแบบสวยงาม
- 8) Comma Separated Values (CSV) เป็นไฟล์ข้อความประเภทหนึ่งที่จัดเก็บข้อมูลในรูปแบบตารางโดยใช้เครื่องหมายจุลภาค หรือคอมม่า (,) เป็นตัวแบ่งเก็บคอลัมน์
- 9) Computer Aided Design (CAD) คือ โปรแกรมคอมพิวเตอร์ที่ใช้ในการออกแบบ
- 10) Excel Spreadsheet คือ เป็นโปรแกรมที่ใช้ในการเก็บข้อมูลในรูปตารางของ Microsoft office
- 11) Extensible Markup Language (XML) หมายถึง ภาษา Markup สำหรับการแลกเปลี่ยนข้อมูลระหว่างเครื่องคอมพิวเตอร์ต่างระบบกัน พัฒนาโดย W3C
- 12) Extensible Stylesheet Language Transformations (XSLT) หมายถึง ภาษาสำหรับการเปลี่ยนเอกสาร XML ลงในเอกสาร XML อื่น ๆ หรือ ในรูปแบบอื่น ๆ เช่น HTML สำหรับหน้าเว็บ, ข้อความธรรมดาหรือรูปแบบ XSL วัตถุซึ่งอาจต่อมาถูกแปลงเป็นรูปแบบอื่น ๆ เช่น PDF, PostScript และ PNG
- 13) Javascript เป็นภาษาสคริปต์เชิงวัตถุ ใช้ในการสร้างและพัฒนาเว็บไซต์ ถูกพัฒนาขึ้นโดย เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation)
- 14) Lempel-Ziv-Welch (LZW) เป็นระบบการอัดบีบแบบไม่สูญเสียรายละเอียดภาพ

- 15) Opentype คือ ฟอนต์ประเภทหนึ่งที่สามารถบรรจุตัวอักษรได้มากถึง 65,000 ตัวอักษร สามารถเก็บตัวอักษรแปลก ๆ อักษรพิเศษได้ และยังสามารถทำงานได้กับทุก platform เป็นผลงานจากความร่วมมือกันของ Adobe และ Microsoft ซึ่งเป็นไปตามมาตรฐาน ISO/IEC 14496-22:2015
- 16) Portable Document Format (PDF) หมายถึง รูปแบบไฟล์ลักษณะหนึ่ง พัฒนาโดย บริษัท Adobe System สำหรับแสดงเอกสารที่สามารถใช้งานได้ในทุกระบบ ยังคงลักษณะเอกสารเหมือนต้นฉบับ โดยสามารถจัดเก็บและแสดง ตัวอักษร รูปภาพ รูปลายเส้น ปัจจุบันประกาศเป็นมาตรฐานสากล
- 17) PDF/A คือ มาตรฐานของ Adobe Portable Document Format (PDF) ที่ศาลรัฐบาลกลางแห่งสหรัฐอเมริกา บังคับใช้สำหรับการนำเสนอและการจัดเก็บข้อมูลถาวร (A ย่อมาจาก Archiving) ในรูปของเอกสารPDF
- 18) PDF Advanced Electronic Signatures (PAdES) คือ มาตรฐานของการเพิ่ม Advanced Electronic Signature บน PDF ที่จัดทำขึ้นโดย European Telecommunications Standards Institute (ETSI)
- 19) sRGB (Standard RGB) เป็นมาตรฐานสำหรับช่องว่างของสี (color space)
- 20) Unicode คือมาตรฐานอุตสาหกรรมที่คอมพิวเตอร์ใช้แสดงผลและจัดการข้อความธรรมดาที่ใช้ในระบบการเขียนของภาษาส่วนใหญ่ในโลกได้อย่างสอดคล้องกัน Unicode ประกอบด้วยรายการอักขระที่แสดงผลได้มากกว่า 100,000 ตัว พัฒนาต่อยอดมาจากมาตรฐานชุดอักขระสากล (Universal Character Set: UCS) และมีการตีพิมพ์ลงในหนังสือ The Unicode Standard เป็นแผนผังรหัสเพื่อใช้เป็นรายการอ้างอิง นอกจากนี้ยังมีการอธิบายวิธีการที่ใช้เข้ารหัสและการนำเสนอมาตรฐานของการเข้ารหัสอักขระอีกจำนวนหนึ่ง การเรียงลำดับอักขระ กฎเกณฑ์ของการรวมและการแยกอักขระ รวมไปถึงลำดับการแสดงผลของอักขระสองทิศทาง เช่นอักขระอาหรับหรืออักขระฮิบรูที่เขียนจากขวาไปซ้าย
- 21) iText คือไลบรารีสำหรับสร้าง PDF ซึ่งมีทั้งลักษณะที่เป็นไลบรารีทางการค้า (Commercial) และ Open source
- 22) Well-Formed คือ เอกสาร XML ที่มีไวยากรณ์ที่ถูกต้อง
- 23) XML Schema คือ โครงสร้างและชนิดของข้อมูลในไฟล์เอกสาร XML โดยตัวอย่างไฟล์นามสกุล (Extension file type) ของไฟล์โครงสร้าง XML ได้แก่ .xsd
- 24) XML Schematron คือ รูปแบบภาษาเพื่อเอาไว้กำหนดข้อกำหนดทางธุรกิจ (Business Rule) ของข้อมูลในไฟล์เอกสาร XML โดยตัวอย่างไฟล์นามสกุล (Extension file type) ของรูปแบบข้อกำหนดดังกล่าวได้แก่ .sch

### 3. ข้อกำหนดทางเทคนิคในการจัดทำใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3

ใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 นั้น สามารถคงลักษณะการแสดงผลได้เช่นเดียวกับตอนสร้าง โดยไม่ขึ้นกับเครื่องมือ หรือระบบที่ใช้ในการสร้าง เก็บรักษา หรือนำมาแสดง เนื่องจากไฟล์ที่ถูกสร้างให้เป็นไปตามมาตรฐานของ PDF/A (PDF for Long-term Archiving) จะบรรจุข้อมูลทั้งหมดที่จำเป็นต่อการแสดงผล เช่น รูปภาพ ฟอนต์ ข้อมูลสี ไว้แล้ว ซึ่งใบกำกับภาษีในรูปแบบ PDF/A-3 นี้ จะยังมีข้อมูลสำหรับการประมวลผลในรูปแบบของไฟล์ XML แบนอยู่ด้วย

ทั้งนี้เอกสารอิเล็กทรอนิกส์ ตามมาตรฐานของ PDF/A ต้องจัดทำให้มีคุณสมบัติ ดังนี้

- 1) ห้ามเข้ารหัสลับ (encrypt) ไฟล์ เพื่อให้พร้อมแสดงผลตลอดเวลา
- 2) ห้ามมีส่วนประกอบที่เป็นไฟล์เสียง ภาพเคลื่อนไหว JavaScript หรือเรียกประมวลผลไฟล์ที่เป็น Executable เนื่องจากการเรียกใช้ซอฟต์แวร์ภายนอกสำหรับการแสดงผลส่วนประกอบเหล่านี้ อาจทำให้ PDF แสดงผลต่างออกไปจากที่ได้สร้างไว้
- 3) ให้แนบ (embed) ฟอนต์ที่ใช้อยู่ในไฟล์ PDF
- 4) ให้กำหนดสีที่ใช้ โดยไม่จำกัดกับอุปกรณ์ใดอุปกรณ์หนึ่ง (ICC color profiles)
- 5) ให้กำหนด Metadata ของไฟล์ ในรูปแบบที่เป็นมาตรฐาน เช่น XMP
- 6) ห้ามบีบอัดไฟล์ (compression) แบบ LZW เนื่องจากประเด็นเรื่องลิขสิทธิ์ แต่ยังสามารถใช้วิธีบีบอัดไฟล์แบบ ZIP หรือ JPEG ในไฟล์ PDF/A ได้

#### 3.1 มาตรฐานเอกสารอิเล็กทรอนิกส์ PDF/A

ปัจจุบันมาตรฐานเอกสารอิเล็กทรอนิกส์ PDF/A ได้มีการจัดทำออกมาเป็นจำนวน 3 รุ่น คือ PDF/A-1, PDF/A-2, และล่าสุดคือ PDF/A-3 ซึ่งสามารถแบ่งย่อยออกไปได้อีกตามระดับของความสอดคล้อง (Conformance Level) ที่จะส่งผลถึงคุณภาพของเอกสารอิเล็กทรอนิกส์ที่เก็บไว้ โดยขึ้นกับส่วนต่าง ๆ ที่ใช้ประกอบกันเป็นเอกสาร และวัตถุประสงค์ของเอกสารนั้น ซึ่งระดับของความสอดคล้อง ถูกแบ่งออกเป็น

- 1) ความสอดคล้องระดับ A (Accessible) ไฟล์ PDF จะต้องมีความสมบูรณ์ตามที่กำหนดให้เป็นมาตรฐานของ PDF/A ประเภทย่อยนั้น ๆ รวมถึงต้องระบุโครงสร้างของเอกสาร เพื่อรองรับการแสดงผลให้ผู้ใช้งานสามารถอ่านเอกสารได้ โดยจะนำตัวอักษรในเอกสารมาใช้งาน หรือค้นหาตัวอักษรได้ ไฟล์ PDF/A ที่มีความสอดคล้องระดับ A นี้ จะได้มาจากการแปลงเอกสารที่ถูกสร้างเป็นอิเล็กทรอนิกส์ตั้งแต่ต้น (born-digital documents) เท่านั้น

- 2) ความสอดคล้องระดับ B (Basic) ไฟล์ PDF/A ที่มีความสอดคล้องระดับ B จะสร้างง่ายกว่าระดับ A แต่หากจะค้นหาตัวอักษร หรือดึงตัวอักษรไปใช้ อาจทำได้ไม่สมบูรณ์นัก เอกสารกระดาษที่นำมาสแกน ก็สามารถแปลงเป็นไฟล์ PDF/A ที่มีความสอดคล้องระดับ B ได้
- 3) ความสอดคล้องระดับ U (Unicode) ถูกสร้างมาในช่วงเวลาเดียวกับ PDF/A-2 โดยปรับมาจาก PDF/A-2b ให้ทุกตัวอักษรสามารถจับคู่ได้กับอักขระ Unicode

เอกสารอิเล็กทรอนิกส์ PDF/A ในแต่ละรุ่น จะมีรายละเอียดที่แตกต่างกันดังนี้

#### 3.1.1 มาตรฐานเอกสารอิเล็กทรอนิกส์ PDF/A-1 (ISO 19005-1)

PDF/A-1 มีพื้นฐานมาจาก PDF เวอร์ชัน 1.4 ถูกเผยแพร่ในปี 2001 มีคุณสมบัติใกล้เคียงกับที่กล่าวมาข้างต้น รวมทั้งสามารถแนบลายมือชื่อดิจิทัล (Digital Signature) และการใช้ hyperlinks ได้ ซึ่ง PDF/A-1 มีระดับของความสอดคล้อง (Conformance Level) อยู่ 2 ระดับ คือ PDF/A-1a และ PDF/A-1b

#### 3.1.2 มาตรฐานเอกสารอิเล็กทรอนิกส์ PDF/A-2 (ISO 19005-2)

PDF/A-2 ถูกเผยแพร่ในปี 2011 มีพื้นฐานมาจาก PDF เวอร์ชัน 1.7 และจัดเป็นมาตรฐานสากล (ISO 32000-1) มีระดับของความสอดคล้อง อยู่ 3 ระดับ คือ PDF/A-2a PDF/A-2b และ PDF/A-2u และมีคุณสมบัติเพิ่มเติมจาก PDF/A-1 ซึ่งได้แก่

- 1) รองรับการบีบอัดรูปภาพแบบ JPEG2000
- 2) รองรับวัตถุที่พื้นหลังโปร่งใสและชั้นการแสดงผลหลายชั้น (PDF Layers) ได้
- 3) รองรับการแนบฟอนต์แบบ OpenType
- 4) รองรับการใส่ลายมือชื่อดิจิทัลที่เป็นไปตามมาตรฐาน PAdES (PDF Advanced Electronic Signature)
- 5) รองรับการแนบไฟล์ PDF/A เพื่อรวมเอกสารที่ต้องการจัดเก็บจากหลาย ๆ ไฟล์ เข้ามาเป็นไฟล์ PDF/A-2 ไฟล์เดียว

#### 3.1.3 มาตรฐานเอกสารอิเล็กทรอนิกส์ PDF/A-3 (ISO 19005-3)

PDF/A-3 ถูกเผยแพร่ในปี 2012 มีคุณสมบัติสำคัญที่ต่างจาก PDF/A-2 หนึ่งเรื่อง คือ รองรับการแนบไฟล์ประเภทอื่น นอกเหนือจาก PDF/A ลงในเอกสารได้ เช่น XML, CSV, CAD, Word Document, Excel Spreadsheet PDF/A-3 มีระดับของความสอดคล้องอยู่ 3 ระดับ เช่นกัน คือ PDF/A-3a PDF/A-3b และ PDF/A-3u



### 3.2 ข้อกำหนดทางเทคนิคของใบกำกับภาษีอิเล็กทรอนิกส์

การจัดทำใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 ให้สอดคล้องกับมาตรฐาน ISO 19005-3 นอกจากคุณสมบัติที่กล่าวมาข้างต้น ยังมีข้อกำหนดทางเทคนิคที่ต้องจัดทำดังนี้

- 1) เป็นไฟล์ PDF/A-3 ที่มีความสอดคล้องระดับ U (PDF/A-3u) เพื่อให้รองรับอักษรภาษาไทย
- 2) ระบุข้อมูล metadata ทั้งที่เกี่ยวกับไฟล์ประเภท PDF/A-3 และบริบทของข้อมูลใบกำกับภาษีในไฟล์ XML ในรูปแบบ Extensible Metadata Platform (XMP) ด้วยค่าตาม ตารางที่ 1 ดังนี้

ตารางที่ 1 แสดงข้อมูล Metadata ที่เกี่ยวกับไฟล์ประเภท PDF/A-3 และไฟล์ข้อมูล XML

No.	ชื่อ Tag ข้อมูล	จำนวน	รายละเอียด	ประเภทข้อมูล	ข้อมูลที่ต้องใส่
1	<pdfaid:part>	1..1	หมายเลข Part ของมาตรฐาน ISO 19005 เช่น ISO 19005-3 หมายถึง PDF/A-3	Integer	3
2	<pdfaid:conformance>	1..1	ระดับความสอดคล้อง (Level of Conformance) ที่ใช้ในการจัดทำ	String	U
3	DocumentFileName	1..1	ชื่อของไฟล์ XML ที่แนบ	String	ETDA-invoice.xml
4	DocumentType	1..1	ประเภทของเอกสาร (เป็นได้ค่าใดค่าหนึ่ง)	String	Tax Invoice Credit Note Debit Note
5	Version	1..1	เวอร์ชันของโครงสร้างข้อมูล	String	2.0

หมายเหตุ: "จำนวน" คือจำนวนครั้งในการปรากฏของข้อมูล (Cardinality) ซึ่งหมายถึงจำนวนครั้งต่ำสุดและสูงสุดของรายการข้อมูลนั้นที่สามารถปรากฏในข้อมูลอิเล็กทรอนิกส์ตามข้อกำหนดใน Cross Industry Invoice Data Model ของ UN/CEFACT โดยตัวเลขทางซ้ายแสดงจำนวนต่ำสุดที่รายการข้อมูลมีได้ในข้อมูลอิเล็กทรอนิกส์ และตัวเลขทางขวาแสดงจำนวนสูงสุดที่รายการข้อมูลมีได้ในข้อมูลอิเล็กทรอนิกส์

```

1 <rdf:Description xmlns:rsm="urn:etda:uncefact:data:standard
2 :Invoice_CrossIndustryInvoice:2" rdf:about="">
3 <rsm:DocumentFileName>ETDA-invoice.xml</rsm:DocumentFileName>
4 <rsm:DocumentType>Tax Invoice</rsm:DocumentType>
5 <rsm:Version>2.0</rsm:Version>
6 </rdf:Description></rdf:RDF>

```

รูปที่ 1 ตัวอย่างข้อความของ XMP

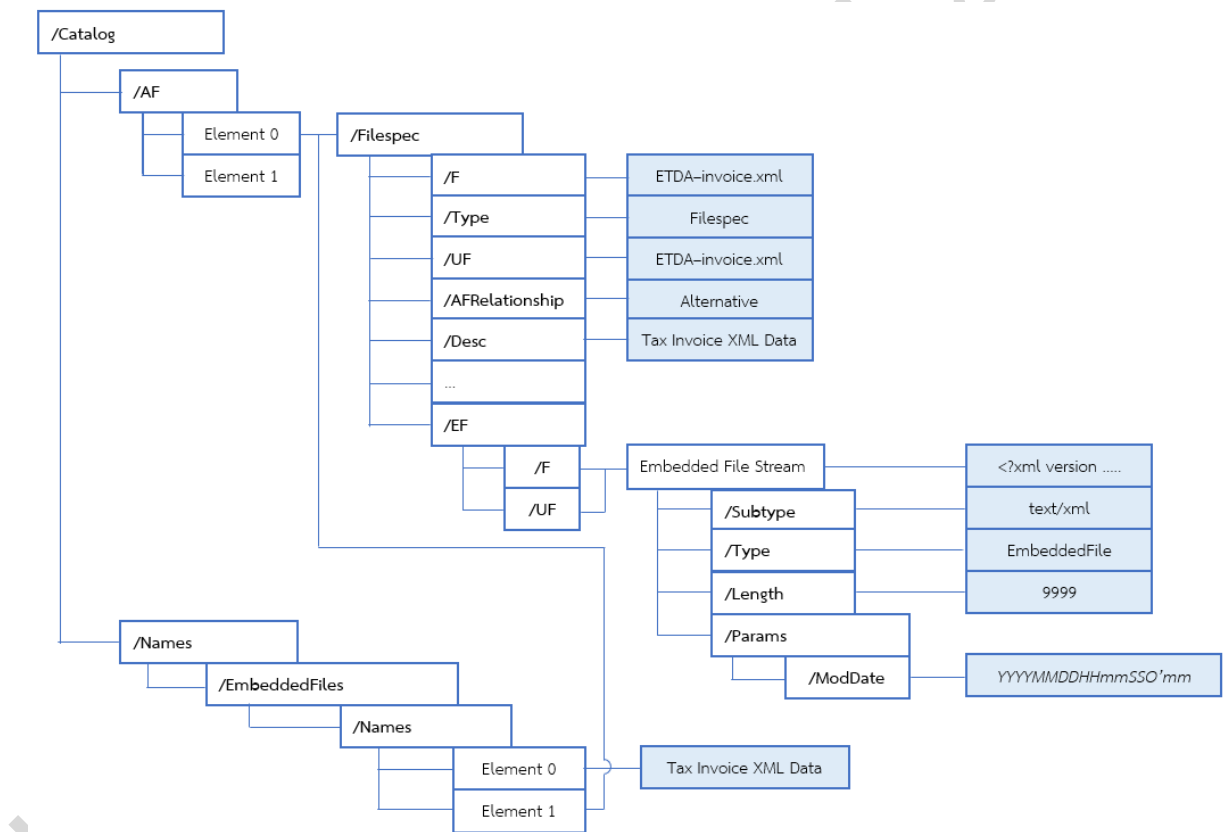
จากรูปที่ 1 เป็นตัวอย่าง XMP ของใบกำกับภาษีอิเล็กทรอนิกส์ที่ถูกแนบกับไฟล์ PDF ในขั้นตอนการสร้าง จะเห็นได้ว่าไฟล์ XMP บรรทัดที่ 3 จะมีระบุชื่อของไฟล์ XML ที่แนบ บรรทัดที่ 4 ระบุประเภทของเอกสาร และบรรทัดที่ 5 จะระบุเวอร์ชันของโครงสร้างข้อมูลข้อมูลดังกล่าวล้วนเป็นข้อมูล metadata ของเอกสารทั้งสิ้น ซึ่งสามารถดูข้อมูล XMP ทั้งหมดได้จาก Appendix E

- 3) มีข้อมูลใบกำกับภาษีที่เป็นไฟล์ XML (Extensible Markup Language) ซึ่งมีโครงสร้างตามข้อเสนอแนะมาตรฐานฯ ชมธอ. 3-2560 โดยต้องระบุรายละเอียดเกี่ยวกับไฟล์ XML ที่แนบดังต่อไปนี้
  - ชื่อไฟล์ XML ของข้อมูลใบกำกับภาษี กำหนดชื่อให้เป็น ETDA-invoice.xml ดังที่ระบุไว้ในฟิลด์ข้อมูล DocumentFileName ใน ตารางที่ 1 และระบุชื่อไฟล์ในส่วนของ Filespec จากรูปที่ 2 ภายได้ /Filespec ในส่วนของ /F และ /UF
  - ไฟล์ XML ของข้อมูลใบกำกับภาษีจะต้องแนบในระดับเอกสาร (Document Level) เนื่องจากแสดงถึงข้อมูลใบกำกับภาษีทั้งฉบับ ไม่ใช่ข้อมูลของเฉพาะหน้าใดหน้าหนึ่ง

ทั้งนี้เพื่อให้ไฟล์ PDF มีรายละเอียดของข้อมูลเอกสารแนบ ดังนั้นจึงจำเป็นต้องกำหนดรายละเอียดเกี่ยวกับไฟล์ XML ที่แนบในส่วนของ PDF File Specification Dictionary ภายได้ Catalog Dictionary ดังต่อไปนี้

- a. /AFRelationship: ระบุความสัมพันธ์ของไฟล์แนบเป็น Alternative เพื่อระบุว่าไฟล์ XML ดังกล่าวเป็นอีกรูปแบบหนึ่งของใบกำกับภาษีดังรูปที่ 2
- b. /Desc : ระบุคำอธิบายของไฟล์แนบเป็น Tax Invoice XML Data
- c. ในส่วนของ Embedded File Stream ให้ระบุรายละเอียดโดยอ้างอิงจากรูปที่ 2 ดังนี้
  - /Subtype: ระบุประเภทเป็น “text/xml” (ตาม MIME Media Type ที่ประกาศโดย IANA)
  - ModDate ในส่วนของ Params: ระบุวันเวลาที่แก้ไขไฟล์ XML ล่าสุดในรูปแบบ YYYYMMDDHHmmSSOHH'mm โดยมีคำอธิบายรูปแบบวันเวลาดังนี้  
 YYYY ตัวเลขระบุปี 4 ตัวอักษร  
 MM เดือน (01-12)

*DD* วันที่ (01-31)  
*HH* ชั่วโมง (00-23)  
*mm* นาที (00-59)  
*SS* วินาที (00-59)  
*O* ความสัมพันธ์ระหว่างเวลาที่ท้องถิ่นกับเวลา UTC  
*HH* offset จากเวลา UTC ในหน่วยชั่วโมง (00-23)  
*mm* offset จากเวลา UTC ในหน่วยนาที (00-59)  
 เช่น วันที่ 31 ธันวาคม 2017 เวลา 18:00 น. , GMT+7  
 จะได้ ModDate เป็น 201712311800+07'00



รูปที่ 2 โครงสร้างของไฟล์ PDF สำหรับใบกำกับภาษีอิเล็กทรอนิกส์

รายละเอียดเกี่ยวกับ XML ที่สำคัญจากข้อที่ 3 ที่จำเป็นต้องระบุในคุณลักษณะของไฟล์ PDF สามารถสรุปรายละเอียดทั้งหมดได้ดังตารางที่ 2

ตารางที่ 2 รายละเอียดของไฟล์ XML ที่กำหนดใน PDF Catalog Directory

No.	Parent Object	Object Name	Field name	Value
1	/AF	/Filespec	/F	ETDA-invoice.xml
2	/AF	/Filespec	/UF	ETDA-invoice.xml
3	/AF	/Filespec	/AFRelationship	Alternative
4	/AF	/Filespec	/Desc	Tax Invoice XML Data
5	/EF	Embedded File Stream	/Subtype	text/xml
6	/EF	/Params	/ModDate	YYYYMMDDHHmmSSO HH'mm

- 4) แบนพอนด์ทั้งหมดที่ถูกใช้ลงไปเอกสาร PDF/A-3 เพื่อว่าเมื่อเวลาผ่านไป ตัวอักษรบนเอกสาร จะแสดงผลได้อย่างสมบูรณ์ เนื่องจากเมื่อทำการเปิดเอกสาร PDF/A-3 แอปพลิเคชันที่ใช้ในการเปิดไฟล์ จะทำการอ่านพอนด์จากเอกสารโดยตรง และนำมาแสดงผล ซึ่งพอนด์ที่จะแนบไปกับเอกสารต้องมีความสอดคล้องกับข้อกำหนดของมาตรฐาน ISO 32000-1:2008 ข้อ 9.6 และ 9.7 (Adobe System Incorporated, 2008)
- 5) กำหนดสีที่ใช้ในเอกสาร (Color Profile) ให้เป็น sRGB แต่ในกรณีที่ผู้ออกใบกำกับภาษีมีความต้องการพิเศษเกี่ยวกับสีของรูปภาพที่ปรากฏในใบกำกับภาษีอิเล็กทรอนิกส์ ให้กำหนดเป็นค่าอื่นได้ในเอกสารนี้จะใช้เป็น sRGB IEC61966-2.1 ซึ่งเป็นโปรไฟล์ที่บรรจุข้อมูลสีขั้นพื้นฐาน สำหรับใช้กับหน้าจอคอมพิวเตอร์, สแกนเนอร์ และเครื่องพิมพ์ระดับล่าง ด้วยรูปแบบแม่สี RGB (red,green,blue) ขั้นตอนการแนบ Color Profile จะอยู่ในขั้นตอนการสร้าง PDF/A-3 ในบทที่ 4

#### 4. กระบวนการสร้างใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3

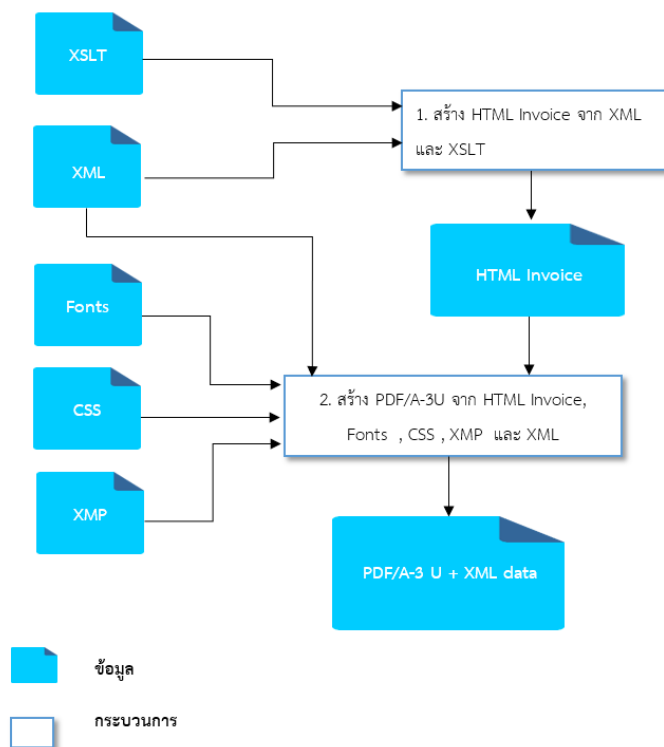
กระบวนการสร้างใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 สามารถสร้างได้ 2 วิธี ได้แก่

- 1) สร้างจากไฟล์ XML โดยใช้ XSLT แปลงไฟล์ XML เป็นไฟล์ HTML จากนั้นแปลงจาก HTML เป็น PDF/A-3 และแนบไฟล์ XML, XMP, Fonts, CSS เพิ่มเติม วิธีนี้เหมาะสำหรับผู้ที่มีความชำนาญในการสร้างเว็บไซต์ เนื่องจากในขั้นตอนการแปลง XML ไปเป็น HTML และ HTML ไปเป็น PDF/A-3 นั้นจะใช้ไฟล์ CSS ในการจัดองค์ประกอบและตกแต่งรูปลักษณ์ของไฟล์ PDF
- 2) สร้างจากไฟล์ PDF โดยแปลงจาก PDF เป็น PDF/A-3 และแนบไฟล์ XML, XMP เหมาะสำหรับผู้ที่มีระบบการสร้างไฟล์ PDF อยู่แล้ว ส่วนประกอบอื่นๆของ PDF/A-3 และแนบเข้าไปภายในไฟล์ PDF เพิ่มเติม

จากวิธีการสร้างทั้ง 2 วิธีจะใช้วิธีการสร้าง รูปลักษณ์ (Appearance) ของใบกำกับภาษีที่แตกต่างกัน อย่างไรก็ตามผลลัพธ์ที่ได้จากการสร้างใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 ทั้ง 2 วิธีจำเป็นต้องมีองค์ประกอบในรูปแบบเดียวกัน (สามารถดูองค์ประกอบได้จากบทที่ 3) ถึงจะทำให้ใบกำกับภาษีที่ถูกสร้างมานั้นอยู่ในรูปแบบ PDF/A-3 โดยสมบูรณ์

#### 4.1. การสร้าง PDF/A-3 จาก XML

สำหรับการสร้าง PDF/A-3 รูปแบบนี้ ประกอบไปด้วยองค์ประกอบ (Component) ที่สำคัญ และขั้นตอนการสร้างไฟล์ตามรูปที่ 3 ดังนี้



รูปที่ 3 การสร้างใบกำกับภาษีอิเล็กทรอนิกส์รูปแบบ PDF/A-3 ด้วย XSLT

กระบวนการสร้างใบกำกับภาษีฯ รูปแบบ PDF/A-3 จากไฟล์ XML มีขั้นตอนดังนี้

- 1) สร้างใบกำกับภาษีในรูปแบบ HTML จากไฟล์ XML และไฟล์
- 2) XSLT ขั้นตอนนี้แอปพลิเคชันจะใช้ไฟล์ XML ที่บรรจุข้อมูลของใบกำกับภาษีที่มีโครงสร้างตามข้อเสนอแนะมาตรฐานฯ ชมธอ. 3-2560 เป็นตัวตั้งต้น จากนั้นจะใช้ไฟล์ XSLT ซึ่งเป็นภาษาสำหรับตกแต่งรูปแบบของข้อมูลของไฟล์ XML แปลงข้อมูลภายใน XML ให้อยู่ในรูปแบบ HTML โดยตัวอย่างขั้นตอนการเปลี่ยนไฟล์ XML ให้เป็น PDF แสดงไว้ในข้อ 4.1.3 และ ตัวอย่างข้อเสนอแนะมาตรฐานฯ ชมธอ. 3-2560 โดยย่อแสดงไว้ในข้อ 4.1.3.1
- 3) เมื่อได้ไฟล์ HTML มาแล้วแอปพลิเคชันจะแนบส่วนประกอบสำหรับการสร้างไฟล์ด้วย PDF/A-3 (Fonts, CSS, XMP และ XML) ซึ่งได้ถูกกล่าวถึงไว้ในบทที่ 3 ข้อ 3.2 และในส่วนของรายละเอียดการสร้าง จะถูกเขียนไว้ในข้อ 4.1.3.2

#### 4.1.1 ตัวอย่างไฟล์ XML , XSLT , XMP

การสร้างใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF/A-3 ในคู่มือนี้จะไม่กล่าวถึงวิธีการสร้างไฟล์ XML ตามรูปแบบข้อเสนอแนะมาตรฐานฯ ชมธอ. 3-2560 เต็มรูปแบบ โดยจะอธิบายเพียง tag ที่เป็นโครงสร้างอย่างง่ายของข้อมูลตามข้อเสนอแนะมาตรฐานฯ ตัวอย่างไฟล์ XSLT และตัวอย่างการสร้างไฟล์ XMP

##### 4.1.2.1 ตัวอย่างโครงสร้างของ XML ตามข้อเสนอแนะมาตรฐานฯ ชมธอ. 3-2560

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <rsm:TaxInvoice_CrossIndustryInvoice
3.   xmlns:ram="urn:etda:unefact:data:standard:TaxInvoice_ReusableAggregateBusinessInformationEntity:1"
4.   xmlns:rsm="urn:etda:unefact:data:standard:TaxInvoice_CrossIndustryInvoice:2"
5.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6.   xsi:schemaLocation="urn:etda:unefact:data:standard:TaxInvoice_CrossIndustryInvoice:2
7.     file:///data/standard/TaxInvoice_CrossIndustryInvoice_2p0.xsd"
8.   xmlns:udt="urn:un:unece:unefact:data:standard:UnqualifiedDataType:15"
9. >
10.   <rsm:ExchangedDocumentContext>
11.     ...
12.   </rsm:ExchangedDocumentContext>
13.   <rsm:ExchangedDocument>
14.     ...
15.   </rsm:ExchangedDocument>
16.   <rsm:SupplyChainTradeTransaction>
17.     ...
18.   </rsm:SupplyChainTradeTransaction>
19.
20. </rsm:TaxInvoice_CrossIndustryInvoice>
```

รูปที่ 4 ตัวอย่างโครงสร้างไฟล์ XML

จากรูปที่ 4 แสดงโครงสร้างข้อมูลตามข้อเสนอแนะมาตรฐานฯ ชมธอ. 3-2560 (สามารถดูโครงสร้างไฟล์ XML รูปแบบสมบูรณ์ได้ที่ Appendix A)

บรรทัดที่ 1 เป็น tag มาตรฐานของ XML แสดงเวอร์ชันของ XML และระบุ encoding ของเอกสาร เป็น UTF-8

บรรทัดที่ 2-8 <rsm:TaxInvoice\_CrossIndustryInvoice ประกาศ namespace ของโครงสร้างข้อมูลชุดนี้

บรรทัดที่ 10-12 <rsm:ExchangedDocumentContext> เป็น tag ข้อมูลที่บรรจุรหัสคู่มือมาตรฐานของ XML ชุดนี้

บรรทัดที่ 13-15 <rsm:ExchangedDocument> เป็น tag ข้อมูลที่บรรจุหัวเรื่องของเอกสาร ยกตัวอย่างเช่น หมายเลขของเอกสาร ชื่อของเอกสาร เวลาในการสร้างเอกสาร

บรรทัดที่ 16 <rsm:SupplyChainTradeTransaction> เป็น tag ข้อมูลที่บรรจุ ข้อมูลภายในเอกสาร เช่นผู้ซื้อ ผู้ขาย ข้อมูลสินค้า

tag ต่างๆที่ปรากฏอยู่ในรูปที่ 3 การสร้างใบกำกับภาษีอิเล็กทรอนิกส์รูปแบบ PDF/A-3 ด้วย XSLT ถูกกำหนดตามข้อเสนอแนะมาตรฐานฯ ชมธอ. 3-2560 โดยสามารถค้นหาความหมายในเอกสารข้อเสนอแนะฯ ได้

#### 4.1.2.2 ตัวอย่างและกระบวนการทำงานของ XSLT

การแปลงไฟล์ XML ไปเป็น HTML โดยใช้ไฟล์ XSLT สามารถดูไฟล์ฉบับเต็มได้ที่รูปที่ 5 โดยไฟล์ดังกล่าว จะถูกแบ่งออกเป็น 2 ส่วน โดยส่วนแรกสำหรับกำหนดโครงสร้างของไฟล์ HTML ดังรูปที่ 5 และส่วนที่สองเป็นการประกาศ template โดยในหัวข้อนี้นี้จะใช้ template ที่มีข้อความว่า <xsl:template match="rsm:CIIHExchangedDocument"> มาเป็นตัวอย่างประกอบในการอธิบายการทำงานของ XSLT file ซึ่งถูกแสดงอยู่ดังรูปที่ 6

```
1. <?xml version="1.0" encoding="utf-8"?>
2. <xsl:stylesheet exclude-result-prefixes="rsm ram udt"
3.           xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4.           xmlns:ram="urn:etda.uncefact.data.standard:TaxInvoice_ReusableAggregateBus
5.                   inessInformationEntity:1"
6.           xmlns:rsm="urn:etda.uncefact.data.standard:TaxInvoice_CrossIndustryInvoice
7.                   :2"
8.           xmlns:udt="urn:un:unece.uncefact.data.standard:UnqualifiedDataType:15"
9.           version="1.0"
10.          xmlns:utils="inbatek:Utils"
11.          >
12. <xsl:output method="xml" indent="yes" encoding="UTF-8" />
13. <xsl:template match="/rsm:TaxInvoice_CrossIndustryInvoice">
14.   <html>
15.     <head>
16.       <link rel="stylesheet" type="text/css" href="invoice.css" />
17.     </head>
18.     <body>
19.       
20.       <br />
21.       <xsl:apply-templates />
22.     </body>
23.   </html>
24. </xsl:template>
```

รูปที่ 5 ตัวอย่างไฟล์ XSLT ส่วนแรกสำหรับกำหนดโครงสร้างของไฟล์ HTML

จากรูปที่ 5 เป็นไฟล์ XSLT ในส่วนแรกโดยจะเห็นได้ว่าภายในไฟล์จะมี tag ในภาษา HTML ปะปนอยู่ บรรทัดที่ 13 ซึ่งมีข้อความว่า <xsl:template match = "/rsm:TaxInvoice\_CrossIndustryInvoice"> นี้เป็นการประกาศว่าภายใน template block นี้จะใช้ข้อมูลทั้งหมดจาก XML tag ชื่อ



rsm:TaxInvoice\_CrossIndustryInvoice รวมถึง element ที่อยู่ภายใต้ tag ดังกล่าว มาประมวลผล ( rsm:TaxInvoice\_CrossIndustryInvoice เป็น root ของ element ทั้งหมดของไฟล์ XML)

บรรทัดที่ 9 ที่มีข้อความว่า <xsl:apply-templates/> บรรทัดนี้จะเป็นการเรียกใช้งาน template ที่ถูกประกาศอยู่ในส่วนที่สอง ซึ่งเป็นส่วนที่ระบุรายละเอียดเบื้องต้นของใบกำกับภาษี เช่น เลขที่ใบกำกับภาษี วันที่ออกใบกำกับภาษี เป็นต้น ทั้งนี้สามารถดู template ทั้งหมดจาก Appendix B

```
1. <xsl:template match="rsm:ExchangedDocument">
2.   <h1 id="header">
3.     <xsl:value-of select="ram:Name" />
4.     <xsl:text> </xsl:text>
5.     <xsl:value-of select="ram:ID" />
6.   </h1>
7.   <br />
8.   <h2 id="date">วันที่ออก:
9.     <xsl:call-template name="calculate-buddisth-date">
10.      <xsl:with-param name="date-time" select="ram:IssueDateTime"/>
11.    </xsl:call-template>
12.   </h2>
13. </xsl:template>
```

รูปที่ 6 ตัวอย่างไฟล์ XSLT ส่วนที่สองสำหรับรายละเอียดเบื้องต้นของใบกำกับภาษี

จากรูปที่ 6 ส่วนนี้ของไฟล์ XSLT จะเป็นส่วนที่นำค่าภายใน XML มาใส่ HTML tag โดยจะใช้ข้อมูลใบกำกับภาษีตามที่ระบุในไฟล์ XML ตามตัวอย่างที่แสดงในรูปที่ 7

บรรทัดที่ 3 <xsl:value of select =ram:Name> จะเป็นการนำข้อมูลจาก XML Tag ชื่อ ram:Name ซึ่งมีค่าเท่ากับ “ใบกำกับภาษี” มาใส่ใน tag H1

บรรทัดที่ 5 <xsl:value of select =ram:ID> จะเป็นการนำข้อมูลจาก XML Tag ชื่อ ram:ID ซึ่งมีค่าเท่ากับ “INWqwe00002” มาใส่ใน tag H1

บรรทัดที่ 9-11 จะเป็นการแปลงค่าวันและเวลาตามปฏิทินคริสตศักราชไปเป็นวันและเวลาตามปฏิทินพุทธศักราช โดยขั้นตอนของการเรียก template สามารถดูได้จาก Appendix B บรรทัดที่ 37-46

```
1. <rsm:ExchangedDocument>
2.   <ram:ID>INVqwe00002</ram:ID>
3.   <ram:Name>ใบกำกับภาษี</ram:Name>
4.   <ram:TypeCode>388</ram:TypeCode>
5.   <ram:IssueDateTime>2017-08-21T00:00:00.000</ram:IssueDateTime>
6.   <ram:CreationDateTime>2017-08-21T00:00:00.000</ram:CreationDateTime>
7. </rsm:ExchangedDocument>
```

#### รูปที่ 7 ตัวอย่างข้อมูลจาก XML ภายใต้ tag rsm:ExchangedDocument

หลังจากมีไฟล์ XML และ XSLT พร้อมเรียบร้อยแล้ว เมื่อนำมาผ่านการประมวลผลด้วยชุดคำสั่งการแปลงไฟล์ตาม Appendix B จะได้ไฟล์ HTML ที่มีค่าวันที่เป็นปีพุทธศักราชตามผลลัพธ์ดังรูปที่ 8 หลังจากนั้นจะนำไฟล์ HTML ดังกล่าวนี้ไปสร้างเป็น PDF ต่อไป

```
1. <h1 id="header">
2.     ใบกำกับภาษี INVqwe00002
3. </h1><br />
4. <h2 id="date">วันที่ออก:
5.     2560-08-21
6. </h2>
```

รูปที่ 8 ตัวอย่าง HTML ที่ได้จากการแปลงไฟล์ XML

ผลลัพธ์ดังกล่าวเป็นเพียงแค่ตัวอย่างการทำงานของ template ชื่อ calculate-buddisth-date เท่านั้น ภายในไฟล์ XSLT ยังมี template อีกหลายชุดที่ทำหน้าที่ประมวลผลข้อมูลจาก XML

#### 4.1.2.3 ตัวอย่างการสร้าง XMP

ภายในหัวข้อนี้การสร้าง XMP จะใช้ method ของ iTextSharp ซึ่งเป็น iText library ใน version C# ในการสร้าง XMP โดย XMP ของใบกำกับภาษีในรูปแบบที่ถูกต้องจะต้องมีข้อมูลของ DocumentFileName DocumentType และ Version ซึ่งสามารถดูตัวอย่างการตั้งค่าข้อมูลดังกล่าวได้ในบรรทัดที่ 4 ,5 และ 6 ตามลำดับ และสามารถดูตัวอย่าง source code แบบสมบูรณ์ได้ที่ Appendix F

```
1. var metadata = writer.XmpMetadata;
2. XmpMetaFactory.SchemaRegistry.RegisterNamespace(EtdaSchemaNS, "rsm");
3.
4. writer.XmpWriter.SetProperty(EtdaSchemaNS, EtdaDocumentFilename, EtdaAttachmentFilename);
5. writer.XmpWriter.SetProperty(EtdaSchemaNS, EtdaDocumentTypeName, EtdaDocumentType);
6. writer.XmpWriter.SetProperty(EtdaSchemaNS, DocumentVersionname, DocumentVersion);
```

รูปที่ 9 ตัวอย่าง code การสร้าง XMP ด้วย iTextSharp

#### 4.1.3 รายละเอียดการสร้าง PDF/A-3 จาก XML

ในหัวข้อนี้ จะอธิบายตัวอย่างการสร้าง PDF/A-3 โดยใช้ iTextSharp ที่เขียนด้วย C# จากที่ได้แสดงไว้ในรูปที่ 3 ในการสร้างไฟล์ PDF นี้จะแบ่งขั้นตอนออกเป็น 2 ส่วนคือ 4.1.3.1 อธิบายวิธีการสร้างไฟล์รูปแบบ HTML จากไฟล์ XML และไฟล์ XSLT และอีกส่วนคือ 4.1.3.2 เป็นการสร้างไฟล์ใบกำกับภาษีในรูปแบบ PDF/A-3 พร้อมแนบไฟล์ใบกำกับภาษี รวมถึงไฟล์อื่นที่กำหนดคุณลักษณะของไฟล์ PDF/A-3 นั้น

#### 4.1.3.1 Source code การสร้าง Invoice จาก XML and XSLT

รูปที่ 10 ตัวอย่าง main function บรรทัดที่ 5 เป็นการแปลงจาก XML ไปเป็น HTML ด้วยฟังก์ชัน createHtml ( สามารถดู source code ฉบับสมบูรณ์ได้จาก Appendix C) จากนั้นในบรรทัดที่ 7 ฟังก์ชัน CreatePdf เริ่มต้นกระบวนการแปลงไฟล์ HTML เป็นไฟล์ PDF

```
1. public void CreatePdf(string xsltfilename, string input, string outputpdffilename)
2. {
3.     var inputstr = File.ReadAllText(input);
4.     var xslstr = File.ReadAllText(xsltfilename);
5.     var html = createHtml(xslstr, inputstr);
6.     var htmlbytes = Encoding.UTF8.GetBytes(html);
7.     createPdf(File.ReadAllBytes(input), htmlbytes, outputpdffilename);
8. }
9.
```

รูปที่ 10 ตัวอย่าง main function สำหรับแปลงไฟล์ XML ไปเป็น HTML

#### 4.1.3.2 การสร้าง PDF/A-3U จาก HTML , Fonts , CSS , XMP , XML

รูปที่ 11 แสดง source code สร้าง PDF จาก HTML เป็น source code ในส่วนของการเพิ่ม XMP , ICC Profile และ XML และ Fonts ให้แก่ PDF การตั้งค่า Conformance ของเอกสารจะอยู่ที่บรรทัด 1-3 และในส่วนของการสร้างรูปลักษณ์ของ PDF จะใช้วิธีการแปลงไฟล์ HTML และไฟล์ CSS ไปเป็นไฟล์ PDF/A-3 ในบรรทัดที่ 9-27 (สามารถดู source code การแปลงไฟล์ฉบับสมบูรณ์ได้ที่ Appendix F) การแนบ Fonts จะใช้คลาส PdfFontProvider โดยเรียกใช้อยู่ในบรรทัดที่ 24 (สามารถดู source code การแนบ Font ได้จาก Appendix G และตัวอย่าง source code สำหรับไฟล์ CSS ได้จาก Appendix M)

```
1. // Create iTextSharp.text.pdf.PdfAWriter and assign Conformance
2. PdfAWriter writer = PdfAWriter.GetInstance(document, new Fileinfo(dest).OpenWrite(),
3. (PdfAConformanceLevel)Etdda.text.pdf.PdfAConformanceLevel.PDF_A_3U);
4. //Set PDF version and create XMP object
5. writer.PdfVersion = PdfWriter.VERSION_1_7;
6.
7. //Set XMP Parameter
8. var metadata = writer.XmpMetadata;
9. XmpMetaFactory.SchemaRegistry.RegisterNamespace(Etdda.SchemaNS, "rsm");
10. writer.XmpWriter.SetProperty(Etdda.SchemaNS, Etdda.DocumentFilename, Etdda.AttachmentFilename);
11. writer.XmpWriter.SetProperty(Etdda.SchemaNS, Etdda.DocumentType, Etdda.DocumentType);
12. writer.XmpWriter.SetProperty(Etdda.SchemaNS, DocumentVersionName, DocumentVersion);
13.
14. // Set ICC Colour Profile
15. ICC_Profile icc = ICC_Profile.GetInstance(File.ReadAllBytes(ICC));
16. writer.SetOutputIntents("Custom", "", "http://www.color.org", "sRGB IEC61966-2.1", icc);
17.
18. // CSS
19. ICSSResolver cssResolver = new StyleAttrCSSResolver();
```

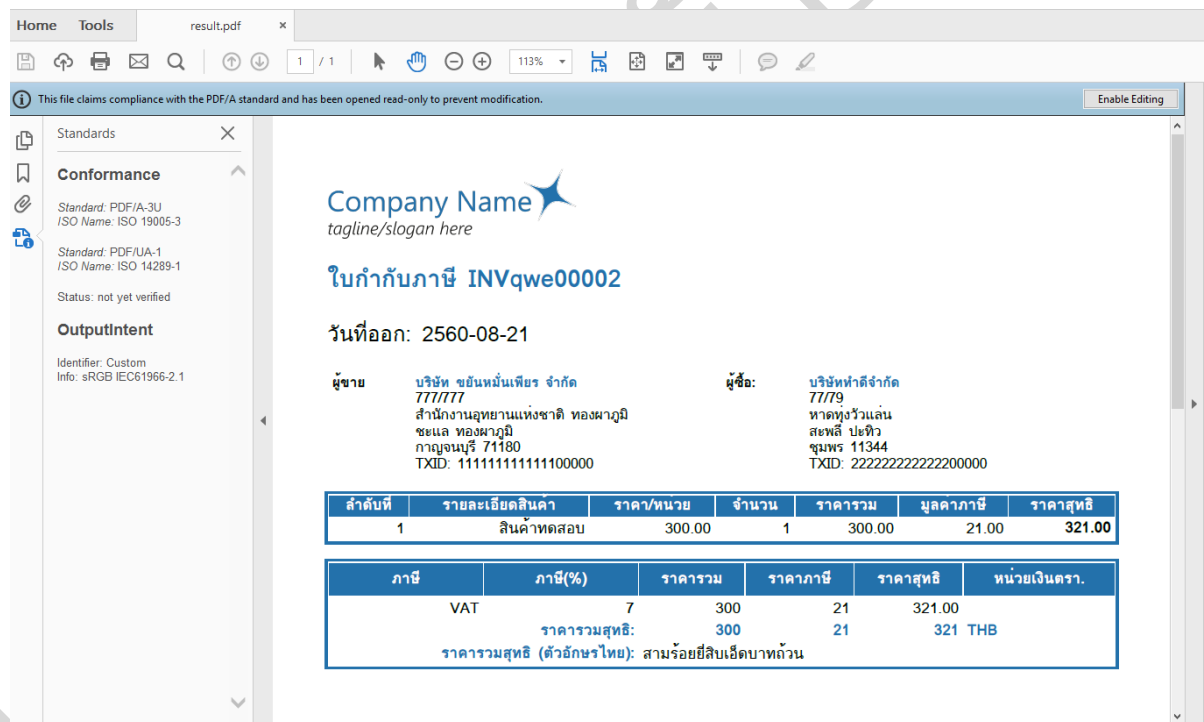
```

20. var cssFile = XMLWorkerHelper.GetCSS(new ด
21. cssResolver.AddCss(cssFile);
22.
23. // HTML and Apply PDF font
24. CssAppliers cssAppliers = new CssAppliersImpl(new PdfFontProvider());
25. HtmlPipelineContext htmlContext = new HtmlPipelineContext(cssAppliers);
26. htmlContext.SetTagFactory(Tags.GetHtmlTagProcessorFactory());
27. htmlContext.SetImageProvider(new PdfImageProvider());
28.
29. // XML Worker
30. XMLWorker worker = new XMLWorker(css, true);
31. XMLParser p = new XMLParser(worker);
32. p.Parse(new MemoryStream(invoice));

```

รูปที่ 11 แสดง source code สร้าง PDF จาก HTML

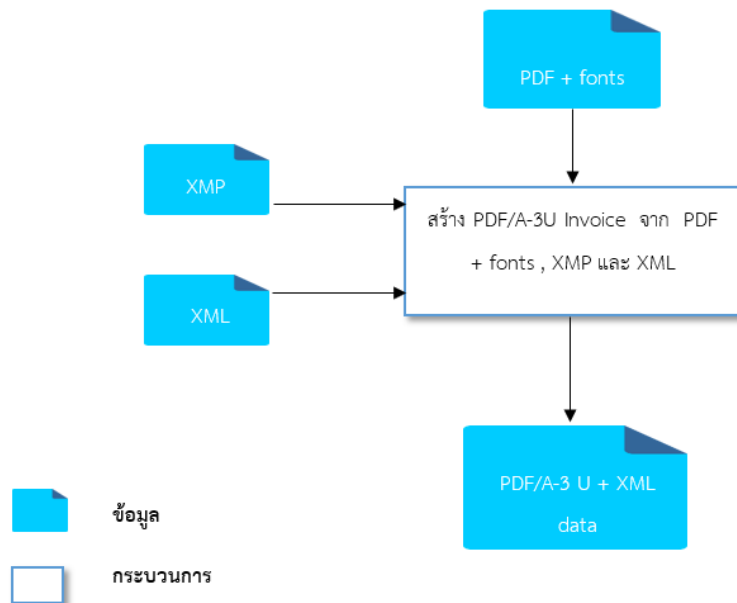
หลังจากผ่านขั้นตอนทั้งหมดจะได้ผลลัพธ์ดังรูปที่ 12 แสดงผลภาพของใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF



รูปที่ 12 ตัวอย่างใบกำกับภาษีอิเล็กทรอนิกส์ ที่ถูกสร้างจาก HTML

## 4.2 การสร้าง PDF/A-3 จากไฟล์ PDF

สำหรับการสร้าง PDF/A-3 อีกรูปแบบหนึ่งเหมาะสำหรับผู้ที่มีไฟล์ PDF อยู่แล้วและต้องการแปลงไฟล์ PDF ดังกล่าวให้อยู่ในรูปแบบ PDF/A-3 ทั้งนี้ในการสร้างไฟล์ PDF/A-3 รูปแบบดังกล่าวประกอบไปด้วยองค์ประกอบและขั้นมีตอนการสร้างไฟล์ดังรูปที่ 13



รูปที่ 13 การสร้างใบกำกับภาษีอิเล็กทรอนิกส์รูปแบบ PDF/A-3 จากไฟล์ PDF

กระบวนการสร้างใบกำกับภาษี ฯ รูปแบบ PDF/A-3 จากไฟล์ PDF มีขั้นตอนดังนี้

- 1) สร้างใบกำกับภาษีในรูปแบบ PDF ธรรมดาพร้อมกับแนบ Fonts
- 2) นำ XMP และ XML ที่มีรูปแบบที่ถูกต้องและสอดคล้องกับรายละเอียดที่ปรากฏในไฟล์ PDF จากข้อ 1. มาแนบกับไฟล์ตามข้อ 1. โดยจะได้ผลลัพธ์ออกมาอยู่ในรูปแบบ PDF/A-3

### 4.2.1 ตัวอย่างการสร้าง XML สำหรับแนบเข้าสู่ไฟล์ PDF

การสร้างไฟล์ XML จะใช้ไฟล์ XML ชุดเดียวกันกับหัวข้อ 4.1 โดยรายละเอียดในการสร้าง XML จะเป็นไปตามข้อ 4.1.2.1

#### 4.2.2.1 ตัวอย่างการสร้าง XMP

ในกรณีที่ผู้ประกอบการมีการออกใบกำกับภาษีในรูปแบบ PDF อยู่แล้ว การแปลง PDF/A-3 จากไฟล์ PDF เดิม ต้องมีการแนบไฟล์ XMP ไปกับไฟล์ PDF ดังนั้นจึงจะต้องมีการเตรียมไฟล์ XMP ขึ้นมาก่อน โดยมีบริบทของข้อมูลใบกำกับภาษีตามตารางที่ 2 ทั้งนี้สามารถดูรายละเอียดของไฟล์ XMP ฉบับเต็มได้จาก Appendix E

```
1. <x:xmpmeta xmlns:x="adobe:meta" x:xmptk="Adobe XMP Core 5.10-jc003...">
2.   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     <rdf:Description xmlns:dc="http://purl.org/dc/elements/1.1/..."
4.       <dc:format>application/pdf</dc:format>
5.       <pdfaid:part>3</pdfaid:part>
6.       <pdfaid:conformance>U</pdfaid:conformance>
7.     </rdf:Description>
8.     <rdf:Description xmlns:pdfaExtension="http://www.aiim.org/pdfans/extension" ...>
9.       <pdfaExtension:schemas>
10.        ...
11.      </pdfaExtension:schemas>
12.    </rdf:Description>
13.    <rdf:Description xmlns:rsm="urn:etda:uncefact:data:standard:Invoice_CrossIndustryInvoice:2#" rdf
14.      :about="">
15.      <rsm:DocumentFileName>ETDA-invoice.xml</rsm:DocumentFileName>
16.      <rsm:DocumentType>Tax Invoice</rsm:DocumentType>
17.      <rsm:Version>2.0</rsm:Version>
18.    </rdf:Description></rdf:RDF>
19. </x:xmpmeta>
```

รูปที่ 14 ตัวอย่างไฟล์ template XMP สำหรับแนบไปกับเอกสาร

จากรูปที่ 14 ตัวอย่างไฟล์ template XMP สำหรับแนบไปกับเอกสาร แสดงตัวอย่างของไฟล์ template XMP ของเอกสาร

บรรทัดที่ 3 – 7 ระบุประเภท และ conformance ของเอกสาร

บรรทัดที่ 8 – 11 บรรจุโครงสร้างข้อมูลของ DocumentFilename , DocumentType และ Version

บรรทัดที่ 14 -16 ระบุค่าของ DocumentFilename , DocumentType และ Version

#### 4.2.2 รายการละเอียดการสร้าง PDF/A-3 จากไฟล์ PDF

ตัวอย่างการสร้าง PDF/A-3 ในที่นี่จะใช้ iTextSharp ซึ่งเป็นไลบรารีสำหรับภาษา C# จากที่แสดงในรูปที่ 13 จะเห็นได้ว่าการสร้าง ใบกำกับภาษี ในรูปแบบ PDF/A-3 จะมีองค์ประกอบอยู่สามส่วนด้วยกันคือ ไฟล์ PDF ที่แนบ Fonts เรียบร้อยแล้ว , ไฟล์ XML และไฟล์ XMP

#### 4.2.2.2 Source code การสร้าง Invoice จาก PDF

```
1. // Create PDF/A-3U writer instance from existing document
2. PdfReader reader = new PdfReader(pdfFilePath);
3. MemoryStream stream = new MemoryStream();
4. Document pdfAdocument = new Document();
5. PdfAWriter writer = this.CreatePDFAIInstance(pdfAdocument, reader, stream);
6.
7. // ICC
8. ICC_Profile icc = ICC_Profile.GetInstance("Resources/sRGB Color Space Profile.icm");
9. writer.SetOutputIntents("sRGB IEC61966-2.1", "", "http://www.color.org", "sRGB IEC61966-2.1", icc);
10.
11. // XML
12. PdfFileSpecification contentSpec = this.EmbeddedAttachment(xmlFilePath, xmlFileName,
13. "text/xml", new PdfName("Alternative"), writer, "Tax Invoice XML Data");
14. array.Add(contentSpec.Reference);
15.
16. //XMP
17. string stringExchangeXMP = File.ReadAllText("Resources/EDocument_PDFAExtensionSchema.xml");
18. byte[] exchangeXMP = Encoding.ASCII.GetBytes(stringExchangeXMP.Replace("@DocumentType", documentType));
19. writer.XmpMetadata = exchangeXMP;
```

รูปที่ 15 ตัวอย่าง source code การสร้าง Invoice จาก PDF และ XML

จากรูปที่ 15 เป็น source code ของการสร้าง PDF/A-3 สามารถดู source code ฉบับเต็มได้จาก Appendix I

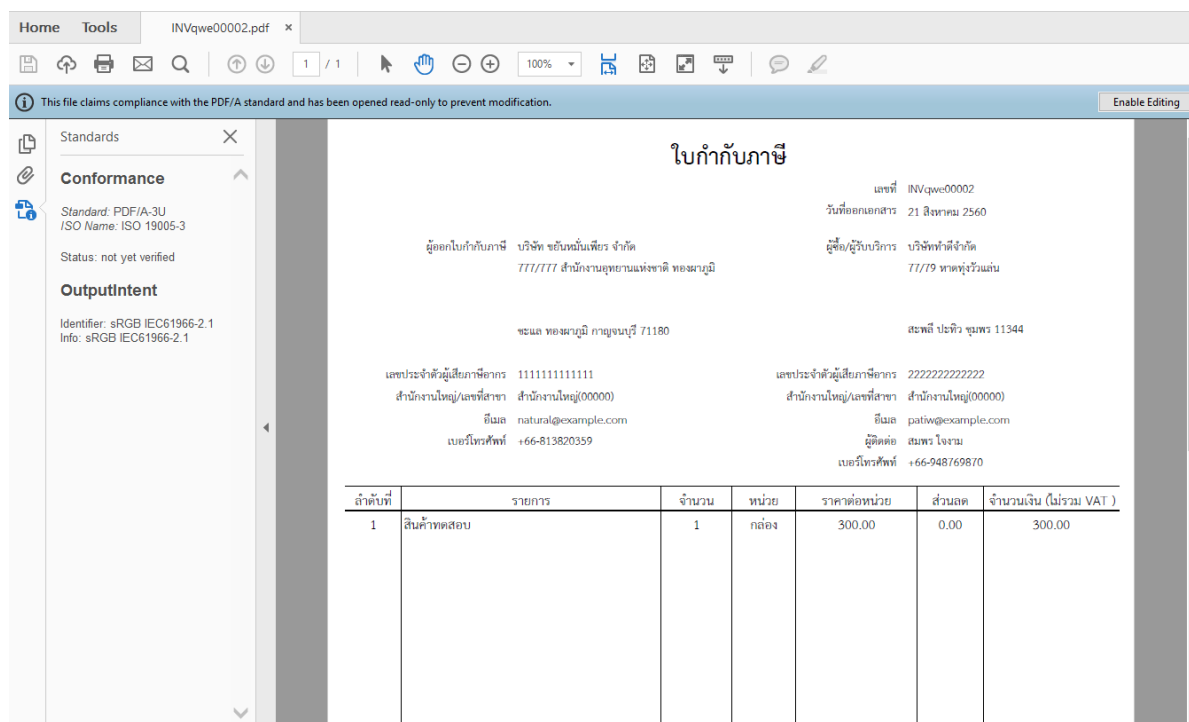
บรรทัดที่ 2 เริ่มต้นการสร้าง PDF โดยไฟล์ PDF ที่มี Fonts จะถูกระบุที่อยู่ของไฟล์ผ่านทาง ตัวแปรชื่อ pdfFilePath

บรรทัดที่ 7-8 แแนบไฟล์ Color Profile และกำหนด Output Intent

บรรทัดที่ 12-14 แแนบไฟล์ XML ,ตั้งค่าชื่อของไฟล์แนบ ,ตั้งค่าประเภทของ subtype เป็น text/xml , ตั้งค่า /AFRelationship เป็น Alternative และ Description ของ XML เป็น Tax Invoice XML Data

บรรทัดที่ 17-19 เป็นการแนบไฟล์ XMP ที่ถูกสร้างเตรียมไว้แล้ว (สามารถดู XMP ได้จาก Appendix E)

หลังจากทำตามขั้นตอนทั้งหมดจะได้ PDF invoice ดังรูปที่ 16



รูปที่ 16 ตัวอย่างใบกำกับภาษีอิเล็กทรอนิกส์ ที่ถูกสร้างจากไฟล์ PDF



## 5. การตรวจสอบความถูกต้องของใบกำกับภาษีอิเล็กทรอนิกส์

การตรวจสอบว่าใบกำกับภาษีอิเล็กทรอนิกส์ใดๆ มีความถูกต้องตามกฎหมายหรือไม่ จะต้องมีการตรวจสอบคุณสมบัติต่างๆ ดังนี้

- 1) การตรวจสอบรายการข้อมูลของใบกำกับภาษีตามกฎหมาย
- 2) การตรวจสอบโครงสร้างของข้อมูลในใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XML (Structural Validation)
- 3) การตรวจสอบข้อกำหนดของข้อมูลในใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XML (Rules-based Validation)
- 4) การตรวจสอบ metadata ของใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XMP
- 5) การตรวจสอบข้อมูล Catalog Dictionary ของใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF
- 6) การเปรียบเทียบ Visual Representation และข้อมูลใน XML

### 5.1 การตรวจสอบรายการข้อมูลของใบกำกับภาษีตามกฎหมาย

ใบกำกับภาษี ใบเพิ่มหนี้ และใบลดหนี้อิเล็กทรอนิกส์ในรูปแบบ PDF ต้องมีรายการข้อมูลอย่างน้อยตามประมวลรัษฎากร (กรมสรรพากร, 2014) มาตรา 86/4, 86/9 และ 86/10 ตามลำดับ โดยสามารถดูรายละเอียดได้ที่ข้อ 1.1, 1.2 และ 1.3 ตามลำดับ

อย่างไรก็ตาม การตรวจสอบในคุณสมบัติข้อนี้ อาจไม่สามารถใช้วิธีทางอิเล็กทรอนิกส์ในการตรวจสอบได้ เนื่องจากไม่สามารถตรวจสอบความถูกต้องของสิ่งที่ปรากฏในการแสดงผลของไฟล์ PDF ได้ด้วยคอมพิวเตอร์โดยตรง

### 5.2 การตรวจสอบโครงสร้างของข้อมูลในใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XML (Structural Validation)

การตรวจสอบโครงสร้างของข้อมูล ใช้ไฟล์ XML Schema Definition (.xsd) ซึ่งเป็นไฟล์ที่กำหนดโครงสร้างที่ถูกต้องของไฟล์ XML แบบต่างๆ ซึ่งในที่นี้จะใช้ XML Schema Definition สำหรับไฟล์ XML ชนิดที่เป็นใบกำกับภาษี ตัวอย่างของไฟล์ XML ของใบกำกับภาษี แสดงใน Appendix A การตรวจสอบโครงสร้างมี 2 ขั้นตอน คือ

- 1) ตรวจสอบโครงสร้างของไฟล์ XML โดยโครงสร้างของข้อมูลต้องถูกต้องตามประเภทใบกำกับภาษี ซึ่งสามารถตรวจสอบได้จากค่า xsi:schemaLocation ของ tag `rsm:TaxInvoice_CrossIndustryInvoice` ที่ระบุไว้ใน header ของใบกำกับภาษีอิเล็กทรอนิกส์ใน

รูปแบบ XML ตามข้อมูลในตารางที่ 4 และสามารถดูตัวอย่างการระบุค่าได้ในรูปที่ 17 การตรวจสอบนี้สามารถใช้วิธีตรวจสอบได้ทั้งโดยบุคคลและวิธีทางอิเล็กทรอนิกส์

- 2) ตรวจสอบไวยากรณ์ของไฟล์ XML ว่ามีไวยากรณ์ที่ถูกต้อง (well-formed) และมีประเภทข้อมูลถูกต้องตามโครงสร้างของข้อมูลที่ระบุไว้ โดยตัวอย่างของวิธีการที่แสดงในเอกสารฉบับนี้ ใช้ภาษา C# อ่าน XML ด้วย XmlReader และตรวจสอบ Schema โดยใช้ XmlSchemaSet โดยแสดงกระบวนการไว้ในรูปที่ 18 และมีขั้นตอนดังนี้
  - a. ระบุ schema ของ XML สำหรับใช้ตรวจสอบ ตามตัวอย่างในรูปที่ 19 และวางไฟล์ schema ทั้งหมดที่ถูกอ้างอิง (การอ้างอิงสามารถมีได้มากกว่าหนึ่งชั้น) ในไฟล์ schema ของ XML ไว้ในพาที่แอปพลิเคชันสามารถอ้างอิงถึงได้
  - b. ตั้งค่าการอ่านไฟล์ XML เป็นการตรวจสอบด้วย schema ของ XML ตามข้อ a. ตามตัวอย่างในรูปที่ 20
  - c. ตั้งค่าให้แสดงผลการตรวจสอบ เมื่อตรวจสอบพบส่วนของไฟล์ XML ที่ไม่ถูกต้องตาม schema ตามตัวอย่างในรูปที่ 21
  - d. ระบุไฟล์ XML ที่ต้องการตรวจสอบ และการตั้งค่าของการอ่าน XML ในข้อ b และสั่งให้ อ่านไฟล์ XML ตามตัวอย่างในรูปที่ 22

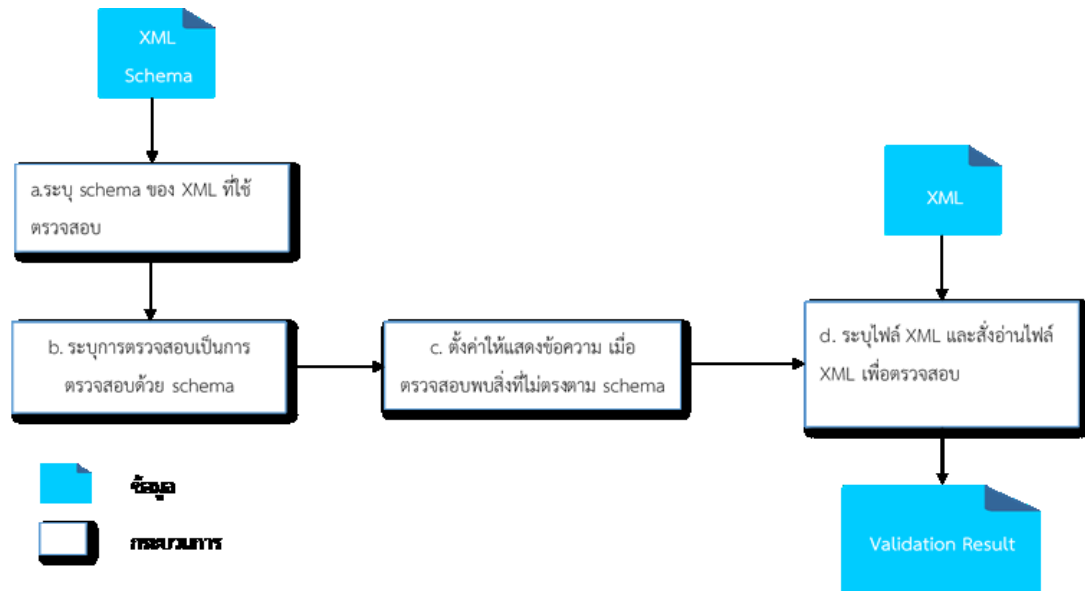
ตารางที่ 3 ค่าสำหรับตรวจสอบโครงสร้างของข้อมูลที่ระบุในใบกำกับภาษี

ประเภทเอกสาร	ชื่อ Tag ข้อมูล	ชื่อค่า	ชื่อโครงสร้างของข้อมูล
ใบกำกับภาษี	<rsm:TaxInvoice_CrossIndustryInvoice>	xsi:schemaLocation	urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2
ใบเพิ่มหนี้	<rsm:DebitCreditNote_CrossIndustryInvoice>	xsi:schemaLocation	urn:etda:uncefact:data:standard:DebitCreditNote_CrossIndustryInvoice:2
ใบลดหนี้	<rsm:DebitCreditNote_CrossIndustryInvoice>	xsi:schemaLocation	urn:etda:uncefact:data:standard:DebitCreditNote_CrossIndustryInvoice:2

1.	<?xml version="1.0" encoding="UTF-8"?>
2.	<rsm:TaxInvoice_CrossIndustryInvoice
3.	xmlns:ram="urn:etda:uncefact:data:standard:TaxInvoice_ReusableAggregateBusinessInformationEntity:2"
4.	xmlns:rsm="urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2"
5.	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6.	xsi:schemaLocation="urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2">
7.	<rsm:ExchangedDocumentContext>

รูปที่ 17 ตัวอย่างโครงสร้างข้อมูลของไฟล์ใบกำกับภาษีอิเล็กทรอนิกส์ที่ระบุใน XML

โดยสามารถดูตัวอย่าง source code การตรวจไฟล์ XML ด้วยภาษา C# แบบสมบูรณ์ได้ที่ Appendix J และสามารถดาวน์โหลดไฟล์ schema ของไฟล์ใบกำกับภาษีสำหรับตรวจสอบ ได้ที่ [https://etax.rd.go.th/etax\\_staticpage/app/index.html#/index/support/download#top](https://etax.rd.go.th/etax_staticpage/app/index.html#/index/support/download#top)



รูปที่ 18 กระบวนการตรวจสอบด้วย Schema โดยใช้ XmlReader และ XmlSchemaSet

```

1. // Create the XmlSchemaSet class.
2. XmlSchemaSet sc = new XmlSchemaSet();
3.
4. // Add the schema to the collection.
5. sc.Add("urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2", "TaxInvoice_CrossIndu
stryInvoice_2p0.xsd");
  
```

รูปที่ 19 ตัวอย่างการระบุไฟล์ schema ของ XML ที่ใช้ตรวจสอบ

```

1. // Set the validation settings.
2. XmlReaderSettings settings = new XmlReaderSettings();
3. settings.ValidationType = ValidationType.Schema;
4. settings.Schemas = sc;
5. settings.ValidationEventHandler += new ValidationEventHandler(ValidationCallback);
  
```

รูปที่ 20 ตัวอย่างการตั้งค่าการตรวจสอบเป็นการตรวจสอบด้วย schema

```

1. // Display any validation errors.
2. private static void ValidationCallback(object sender, ValidationEventArgs e)
3. {
4. Console.WriteLine("Validation Error: {0}", e.Message);
5. }
  
```

รูปที่ 21 ตัวอย่างการตั้งค่าให้แสดงข้อความ เมื่อตรวจสอบพบสิ่งที่ไม่ตรงตาม schema

```

1. // Create the XmlReader object.
2. XmlReader reader = XmlReader.Create("ETDA-invoice.xml", settings);
3.
4. // Parse the file.
5. while (reader.Read()) ;

```

รูปที่ 22 ตัวอย่างการสร้าง XmlReader สำหรับไฟล์ XML ที่ต้องการตรวจสอบ เลือกการตั้งค่าที่ระบุไว้ และการสั่งอ่านไฟล์

### 5.3 การตรวจสอบข้อกำหนดของใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XML (Rules-based Validation)

การตรวจสอบข้อกำหนดของข้อมูลในรูปแบบ XML นั้นใช้ไฟล์ schematron (.sch) ซึ่งระบุข้อกำหนดของไฟล์ข้อมูลใบกำกับภาษีอิเล็กทรอนิกส์ โดยไฟล์ schematron ที่ใช้ตรวจสอบในเอกสารฉบับนี้เป็นเวอร์ชันที่สอดคล้องตาม ISO/IEC 19757-3

ตัวอย่างของไฟล์ schematron นั้นแสดงอยู่ใน Appendix K โดยการตรวจสอบด้วย Schematron นั้นสามารถใช้ตรวจสอบโครงสร้างข้อมูลของ XML ได้ด้วย

การตรวจสอบข้อกำหนดด้วย Schematron มีไลบรารีหลายตัวที่สามารถใช้งานได้ทั้งที่ไลบรารีที่เป็น open source และ commercial โดยในตัวอย่างนี้ เลือกใช้ไลบรารีที่เป็น open source สำหรับภาษา Java และ Apache Ant (Apache Ant, 2017) ที่มีชื่อว่า Schematron for Ant (Schematron, 2017) โดยสามารถดูตัวอย่าง, รายละเอียดความต้องการและวิธีการติดตั้งของไลบรารีดังกล่าวได้ในคู่มือของ Schematron for Ant ที่ <https://github.com/Schematron/schematron/> และสามารถศึกษาวิธีการใช้งาน Apache Ant ได้ที่ <https://ant.apache.org/manual/>

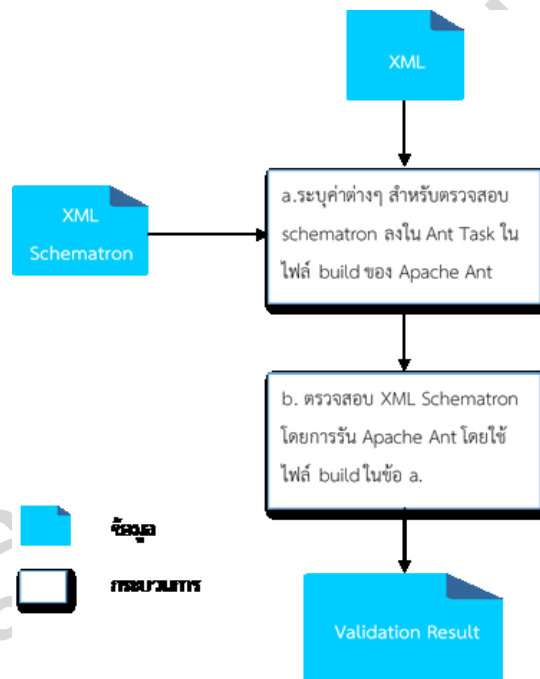
วิธีการใช้งาน Schematron for Ant มีขั้นตอนดังนี้ และแสดงไว้ในรูปที่ 23

- 1) ระบุค่าต่างๆ ที่ใช้ในการตรวจสอบ schematron ลงใน Ant Task ซึ่งเป็นที่ๆ กำหนดสิ่งต่างๆ ที่ต้องการทำงานด้วย Apache Ant ลงในไฟล์ build ของ Apache Ant โดยสามารถดูตัวอย่าง Ant Task ได้ในรูปที่ 24 และตัวอย่างของไฟล์ build ได้ใน Appendix L โดยตารางที่ 4 แสดงค่าที่จำเป็นต้องระบุใน Ant task และสามารถดูรายละเอียดของค่าอื่นๆ ที่สามารถระบุเพิ่มเติมได้ในคู่มือของ Schematron for Ant (Apache Ant, 2017)

ตารางที่ 4 ค่า ที่ระบุใน Ant Task สำหรับการตรวจสอบ

No.	ชื่อ Tag ข้อมูล	ชื่อค่า	รายละเอียด
1	<target>	Name	ระบุชื่อของ target
2	<target>	Description	ระบุข้อความอธิบาย target
3	<taskdef	Name	ระบุชื่อของ Ant Task

No.	ชื่อ Tag ข้อมูล	ชื่อค่า	รายละเอียด
4	<taskdef>	Classname	ระบุชื่อ class ที่จะใช้รัน ในที่นี้จะป็น class สำหรับใช้ตรวจสอบไฟล์ XML ด้วย schematron ซึ่งมีชื่อว่า "com.schematron.ant.SchematronTask"
5	<taskdef>	Classpath	ระบุพารที่เก็บไฟล์ "ant-schematron.jar"
6	<schematron>	Schema	ระบุพารที่เก็บไฟล์ schematron ที่ใช้ตรวจสอบ
7	<Schematron>	File	ระบุพารที่เก็บไฟล์ XML ที่ต้องการตรวจสอบ



รูปที่ 23 วิธีการใช้งาน Schematron for Ant

```

1. <target name="schematron-validation" description="Schematron Validation Example">
2.   <taskdef name="schematron" classname="com.schematron.ant.SchematronTask"
3.     classpath="lib/ant-schematron.jar"/>
4.   <schematron schema="schemas/TaxInvoice_Schematron_2p0.sch"
5.     file="xml/ETDA-invoice.xml"/>
6. </target>

```

รูปที่ 24 ตัวอย่าง Ant Task ในไฟล์ build

- ตรวจสอบ schematron ด้วยการรัน Apache Ant ด้วยคำสั่ง ant [build.xml] บน Command Prompt ที่ directory ที่เก็บไฟล์ build หรือระบุพารที่เก็บไฟล์ build ซึ่งแสดงตัวอย่างผลการ

ตรวจสอบไฟล์ XML ที่ไม่ถูกต้องตามข้อกำหนดของ schematron ในรูปที่ 25 โดยแสดงชื่อไฟล์ที่ไม่ถูกต้องในบรรทัดที่ 2 และข้อความแสดงผลการตรวจสอบในบรรทัดที่ 3 ถึงบรรทัดที่ 5 และแสดงตัวอย่างผลการตรวจสอบไฟล์ XML ที่ถูกต้องตามข้อกำหนด ในรูปที่ 26

```
1. schematron-validation:
2. [schematron] Source file: ETDA-invoice.xml
3. [schematron] (report) /*[local-name()='TaxInvoice_CrossIndustryInvoice' and namespace-
uri()='urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2']/*[local-
name()='ExchangedDocument' and namespace-
uri()='urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2'] -
4. [schematron] TIV-Document-003 รหัสประเภทเอกสาร (ExchangedDocument/TypeCode) และชื่อประเภท
เอกสาร (ExchangedDocument/Name) ไม่สอดคล้องกัน
5. [schematron] (TypeCode and Name Not Match.)
6. [schematron]
7. [schematron] 1 file(s) have been successfully validated.
```

รูปที่ 25 ตัวอย่างผลการตรวจสอบไฟล์ XML ที่ไม่ถูกต้องตามข้อกำหนดของ Schematron

```
1. schematron-validation:
2. [schematron] 1 file(s) have been successfully validated.
```

รูปที่ 26 ตัวอย่างผลการตรวจสอบไฟล์ XML ที่ถูกต้องตามข้อกำหนดของ Schematron

โดยสามารถดาวน์โหลดไฟล์ schematron ของใบกำกับภาษีอิเล็กทรอนิกส์สำหรับใช้ตรวจสอบได้ที่ [https://etax.rd.go.th/etax\\_staticpage/app/index.html#/index/support/download#top](https://etax.rd.go.th/etax_staticpage/app/index.html#/index/support/download#top)

หรือสามารถตรวจสอบข้อกำหนดและโครงสร้างของข้อมูลใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XML ได้ที่ [https://etax.rd.go.th/etax\\_staticpage/app/#/index/support/validateschema#top](https://etax.rd.go.th/etax_staticpage/app/#/index/support/validateschema#top)

#### 5.4 การตรวจสอบ metadata ของใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XMP

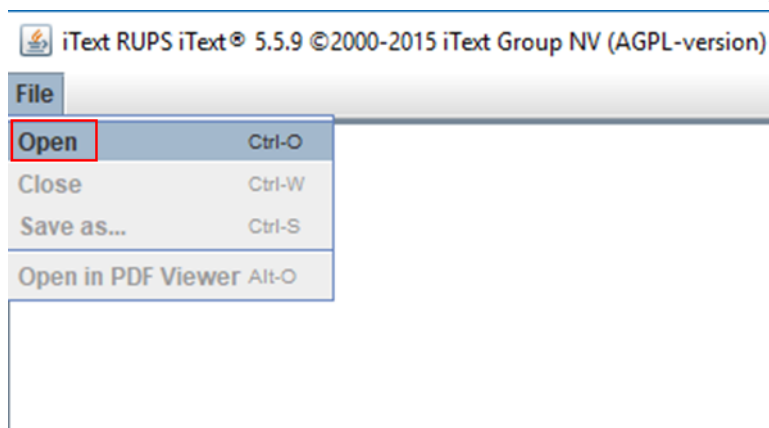
สามารถตรวจสอบ metadata ในรูปแบบ XMP โดยมี tag อย่างน้อยตามที่ระบุในบทที่ 3 ข้อ 3.2 ข้อ ย่อยที่ 2 และตรวจสอบชื่อและค่าของ tag ว่าตรงตามที่ระบุไว้หรือไม่

การตรวจสอบข้อมูล metadata ในรูปแบบ XMP มีเครื่องมือและไลบรารีหลายตัวที่สามารถใช้งานได้ทั้งที่เป็น open source และ commercial โดยในตัวอย่างนี้ เลือกใช้เครื่องมือที่เป็น AGPL license บน Microsoft Windows ที่มีชื่อว่า iText RUPS (iText, 2017) ในตัวอย่างนี้แสดงเฉพาะวิธีการใช้ iText RUPS ในรูปแบบเครื่องมือ (.exe) โดยวิธีตรวจสอบ metadata ในรูปแบบ XMP ของ ไฟล์ PDF มีขั้นตอนดังนี้

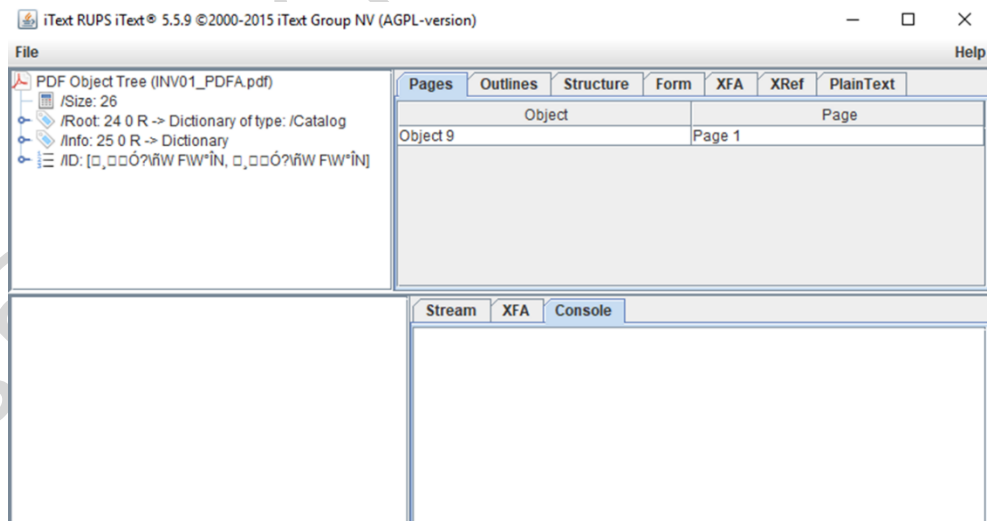
- 1) เปิดเครื่องมือ iText RUPS และเลือกเมนู File แล้ว Open เพื่อเลือกไฟล์ PDF ที่ต้องการตรวจสอบ โดยแสดงตัวอย่างหน้าจอของ iText RUPS ไว้ในรูปที่ 27

- 2) หน้าจอ iText RUPS จะแสดงไฟล์ตามตัวอย่างในรูปที่ 28 ให้เลือก object ภายในไฟล์ PDF ตามลำดับขั้นดังนี้ “/Root”, “Dictionary of type: /Catalog”, “/Metadata” และ “Stream” โดยแสดงลำดับขั้นที่เลือกในรูปที่ 29

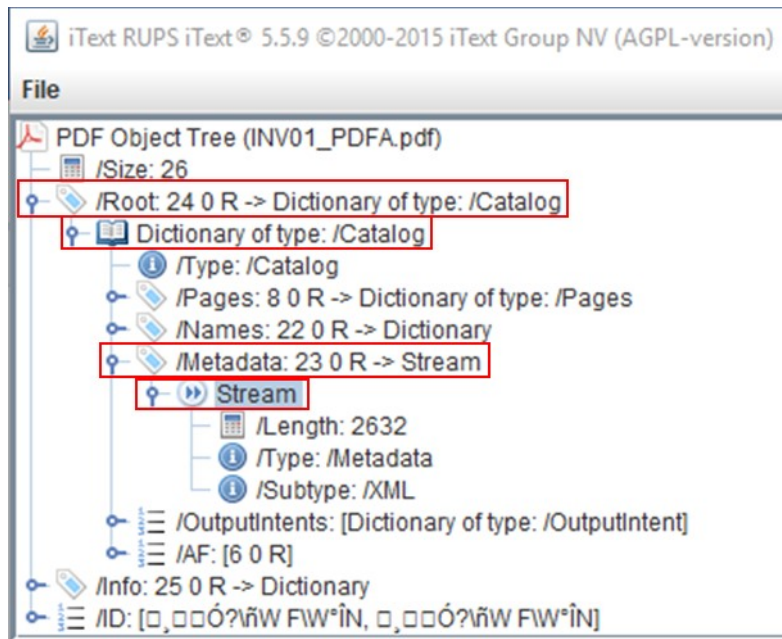
iText RUPS จะแสดงข้อมูล metadata ในหน้าจอ stream ตามตัวอย่างในรูปที่ 30 โดยสามารถตรวจสอบค่าชื่อและค่าของ tag ว่าตรงตามที่ระบุไว้ตามตารางที่ 1 หรือไม่ และดูตัวอย่างของข้อมูล metadata ในรูปแบบ XMP ได้ใน Appendix E



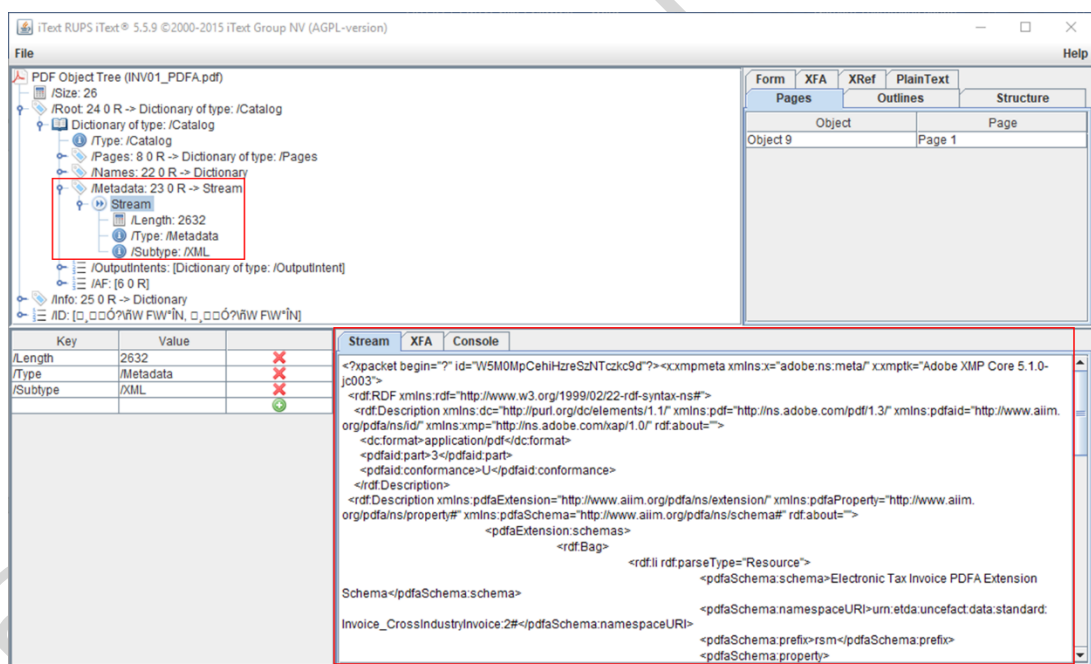
รูปที่ 27 ตัวอย่างหน้าจอ iText RUPS และเมนูสำหรับเลือกไฟล์ PDF



รูปที่ 28 ตัวอย่างหน้าจอเมื่อเลือก browse ไฟล์ PDF สำเร็จ



รูปที่ 29 ตัวอย่างการเลือก object เพื่อแสดงข้อมูล metadata ในรูปแบบ XMP บน iText RUPS



รูปที่ 30 ตัวอย่างหน้าจอแสดงข้อมูล metadata ในรูปแบบ XMP บน iText RUPS

## 5.5 การตรวจสอบข้อมูล Catalog Dictionary ของใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF

สามารถตรวจสอบข้อมูล Catalog Dictionary โดยมีข้อมูลอย่างน้อยตามที่ระบุในบทที่ 3 ข้อ 3.2 ข้อ ย่อยที่ 3 และตรวจสอบค่าของข้อมูลว่าตรงตามที่ระบุไว้หรือไม่ สามารถดูรายละเอียดของข้อมูล Catalog Dictionary ของไฟล์ PDF และค่าสำหรับตรวจสอบได้ในตารางที่ 5



ตารางที่ 5 ข้อมูล catalog dictionary ของไฟล์ไบบ์กำกับภาษา และค่าที่แสดงในการตรวจสอบด้วยเครื่องมือ iText RUPS

No.	Parent Object	Object Name	Field name	Value	Value represented in iText RUPS
1	/AF	/Filespec	/F	ETDA-invoice.xml	ETDA-invoice.xml
2	/AF	/Filespec	/UF	ETDA-invoice.xml	ETDA-invoice.xml
3	/AF	/Filespec	/AFRelationship	Alternative	/Alternative
4	/AF	/Filespec	/Desc	Tax Invoice XML Data	Tax Invoice XML Data
5	/EF	Embedded File Stream	/Subtype	text/xml	/text#2fxml
6	/EF	/Params	/ModDate	YYYYMMDDHHm mSSOHH'mm	D:YYYYMMDDHHm mSSOHH'mm

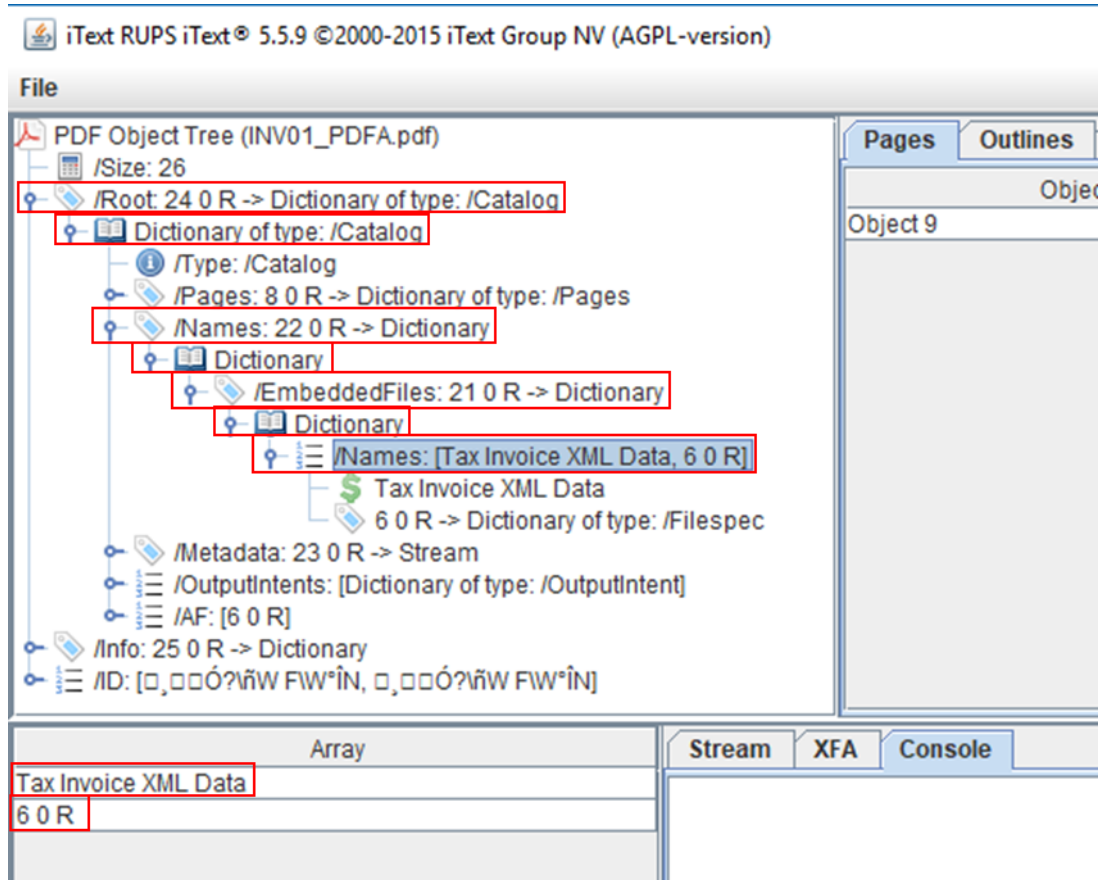
การตรวจสอบข้อมูล Catalog Dictionary มีเครื่องมือและไลบรารีหลายตัวที่สามารถใช้งานได้ทั้งที่เป็น open source และ commercial โดยในตัวอย่างนี้ เลือกใช้เครื่องมือที่เป็น AGPL license บน Microsoft Windows ที่มีชื่อว่า iText RUPS เช่นเดียวกับการตรวจสอบ metadata ในรูปแบบ XMP ในตัวอย่างนี้แสดงเฉพาะวิธีการใช้ iText RUPS ในรูปแบบเครื่องมือ (.exe) โดยสามารถดูรายละเอียดของข้อมูล Catalog Dictionary ของไฟล์ PDF แต่ละค่าได้ใน มีขั้นตอนตรวจสอบดังนี้

#### 5.5.1 การค้นหา indirect reference ของไฟล์ไบบ์กำกับภาษาในรูปแบบ XML

indirect reference เปรียบเสมือน link ของ object ภายในไฟล์ PDF โดยในตัวอย่างนี้แสดงการค้นหา indirect reference ของไฟล์ไบบ์กำกับภาษาในรูปแบบ XML เพื่อใช้สำหรับระบุข้อมูล ใน catalog dictionary ได้

- 1) เปิดเครื่องมือ iText RUPS และเลือกเมนู File แล้ว Open เพื่อเลือกไฟล์ PDF ที่ต้องการตรวจสอบ โดยแสดงตัวอย่างหน้าจอของ iText RUPS ไว้ในรูปที่ 27
- 2) หน้าจอ iText RUPS จะแสดงไฟล์ตามตัวอย่างในรูปที่ 28 ให้เลือก object ภายในไฟล์ PDF ตามลำดับขั้นดังนี้ “/Root”, “/Names”, “Dictionary”, “/EmbeddedFiles”, “Dictionary” และ “/Names” โดยแสดงลำดับขั้นที่เลือกและแสดงข้อมูล “/Names” บนหน้าจอตามตัวอย่างในรูปที่ 30 โดย Indirect reference ของไฟล์ไบบ์กำกับภาษาในรูปแบบ XML จะแสดงเป็น element ที่

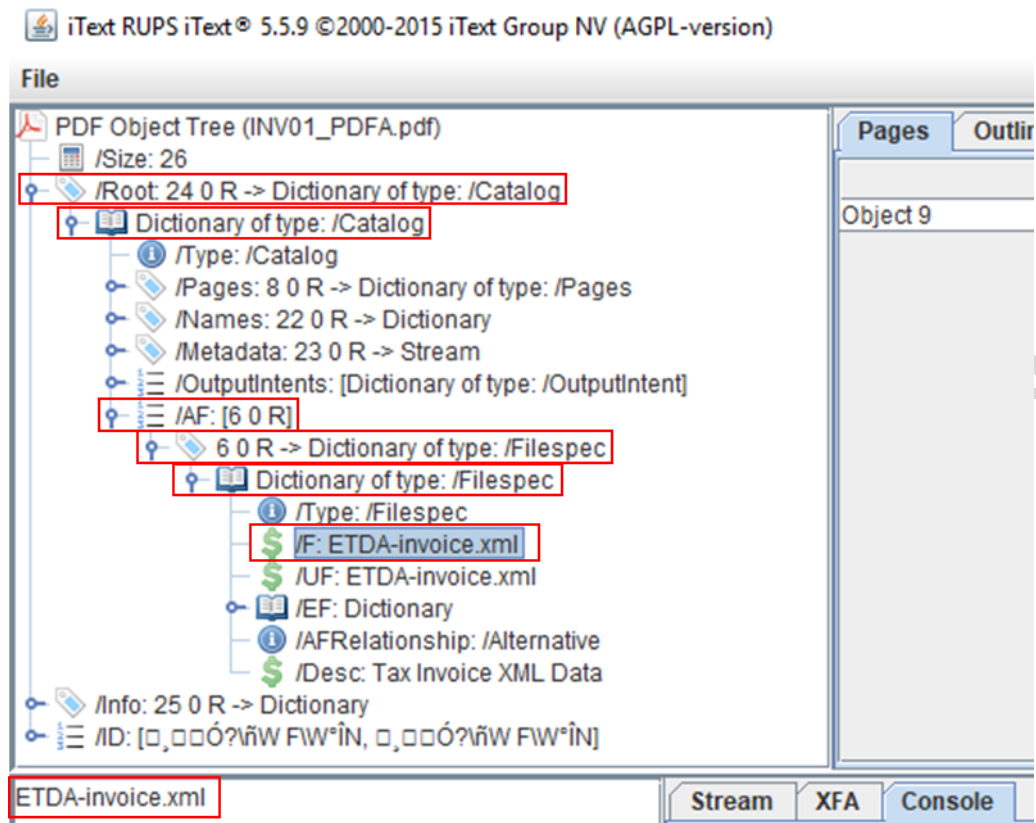
อยู่ถัดมาจาก element ที่มีค่าเป็น “Tax รูปที่ 31 Invoice XML Data” โดยในตัวอย่าง แสดง indirect reference ของไฟล์ใบกำกับภาษีในรูปแบบ XML มีค่าเท่ากับ “6 0 R”



รูปที่ 31 ตัวอย่างการเลือก object เพื่อค้นหา Indirect reference ของ ไฟล์ใบกำกับภาษีในรูปแบบ XML บน iText RUPS

#### 5.5.2 การตรวจสอบข้อมูล Catalog Dictionary ของชื่อไฟล์ใบกำกับภาษีในรูปแบบ XML (/F)

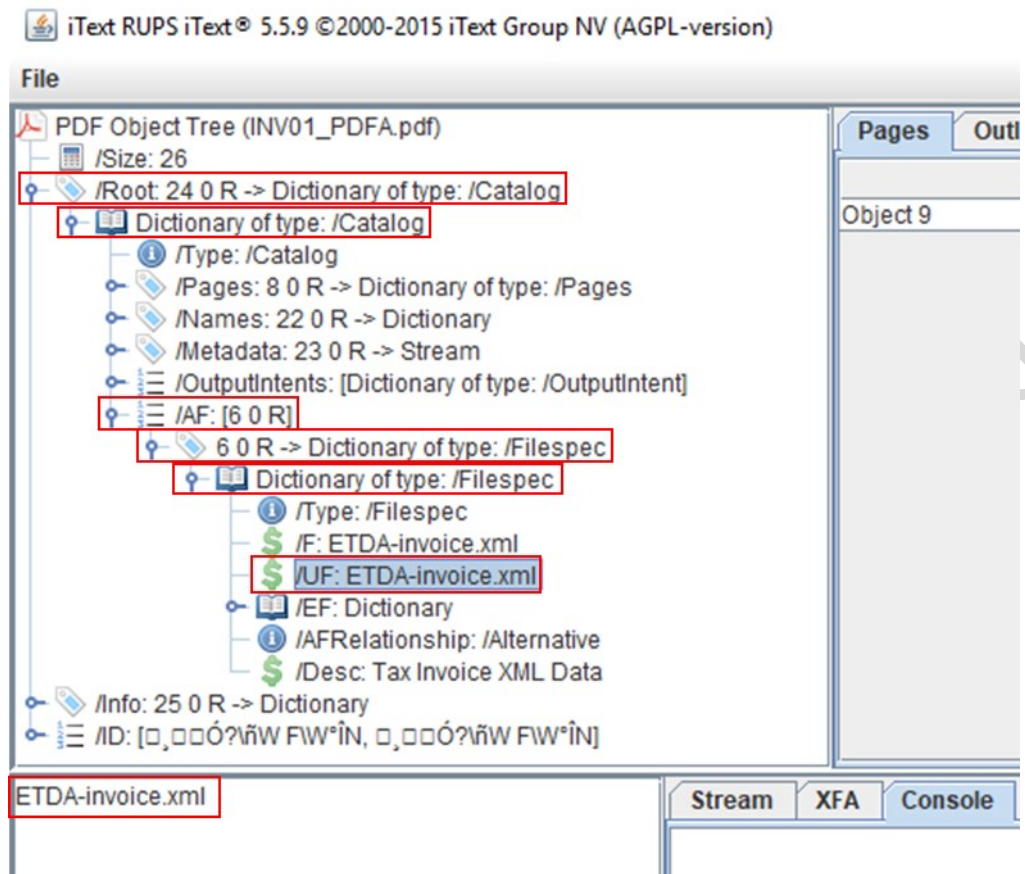
- 1) เปิดเครื่องมือ iText RUPS และเลือกเมนู File แล้ว Open เพื่อเลือกไฟล์ PDF ที่ต้องการตรวจสอบ โดยแสดงตัวอย่างหน้าจอของ iText RUPS ไว้ในรูปที่ 27
- 2) หน้าจอ iText RUPS จะแสดงไฟล์ตามตัวอย่างในรูปที่ 28 ให้เลือก object ภายในไฟล์ PDF ตามลำดับชั้นดังนี้ “/Root”, “Dictionary of type: /Catalog”, “/AF”, “Dictionary of type: /Filespec” ของ element ที่มีค่า indirect reference ของไฟล์ใบกำกับภาษีในรูปแบบ XML โดยตัวอย่างในรูปที่ 32 มีค่าเท่ากับ “6 0 R”, “Dictionary of type: /Filespec” และ “/F” โดยแสดงลำดับชั้นที่เลือกและแสดงข้อมูล “/F” บนหน้าจอตามตัวอย่างในรูปที่ 32 โดยข้อมูลดังกล่าวต้องมีค่าเท่ากับค่าที่ระบุไว้ตามตารางที่ 5



รูปที่ 32 ตัวอย่างการเลือก object เพื่อแสดงข้อมูลชื่อไฟล์ใบกำกับภาษีในรูปแบบ XML (/F) บน iText RUPS

### 5.5.3 การตรวจสอบข้อมูล Catalog Dictionary ของชื่อไฟล์ใบกำกับภาษีในรูปแบบ XML ที่เป็น Unicode encoding (/UF)

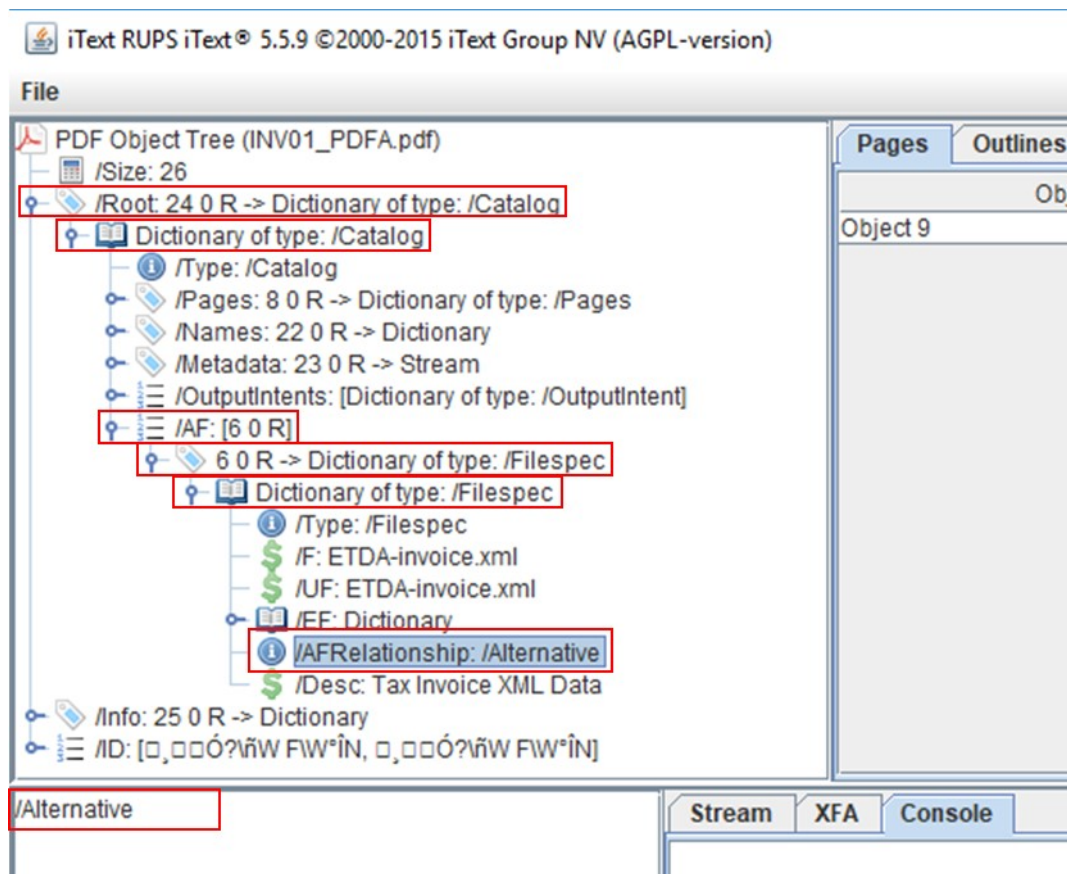
- 1) เปิดเครื่องมือ iText RUPS และเลือกเมนู File แล้ว Open เพื่อเลือกไฟล์ PDF ที่ต้องการตรวจสอบ โดยแสดงตัวอย่างหน้าจอของ iText RUPS ไว้ในรูปที่ 27
- 2) หน้าจอ iText RUPS จะแสดงไฟล์ตามตัวอย่างในรูปที่ 28 ให้เลือก object ภายในไฟล์ PDF ตามลำดับขั้นดังนี้ “/Root”, “Dictionary of type: /Catalog”, “/AF”, “Dictionary of type: /Filespec” ของ element ที่มีค่า indirect reference ของไฟล์ใบกำกับภาษีในรูปแบบ XML โดยตัวอย่างในรูปที่ 33 มีค่าเท่ากับ “6 0 R”, “Dictionary of type: /Filespec” และ “/UF” โดยแสดงลำดับขั้นที่เลือกและแสดงข้อมูล “/UF” บนหน้าจอตามตัวอย่างในรูปที่ 33 โดยข้อมูลดังกล่าวต้องมีค่าเท่ากับค่าที่ระบุไว้ในตารางที่ 5



รูปที่ 33 ตัวอย่างการเลือก object เพื่อแสดงข้อมูลชื่อไฟล์ที่เป็น Unicode encoding (/UF) บน iText RUPS

#### 5.5.4 การตรวจสอบข้อมูล Catalog Dictionary แสดงความสัมพันธ์ระหว่างไฟล์ PDF และไฟล์ใบกำกับภาษีในรูปแบบ XML (/AFRelationship)

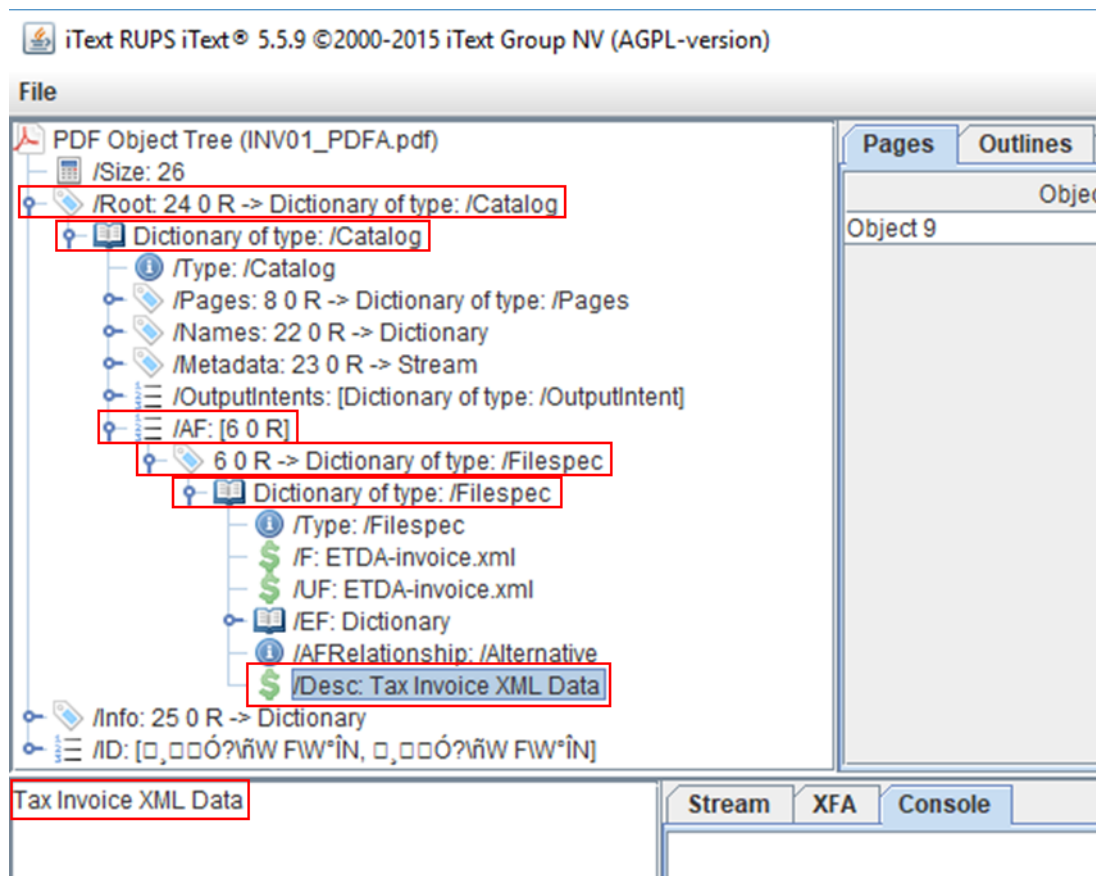
- 1) เปิดเครื่องมือ iText RUPS และเลือกเมนู File แล้ว Open เพื่อเลือกไฟล์ PDF ที่ต้องการตรวจสอบ โดยแสดงตัวอย่างหน้าจอของ iText RUPS ไว้ในรูปที่ 27
- 2) หน้าจอ iText RUPS จะแสดงไฟล์ตามตัวอย่างในรูปที่ 28 ให้เลือก object ภายในไฟล์ PDF ตามลำดับขั้นดังนี้ “/Root”, “Dictionary of type: /Catalog”, “/AF”, “Dictionary of type: /Filespec” ของ element ที่มีค่า indirect reference ของไฟล์ใบกำกับภาษีในรูปแบบ XML โดยตัวอย่างในรูปที่ 34 มีค่าเท่ากับ “6 0 R”, “Dictionary of type: /Filespec” และ “/AFRelationship” โดยแสดงลำดับขั้นที่เลือกและแสดงข้อมูล “/AFRelationship” บนหน้าจอตามตัวอย่างในรูปที่ 34 โดยข้อมูลดังกล่าวต้องมีค่าเท่ากับค่าที่ระบุไว้ตามตารางที่ 5



รูปที่ 34 ตัวอย่างการเลือก object เพื่อแสดงความสัมพันธ์ระหว่างไฟล์ PDF และไฟล์ใบกำกับภาษีบน iText RUPS

#### 5.5.5 การตรวจสอบข้อมูล Catalog Dictionary แสดงคำอธิบาย (Description) ของไฟล์ใบกำกับภาษีในรูปแบบ XML (/Desc)

- 1) เปิดเครื่องมือ iText RUPS และเลือกเมนู File แล้ว Open เพื่อเลือกไฟล์ PDF ที่ต้องการตรวจสอบ โดยแสดงตัวอย่างหน้าจอของ iText RUPS ไว้ในรูปที่ 27
- 2) หน้าจอ iText RUPS จะแสดงไฟล์ตามตัวอย่างในรูปที่ 28 ให้เลือก object ภายในไฟล์ PDF ตามลำดับขั้นดังนี้ “/Root”, “Dictionary of type: /Catalog”, “/AF”, “Dictionary of type: /Filespec” ของ element ที่มีค่า indirect reference ของไฟล์ใบกำกับภาษีในรูปแบบ XML โดยตัวอย่างในรูปที่ 35 มีค่าเท่ากับ “6 0 R”, “Dictionary of type: /Filespec” และ “/Desc” โดยแสดงลำดับขั้นที่เลือกและแสดงข้อมูล “/Desc” บนหน้าจอตามตัวอย่างในรูปที่ 35 โดยข้อมูลดังกล่าวต้องมีค่าเท่ากับค่าที่ระบุไว้ตามตารางที่ 5

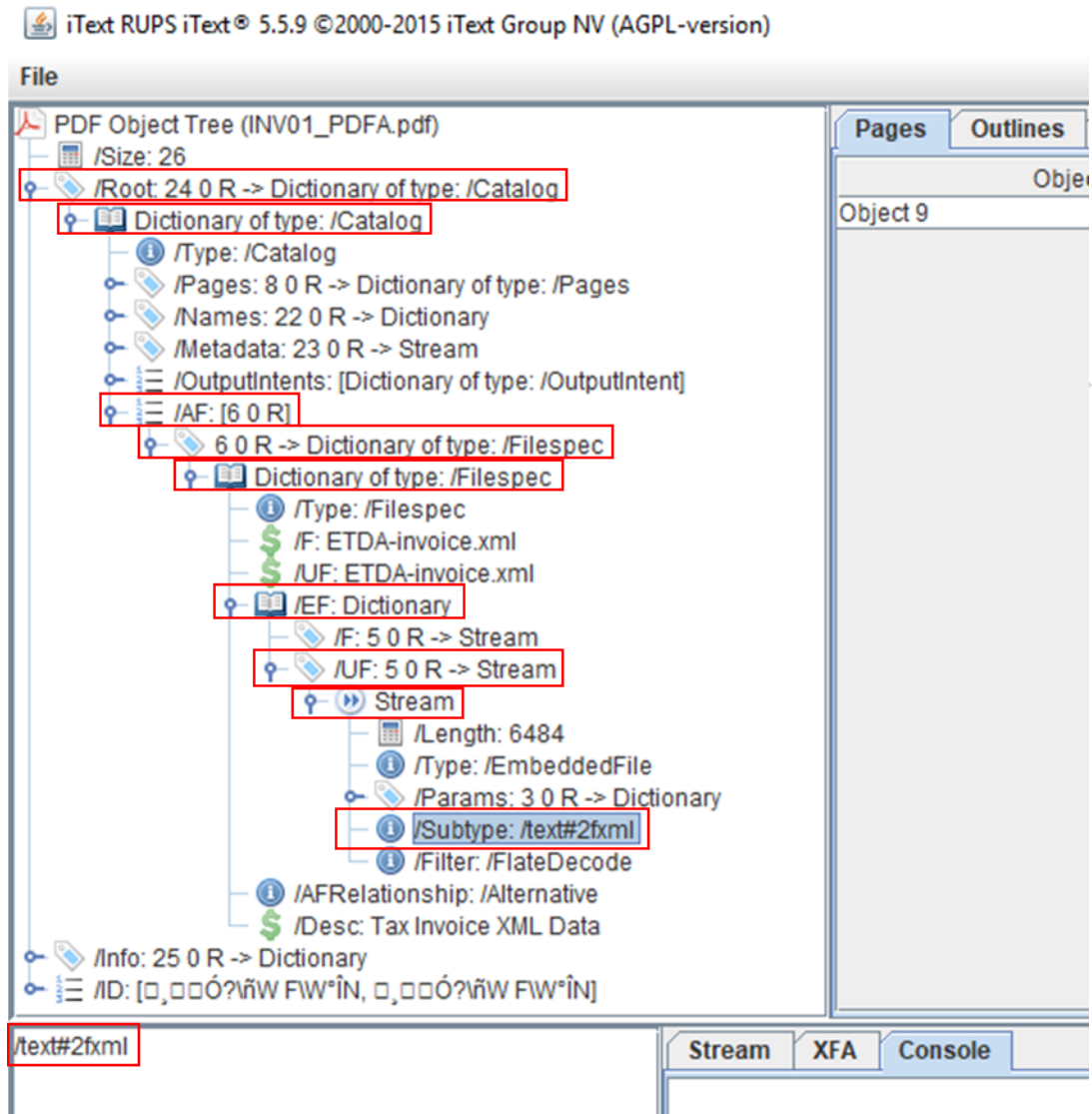


รูปที่ 35 ตัวอย่างการเลือก object เพื่อแสดงคำอธิบาย (Description) ของไฟล์ใบกำกับภาษีบน iText RUPS

#### 5.5.6 การตรวจสอบข้อมูล Catalog Dictionary ของประเภทของไฟล์ใบกำกับภาษีในรูปแบบ XML (/Subtype)

- 1) เปิดเครื่องมือ iText RUPS และเลือกเมนู File แล้ว Open เพื่อเลือกไฟล์ PDF ที่ต้องการตรวจสอบ โดยแสดงตัวอย่างหน้าจอของ iText RUPS ไว้ในรูปที่ 27
- 2) หน้าจอ iText RUPS จะแสดงไฟล์ตามตัวอย่างในรูปที่ 28 ให้เลือก object ภายในไฟล์ PDF ตามลำดับขั้นดังนี้ “/Root”, “Dictionary of type: /Catalog”, “/AF”, “Dictionary of type: /Filespec” ของ element ที่มีค่า indirect reference ของไฟล์ใบกำกับภาษีในรูปแบบ XML โดยตัวอย่างในรูปที่ 36 มีค่าเท่ากับ “6 0 R”, “Dictionary of type: /Filespec”, “/EF”, “/UF”, “Stream” และ “/Subtype” โดยแสดงลำดับขั้นที่เลือกและแสดงข้อมูล “/Subtype” บนหน้าจอตามตัวอย่างในรูปที่ 36 โดยข้อมูลดังกล่าวต้องมีค่าเท่ากับค่าที่ระบุไว้ตามตารางที่ 5



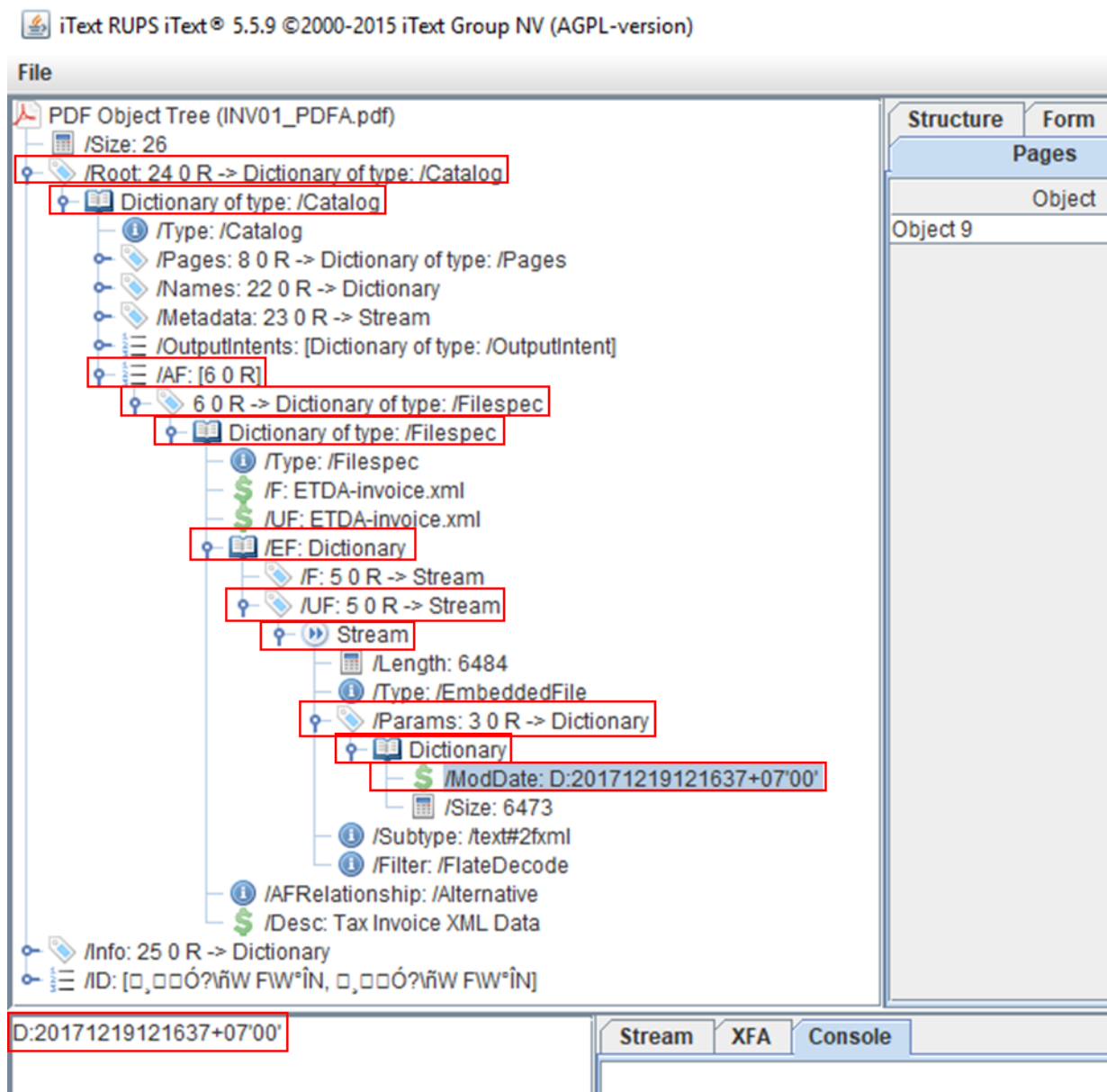


รูปที่ 36 ตัวอย่างการเลือก object เพื่อแสดงข้อมูลประเภทของไฟล์ (/Subtype) บน iText RUPS

#### 5.5.7 การตรวจสอบข้อมูล Catalog Dictionary ของวันที่ที่แก้ไขไฟล์ใบกำกับภาษีในรูปแบบ XML (/ModDate)

- 1) เปิดเครื่องมือ iText RUPS และเลือกเมนู File แล้ว Open เพื่อเลือกไฟล์ PDF ที่ต้องการตรวจสอบ โดยแสดงตัวอย่างหน้าจอของ iText RUPS ไว้ในรูปที่ 27
- 2) หน้าจอ iText RUPS จะแสดงไฟล์ตามตัวอย่างในรูปที่ 28 ให้เลือก object ภายในไฟล์ PDF ตามลำดับขั้นดังนี้ “/Root”, “Dictionary of type: /Catalog”, “/AF”, “Dictionary of type: /Filespec” ของ element ที่มีค่า indirect reference ของไฟล์ใบกำกับภาษีในรูปแบบ XML โดยตัวอย่างในรูปที่ 37 มีค่าเท่ากับ “6 0 R”, “Dictionary of type: /Filespec”, “/EF”, “/UF”, “Stream” “/Params” “Dictionary” และ “/ModDate” โดยแสดงลำดับขั้นที่เลือกและแสดง

ข้อมูล “/ModDate” บนหน้าจอตามตัวอย่างในรูปที่ 37 รูปที่ 30 โดยข้อมูลดังกล่าวต้องมีค่าเท่ากับค่าที่ระบุไว้ตามตารางที่ 5



รูปที่ 37 ตัวอย่างการเลือก object เพื่อแสดงข้อมูลวันที่ที่แก้ไขไฟล์ (/ModDate) บน iText RUPS

## 5.6 การเปรียบเทียบ Visual Representation และข้อมูลใน XML

ข้อมูลใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ XML และข้อมูลใบกำกับภาษีอิเล็กทรอนิกส์ในรูปแบบ PDF ที่ถูกต้องนั้น ต้องมีข้อมูลที่เหมือนกัน การตรวจสอบสามารถทำได้ด้วยการเปรียบเทียบรายการข้อมูลใบกำกับภาษีที่จำเป็นในรูปแบบ PDF และในรูปแบบ XML ตามที่แสดงในตารางที่ 6



ตารางที่ 6 เปรียบเทียบระหว่างข้อมูลในรูปแบบ PDF และในรูปแบบ XML

No.	รายการข้อมูลใบกำกับภาษีในรูปแบบ PDF	Index ของตารางที่ 2 แสดงโครงสร้างใบกำกับภาษี ใบเพิ่มหนี้/ใบลดหนี้ ใน ชมธอ. 3-2560	Tag ในใบกำกับภาษีในรูปแบบ XML
1	ใบกำกับภาษี, ใบเพิ่มหนี้, ใบลดหนี้	2.3	<ram:TypeCode>*
2	ชื่อผู้ออกใบกำกับภาษี	3.1.1.3	<ram:Name>
3	รหัสไปรษณีย์ของผู้ออกใบกำกับภาษี	3.1.1.6.1	<ram:PostcodeCode>
4	ชื่ออาคารของผู้ออกใบกำกับภาษี	3.1.1.6.2	<ram:BuildingName>
5	ที่อยู่บรรทัดที่ 1 ของผู้ออกใบกำกับภาษี	3.1.1.6.3	<ram:LineOne>
6	ที่อยู่บรรทัดที่ 2 ของผู้ออกใบกำกับภาษี	3.1.1.6.4	<ram:LineTwo>
7	ซอยของผู้ออกใบกำกับภาษี	3.1.1.6.5	<ram:LineThree>
8	หมู่บ้านของผู้ออกใบกำกับภาษี	3.1.1.6.6	<ram:LineFour>
9	หมู่ที่ของผู้ออกใบกำกับภาษี	3.1.1.6.7	<ram:LineFive>
10	ถนนของผู้ออกใบกำกับภาษี	3.1.1.6.8	<ram:StreetName>
11	ชื่ออำเภอของผู้ออกใบกำกับภาษี	3.1.1.6.9	<ram:CityName>
12	ชื่อตำบลของผู้ออกใบกำกับภาษี	3.1.1.6.10	<ram:CitySubDivisionName>
13	รหัสประเทศของผู้ออกใบกำกับภาษี	3.1.1.6.11	<ram:CountryID> *
14	รหัสจังหวัดของผู้ออกใบกำกับภาษี	3.1.1.6.12	<ram:CountrySubDivisionID> *
15	บ้านเลขที่ของผู้ออกใบกำกับภาษี	3.1.1.6.13	<ram:BuildingNumber>
16	เลขประจำตัวผู้เสียภาษีอากรของผู้ออกใบกำกับภาษี	3.1.1.4.1	<ram:ID>

No.	รายการข้อมูลใบกำกับภาษีในรูปแบบ PDF	Index ของตารางที่ 2 แสดงโครงสร้าง ใบกำกับภาษี ใบเพิ่มหนี้/ใบลดหนี้ ใน ชมธอ. 3-2560	Tag ในใบกำกับภาษีในรูปแบบ XML
17	ชื่อผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.3	<ram:Name>
18	รหัสไปรษณีย์ของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.1	<ram:PostcodeCode>
19	ชื่ออาคารของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.2	<ram:BuildingName>
20	ที่อยู่บรรทัดที่ 1 ของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.3	<ram:LineOne>
21	ที่อยู่บรรทัดที่ 2 ของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.4	<ram:LineTwo>
22	ซอยของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.5	<ram:LineThree>
23	หมู่บ้านของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.6	<ram:LineFour>
24	หมู่ที่ของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.7	<ram:LineFive>
25	ถนนของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.8	<ram:StreetName>
26	ชื่ออำเภอของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.9	<ram:CityName>
27	ชื่อตำบลของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.10	<ram:CitySubDivisionName>
28	รหัสประเทศของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.11	<ram:CountryID> *
29	รหัสจังหวัดของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.12	<ram:CountrySubDivisionID> *
30	บ้านเลขที่ของผู้ซื้อสินค้าหรือผู้รับบริการ	3.1.2.6.13	<ram:BuildingNumber>
31	หมายเลขลำดับของใบกำกับภาษี	2.1	<ram:ID>

No.	รายการข้อมูลใบกำกับภาษีในรูปแบบ PDF	Index ของตารางที่ 2 แสดงโครงสร้าง ใบกำกับภาษี ใบเพิ่มหนี้/ใบลดหนี้ ใน ชมธอ. 3-2560	Tag ในใบกำกับภาษีในรูปแบบ XML
32	ชื่อของสินค้าหรือของบริการ	3.4.5.3	<ram:Name>
33	ประเภทของสินค้าหรือของบริการ	3.4.5.6.2	<ram:ClassName>
34	ปริมาณสินค้าหรือของบริการ	3.4.3.1	<ram:BilledQuantity>
35	มูลค่าของสินค้าหรือของบริการ	3.4.4.3.2	<ram:NetLineTotalAmount>
36	จำนวนภาษีมูลค่าเพิ่มที่คำนวณจากมูลค่าของสินค้าหรือของบริการ	3.4.4.3.1	<ram:TaxTotalAmount>
37	วัน เดือน ปี ที่ออกใบกำกับภาษี	2.4	<ram:IssueDateTime>
38	ข้อความอื่นที่อธิบติกำหนด	**	**

หมายเหตุ: \*ค่าที่แสดงใน XML เป็นรหัส สามารถเปรียบเทียบค่าและรหัสได้ในตารางใน ชมธอ 3-2560

\*\* รายละเอียดขึ้นอยู่กับธุรกิจแต่ละประเภท

## 6. Appendix

### Appendix A

#### E-Tax invoice XML file

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <rsm:TaxInvoice_CrossIndustryInvoice
3.   xmlns:ram="urn:etda:uncefact:data:standard:TaxInvoice_ReusableAggregateBusinessInformationEnti
   ty:1"
4.   xmlns:rsm="urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2"
5.   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6.   xsi:schemaLocation="urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2 file:///data
   /standard/TaxInvoice_CrossIndustryInvoice_2p0.xsd"
7.   xmlns:udt="urn:un:unece:uncefact:data:standard:UnqualifiedDataType:15"
8. >
9.   <rsm:ExchangedDocumentContext>
10.     <ram:GuidelineSpecifiedDocumentContextParameter>
11.       <ram:ID schemeAgencyID="ETDA" schemeVersionID="v2.0">ER3-2560</ram:ID>
12.     </ram:GuidelineSpecifiedDocumentContextParameter>
13.   </rsm:ExchangedDocumentContext>
14.   <rsm:ExchangedDocument>
15.     <ram:ID>INVqwe00002</ram:ID>
16.     <ram:Name>ใบกำกับภาษี</ram:Name>
17.     <ram:TypeCode>388</ram:TypeCode>
18.     <ram:IssueDateTime>2017-08-21T00:00:00.000</ram:IssueDateTime>
19.     <ram:CreationDateTime>2017-08-21T00:00:00.000</ram:CreationDateTime>
20.   </rsm:ExchangedDocument>
21.   <rsm:SupplyChainTradeTransaction>
22.     <ram:ApplicableHeaderTradeAgreement>
23.       <ram:SellerTradeParty>
24.         <ram:Name>บริษัท ขยันหมั่นเพียร จำกัด</ram:Name>
25.         <ram:SpecifiedTaxRegistration>
26.           <ram:ID schemeID="TXID">1111111111110000</ram:ID>
27.         </ram:SpecifiedTaxRegistration>
28.         <ram:DefinedTradeContact>
29.           <ram:EmailURIUniversalCommunication>
30.             <ram:URIID>natural@example.com</ram:URIID>
31.           </ram:EmailURIUniversalCommunication>
32.           <ram:TelephoneUniversalCommunication>
33.             <ram:CompleteNumber>+66-813820359</ram:CompleteNumber>
34.           </ram:TelephoneUniversalCommunication>
35.         </ram:DefinedTradeContact>
36.         <ram:PostalTradeAddress>
37.           <ram:PostcodeCode>71180</ram:PostcodeCode>
38.           <ram:LineOne>สำนักงานอุทยานแห่งชาติ ทองผาภูมิ</ram:LineOne>
39.           <ram:CityName>7107</ram:CityName>
40.           <ram:CitySubDivisionName>710705</ram:CitySubDivisionName>
41.           <ram:CountryID schemeID="3166-1 alpha-2">TH</ram:CountryID>
42.           <ram:CountrySubDivisionID>71</ram:CountrySubDivisionID>
43.           <ram:BuildingNumber>777/777</ram:BuildingNumber>
44.         </ram:PostalTradeAddress>
```

```

45.         </ram:SellerTradeParty>
46.         <ram:BuyerTradeParty>
47.             <ram:Name>บริษัททำดีจำกัด</ram:Name>
48.             <ram:SpecifiedTaxRegistration>
49.                 <ram:ID schemeID="TXID">2222222222220000</ram:ID>
50.             </ram:SpecifiedTaxRegistration>
51.             <ram:DefinedTradeContact>                 <ram:EmailURIUniversalCommunication>
52.                 <ram:URIID>patiw@example.com</ram:URIID>
53.             </ram:EmailURIUniversalCommunication>
54.             <ram:TelephoneUniversalCommunication>
55.                 <ram:CompleteNumber>+66-948769870</ram:CompleteNumber>
56.             </ram:TelephoneUniversalCommunication>
57.             </ram:DefinedTradeContact>
58.             <ram:PostalTradeAddress>
59.                 <ram:PostcodeCode>11344</ram:PostcodeCode>
60.                 <ram:LineOne>หาดทุ่งวัวแล่น </ram:LineOne>
61.                 <ram:CityName>8603</ram:CityName>
62.                 <ram:CitySubDivisionName>860303</ram:CitySubDivisionName>
63.                 <ram:CountryID schemeID="3166-1 alpha-2">TH</ram:CountryID>
64.                 <ram:CountrySubDivisionID>86</ram:CountrySubDivisionID>
65.                 <ram:BuildingNumber>77/79</ram:BuildingNumber>
66.             </ram:PostalTradeAddress>
67.         </ram:BuyerTradeParty>
68.     </ram:ApplicableHeaderTradeAgreement>
69.     <ram:ApplicableHeaderTradeDelivery>
70.         <ram:ShipToTradeParty>
71.             <ram:DefinedTradeContact>
72.                 <ram:PersonName>สมพร ใจงาม</ram:PersonName>
73.             </ram:DefinedTradeContact>
74.         </ram:ShipToTradeParty>
75.     </ram:ApplicableHeaderTradeDelivery>
76.     <ram:ApplicableHeaderTradeSettlement>
77.         <ram:InvoiceCurrencyCode listID="ISO 4217 3A">THB</ram:InvoiceCurrencyCode>
78.         <ram:ApplicableTradeTax>
79.             <ram:TypeCode>VAT</ram:TypeCode>
80.             <ram:CalculatedRate>7</ram:CalculatedRate>
81.             <ram:BasisAmount>300</ram:BasisAmount>
82.             <ram:CalculatedAmount>21</ram:CalculatedAmount>
83.         </ram:ApplicableTradeTax>
84.         <ram:SpecifiedTradeSettlementHeaderMonetarySummation>
85.             <ram:LineTotalAmount>300</ram:LineTotalAmount>
86.             <ram:TaxBasisTotalAmount>300</ram:TaxBasisTotalAmount>
87.             <ram:TaxTotalAmount>21</ram:TaxTotalAmount>
88.             <ram:GrandTotalAmount>321</ram:GrandTotalAmount>
89.         </ram:SpecifiedTradeSettlementHeaderMonetarySummation>
90.     </ram:ApplicableHeaderTradeSettlement>
91.     <ram:IncludedSupplyChainTradeLineItem>
92.         <ram:AssociatedDocumentLineDocument>
93.             <ram:LineID>1</ram:LineID>
94.         </ram:AssociatedDocumentLineDocument>
95.         <ram:SpecifiedTradeProduct>
96.             <ram:Name>สินค้าทดสอบ </ram:Name>

```

```

97.         </ram:SpecifiedTradeProduct>
98.         <ram:SpecifiedLineTradeAgreement>
99.             <ram:GrossPriceProductTradePrice>
100.                 <ram:ChargeAmount>300</ram:ChargeAmount>
101.             </ram:GrossPriceProductTradePrice>
102.         </ram:SpecifiedLineTradeAgreement>
103.         <ram:SpecifiedLineTradeDelivery>
104.             <ram:BilledQuantity unitCode="XCS">1</ram:BilledQuantity>
105.         </ram:SpecifiedLineTradeDelivery>
106.         <ram:SpecifiedLineTradeSettlement>
107.             <ram:SpecifiedTradeSettlementLineMonetarySummation>
108.                 <ram:NetLineTotalAmount>300</ram:NetLineTotalAmount>
109.                 <ram:NetIncludingTaxesLineTotalAmount>321</ram:NetIncludingTaxesLineT
otalAmount>
110.             </ram:SpecifiedTradeSettlementLineMonetarySummation>
111.         </ram:SpecifiedLineTradeSettlement>
112.     </ram:IncludedSupplyChainTradeLineItem>
113. </rsm:SupplyChainTradeTransaction>
114. </rsm:TaxInvoice_CrossIndustryInvoice>

```

## Appendix B

### XSLT file

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <xsl:stylesheet exclude-result-prefixes="rsm ram udt"
3.     xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4.     xmlns:ram="urn:etda:uncefact:data:standard:TaxInvoice_ReusableAggregateBusiness
InformationEntity:1"
5.     xmlns:rsm="urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2"
6.     xmlns:udt="urn:un:unece:uncefact:data:standard:UnqualifiedDataType:15" version="1.0"
7.     xmlns:utils="inbatek:Utils"
8. >
9. <xsl:output method="xml" indent="yes" encoding="UTF-8" />
10.
11. <xsl:template match="/rsm:TaxInvoice_CrossIndustryInvoice">
12.     <html>
13.         <head>
14.             <link rel="stylesheet" type="text/css" href="invoice.css" />
15.         </head>
16.         <body>
17.             
18.             <br />
19.             <xsl:apply-templates />
20.         </body>
21.     </html>
22. </xsl:template>
23. <xsl:template match="rsm:ExchangedDocumentContext" />
24. <xsl:template match="rsm:ExchangedDocument">
25.     <h1 id="header">

```

```

26.     <xsl:value-of select="ram.Name" />
27.     <xsl:text> </xsl:text>
28.     <xsl:value-of select="ram.ID" />
29.     </h1>
30.     <br />
31.     <h2 id="date">วันที่ออก:</h2>
32.     <xsl:call-template name="calculate-buddisth-date">
33.       <xsl:with-param name="date-time" select="ram.IssueDateTime"/>
34.     </xsl:call-template>
35.     </h2>
36.   </xsl:template>
37.   <xsl:template name="calculate-buddisth-date">
38.     <xsl:param name="date-time"/>
39.     <xsl:param name="date" select="substring-before($date-time, 'T')"/>
40.     <xsl:param name="year" select="substring-before($date, '-')"/>
41.     <xsl:param name="month"
42.       select="substring-before(substring-after($date, '-'), '-')"/>
43.     <xsl:param name="day" select="substring-after(substring-after($date, '-'), '-')"/>
44.     <xsl:variable name="by" select="concat(($year + 543), '-', $month, '-', $day)"/>
45.     <xsl:value-of select="$by"/>
46.   </xsl:template>
47.   <xsl:template match="rsm:SupplyChainTradeTransaction">
48.     <table width="100%" border="0" id="addresses">
49.       <tr>
50.         <td valign="Top" width="10%">
51.           <b>ผู้ขาย</b>
52.         </td>
53.         <td valign="Top" width="40%">
54.           <xsl:for-each select="ram.ApplicableHeaderTradeAgreement/ram.SellerTradeParty">
55.             <xsl:call-template name="address" />
56.           </xsl:for-each>
57.         </td>
58.         <td valign="Top" width="10%">
59.           <b>ผู้ซื้อ</b>
60.         </td>
61.         <td valign="Top" width="40%">
62.           <xsl:for-each select="ram.ApplicableHeaderTradeAgreement/ram.BuyerTradeParty">
63.             <xsl:call-template name="address" />
64.           </xsl:for-each>
65.         </td>
66.       </tr>
67.     </table>
68.     <table width="100%" id="products">
69.       <tr class="headerrow">
70.         <th>ลำดับที่</th>
71.         <th>รายละเอียดสินค้า</th>
72.         <th>ราคา/หน่วย</th>
73.         <th>จำนวน</th>
74.         <th>ราคารวม</th>
75.         <th>มูลค่าภาษี</th>
76.         <th>ราคาสุทธิ</th>
77.       </tr>
78.       <xsl:for-each select="ram.IncludedSupplyChainTradeLineItem">

```

```

79.     <tr>
80.         <td align="Right">
81.             <xsl:value-of select="ram.AssociatedDocumentLineDocument/ram.LineID" />
82.         </td>
83.         <td align="Right">
84.             <xsl:value-of select="ram.SpecifiedTradeProduct/ram.Name" />
85.         </td>
86.         <td align="Right">
87.             <xsl:call-template name="twodecimals">
88.                 <xsl:with-
param name="number" select="ram.SpecifiedLineTradeAgreement/ram.GrossPriceProductTradePrice/ram.
ChargeAmount" />
89.                 <xsl:call-template>
90.             </td>
91.         <td align="Right">
92.             <xsl:value-of select="ram.SpecifiedLineTradeDelivery/ram.BilledQuantity" />
93.         </td>
94.         <td align="Right">
95.             <xsl:call-template name="twodecimals">
96.                 <xsl:with-
param name="number" select="ram.SpecifiedLineTradeSettlement/ram.SpecifiedTradeSettlementLineMo
netarySummation/ram.NetLineTotalAmount" />
97.                 <xsl:call-template>
98.             </td>
99.         <td align="Right">
100.            <xsl:call-template name="calcTax">
101.                <xsl:with-
param name="noTax" select="ram.SpecifiedLineTradeSettlement/ram.SpecifiedTradeSettlementLineMon
etarySummation/ram.NetLineTotalAmount" />
102.                <xsl:with-
param name="withTax" select="ram.SpecifiedLineTradeSettlement/ram.SpecifiedTradeSettlementLineM
onetarySummation/ram.NetIncludingTaxesLineTotalAmount" />
103.                <xsl:call-template>
104.            </td>
105.            <td align="Right" class="bold">
106.                <xsl:call-template name="twodecimals">
107.                    <xsl:with-
param name="number" select="ram.SpecifiedLineTradeSettlement/ram.SpecifiedTradeSettlementLineMo
netarySummation/ram.NetIncludingTaxesLineTotalAmount" />
108.                    <xsl:call-template>
109.                </td>
110.        </tr>
111.        <xsl:for-each>
112.            <table>
113.                <table width="100%" id="totals">
114.                    <tr class="headerrow">
115.                        <th height="30">ภาษี</th>
116.                        <th>ภาษี(<th>ภาษี(</th>
117.                        <th>ราคารวม</th>
118.                        <th>ราคาภาษี</th>
119.                        <th>ราคาสุทธิ</th>
120.                        <th>หน่วยเงินตรา.</th>
121.                    </tr>

```



```

122.         <xsl:for-
each select="ram:ApplicableHeaderTradeSettlement/ram:ApplicableTradeTax">
123.             <tr>
124.                 <td align="Right">
125.                     <xsl:value-of select="ram:TypeCode" />
126.                 </td>
127.                 <td align="Right">
128.                     <xsl:value-of select="ram:CalculatedRate" />
129.                 </td>
130.                 <td align="Right">
131.                     <xsl:value-of select="ram:BasisAmount" />
132.                 </td>
133.                 <td align="Right">
134.                     <xsl:value-of select="ram:CalculatedAmount" />
135.                 </td>
136.                 <td align="Right">
137.                     <xsl:call-template name="calcWithTax">
138.                         <xsl:with-param name="basis" select="ram:BasisAmount" />
139.                         <xsl:with-param name="tax" select="ram:CalculatedRate" />
140.                     </xsl:call-template>
141.                 </td>
142.                 <td>
143.                     <xsl:value-of select="@ram:InvoiceCurrencyCode" />
144.                 </td>
145.             </tr>
146.         </xsl:for-each>
147.         <tr>
148.             <td class="total" align="Right" colspan="2">ราคารวมสุทธิ:</td>
149.             <td class="total" align="Right">
150.                 <xsl:value-
of select="ram:ApplicableHeaderTradeSettlement/ram:SpecifiedTradeSettlementHeaderMonetarySumm
ation/ram:TaxBasisTotalAmount" />
151.             </td>
152.             <td class="total" align="Right">
153.                 <xsl:value-
of select="ram:ApplicableHeaderTradeSettlement/ram:SpecifiedTradeSettlementHeaderMonetarySumm
ation/ram:TaxTotalAmount" />
154.             </td>
155.             <td class="total" align="Right">
156.                 <xsl:value-
of select="ram:ApplicableHeaderTradeSettlement/ram:SpecifiedTradeSettlementHeaderMonetarySumm
ation/ram:GrandTotalAmount" />
157.             </td>
158.             <td class="total">
159.                 <xsl:value-
of select="ram:ApplicableHeaderTradeSettlement/ram:InvoiceCurrencyCode" />
160.             </td>
161.         </tr>
162.         <tr>
163.             <td class="total" align="Right" colspan="2">ราคารวมสุทธิ (ตัวอักษรไทย):</td>
164.             <td colspan="3" align="Left">
165.                 <xsl:value-
of select="utils:BahtText(ram:ApplicableHeaderTradeSettlement/ram:SpecifiedTradeSettlementHeade
rMonetarySummation/ram:GrandTotalAmount)" />

```

```

166.         </td>
167.     </tr>
168. </table>
169. </xsl:template>
170. <xsl:template match="udt:DateTimeString">
171.     <h2 id="date">
172.         <xsl:choose>
173.             <xsl:when test="@format='610'">
174.                 <xsl:call-template name="YYYYMM">
175.                     <xsl:with-param name="date" select="." />
176.                 </xsl:call-template>
177.             </xsl:when>
178.             <xsl:when test="@format='616'">
179.                 <xsl:call-template name="YYYYWW">
180.                     <xsl:with-param name="date" select="." />
181.                 </xsl:call-template>
182.             </xsl:when>
183.             <xsl:otherwise>
184.                 <xsl:call-template name="YYYYMMDD">
185.                     <xsl:with-param name="date" select="." />
186.                 </xsl:call-template>
187.             </xsl:otherwise>
188.         </xsl:choose>
189.     </h2>
190. </xsl:template>
191. <xsl:template name="address">
192.     <span class="name">
193.         <xsl:value-of select="ram:Name" />
194.     </span><br />
195.     <xsl:value-of select="ram:PostalTradeAddress/ram:BuildingNumber" /><br />
196.     <xsl:value-of select="ram:PostalTradeAddress/ram:LineOne" /><br />
197.     <xsl:value-
of select="utils:CitySubNameText(ram:PostalTradeAddress/ram:CitySubDivisionName)" /> <xsl:text> </
xsl:text>
198.     <xsl:value-
of select="utils:CityNameText(ram:PostalTradeAddress/ram:CityName)" /><br />
199.     <xsl:value-
of select="utils:CountrySubIDText(ram:PostalTradeAddress/ram:CountrySubDivisionID)" /> <xsl:text>
</xsl:text>
200.     <xsl:value-of select="ram:PostalTradeAddress/ram:PostcodeCode" /><br />
201.     <xsl:for-each select="ram:SpecifiedTaxRegistration/ram:ID">
202.         <xsl:value-of select="@schemeID" />: <xsl:value-of select="." /><br />
203.     </xsl:for-each>
204. </xsl:template>
205. <xsl:template name="tax">
206.     <xsl:value-of select="@schemeID"/>
207. </xsl:template>
208. <xsl:template name="calcTax">
209.     <xsl:param name="noTax" />
210.     <xsl:param name="withTax" />
211.     <xsl:call-template name="twodecimals">
212.         <xsl:with-param name="number" select="($withTax - $noTax)/>
213.     </xsl:call-template>

```

```

214.     </xsl:template>
215.     <xsl:template name="calcWithTax">
216.         <xsl:param name="basis" />
217.         <xsl:param name="tax" />
218.         <xsl:call-template name="twodecimals">
219.             <xsl:with-param name="number" select="($basis + (($basis * $tax) div 100))" />
220.         </xsl:call-template>
221.     </xsl:template>
222.     <xsl:template name="twodecimals">
223.         <xsl:param name="number" />
224.         <xsl:value-of select="format-number(round($number*100) div 100, '0.00')"/>
225.     </xsl:template>
226.     <xsl:template name="YYYYMMDD">
227.         <xsl:param name="date" />
228.         <xsl:value-of select="substring($date,1,4)" /><xsl:value-
of select="substring($date,5,2)" /><xsl:value-of select="substring($date,7,2)" />
229.     </xsl:template>
230.     <xsl:template name="YYYYMM">
231.         <xsl:param name="date" />
232.         <xsl:value-of select="substring($date,1,4)" /><xsl:value-
of select="substring($date,5,2)" />
233.     </xsl:template>
234.     <xsl:template name="YYYYWW">
235.         <xsl:param name="date" />
236.         <xsl:value-of select="substring($date,1,4)" />; week <xsl:value-
of select="substring($date,5,2)" />
237.     </xsl:template>
238. </xsl:stylesheet>

```

## Appendix C

### Create HTML Function

```

1. public virtual string createHtml(string transformXML, string xmlinvoice)
2. {
3.     XsltArgumentList arguments = new XsltArgumentList();
4.     arguments.AddExtensionObject("inbatek.Utills", new ThaiBahtText());
5.
6.     XslCompiledTransform proc = new XslCompiledTransform();
7.
8.     using (StringReader sr = new StringReader(transformXML))
9.     {
10.         using (XmlReader xr = XmlReader.Create(sr))
11.         {
12.             proc.Load(xr);
13.         }
14.     }
15.
16.     string resultXML;
17.
18.     using (StringReader sr = new StringReader(xmlinvoice))
19.     {

```

```

20.     using (XmlReader xr = XmlReader.Create(sr))
21.     {
22.         using (StringWriter sw = new StringWriter())
23.         {
24.             proc.Transform(xr, arguments, sw);
25.             resultXML = sw.ToString();
26.             return resultXML;
27.         }
28.     }
29. }
30. }

```

ภายในขั้นตอนการสร้าง HTML เราสามารถดึงข้อมูลจากไฟล์ XML มาประมวลผลด้วย C# ได้ โดยใช้ XSLT เป็นผู้เรียกใช้คลาสของ C# (สามารถดู source code ได้จาก Appendix B บรรทัดที่ 7 ,197,198 และ 199) ในตัวอย่างนี้จะใช้ C# คลาสชื่อ ThaiBahtText สำหรับการประมวลผลข้อมูลจาก XML ( สามารถดู source code ได้จาก Appendix H )

## Appendix D

### Create PDF Function

```

1. public void CreatePdf(string xsltfilename, string input, string outputpdffilename)
2. {
3.     var inputstr = File.ReadAllText(input);
4.     var xslstr = File.ReadAllText(xsltfilename);
5.     var html = createHtml(xslstr, inputstr);
6.     var htmlbytes = Encoding.UTF8.GetBytes(html);
7.     createPdf(File.ReadAllBytes(input), htmlbytes, outputpdffilename);
8. }

```

## Appendix E

### XMP file

```

1. <?xpacket begin="" id="W5M0MpCehiHzreSzNTczkc9d"?><x:xmpmeta xmlns:x="adobe:ns:meta" x:xmptk="Adobe XMP Core 5.10-jc003">
2.   <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
3.     <rdf:Description xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:pdf="http://ns.adobe.com/pdf/1.3/" xmlns:pdfaid="http://www.aiim.org/pdfans/id/" xmlns:xmp="http://ns.adobe.com/xap/1.0/" rdf:about="">
4.       <dc:format>application/pdf</dc:format>
5.       <pdfaid:part>3</pdfaid:part>
6.       <pdfaid:conformance>U</pdfaid:conformance>
7.     </rdf:Description>
8.     <rdf:Description xmlns:pdfaExtension="http://www.aiim.org/pdfans/extension/" xmlns:pdfaProperty="http://www.aiim.org/pdfans/property/" xmlns:pdfaSchema="http://www.aiim.org/pdfans/schema/" rdf:about="">

```

```

9.         <pdfaExtension:schemas>
10.             <rdf:Bag>
11.                 <rdf:li rdf:parseType="Resource">
12.                     <pdfaSchema:schema>Electronic Tax Invoice PDFa Extension Schema</pdfaSch
ema:schema>
13.                     <pdfaSchema:namespaceURI>urn:etda:uncefact:data:standard:Invoice_CrossIndu
stryInvoice.2#</pdfaSchema:namespaceURI>
14.                     <pdfaSchema:prefix>rsm</pdfaSchema:prefix>
15.                     <pdfaSchema:property>
16.                         <rdf:Seq>
17.                             <rdf:li rdf:parseType="Resource">
18.                                 <pdfaProperty:name>DocumentFileName</pdfaProperty:name>
19.                                 <pdfaProperty:valueType>Text</pdfaProperty:valueType>
20.                                 <pdfaProperty:category>external</pdfaProperty:category>
21.                                 <pdfaProperty:description>Name of the embedded XML invoice
file</pdfaProperty:description>
22.                             </rdf:li>
23.                             <rdf:li rdf:parseType="Resource">
24.                                 <pdfaProperty:name>DocumentType</pdfaProperty:name>
25.                                 <pdfaProperty:valueType>Text</pdfaProperty:valueType>
26.                                 <pdfaProperty:category>external</pdfaProperty:category>
27.                                 <pdfaProperty:description>Type of the document</pdfaProperty
:description>
28.                             </rdf:li>
29.                             <rdf:li rdf:parseType="Resource">
30.                                 <pdfaProperty:name>Version</pdfaProperty:name>
31.                                 <pdfaProperty:valueType>Text</pdfaProperty:valueType>
32.                                 <pdfaProperty:category>external</pdfaProperty:category>
33.                                 <pdfaProperty:description>Version of the ETDA XML data</pdfa
Property:description>
34.                             </rdf:li>
35.                         </rdf:Seq>
36.                     </pdfaSchema:property>
37.                 </rdf:li>
38.             </rdf:Bag>
39.         </pdfaExtension:schemas>
40.     </rdf:Description>
41.     <rdf:Description xmlns:rsm="urn:etda:uncefact:data:standard:Invoice_CrossIndustryInvoice.2#"
rdf:about="">
42.         <rsm:DocumentFileName>ETDA-invoice.xml</rsm:DocumentFileName>
43.         <rsm:DocumentType>@DocumentType</rsm:DocumentType>
44.         <rsm:Version>2.0</rsm:Version>
45.     </rdf:Description></rdf:RDF>
46. </x:xmpmeta><?xpacket end="r"?>

```

## Appendix F

### PDFa-3 Generator source code

```

1. public const string CSS = @"css\invoice.css";

```

```

2. public const string ICC = @"profile\sRGB_CS_profile.icm";
3. public const string EtdaSchemaNS = "urn:etda:uncefact:data:standard:Invoice_CrossIndustryInvoice2";
4. public const string EtdaDocumentFilename = "DocumentFileName";
5. public const string EtdaAttachmentFilename = "ETDA-invoice.xml";
6. public const string EtdaDocumentType = "DocumentType";
7. public const string EtdaDocumentType = "Tax Invoice";
8. public const string DocumentVersionname = "Version";
9. public const string DocumentVersion = "2.0";
10.
11. public virtual void createPdf(byte[] xml, byte[] invoice, string dest)
12. {
13.     // Create iTextSharp.text.Document
14.     Document document = new Document();
15.
16.     // Create iTextSharp.text.pdf.PdfWriter and assign Conformance
17.     PdfWriter writer = PdfWriter.GetInstance(document, new FileInfor(dest).OpenWrite(), (PdfACo
nformanceLevel)Etdata.text.pdf.PdfAConformanceLevel.PDF_A_3U);
18.
19.     //Set PDF version and create XMP object
20.     writer.SetTagged();
21.     writer.PdfVersion = PdfWriter.VERSION_1_7;
22.     writer.CreateXmpMetadata();
23.
24.     //Set XMP Parameter
25.     var metadata = writer.XmpMetadata;
26.     XmpMetaFactory.SchemaRegistry.RegisterNamespace(EtdaSchemaNS, "rsm");
27.     writer.XmpWriter.SetProperty(EtdaSchemaNS, EtdaDocumentFilename, EtdaAttachmentFilename);
28.
29.     writer.XmpWriter.SetProperty(EtdaSchemaNS, EtdaDocumentType, EtdaDocumentType);
30.     writer.XmpWriter.SetProperty(EtdaSchemaNS, DocumentVersionname, DocumentVersion);
31.
32.     document.Open();
33.
34.     // Set ICC Colour Profile
35.     ICC_Profile icc = ICC_Profile.GetInstance(File.ReadAllBytes(ICC));
36.     writer.SetOutputIntents("Custom", "", "http://www.color.org", "sRGB IEC61966-2.1", icc);
37.
38.     // CSS
39.     ICSSResolver cssResolver = new StyleAttrCSSResolver();
40.     var cssFile = XMLWorkerHelper.GetCSS(new FileStream(CSS, FileMode.Open));
41.     cssResolver.AddCss(cssFile);
42.
43.     // HTML and Apply PDF font
44.     CssAppliers cssAppliers = new CssAppliersImpl(new PdfFontProvider());
45.     HtmlPipelineContext htmlContext = new HtmlPipelineContext(cssAppliers);
46.     htmlContext.SetTagFactory(Tags.GetHtmlTagProcessorFactory());
47.     htmlContext.SetImageProvider(new PdfImageProvider());
48.
49.     // Pipelines
50.     PdfWriterPipeline pdf = new PdfWriterPipeline(document, writer);
51.     HtmlPipeline html = new HtmlPipeline(htmlContext, pdf);
52.     CssResolverPipeline css = new CssResolverPipeline(cssResolver, html);
53.
54.     // XML Worker

```

```

55. XMLWorker worker = new XMLWorker(css, true);
56. XMLParser p = new XMLParser(worker);
57. p.Parse(new MemoryStream(invoice));
58.
59. PdfDictionary parameters = new PdfDictionary();
60. parameters.Put(PdfName.MODDATE, new PdfDate());
61. PdfFileSpecification fileSpec = writer.AddFileAttachment("Tax Invoice XML Data", xml, null
1, EtداAttachmentFilename, "text/xml", AFRelationshipValue.Alternative, parameters);
62. PdfArray array = new PdfArray();
63. array.Add(fileSpec.Reference);
64. writer.ExtraCatalog.Put(PdfName.AF, array);
65.
66. document.Close();
67. }

```

## Appendix G

### PDF/A-3 Font Provider

```

1. public class PdfFontProvider : IFontProvider
2. {
3.     protected internal BaseFont regular;
4.     protected internal BaseFont bold;
5.
6.     public PdfFontProvider()
7.     {
8.         regular = BaseFont.CreateFont(@"fonts\Loma.ttf", BaseFont.IDENTITY_H, true);
9.         bold = BaseFont.CreateFont(@"fonts\LomaBold.ttf", BaseFont.IDENTITY_H, true);
10.    }
11.
12.    public virtual bool IsRegistered(string fontname)
13.    {
14.        return true;
15.    }
16.
17.    public virtual Font GetFont(string fontname, string encoding, bool embedded, float size
, int style, BaseColor color)
18.    {
19.        Font font;
20.        switch (style)
21.        {
22.            case Font.BOLD:
23.                font = new Font(bold, size-2);
24.                break;
25.            default:
26.                font = new Font(regular, size-2);
27.                break;
28.        }
29.        font.Color = color;
30.        return font;
31.    }
32. }

```

## Appendix H

### ThaiBahtText Class

```
1. public class ThaiBahtText
2. {
3.     public string BahtText(decimal number)
4.     {
5.         return number.ThaiBahtText();
6.     }
7.
8.     public string CitySubNameText(string id)
9.     {
10.        Dictionary<string, string> d = new Dictionary<string, string>()
11.        {
12.            {"710705", "ชะแล"},
13.            {"860303", "สะพลี"}
14.        };
15.        string value;
16.        d.TryGetValue(id, out value);
17.        return (string.IsNullOrEmpty(value)) ? " " : value;
18.    }
19.
20.    public string CityNameText(string id)
21.    {
22.        Dictionary<string, string> d = new Dictionary<string, string>()
23.        {
24.            {"7107", "ทองผาภูมิ"},
25.            {"8603", "ปะทิว"}
26.        };
27.        string value;
28.        d.TryGetValue(id, out value);
29.        return (string.IsNullOrEmpty(value)) ? " " : value;
30.    }
31.
32.
33.    public string CountrySubIDText(string id)
34.    {
35.
36.        Dictionary<string, string> d = new Dictionary<string, string>()
37.        {
38.            {"71", "กาญจนบุรี"},
39.            {"86", "ชุมพร"}
40.        };
41.        string value;
42.        d.TryGetValue(id, out value);
43.        return (string.IsNullOrEmpty(value)) ? " " : value ;
44.    }
45. }
```



## Appendix I

### PDF/A-3 Convertor source code

```
1. class PDFa3Invoice
2. {
3.
4.     public void CreatePDFa3Invoice(string pdfFilePath, string xmlFilePath, string xmlFileName,
5.         string xmlVersion, string documentID, string documentOID, string outputPath, string documentType)
6.     {
7.         // ===== Create PDF/A-3U Document =====
8.         // Create PDF/A-3U writer instance from existing document
9.         PdfReader reader = new PdfReader(pdfFilePath);
10.        MemoryStream stream = new MemoryStream();
11.        Document pdfAdocument = new Document();
12.        PdfAWriter writer = this.CreatePDFaInstance(pdfAdocument, reader, stream);
13.
14.        // Create Output Intents
15.        ICC_Profile icc = ICC_Profile.GetInstance("Resources/sRGB Color Space Profile.icm");
16.        ;
17.        writer.SetOutputIntents("sRGB IEC61966-2.1", "", "http://www.color.org", "sRGB IEC61966-2.1", icc);
18.
19.        PdfArray array = new PdfArray();
20.        writer.ExtraCatalog.Put(new PdfName("AF"), array);
21.
22.        //===== Create Exchange Invoice =====
23.        //add xml to document
24.        PdfFileSpecification contentSpec = this.EmbeddedAttachment(xmlFilePath, xmlFileName,
25.            "text/xml", new PdfName("Alternative"), writer, "Tax Invoice XML Data");
26.        array.Add(contentSpec.Reference);
27.
28.        //add Electronic Document XMP Metadata
29.        string stringExchangeXMP = File.ReadAllText("Resources/EDocument_PDFaExtensionSchema.xml");
30.        byte[] exchangeXMP = Encoding.ASCII.GetBytes(stringExchangeXMP.Replace("@DocumentType", documentType));
31.        writer.XmpMetadata = exchangeXMP;
32.
33.        pdfAdocument.Close();
34.        reader.Close();
35.
36.        File.WriteAllBytes(outputPath, stream.ToArray());
37.    }
38.
39.    public PdfAWriter CreatePDFaInstance(Document targetDocument, PdfReader originalDocument, Stream os)
40.    {
41.        PdfAWriter writer = PdfAWriter.GetInstance(targetDocument, os, PdfAConformanceLevel.PDF_A_3U);
42.        writer.CreateXmpMetadata();
43.
44.        if (!targetDocument.IsOpen())
```

```

45.         targetDocument.Open();
46.
47.         PdfContentByte cb = writer.DirectContent; // Holds the PDF data
48.         PdfImportedPage page;
49.         int pageCount = originalDocument.NumberOfPages;
50.         for (int i = 0; i < pageCount; i++)
51.         {
52.             targetDocument.NewPage();
53.             page = writer.GetImportedPage(originalDocument, i + 1);
54.             cb.AddTemplate(page, 0, 0);
55.         }
56.         return writer;
57.     }
58.     public PdfFileSpecification EmbeddedAttachment(string filePath, string fileName, string mimeType,
59.         PdfName afRelationship, PdfAWriter writer, string description)
60.     {
61.         PdfDictionary parameters = new PdfDictionary();
62.         parameters.Put(PdfName.MODDATE, new PdfDate(File.GetLastWriteTime(filePath)));
63.         PdfFileSpecification fileSpec = PdfFileSpecification.FileEmbedded(writer, filePath, fileName, null, mimeType,
64.             parameters, 0);
65.         fileSpec.Put(new PdfName("AFRelationship"), afRelationship);
66.         writer.AddFileAttachment(description, fileSpec);
67.         return fileSpec;
68.     }
69. }

```

## Appendix J

ตัวอย่าง source code สำหรับการตรวจสอบโครงสร้างของ XML ด้วย Schema

```

1.     class main
2.     {
3.         public static void Main()
4.         {
5.
6.             // Create the XmlSchemaSet class.
7.             XmlSchemaSet sc = new XmlSchemaSet();
8.
9.             // Add the schema to the collection.
10.            sc.Add("urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2", "TaxInvoice_CrossIndustryInvoice_2p0.xsd");
11.
12.            // Set the validation settings.
13.            XmlReaderSettings settings = new XmlReaderSettings();
14.            settings.ValidationType = ValidationType.Schema;
15.            settings.Schemas = sc;
16.            settings.ValidationEventHandler += new ValidationEventHandler(ValidationCallBack);
17.        };
18.
19.        // Create the XmlReader object.
20.        XmlReader reader = XmlReader.Create("ETDA-invoice.xml", settings);

```

```

21.         // Parse the file.
22.         while (reader.Read()) ;
23.
24.     }
25.
26.     // Display any validation errors.
27.     private static void ValidationCallback(object sender, ValidationEventArgs e)
28.     {
29.         Console.WriteLine("Validation Error: {0}", e.Message);
30.     }
31. }

```

## Appendix K

ตัวอย่าง Schematron สำหรับใบกำกับภาษีอิเล็กทรอนิกส์

```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <!-- edited by Office Of Standard,ETDA -->
3. <!-- ===== -->
4. <!-- ===== ETDA TaxInvoice_ CrossIndustryInvoice Schema ===== -->
5. <!-- ===== -->
6. <!--
7. Schema agency: ETDA
8. Schema version: 1.0
9. Schema date: Aug 2015
10.
11. © Copyright ETDA, 2015
12.
13. This XML Schema Definition file was developed as a subset profile of the UNCEFACT CrossIndu
    stryInvoice Schema , version 9.1.
14.
15. -->
16.
17. <!-- ===== -->
18. <!-- ===== CrossIndustryInvoice Schema Module ===== -->
19. <!-- ===== -->
20. <!--
21. Schema agency: UNCEFACT
22. Schema version: 9.1
23. Schema date: 15 Nov 2014
24.
25. Copyright (C) UNCEFACT (2014). All Rights Reserved.
26.
27. This document and translations of it may be copied and furnished to others, and derivative
    works that comment on or otherwise explain it or assist in its implementation may be prepar
    ed, copied, published and distributed, in whole or in part, without restriction of any kind
    , provided that the above copyright notice and this paragraph are included on all such copi
    es and derivative works. However, this document itself may not be modified in any way, such
    as by removing the copyright notice or references to UNCEFACT, except as needed for the pur
    pose of developing UNCEFACT specifications, in which case the procedures for copyrights def
    ined in the UNCEFACT Intellectual Property Rights document must be followed, or as required
    to translate it into languages other than English.
28.

```

```

29. The limited permissions granted above are perpetual and will not be revoked by UNCEFACT or
    its successors or assigns.
30.
31. This document and the information contained herein is provided on an "AS IS" basis and UNCEFACT
    DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY
    THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES
    OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
32. -->
33. <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
34.   xmlns:ccts="urn:un:unece:uncefact:documentation:standard:CoreComponentsTechnicalSpecification:2"
35.   xmlns:udt="urn:un:unece:uncefact:data:standard:UnqualifiedDataType:16"
36.   xmlns:qdt="urn:etda:uncefact:data:standard:QualifiedDataType:1"
37.   xmlns:ram="urn:etda:uncefact:data:standard:TaxInvoice_ReusableAggregateBusinessInformationEntity:2"
38.   targetNamespace="urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2"
39.   elementFormDefault="qualified" attributeFormDefault="unqualified" version="2.0"
40.   xmlns:rsm="urn:etda:uncefact:data:standard:TaxInvoice_CrossIndustryInvoice:2"
41.   xmlns:ns1="urn:etda:uncefact:data:standard:TaxInvoice_ReusableAggregateBusinessInformationEntity:2"
42.   xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
43.   <!-- ===== -->
44.   <!-- ===== Imports ===== -->
45.   <!-- ===== -->
46.   <!-- ===== -->
47.   <!-- ===== Import of Unqualified Data Type Schema Module ===== -->
48.   <!-- ===== -->
49.   <xsd:import namespace="urn:un:unece:uncefact:data:standard:UnqualifiedDataType:16"
50.     schemaLocation="../../../../uncefact/data/standard/UnqualifiedDataType_16p0.xsd"/>
51.   <!-- ===== -->
52.   <!-- ===== Import of Qualified Data Type Schema Module ===== -->
53.   <!-- ===== -->
54.   <xsd:import namespace="urn:etda:uncefact:data:standard:QualifiedDataType:1"
55.     schemaLocation="QualifiedDataType_1p0.xsd"/>
56.   <!-- ===== -->
57.   <!--
58.   - ===== Import of TaxInvoice_Reusable Aggregate Business Information Entity Schema Module ===== -->
59.   <xsd:import
60.     namespace="urn:etda:uncefact:data:standard:TaxInvoice_ReusableAggregateBusinessInformationEntity:2"
61.     schemaLocation="TaxInvoice_ReusableAggregateBusinessInformationEntity_2p0.xsd"/>
62.   <!-- ===== -->
63.   <!-- ===== Element Declarations ===== -->
64.   <!-- ===== -->
65.   <!-- ===== Root Element Declarations ===== -->
66.   <!-- ===== -->
67.   <!-- ===== -->
68.   <!-- ===== Type Definitions ===== -->
69.   <!-- ===== -->
70.   <!-- ===== Type Definition: CrossIndustryInvoiceType ===== -->
71.   <!-- ===== -->
72.   <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
73.     schemaLocation="xmldsig-core-schema.xsd"/>

```

```

74. <xsd:element name="TaxInvoice_CrossIndustryInvoice" type="rsm:TaxInvoice_CrossIndustryInvoiceType">
75.   <xsd:annotation>
76.     <xsd:documentation xml:lang="en">
77.       <ccts:UniqueID>CII</ccts:UniqueID>
78.       <ccts:Acronym>RSM</ccts:Acronym>
79.       <ccts:Name>CrossIndustryInvoice</ccts:Name>
80.       <ccts:Version>1.0</ccts:Version>
81.       <ccts:Definition>A message used as a request for payment, or modification of a request for payment, for the supply of goods or services ordered, delivered, received, consumed.
82.       <ccts:BusinessProcessContextValue>Invoicing</ccts:BusinessProcessContextValue>
83.       <ccts:GeopoliticalOrRegionContextValue>In All Contexts</ccts:GeopoliticalOrRegionContextValue>
84.       <ccts:OfficialConstraintContextValue>None</ccts:OfficialConstraintContextValue>
85.       <ccts:ProductContextValue>In All Contexts</ccts:ProductContextValue>
86.       <ccts:IndustryContextValue>In All Contexts</ccts:IndustryContextValue>
87.       <ccts:BusinessProcessRoleContextValue>In All Contexts</ccts:BusinessProcessRoleContextValue>
88.       <ccts:SupportingRoleContextValue>In All Contexts</ccts:SupportingRoleContextValue>
89.       <ccts:SystemCapabilitiesContextValue>In All Contexts</ccts:SystemCapabilitiesContextValue>
90.     </xsd:documentation>
91.   </xsd:annotation>
92. </xsd:element>
93. <xsd:complexType name="TaxInvoice_CrossIndustryInvoiceType">
94.   <xsd:annotation>
95.     <xsd:documentation xml:lang="en">
96.       <ccts:UniqueID>CII-2</ccts:UniqueID>
97.       <ccts:Acronym>MA</ccts:Acronym>
98.       <ccts:DictionaryEntryName>Cross Industry_ Invoice. Details</ccts:DictionaryEntryName>
99.       <ccts:Version>1.0</ccts:Version>
100.      <ccts:Definition>A message used as a request for payment, or modification of a request for payment, for the supply of goods or services ordered, delivered, received, consumed.
101.      <ccts:ObjectClassTerm>Invoice</ccts:ObjectClassTerm>
102.      <ccts:ObjectClassQualifierTerm>Cross Industry</ccts:ObjectClassQualifierTerm>
103.      <ccts:BusinessProcessContextValue>Invoicing</ccts:BusinessProcessContextValue>
104.      <ccts:GeopoliticalOrRegionContextValue>In All Contexts</ccts:GeopoliticalOrRegionContextValue>
105.      <ccts:OfficialConstraintContextValue>None</ccts:OfficialConstraintContextValue>
106.      <ccts:ProductContextValue>In All Contexts</ccts:ProductContextValue>
107.      <ccts:IndustryContextValue>In All Contexts</ccts:IndustryContextValue>
108.      <ccts:BusinessProcessRoleContextValue>In All Contexts</ccts:BusinessProcessRoleContextValue>
109.      <ccts:SupportingRoleContextValue>In All Contexts</ccts:SupportingRoleContextValue>
110.      <ccts:SystemCapabilitiesContextValue>In All Contexts</ccts:SystemCapabilitiesContextValue>
111.    </xsd:documentation>
112.   </xsd:annotation>
113.   <xsd:sequence>
114.     <xsd:element name="ExchangedDocumentContext" type="ram:ExchangedDocumentContextType">
115.       <xsd:annotation>

```

```

116.         <xsd:documentation xml:lang="en">
117.             <ccts:UniqueID>CII01</ccts:UniqueID>
118.             <ccts:Acronym>ASMA</ccts:Acronym>
119.             <ccts:DictionaryEntryName>Cross Industry_ Invoice. CI_ Exchanged Document_ Conte
xt</ccts:DictionaryEntryName>
120.             <ccts:Version>1.0</ccts:Version>
121.             <ccts:Definition>The scenario or setting of a Cross Industry Invoice (CII) excha
nged document, such as its business process application context.</ccts:Definition>
122.             <ccts:Cardinality>1</ccts:Cardinality>
123.             <ccts:ObjectClassTerm>Invoice</ccts:ObjectClassTerm>
124.             <ccts:ObjectClassQualifierTerm>Cross Industry</ccts:ObjectClassQualifierTerm>
125.             <ccts:AssociationType>composition</ccts:AssociationType>
126.             <ccts:PropertyTerm/>
127.             <ccts:AssociatedObjectClassTerm>Context</ccts:AssociatedObjectClassTerm>
128.             <ccts:AssociatedObjectClassQualifierTerm>CI</ccts:AssociatedObjectClassQualifier
Term>
129.             <ccts:AssociatedObjectClassQualifierTerm>Exchanged Document</ccts:AssociatedObje
ctClassQualifierTerm>
130.             <ccts:BusinessProcessContextValue>Invoicing</ccts:BusinessProcessContextValue>
131.             <ccts:GeopoliticalOrRegionContextValue>In All Contexts</ccts:GeopoliticalOrRegio
nContextValue>
132.             <ccts:OfficialConstraintContextValue>None</ccts:OfficialConstraintContextValue>
133.             <ccts:ProductContextValue>In All Contexts</ccts:ProductContextValue>
134.             <ccts:IndustryContextValue>In All Contexts</ccts:IndustryContextValue>
135.             <ccts:BusinessProcessRoleContextValue>In All Contexts</ccts:BusinessProcessRoleC
ontextValue>
136.             <ccts:SupportingRoleContextValue>In All Contexts</ccts:SupportingRoleContextValu
e>
137.             <ccts:SystemCapabilitiesContextValue>In All Contexts</ccts:SystemCapabilitiesCon
textValue>
138.         </xsd:documentation>
139.         </xsd:annotation>
140.     </xsd:element>
141.     <xsd:element name="ExchangedDocument" type="ram:ExchangedDocumentType">
142.         <xsd:annotation>
143.             <xsd:documentation xml:lang="en">
144.                 <ccts:UniqueID>CII02</ccts:UniqueID>
145.                 <ccts:Acronym>ASMA</ccts:Acronym>
146.                 <ccts:DictionaryEntryName>Cross Industry_ Invoice. Exchanged_ Document</ccts:Dict
ionaryEntryName>
147.                 <ccts:Version>1.0</ccts:Version>
148.                 <ccts:Definition>A collection of data for a Cross Industry Invoice Header that
is exchanged between two or more parties in written, printed or electronic form.</ccts:Defini
tion>
149.                 <ccts:Cardinality>1</ccts:Cardinality>
150.                 <ccts:ObjectClassTerm>Invoice</ccts:ObjectClassTerm>
151.                 <ccts:ObjectClassQualifierTerm>Cross Industry</ccts:ObjectClassQualifierTerm>
152.                 <ccts:AssociationType>composition</ccts:AssociationType>
153.                 <ccts:PropertyTerm/>
154.                 <ccts:AssociatedObjectClassTerm>Document</ccts:AssociatedObjectClassTerm>
155.                 <ccts:AssociatedObjectClassQualifierTerm>Header</ccts:AssociatedObjectClassQuali
fierTerm>
156.                 <ccts:AssociatedObjectClassQualifierTerm>Exchanged</ccts:AssociatedObjectClassQu
alifierTerm>

```

```

157.         <ccts:BusinessProcessContextValue>Invoicing</ccts:BusinessProcessContextValue>
158.         <ccts:GeopoliticalOrRegionContextValue>In All Contexts</ccts:GeopoliticalOrRegionContextValue>
159.         <ccts:OfficialConstraintContextValue>None</ccts:OfficialConstraintContextValue>
160.         <ccts:ProductContextValue>In All Contexts</ccts:ProductContextValue>
161.         <ccts:IndustryContextValue>In All Contexts</ccts:IndustryContextValue>
162.         <ccts:BusinessProcessRoleContextValue>In All Contexts</ccts:BusinessProcessRoleContextValue>
163.         <ccts:SupportingRoleContextValue>In All Contexts</ccts:SupportingRoleContextValue>
164.         <ccts:SystemCapabilitiesContextValue>In All Contexts</ccts:SystemCapabilitiesContextValue>
165.     </xsd:documentation>
166.     </xsd:annotation>
167. </xsd:element>
168. <xsd:element name="SupplyChainTradeTransaction" type="ns1:SupplyChainTradeTransactionType">
169.     <xsd:annotation>
170.         <xsd:documentation xml:lang="en">
171.             <ccts:UniqueID>CII03</ccts:UniqueID>
172.             <ccts:Acronym>ASMA</ccts:Acronym>
173.             <ccts:DictionaryEntryName>Cross Industry_ Invoice. Supply Chain_ Trade Transaction</ccts:DictionaryEntryName>
174.             <ccts:Version>1.0</ccts:Version>
175.             <ccts:Definition>The trade line items, trade agreement, trade delivery and trade settlement details included in this Cross Industry Invoice Header (Header).</ccts:Definition>
176.             <ccts:Cardinality>1</ccts:Cardinality>
177.             <ccts:ObjectClassTerm>Invoice</ccts:ObjectClassTerm>
178.             <ccts:ObjectClassQualifierTerm>Cross Industry</ccts:ObjectClassQualifierTerm>
179.             <ccts:AssociationType>composition</ccts:AssociationType>
180.             <ccts:PropertyTerm>
181.                 <ccts:AssociatedObjectClassTerm>Trade Transaction</ccts:AssociatedObjectClassTerm>
182.                 <ccts:AssociatedObjectClassQualifierTerm>Header</ccts:AssociatedObjectClassQualifierTerm>
183.                 <ccts:AssociatedObjectClassQualifierTerm>Supply Chain</ccts:AssociatedObjectClassQualifierTerm>
184.                 <ccts:BusinessProcessContextValue>Invoicing</ccts:BusinessProcessContextValue>
185.                 <ccts:GeopoliticalOrRegionContextValue>In All Contexts</ccts:GeopoliticalOrRegionContextValue>
186.                 <ccts:OfficialConstraintContextValue>None</ccts:OfficialConstraintContextValue>
187.                 <ccts:ProductContextValue>In All Contexts</ccts:ProductContextValue>
188.                 <ccts:IndustryContextValue>In All Contexts</ccts:IndustryContextValue>
189.                 <ccts:BusinessProcessRoleContextValue>In All Contexts</ccts:BusinessProcessRoleContextValue>
190.                 <ccts:SupportingRoleContextValue>In All Contexts</ccts:SupportingRoleContextValue>
191.                 <ccts:SystemCapabilitiesContextValue>In All Contexts</ccts:SystemCapabilitiesContextValue>
192.             </xsd:documentation>
193.             </xsd:annotation>
194.         </xsd:element>

```

```

195.     <xsd:element minOccurs="0" ref="ds:Signature"/>
196.   </xsd:sequence>
197. </xsd:complexType>
198. </xsd:schema>

```

## Appendix L

ตัวอย่าง ไฟล์ build ของการตรวจสอบข้อกำหนดด้วย Apache Ant

```

1. <?xml version="1.0" encoding="UTF-8"?>
2. <project name="schematron-ant-test" default="schematron-validation">
3.
4.   <!-- ISO namespace -->
5.   <target name="schematron-validation" description="Schematron Validation Example">
6.     <taskdef name="schematron" classname="com.schematron.ant.SchematronTask"
7.       classpath="lib/ant-schematron.jar"/>
8.     <schematron schema="schemas/TaxInvoice_Schematron_2p0.sch"
9.       file="xml/ETDA-invoice.xml"/>
10.    </target>
11. </project>

```

## Appendix M

ตัวอย่างไฟล์ CSS สำหรับใบกำกับภาษีอิเล็กทรอนิกส์

```

1. #header { color: #2371ab; font-size: 18pt; font-weight: bold; }
2. #date { font-size: 16pt; }
3. #addresses { margin-top: 20pt; font-size: 11pt; }
4. #products { margin-top: 10pt; border: 2px solid #2371ab; }
5. #totals { margin-top: 10pt; border: 2px solid #2371ab; }
6. #wireinfo { margin-top: 10pt; margin-left: 72pt; }
7. .name { font-weight: bold; color: #2371ab; }
8. .total { font-weight: bold; color: #2371ab; }
9. .headerrow { line-height: 150%; background-color: #2371ab; color: #FFFFFF; }
10. .bold { font-weight: bold; }
11. .wireheader { font-weight: bold; text-align: left; }
12. th { padding: 2pt; font-weight: bold; text-align: center; }
13. td { padding: 2pt; }

```



## 7. เอกสารอ้างอิง

**Adobe System Incorporated. 2008.** PDF 32000-1:2008. [Online] 7 1, 2008.  
[http://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/PDF32000\\_2008.pdf](http://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/PDF32000_2008.pdf).

**Apache Ant. 2017.** Apache Ant. [Online] The Apache Software Foundation, 2017.  
[ant.apache.org/](http://ant.apache.org/).

**iText, Rups. 2017.** iText RUPS. [Online] 2017. <https://github.com/itext/rups>.

**Schematron. 2017.** Schematron Git Hub. [Online] 2017.  
<https://github.com/Schematron/schematron>.

**กรมสรรพากร. 2014.** บทบัญญัติแห่งประมวลรัษฎากร. [Online] July 29, 2014.  
<http://www.rd.go.th/publish/5208.0.html>.