# 8 자산의 패키지 파일 형식 관리 셸(AASX)

## 8.1 일반

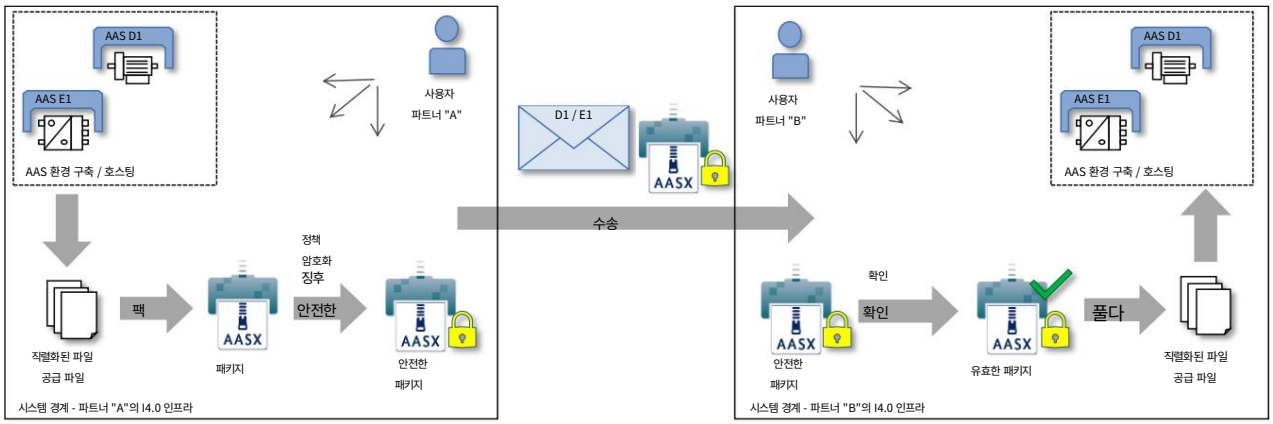
일부 사용 사례에서는 자산 관리 셸의 전체 또는 부분 구조를 교환해야 합니다.
연관된 값의 유무에 관계없이 및/또는 정보를 영구적으로 만듭니다(예: 파일 서버에 저장). 이는 이 정보를 보유하고 저장할 수 있는 파일 형식을 정의해야 함을 의미합니다. 따라서 AASX(자산 관리 셸)의 패키지 파일 형식은 다음 요구 사항을 기반으로 정의됩니다.

- Asset Administration Shell 구조, 데이터 및 기타 관련 항목을 포함하는 일반 패키지 파일 형식
  파일
- 주요 사용 사례는 조직/파트너 간의 교환 및
  자산 관리 셸의 정보입니다.
- 법적 제한 및 로열티가 없습니다. 바람직하게는 해당 형식의 향후 유지 관리 가능성이 높은 국제 표준을 기반으로 합니다.

- 이 형식을 만들고 읽고 쓸 수 있는 API의 존재
- 디지털 서명 및 암호화 기능이 제공되어야 합니다.
- 패키지 파일의 신뢰성 및 통합을 위한 정책36

그림 74의 다음 프로세스는 AASX 패키지를 만들고 사용하기 위해 정의됩니다.

그림 74 AASX 패키지 생성 및 사용 프로세스




프로세스는 기존 AAS(예: D1 및 E1)를 파일로 직렬화하고(이 문서에 설명된 직렬화 메커니즘에 따라) 다른 추가 파일(매뉴얼과 같은 AAS 구조에서 언급된 파일)을 내보내는 것으로 시작됩니다. , CAD 파일 등). 이러한 모든 파일은 AASX ZIP 파일 형식으로 함께 패키징되며 AASX 내부의 파일 수정, 암호화 및 디지털 서명에 대한 정책을 정의하는 몇 가지 보안 단계가 따릅니다. 그런 다음 최종 AASX는 전자 메일, USB 스틱 등과 같은 디지털 미디어를 통해 AASX 생산자(이 경우 파트너 A)에서 AASX 소비자(파트너 B)로 전송될 수 있습니다. 소비자는 먼저 유효성을 검사하고 확인해야 합니다. 들어오는 AASX, 포함된 파일의 압축을 푼 다음 이를 가져와 소비자 환경에서 새 AAS를 생성합니다. 이 프로세스는 다음 하위 섹션에서 자세히 설명합니다.

## 8.2 오픈 패키징 협약의 기본 개념

Open Packaging Convention에서 지정한 패키징 모델은 패키지, 부품 및 관계를 설명합니다. 패키지는 파일과 같은 콘텐츠와 리소스를 포함하는 부분을 포함합니다 37. 패키지의 모든 파일

---

36 이 패키지에 액세스하기 위한 역할 기반 정책은 정의되어 있지 않습니다.
AAS(섹션 7 참조)

37 "파트" 대신 "파일"이라는 용어가 사용됩니다.

MIME 미디어 유형의 형식으로 표현된 지정된 콘텐츠 유형과 함께 고유한 URI 호환 파일 이름이 있습니다.

패키지를 파일에 연결하고 패키지에 있는 다양한 파일을 연결하기 위해 관계가 정의됩니다. 관계의 정의(파일 이름과 함께)는 패키지 의 논리적 모델 입니다. 관계의 소스가 되는 리소스는 패키지 자체 또는 패키지 내부의 데이터 구성 요소(파일)여야 합니다. 관계의 대상 리소스는 패키지 내부 또는 외부의 URI 주소 지정 가능한 리소스일 수 있습니다. 동일한 대상 파일을 공유하는 둘 이상의 관계가 있을 수 있습니다(ISO/IEC 29500-2: 2012의 예 9–6 참조).

물리적 모델 은 이러한 논리적 개념을 물리적 형식에 매핑합니다. 이 매핑의 결과는 파일이 디렉토리와 같은 계층 구조로 나타나는 물리적 패키지 형식(ZIP 아카이브 형식)입니다([27] 및 [28]에서 채택).

## 8.3 자산 관리 셸 패키지 파일 형식에 대한 규칙 (AASX)

AASX(Asset Administration Shell Package) 형식은 Open Package Conventions 표준에서 파생되어 결과적으로 그 특성을 계승합니다. 그럼에도 불구하고 AASX에 대해 몇 가지 규칙을 정의해야 합니다.

- ISO/IEC 29500-2:2012에 따른 패키지 형식 및 규칙. 이 표준의 파생 형식(예: AASX 형식)에는 논리적 모델, 물리적 모델 및 보안 모델의 정의가 필요합니다. 이러한 특정 규칙은 다음 하위 섹션에 설명되어 있습니다.
- AASX 형식의 파일 확장자: .aasx
- AASX 형식의 MIME 유형: application/asset-administration-shell-package38
- AASX의 아이콘 .
- AASX 형식은 MIME(파일 확장명 및 콘텐츠) 유형으로 식별할 수 있습니다. 콘텐츠 측면에서, 첫 번째 관계 파일 /_rels/.rels(Open Packaging Conventions에 정의됨)를 읽고 관계 유형 http://admin-shell.io/aasx/relationships/aasx 를 찾을 때 식별할 수 있습니다. origin (자산 관리 셸의 논리적 모델에 대한 진입점).
- 패키지의 다음 경로와 파일 이름은 이미 Open Packaging Conventions 사양에 따라 예약되어 있으므로 파생 형식에 사용할 수 없습니다.

  /[콘텐츠 유형].xml; //rels/.rels; /<file_path>/_rels/<filename> .rels (여기서 <filename> 은 관계의 소스인 패키지의 파일이고 <file_path> 는 해당 파일의 경로입니다.)
- 기존 Office Open XML/Open Packaging Conventions 호환 가능한 사무실 응용 프로그램(예: Microsoft Office, LibreOffice)에서 AASX 형식을 반드시 열 필요는 없습니다. 다른 사무실 "모델"에 필요한 관계 및 파일이 없을 수 있기 때문입니다(예: http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument "docx" 문서).

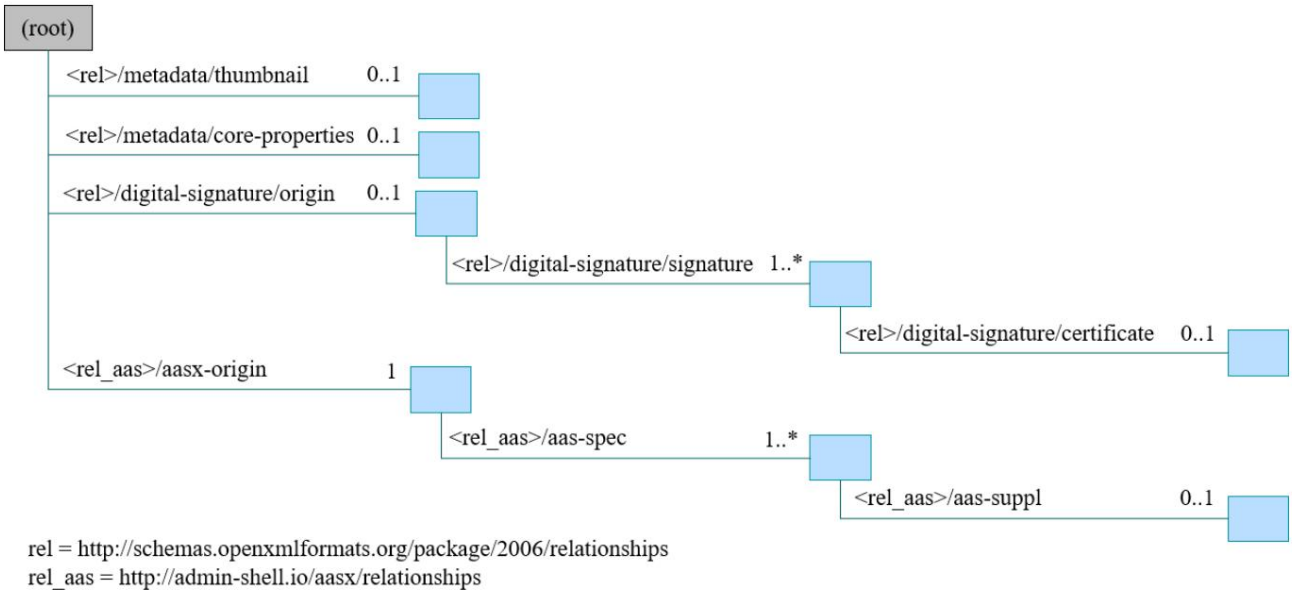
## 8.4 ECMA-376 관계

앞서 언급했듯이 개방형 패키징 규칙 위에 형식에 대한 논리적 모델을 정의하는 것이 필요합니다. 그림 75는 관계 유형(URI) 세트와 해당 소스 파일을 AASX 형식에 대한 논리 모델의 일부로 정의합니다. 또한(그림 75에는 표시되지 않음) 특정 관계 인스턴스에는 고유 ID와 대상 리소스(패키지 내부 또는 외부에 있는 대상 파일의 URI)도 있습니다.

---

38 현재 MIME 유형은 잠정적이며 공식적으로 요청해야 합니다.

그림 75 AASX 패키지의 관계 유형




썸네일, 핵심 속성, 디지털 서명(원본, 서명 및 인증서)에 대한 관계 유형은 Open Packaging Conventions에 의해 정의되므로 재창조할 필요가 없습니다. 다른 관계 유형은 AASX 패키지 형식을 지원하기 위해 특별히 정의되었습니다. 다음 은 그림 75의 각 관계 유형39에 대한 간단한 설명입니다 .

- 썸네일 – 선택 사항입니다. 해당 패키지의 썸네일을 정의하는 데 필요합니다(예: 관리되는 장치의 사진). 확장자 및/또는 콘텐츠 유형에 따라 패키지 아이콘 대신 축소판 그림이 표시될 수 있습니다.

- core-properties – 선택 사항입니다. 일부 오픈 패키징 규약 고유의 요소 외에 선택된 더블린 코어 메타데이터 요소를 사용하는 "핵심 속성"을 통해 패키지를 설명하는 스키마가 있습니다. 핵심 속성은 관리 셸을 설명하지 않고 패키지 자체를 설명합니다. 핵심 속성의 일부 요소는 관리 셸의 요소와 유사/동일할 수 있습니다. 일부 핵심 속성은 Title, Subject, Creator, Keywords, Description, LastModifiedBy, Revision, LastPrinted, Created, Modified, Category, Identifier, ContentType, Language, Version, ContentStatus입니다.

- 디지털 서명/원본, 디지털 서명/서명 및 디지털 서명/인증서 – 선택 사항.
  패키지 내의 파일 및 관계에 서명해야 하는 경우 필요합니다. 그들의 관계는 기본적으로 서명에 대한 데이터(예: 인증서, 다이제스트 등)를 포함하는 파일을 대상으로 합니다. 디지털 서명에 대한 이 문서 뒷부분의 설명을 참조하십시오.

- aasx-origin – 필수. 이 관계는 "의도적으로 비어 있음"이라는 텍스트를 포함하는 빈 파일 또는 일반 텍스트 파일인 aasx-origin 파일을 대상으로 합니다. 패키지 내부의 특정 관계 및 파일에 대한 진입점입니다. aasx-origin 관계의 소스는 패키지 루트여야 합니다.

- aas-spec – 필수: 이 문서에 정의된 XML 또는 JSON 형식에 따라 하나 이상의 식별 가능한 요소(예: AAS, Submodel 또는 ConceptDescription)의 구조/사양을 포함하는 파일("aasenv")을 대상으로 합니다. aasx-spec 관계의 소스는 aasx 원본 파일이어야 합니다.

- aas-suppl – 선택 사항입니다. File 요소를 통해 AAS의 데이터 내에서 참조되는(BLOB으로 저장되지 않은) 추가 파일을 대상으로 합니다(5.7.7.8절 참조). 모든 aasx-suppl 관계의 소스는 AAS 구조/사양을 포함하는 파일이어야 합니다.

---

39 관계 유형의 긴 이름을 피하기 위해 텍스트를 따라 짧은 이름을 사용합니다.

참고: 하위 모델 사양 내의 모든 파일 요소가 동일한 AASX 패키지에 저장된 파일을 대상으로 하는 것은 아닙니다. 상대 URI 참조(절대 경로 또는 상대 경로 참조)만 AASX 패키지 내의 추가 파일에 대한 참조로 해석됩니다.

## 8.5 파일 이름 규칙

ECMA-375 관계(8.4)를 사용하면 파일 이름과 독립적으로 AASX 패키지 내에서 파일을 찾을 수 있습니다. 예를 들어 한 패키지 생산자는 /aasx/device.xml에 aas-spec 파일을 저장하고 다른 하나는 /asset-admin-shell/productX123.xml에 저장할 수 있지만 둘 다 동일한 관계 유형을 사용하여 해당 파일을 대상으로 합니다. 보다 일관된 접근 방식을 위해 AASX 패키지 내에서 파일 이름 지정에 대해 다음 규칙이 정의됩니다.

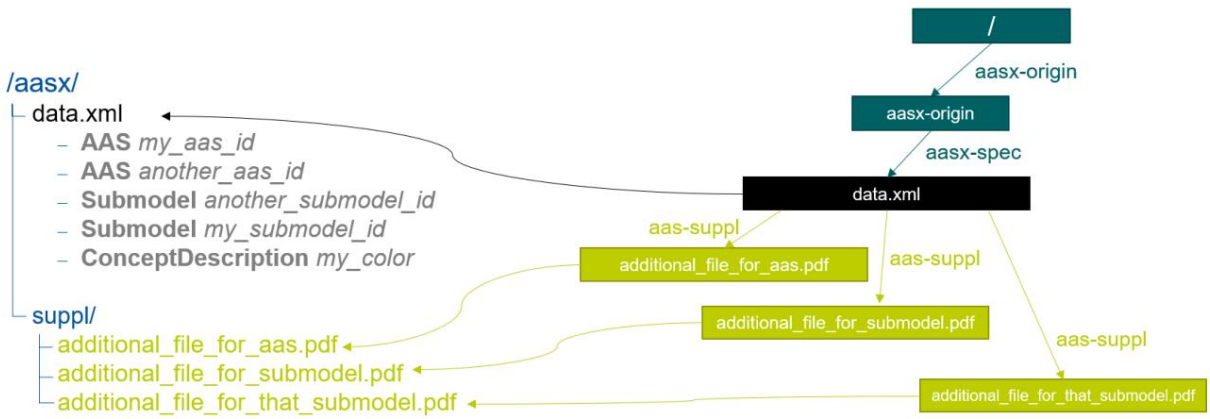
- /aasx/ 는 AASX 패키지 특정 정보를 포함하는 모든 파일의 공통 접두사입니다.
- /aasx/aasx-origin 은 내용이 없는(빈 파일) asx-origin 관계의 대상이 됩니다.
- /aasx/data.<extension> 은 aas-spec 관계의 대상이 됩니다. 여기서 <extension> 은 직렬화 유형에 따라 "xml" 또는 "json"입니다.

- 동일한 AASX 패키지에 저장된 두 직렬화 형식(xml, json) 모두에서 동일한 데이터의 직렬화가 가능합니다. 이 경우 앞서 언급한 확장과 적절한 ECMA-376 콘텐츠 유형(MIME 유형)을 사용하여 다른 직렬화 형식을 병렬로 저장할 수 있습니다.

  이 경우 이 두 파일에 대해 보조 파일을 대상으로 하는 적절한 as-suppl 관계를 만들어야 합니다.

AASX 패키지의 예는 그림 76에 나와 있습니다. 이는 그림 75에 정의된 ECMA-376 관계 유형을 사용하고 위에 정의된 파일 이름 규칙을 따라 트리 보기에 나열된 AASX 패키지의 내용을 보여줍니다. 이 예에서는 AAS 사양 파일이 XML로 직렬화되어 있다고 가정합니다.

그림 76 AASX 패키지 콘텐츠의 예 - 트리 보기(왼쪽) 및 ECMA-376 관계 유형(오른쪽)




AASX 특정 파일 외에도 모든 ECMA-376 패키지에 공통적인 파일(예: 관계 부분(*.rels) 및 콘텐츠 유형 스트림([Content_Types].xml))은 물리적 표현으로 AASX 패키지에 포함되어야 합니다. .zip 아카이브로. 이러한 파일에 대한 자세한 내용은 ECMA-376 사양을 참조하십시오.

## 8.6 디지털 서명

디지털 서명 기능은 이미 Open Packaging Conventions 사양[27]에 의해 제공됩니다. 따라서 이 패키지 서명 프레임워크는 AASX 패키지에도 사용할 수 있습니다. AAS 데이터의 무결성을 보장하기 위해 패키지 내의 모든 관련 파일(aasx-origin 파일, AAS 구조 사양 파일, 추가 파일) 및 관련 관계 부분에 서명해야 합니다.

## 8.7 암호화

Open Packaging Conventions 사양(ISO/IEC 29500-2:2012)은 “ZIP 기반 패키지는 ZIP 사양에 설명된 암호화를 포함하지 않아야 합니다. 패키지 구현자는 이 제한[M3.9]”40을 시행해야 합니다. 그러나 Open Packaging Conventions 패키지는 다른 수단으로 암호화될 수 있으며 이 패키지 형식을 보다 구체적인 형식의 기초로 사용하는 일부 응용 프로그램은 교환 시 암호화를 사용하거나 배포를 위해 DRM을 사용할 수 있습니다[24].

한 예로 파생 오피스 형식에서 사용하는 MS-OFFCRYPTO(Office 문서 암호화 구조)가 있습니다. 사용된 일부 기술은 Microsoft의 특허가 적용될 수 있으므로 AASX 형식에는 권장되지 않습니다. DRM(디지털 권한 관리)은 사용자에게 권한을 부여하기 위해 부여된 특정 액세스 권한으로 패키지의 콘텐츠 요소를 암호화하는 데에도 사용할 수 있습니다(system.io.packaging 네임스페이스의 구현 참조[31]).

암호화 및 기밀 유지와 관련하여 다음 규칙을 따라야 합니다.

1. 패키지에 기밀 콘텐츠를 포함할 필요가 있는지 결정합니다. 이유가 없으면 기밀 내용을 포함하지 않아야 합니다.

2. 일시적인 통신 행위(예: 이메일 교환 등)에 암호화가 필요한 경우 또는 AASX를 동일한 엔티티가 나중에 열 수 있도록 어딘가에 저장해야 하는 경우 해당 특정에 암호화 방법을 사용할 수 있습니다. 의미(예: BitLocker를 지원하는 Windows 기반 시스템에 AASX를 저장할 때 BitLocker 사용, 엔터티 간에 암호화된 전자 메일을 교환하기 위해 S/MIME 사용 등).

3. AASX 의 일부 또는 전체 콘텐츠에 암호화가 필요한 기타 모든 사용 사례41의 경우:

   - "암호화된" 버전이 패키지의 원본 파일을 대체하는 즉시 AASX 패키지의 개별 파일에 암호화 방법을 사용할 수 있으며 암호화 형식의 콘텐츠 유형이 알려지고 콘텐츠 유형이 [ 콘텐츠 유형].xml. 이 문서에 정의된 관계는 콘텐츠가 암호화되었는지 여부에 관계없이 동일하게 유지됩니다. Open Packaging Conventions 관련 파일 및 관계 파일은 암호화되지 않으며 암호화 후 디지털 서명을 수행해야 합니다. 암호화 표준의 한 예는 암호화된 콘텐츠가 RFC 5652에 정의된 application/pkcs7-mime 형식으로 저장되고 파일 확장자 *.p7m을 사용해야 하는 Secure MIME(S/MIME)입니다.

   - 패키지의 내용(개별 파일)을 암호화하는 것 외에도 전체 패키지를 암호화할 수 있습니다(예: Secure MIME을 사용하고 암호화된 패키지를 application/pkcs7-mime 파일 형식으로 저장). 이 경우 암호화 전에 패키지 내용에 대한 서명이 이루어져야 합니다.

---

40 그 이유는 패키지 형식에 대한 투명성 요구 사항 및 PKWARE의 라이선스 요구 사항과 관련이 있을 수 있습니다. ISO/IEC 21320-1(Document Container File: Core)의 경우 "개별 파일 및 중앙 디렉토리의 암호화는 금지됩니다. 따라서 ZIP_PK의 이 프로필은 상위 형식보다 더 투명합니다.” [30]

41 사용 사례는 하위 모델을 암호화하고 요금을 지불한 후에만 암호화되지 않은 데이터에 대한 액세스를 제공하는 것일 수 있습니다.