Korea Communications Standard

방송통신표준

KCS.OT-07.0001/R1 개정일: 2013 년 12 월 31 일

지상파 데이터 방송

Standard for Terrestrial Data Broadcasting

미래창조과학부

국립전파연구원

방송통신표준
KCS.OT-07.0001/R1 개정일: 2013 년 12 월 31 일

지상파 데이터 방송

Standard for Terrestrial Data Broadcasting

미래창조과학부
국립전파연구원

서 문

1. 표준의 목적

본 표준은 국내 지상파 방송 매체를 통한 데이터 방송 서비스를 제공하기 위해 필요한 전송 시스템 및 수신 단말에 적용되는 전반적인 기술 규격을 정의하는 것을 목적으로 한다.

2. 주요 내용 요약

데이터 방송 서비스는 프로그램 관련 데이터 서비스, 독립 데이터 서비스, 양방향 서비스 등을 제공할 수 있어야 한다. 본 표준에는 이러한 데이터 서비스를 제공하기 위한 지상파 데이터 방송 전송 시스템 및 수신 단말에 적용되는 애플리케이션 규격, 전송 및 시그널링 규격 등의 제반 사항을 기술한다.

3. 표준 적용 산업 분야 및 산업에 미치는 영향

본 표준은 국내 데이터 방송 서비스를 위한 체계를 구축할 경우 발생할 수 있는 혼란을 최소화할 수 있으며 이는 디지털 방송 기술 개발 관련 응용 서비스 활성화에 기여할 것이다. 또한 양방향 데이터 방송 서비스를 통한 TV 전자상거래를 통해 전자상거래 시장을 자연스럽게 활성화시켜 나갈 것이다.

4. 참조 표준(권고)

4.1. 국제 표준(권고)

- ATSC, A/101A, 'ATSC Standard: Advanced Common Application Platform (ACAP)', 2009. 2. 12.

4.2. 국내 표준

- TTAK.OT-07.0111/R4, '지상파 데이터방송', 2009.11.

5. 참조 표준(권고)과의 비교

5.1. 참조 표준(권고)과의 관련성

본 표준은 'TTAK.KO-07.0111/R4' '지상파 데이터방송' 표준을 따르고 있으며, 'TTAK.KO-07.0111/R4' 표준은 국내 지상파 데이터 방송 서비스를 위하여 ATSC 표준 'A/101A' 표준을 참조하였다.

5.2. 참조한 표준(권고)과 본 표준의 비교표

본 표준은 'TTAK.OT-07.0111/R4' 표준을 국가 표준으로 개정 제안한 사항으로 TTA 표준이 ATSC 'A/101A'를 기준으로 국내 방송 현실에 적합하도록 추가 및 수정 보완되어 'ATSC A/101A'와의 비교 사항을 아래와 같이 기재한다.

KCS.OT-07.0001/R1	ATSC A/101A	비고
1. 개요	-	추가
2. 표준의 구성 및 범의	-	추가
3. 용어 정의 및 약어	-	추가
4. 일반 요구 사항	-	추가
5. 구조	5. Basic Architecture	동일(참조)
6. 전송 프로토콜	6. Transport Protocols	확장
7. 콘텐츠 형식	7. Content Formats	'7.3 방송 스트림 형식' 확장
8. ACAP-X ~ 10. 애플리케이션 시그널링	8. ACAP-X ~ 10. Application Signaling	동일(참조)
11. ACAP-J Platform ~ 12. Security	11. ACAP-J Platform ~ 12. Security	확장
13. 그래픽 참조 모델 ~ 14. 시스템 통합	13. Graphics Reference Model ~ 14. System Integration	동일(참조)
15. 최소 수신기 요구사항 ~ 16. 상세한 플랫폼 프로파일 정의	15. Minimum Receiver Requirements ~ 16. Detailed Platform Profile Definitions	확장
17. 정합 ~ 18. 모니터 애플리케이션	17. Conformance ~ 18. Monitor Application	동일(참조)
부속서 A. Content Identification API ~ 부속서 B. Document Type Definitions	Annex A Content Identification API ~ Annex B Document Type Definitions	동일(참조)
부속서 C. Data Piping 프로토콜	-	추가
부속서 D. 수신기의 표준 준수 시험을 위한 인터페이스	Annex C Test Support	동일
부속서 E. OCAP 1.0 Storage API ~ 부속서 J. 지상파 데이터 방송 표준 API 리스트	-	추가

6. 지식 재산권 관련 사항

본 표준의 '지적 재산권 확약서' 제출 현황은 국립전파연구원 웹사이트에서 확인할 수 있다.

※ 본 표준을 이용하는 자는 이용함에 있어 지식 재산권이 포함되어 있을 수 있으므로, 확인 후 이용한다.

※ 본 표준과 관련하여 접수된 확약서 이외에도 지식 재산권이 존재할 수 있다.

7. 시험 인증 관련 사항

7.1. 시험 인증 대상 여부

해당 사항 없음.

7.2. 시험 표준 제정 현황

해당 사항 없음.

8. 표준의 이력 정보

8.1. 표준의 이력

판수	제정·개정일	제정·개정 내역
제 1 판	2007.02.01.	제정 KCS.OT-07.0001
제 2 판	2013.12.31.	개정 KCS.OT-07.0001/R1

8.2. 주요 개정 사항

본 표준 개정판의 주요 개정사항은 아래와 같다.

- 참조 표준 개정 반영(ATSC A/101→ATSC A/101A)
- 코드 사인 검증 인증체계 추가(12.8.6절, 부속서 I)

- OCAP 1.0.0 Persistent Storage API 관련 사항 추가
- Java 플랫폼 참조 표준 변경(JAE 1.1.8 → PBP, CDC 1.1)
- ACAP Test API 추가(부속서 D)
- Version 및 System Property 정의 추가(11.4.2절, 11.4.3절)

Preface

1. Purpose of Standard

This standard aims to define overall technical specifications that are applied to transmission systems and receiver terminals for providing data broadcasting services through national terrestrial broadcasting media.

2. Summary of Contents

Data broadcasting service should provide a program related data broadcasting service, Non program related data broadcasting service, bi-directional data broadcasting service with interaction channel, etc. Herein, to support there services above, this standard describes application, transmission, and signaling specifications for developing transmission system and receiver terminals for terrestrial data broadcasting.

3. Applicable Fields of Industry and its Effect

The present standard can resolve issues that may arise as operation when a data broadcasting service is launched. Also, T-commerce, using interaction channel in data broadcasting service, will be deployed, actively as a case of data broadcasting service.

4. Reference Standards(Recommendations)

4.1. International Standards

- ACAP, A/101A, ATSC Advanced Common Application Platform (ACAP), 2009. 2. 12.

4.2. Domestic Standards

- TTAK.OT-07.0111/R4, ‘Standard for Terrestrial Data Broadcasting’, 2009.11.

5. Comparison between Reference Standards(Recommendations) and this Standard

5.1. Relevance of this standard with Reference Standards(Recommendations)

This standard refers to the recommended references and international standards listed in 4.1 for the standardization of data service for satellite digital multimedia broadcasting.

5.2. A Comparative Table of Reference Standard(Recommendation) and this Standard

This standard refers to TTAK.OT-07.0111/R4 which refers A/101A ATSC standard for national terrestrial data broadcasting service.

KCS.OT-07.0001/R1	ATSC A/101A	Remarks
1. Introduction	-	Added
2. Constitution and Scope	-	Added
3. Terms Definition and Abbreviations	-	Added
4. General Considerations	-	Added
5. Architecture	5. Basic Architecture	Referred
6. Transport Protocols	6. Transport Protocols	Extended
7. Content Formats	7. Content Formats	“7.3 Broadcast Stream Format” Extended
8. ACAP-X ~ 10. Application Signalling	8. ACAP-X ~ 10. Application Signalling	Referred
11. ACAP-J Platform ~ 12. Security	11. ACAP-J Platform ~ 12. Security	Extended
13. Graphics Reference Model ~ 14. System Integration	13. Graphics Reference Model ~ 14. System Integration	Referred
15. Minimum Receiver Requirements ~ 16. Detailed Platform Profile Definitions	15. Minimum Receiver Requirements ~ 16. Detailed Platform Profile Definitions	Extended
17. Conformance ~ 18. Monitor Application	17. Conformance ~ 18. Monitor Application	Referred
Annex A Content Identification API ~ Annex B Document Type Definitions	Annex A Content Identification API ~ Annex B Document Type Definitions	Referred
Annex C Data Piping Protocol	-	Added
Annex D Interface for Conformance Testing of a Receiver	Annex C Test Support	Referred
Annex E OCAP 1.0 Storage API ~ Annex J Standard API List for Terrestrial Data Broadcasting	-	Added

6. Statement of Intellectual Property Rights

"Written Confirmation of Intellectual Property Rights" for this standard can be referenced to the website of the National Radio Research Agency.

Those using this standard must confirm that whether intellectual property rights are included in this standard.

Other intellectual property rights may exist in relation to written confirmation received for this standard.

7. Statement of Testing and Certification

7.1. Object of Testing and Certification

None

7.2. Standards of Testing and Certification

None

8. History of Standard

8.1. Change History

Edition	Issued date	Outline
The 1st edition	2007.02.01.	Established KCS.OT-07.0001
The 2nd edition	2013.12.31.	Revised KCS.OT-07.0001/R1

8.2. Revision Related Details

The revision details of this standard since the 1st edition are;

- Reflection of the reference standard revision(ATSC A/101→ATSC A/101A)

- Addition of the code siging authentication framework(12.8.6, Annex I)
- Addition of the OCAP 1.0.0 Persistent Storage API and related items
- Modification of the reference java platform(JAE 1.1.8 → PBP, CDC 1.1)
- Addtion of the ACAP Test API(Annex D)
- Addition of the version and system properties definition(11.4.2, 11.4.3)

목 차

Contents

지상파 데이터 방송
(Standard for Terrestrial Data Broadcasting)

1. 개요

본 표준은 국내 지상파 방송 매체를 통한 디지털 데이터 방송 서비스를 제공하기 위한 국내 지상파 데이터 방송 표준이다.

2. 표준의 구성 및 범위

데이터 방송 서비스는 프로그램 관련 데이터 서비스, 독립 데이터 서비스, 양방향 서비스 등을 제공할 수 있어야 한다. 본 표준에는 이러한 데이터 서비스를 제공하기 위한 지상파 데이터 방송 전송 시스템 및 수신 단말에 적용되는 애플리케이션 규격, 전송 및 시그널링 규격 등의 제반 사항을 기술한다.

3. 용어 정의 및 약어

용어나 약어는 ATSC 표준 'A/101A' ACAP 문서에 서술되어 있는 것을 제외한 국내 데이터 방송에서 사용되는 내용을 풀이하는 것으로 한다.

3.1. 용어 정의

SEED	128 비트 블록 암호 알고리즘 표준(KISA/TTA/IETF RFC4009 참조)

3.2. 약어

ACAP	Advanced Common Application Platform
DNS	Domain Name Service
GEM	Globally Executable MHP
HTTP	Hyper-Text Transfer Protocol
JSSE	Java Secure Socket Extension

KISA	Korea Information Security Agency(한국정보보호진흥원)
MD5	Message Digest 5
PRF	Permission Request File
RFC	Request for Comments
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
SSL	Secure Socket Layer
TLS	Transport Layer Security
UCS	The Universal Character Set
UTF-8	UCS Transformation Format-8
URI	Uniform Resource Identifier

4. 일반 요구 사항

4.1. 개방형 국제 표준 정립

현재 진행되고 있는 관련 국제 규격의 표준화 동향을 파악하고 필수적이고 공통적인 부분을 우선 표준화에 반영하여야 한다.

기술의 발전에 따라 부가되는 우수한 기술을 차기 표준화에 반영하는 프로파일 개념으로 표준화를 진행하여야 한다.

현재 시장에서 널리 활용되고 있거나 향후 표준으로 예상되는 기술이 우선적으로 고려되어야 한다.

4.2. 상호 운용성 보장

네트워크 및 하드웨어 플랫폼에 독립적인 표준이어야 한다.

4.3. 인터넷과의 연계

다양한 인터넷 콘텐츠를 활용하여 데이터 방송의 콘텐츠를 제작 및 보급 가능하여야 한다.

4.4. 향후 기술 발전의 수용성 및 확장성

급속한 기술 발전에 따른 사용자의 요구 사항에 효율적으로 대응하여야 한다.

장래에 요구되는 기능을 쉽게 수용할 수 있도록 확장이 용이해야 하며, 확장성(scalability) 및 역방향 호환성(backward compatibility)을 보장하여야 한다.

4.5. 저렴한 가격의 가입자 단말 장치 보급 가능성

국민 편의 제고를 위해 기존의 가입자 단말 장치에 하드웨어 및 소프트웨어의 부담이 적은 기술이어야 한다.

4.6. 콘텐츠의 다용도 허용

기존에 개발된 서비스 및 콘텐츠의 재활용성을 높이기 위해 기존의 규격을 용이하게 수용할 수 있어야 한다.

4.7. 데이터 방송 서비스 프로파일 정의

4.7.1. 배경

기술 발전 및 관련 국제 규격의 변화에 따라 현재 구현 가능한 기술부터 단계적으로 표준화에 도입하여 국내 데이터 방송의 조기 활성화 및 국제적인 기술 경쟁력의 확보를 위하여 프로파일 개념을 도입한다.

프로파일 내의 표준 기술 적용 시의 고려 사항은 다음과 같다.

- 각 프로파일에 적용되는 표준 기술은 용도에 따른 데이터 표현 및 전송 방법, 애플리케이션 프로그램 실행 엔진 등을 포함하여 필요 기술의 전체를 기술하거나 관련 국제 규격을 규정하여 적용 시의 혼란을 최소화하여야 한다.
- 각 프로파일에 적용되는 표준 기술은 현재의 국제 표준 단체들이 정의하는 관련 규격과 전송 및 콘텐츠 사용 면에서 호환성을 가져야 한다.

4.7.2. 프로파일

본 표준에서는 국내 디지털 방송의 데이터 방송을 위한 프로파일을 정의한다. 프로파일 1 은 현재 관련 국제 규격에서 의무적으로 채택하고 있는 기술이고, 프로파일 2 는 선택적으로 채택하고 있는 기술이며, 그 이상은 향후 기술 발전에 따라 추후 정의한다.

- 프로파일 1 : 국제 규격에서 의무적으로 정의하고 있는 기술 항목
- 프로파일 2 : 국제 규격에서 선택적으로 정의하고 있는 기술 항목

상위 프로파일을 정의하는 경우에 상위 프로파일은 하위 프로파일의 서비스 및 기술 요소를 포함하여야 한다.

표 4.1 디지털 데이터 방송 프로파일

디지털 데이터 방송 프로파일	적용 기술 규격
프로파일 1	ACAP-J
프로파일 2	ACAP-J & ACAP-X

4.8. 텍스트 인코딩

데이터 방송('Transport protocol' 포함)에서 사용하는 텍스트의 인코딩 방식은 UTF-8 을 사용한다.

5. 구조

5.1. ACAP-J 애플리케이션 지원 환경

ACAP 문서 'A/101A'의 '5.1 Support for ACAP-J Applications'을 따라야 한다.

5.2. ACAP-X 애플리케이션 지원 환경

ACAP 문서 'A/101A'의 '5.2 Support for ACAP-X Applications'을 따라야 한다.

6. 전송 프로토콜

6.1. 소개

ACAP 문서 'A/101A'의 '6.1 Introduction'을 따라야 한다.

6.2. 방송 채널 프로토콜

6.2.1. Object Carousel Protocol

6.2.1.1. Message Template

ACAP 문서 'A/101A'의 '6.2.1.1 Message Template'을 따라야 한다.

6.2.1.2. Service Gateway Message

ACAP 문서 'A/101A'의 '6.2.1.2 Service Gateway Message'를 따라야 한다.

6.2.1.3. Directory Message

ACAP 문서 'A/101A'의 '6.2.1.3 Directory Message'를 따라야 한다.

6.2.1.4. Message Descriptors

ACAP 문서 'A/101A'의 '6.2.1.4 Message Descriptors'를 따라야 한다.

6.2.1.5. File Message

ACAP 문서 'A/101A'의 '6.2.1.5 File Message'를 따라야 한다.

6.2.1.6. Stream Message

ACAP 문서 'A/101A'의 '6.2.1.6 Stream Message'를 따라야 한다.

6.2.1.7. Stream Event Message

ACAP 문서 'A/101A'의 '6.2.1.7 Stream Event Message'를 따라야 한다.

6.2.2. Data Carousel Protocol

6.2.2.1. The Message Template

ACAP 문서 'A/101A'의 '6.2.2.1 The Message Template'을 따라야 한다.

6.2.2.2. Download Info Indication Message

ACAP 문서 'A/101A'의 '6.2.2.2 Download Info Indication Message'를 따라야 한다.

6.2.2.3. Download Server Initiate Message

ACAP 문서 'A/101A'의 '6.2.2.3 Download Server Initiate Message'를 따라야 한다.

6.3. 리턴 채널 프로토콜

6.3.1. Network Specific Protocols

ACAP 문서 'A/101A'의 '6.3.1 Network Specific Protocols'을 따라야 한다.

6.3.2. Internet Protocol

ACAP 문서 'A/101A'의 '6.3.2 Internet Protocol'을 따라야 한다.

6.3.3. User Datagram Protocol(UDP)

ACAP 문서 'A/101A'의 '6.3.3 User Datagram Protocol(UDP)'을 따라야 한다.

6.3.4. Transmission Control Protocol(TCP)

ACAP 문서 'A/101A'의 '6.3.4 Transmission Control Protocol(TCP)'을 따라야 한다.

6.3.5. Hyper-Text Transfer Protocol(HTTP)

ACAP 문서 'A/101A'의 '6.3.5 Hyper-Text Transfer Protocol(HTTP)'을 따라야 한다.

6.3.6. Domain Name Service(DNS)

ACAP 문서 'A/101A'의 '6.3.6 Domain Name Service(DNS)'를 따라야 한다.

6.3.7. 국내 적용 사항

6.3.7.1. 리턴 채널 네트워크 프로토콜

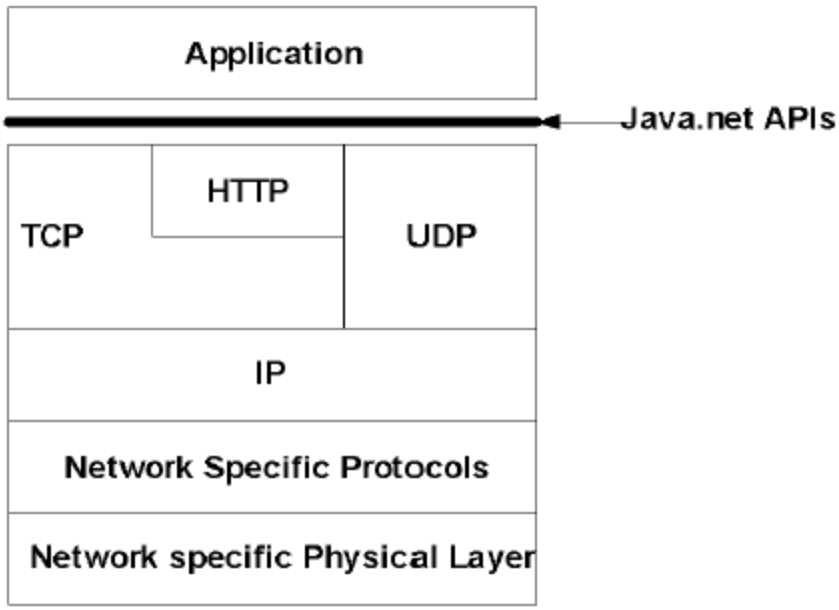

그림 6.1 리턴 채널 네트워크 프로토콜

위 그림 6.1 은 리턴 채널 네트워크에 해당하는 프로토콜을 나타내고 있으며, 그림에서 표시된 프로토콜을 지원하여야 한다. 또한, 추가적으로 6.3.7.2 절에 정의한 내용을 따라야 한다.

6.3.7.2. 리턴 채널 지원 프로토콜

6.3.7.2.1. Physical and Data-Link Layer

Physical and Data-Link Layer 를 위한 프로토콜로 Ethernet 프로토콜 10base-T('IEEE 802.3i') 또는 100base-T('IEEE 802.3u')를 권고한다.

이 외의 프로토콜은 필요시 지원할 수 있으며, 본 표준에서는 정의하지 않음을 원칙으로 한다.

6.3.7.2.2. Network Layer

ICMP(Internet Control Message Protocol, 'RFC 792')를 지원하여야 한다.

6.3.7.2.3. Application Protocol Layer

HTTP 1.1(HyperText Transfer Protocol, 'RFC 2616')을 지원하여야 한다.

사용자가 수동 및 자동으로 수신기(STB)의 IP 주소를 설정하는 방안을 제공하여야 하며, 자동 설정을 위한 프로토콜인 DHCP 를 지원하여야 한다.

7. 콘텐츠 형식

7.1. 일반사항

ACAP 문서 'A/101A'의 '7.1 General'을 따라야 한다.

7.2. 정적 형식

ACAP 문서 'A/101A'의 '7.2 Static Formats'을 따라야 한다.

7.3. 방송 스트림 형식

비디오, 오디오, 폐쇄 자막 등의 방송 스트림 형식은 지상파 디지털 TV 방송 송수신 정합 표준 문서 'TTAK.KO-07.0014/R2'를 따른다.

7.4. 내장 폰트[1]

폰트 종류	폰트 이름 또는 내용
기본 폰트	Korea iTV SansSerifD
추가 폰트(선택사항)	Korea iTV SerifD, Korea iTV SansSerifB

위의 정의된 내장 폰트 서체에 대하여 TTF 타입 기준의 폰트를 이용한다. UTF 타입의 경우 TTF 타입 기준으로 동일한 정보를 제공하는 경우 이용가능하다.

7.5. 다운로더블 폰트(선택 사항)

다운로더블 폰트의 운용 방식은 'ETSI TS 101 812 V1.3.1' MHP1.0.3 표준의 '7.4 Downlaodable Font'를 따르며 지원은 선택 사항으로 한다.

8. ACAP-X

8.1. 정의

8.1.1. 애플리케이션 정의

ACAP 문서 'A/101A'의 '8.1.1 Application Behavior'을 따라야 한다.

8.1.2. 자원 지정 스키마

ACAP 문서 'A/101A'의 '8.1.2 Resource Identifier Schemes'을 따라야 한다.

8.1.3. 이벤트 처리 방식

ACAP 문서 'A/101A'의 '8.1.3 Event Processing'을 따라야 한다.

8.1.4. Trigger 처리 방식

ACAP 문서 'A/101A'의 '8.1.4 Trigger processing'을 따라야 한다.

---

[1] 내장 폰트 사용과 관련된 문의 사항은 TTA 사무국을 통해 확인할 수 있다.

8.2. 기타 사항

8.2.1. Application Metadata Content

ACAP 문서 'A/101A'의 '8.2.1 Application Metadata Content'을 따라야 한다.

8.2.2. Graphics Content

ACAP 문서 'A/101A'의 '8.2.2 Graphics Content'을 따라야 한다.

8.2.3. Non-Streaming Video Content

ACAP 문서 'A/101A'의 '8.2.3 Non-Streaming Video Content'을 따라야 한다.

8.2.4. Non-Streaming Audio Content

ACAP 문서 'A/101A'의 '8.2.4 Non-Streaming Audio Content'을 따라야 한다.

8.2.5. Streaming Video Content

ACAP 문서 'A/101A'의 '8.2.5 Streaming Video Content'을 따라야 한다.

8.2.6. Streaming Audio Content

ACAP 문서 'A/101A'의 '8.2.6 Streaming Audio Content'를 따라야 한다.

8.2.7. Font Content

ACAP 문서 'A/101A'의 '8.2.7 Font Content'를 따라야 한다.

8.2.8. Archive Content

ACAP 문서 'A/101A'의 '8.2.8 Archive Content'를 따라야 한다.

8.2.9. Markup Content

ACAP 문서 'A/101A'의 '8.2.9 Markup Content'를 따라야 한다.

8.2.10. Stylesheet Content

ACAP1.0 문서 'A/101A'의 '8.2.10 Stylesheet Content'를 따라야 한다.

8.2.11. Script Content

ACAP 문서 'A/101A'의 '8.2.11 Script Content'를 따라야 한다.

8.3. ACAP-X 보안

8.3.1. 쿠키 접근

ACAP 문서 'A/101A'의 '8.3.1 Cookie Access'를 따라야 한다.

8.3.2. Inter-Environment Bridge Access

ACAP 문서 'A/101A'의 '8.3.2 Inter-Environment Bridge Access'를 따라야 한다.

8.3.3. Runtime Code Extension Access

ACAP 문서 'A/101A'의 '8.3.3 Runtime Code Extension Access'를 따라야 한다.

8.4. ACAP-X 전송(ACAP-X Transport Specifics)

ACAP 문서 'A/101A'의 '8.4 ACAP-X Transport Specifics'을 따라야 한다.

8.4.1. ACAP-X 전송 바인딩(ACAP-X Transport Binding)

ACAP 문서 'A/101A'의 '8.4.1 ACAP-X Transport binding'을 따라야 한다.

9. 애플리케이션 모델

9.1. 방송 ACAP 애플리케이션

ACAP 문서 'A/101A'의 '9.1 Broadcast ACAP Application'을 따라야 한다.

9.2. ACAP-J Model

ACAP 문서 'A/101A'의 '9.2 ACAP-J Model'을 따라야 한다.

9.3. ACAP-X Model

ACAP 문서 'A/101A'의 '9.3 ACAP-X Model'을 따라야 한다.

9.4. 애플리케이션 간 리소스 관리

ACAP 문서 'A/101A'의 '9.4 Inter-application Resource Management'를 따라야 한다.

10. 애플리케이션 시그널링

10.1. 소개

ACAP 문서 'A/101A'의 '10.1 Introduction'을 따라야 한다.

10.2. Program Map Table

ACAP 문서 'A/101A'의 '10.2 Program Map Table'을 따라야 한다.

10.3. Application Information Table

ACAP 문서 'A/101A'의 '10.3 Application Information Table'을 따라야 한다.

10.4. Application Specific Descriptor Sequence

ACAP 문서 'A/101A'의 '10.4 Application Specific Descriptor Sequence'를 따라야 한다.

10.5. Application Representation Specific Descriptor Sequences

ACAP 문서 'A/101A'의 '10.5 Application Representation Specific Descriptor Sequences'를 따라야 한다.

11. ACAP-J 플랫폼

ACAP 문서 'A/101A'의 '11 ACAP-J Platform'를 본 표준 11.4 절 국내 적용 사항을 적용하여 따라야 한다.

11.1. Java Content

ACAP 문서 'A/101A'의 '11.1 Java Content'를 본 표준 11.4 절 국내 적용 사항을 적용하여 따라야 한다.

11.2. 폰트 색인 콘텐츠

ACAP 문서 'A/101A'의 '11.2 Font Index Content'를 따라야 한다.

11.3. 아카이브 콘텐츠

ACAP 문서 'A/101A'의 '11.3 Archive Content'를 따라야 한다.

11.4. 국내 적용 사항

국내 지상파 데이터 방송 규격의 Java 플랫폼은 'JSR-218' CDC1.1/'JSR-217' PBP 1.1 을 따르며, 'JSR-927' Java TV 1.1 를 따른다. 그러나, 'JSR-218' CDC 1.1/'JSR-217' PBP 1.1/'JSR-927' Java TV 1.1 에 정의되어 있는 모든 패키지, 클래스 그리고 메소드들을 수용하지는 않는다. 본 표준 부 속 서 J 와 참조 표준인 ACAP 'A/101A'에서

정의하고 있는 ACAP-J Platform 에 준하는 패키지, 클래스 및 메소드들만 수용하며 또한, 본 표준과 참조 표준인 ACAP ‘A/101A’에서 정의하고 있는 ACAP-J Platform 의 제약사항을 따라야 한다.

11.4.1. OCAP 1.0.0 반영 사항

본 표준의 기능 향상을 위해 OCAP 1.0.0(‘OC-SP-OCAP1.0.0-070814’)의 아래의 내용을 반영한다.

11.4.1.1. OcapLocator

OCAP 1.0.0(‘OC-SP-OCAP1.0.0-070814’)의 ‘16.2.1.1 OCAP 1.0 Locators’를 따른다.

11.4.1.2. Persistent Storage API

OCAP 1.0.0(‘OC-SP-OCAP1.0.0-070814’)의 ‘13.3.7.5 Persistent Storage API’를 따른다.

11.4.2. Version Properties

수신기에서는 아래 표 11.1 과 같이 정의된 수신기가 지원하는 데이터 방송 표준 버전에 대한 시스템 속성값들을 탑재하여야 한다.

표 11.1 Version properties

속성	설명	형식	값	애플리케이션 접근
acap.kr.version	수신기가 지원하는 데이터 방송 표준 버전	String(‘x.x.x’), major, minor, micro version을 ‘.’로 연결한 문자열	‘1.0.4’	unsigned and signed
acap.kr.version.major	수신기가 지원하는 미들웨어 major 버전	String(‘x’), 숫자를 나타내는 문자열	‘1’	unsigned and signed
acap.kr.version.minor	수신기가 지원하는 미들웨어 minor 버전	String(‘x’), 숫자를 나타내는 문자열	‘0’	unsigned and signed
acap.kr.version.micro	수신기가 지원하는 미들웨어 micro 버전	String(‘x’), 숫자를 나타내는 문자열	‘4’	unsigned and signed

버전 속성값의 형식은 숫자를 나타내는 문자열로서 숫자 10 진수(예, ‘1.0.4’, ‘1’, ‘0’, ‘4’)로 표현한다.

11.4.3. System Properties

본 표준에서는 아래 표 11.2 과 같이 시스템 속성값들을 정의하며 수신기에서는 아래 속성값들을 탑재하고 있어야 하며 정확한 값으로 설정되어 있어야 한다.

hardware.terminal_id 는 다른 수신기와 차별된 값을 지정하기 위해 수신기의 MAC address 를 사용하며 형태는 ‘:’이 포함된 스트링이며 숫자는 대문자의 16 진수로 길이가 17 인 스트링(예, ‘12:34:56:78:9A:BC’)으로 한다.

표 11.2 시스템 프로퍼티

속성	설명	형식	애플리케이션 접근
tcommerce.enable	t-commerce 애플리케이션 지원 여부	String(‘true’/’false’)	unsigned and signed
storageapi.enable	Org.ocap.storage API 지원 여부, 즉, USB를 통한 외부 저장장치의 지원 여부	String(‘true’/’false’)	unsigned and signed
hardware.terminal_id	수신기 ID	String(‘XX:XX:XX:XX:XX:XX’)	unsigned and signed

12. 보안

12.1. 소개

ACAP 문서 ‘A/101A’의 ‘12.1 Introduction’을 따라야 한다.

12.2. ACAP 신뢰 모델

ACAP 문서 ‘A/101A’의 ‘12.2 ACAP Trust Model’을 따라야 한다.

12.2.1. 일반 규칙

ACAP 문서 ‘A/101A’의 ‘12.2.1 General Rules’을 따라야 한다.

12.2.2. 지상파를 통해 수신된 애플리케이션

ACAP 문서 'A/101A'의 '12.2.2 Applications Received Over a Terrestrial Interface'를 따라야 한다.

12.2.3. 케이블 인터페이스를 통해 수신된 애플리케이션

ACAP 문서 'A/101A'의 '12.2.3 Applications Received Over a Cable Interface'를 따라야 한다.

12.3. 애플리케이션 보안 정책

ACAP 문서 'A/101A'의 '12.3 Security Policy for Applications'을 따라야 한다.

12.4. GEM 보안 모델의 ACAP 확장

ACAP 문서 'A/101A'의 '12.4 ACAP Extensions to GEM Security Model'을 따라야 한다.

12.4.1. ACAP Signing Framework

ACAP 문서 'A/101A'의 '12.4.1 ACAP Signing Framework'를 따라야 한다.

12.4.2. ACAP Extensions to Security Policies for Applications

ACAP 문서 'A/101A'의 '12.4.2 ACAP Extensions to Security Policies for Applications'을 따라야 한다.

12.5. 리턴 채널 보안

ACAP 문서 'A/101A'의 '12.5 Security over the Interaction Channel'을 따라야 한다.

12.6. 플랫폼 최소 요구사항

ACAP 문서 'A/101A'의 '12.6 Platform Minima'를 따라야 한다.

12.7. ACAP 보안 운용 모델

ACAP 문서 'A/101A'의 '12.7 ACAP Security Operational Model'을 따라야 한다.

12.8. 국내 적용 사항

12.8.1. Certificates and Certificate Revocation Lists

ACAP 문서 'A/101A'의 '12 Security'를 아래 사항을 적용하여 따라야 한다.

- 인증서 버전 : X.509 인증서 버전 3
- 전자서명을 위한 cryptographic algorithms : RSA with SHA-1, RSA with MD5 지원
- Key Agreement Algorithms : DH(Diffie-Hellman) 알고리즘 지원

12.8.2. Secure Channel Protocols

12.8.2.1. TLS 1.0 지원

TLS Cipher Suites 는 'ETSI TS 101 812 V1.3.1' DVB MHP 1.0.3 의 'Table 60 : Profile of cipher suites that implementations are required to support'를 기본으로 지원한다. 단, 암호 알고리즘은 한국정보보호진흥원에서 규정하고 있는 128 비트 SEED 블록 암호 알고리즘을 지원한다.

SUN 의 JSSE 상에서 SEED 를 사용하기 위해서는 OID 등을 자체적으로 결정하고 SEED 에 대한 provider 를 만들어서 추가해야 한다.

SEED 알고리즘을 지원하는 cipher suite 는 다음과 같이 정의한다.

- TLS_RSA_WITH_SEED_CBC_SHA = {0x00, 0x96}

리턴 채널 보안과 관련하여 서버 인증이 기본이며 클라이언트 인증은 선택 사항이다.

12.8.2.2. 루트 인증서

서버 인증을 위한 루트 인증서는 애플리케이션과 함께 전송된다. 이때, 루트 인증서의 위치는 애플리케이션의 base directory 로 한다.

루트 인증서가 동반되지 않은 경우, 미들웨어는 서버 인증을 생략하며, 필요시 애플리케이션에서 자체적으로 JSSE API 를 이용해서 수행할 수 있다.

루트 인증서의 파일 이름의 형식은 atsc.tls.<organization_id>.<application_id>.x 이며, id 는 16 진수로 표시한다.

예 : atsc.tls.00000004.0045.1

12.8.3. 사용자 인증과 전자서명

데이터 방송을 이용하여 T-커머스와 T-뱅킹 등과 같은 양방향 서비스를 실시하기 위해서는 사용자 인증과 전자서명을 반드시 지원해야 하며 이를 위해 지원되어야 하는 서버와 수신기 사양은 다음과 같다.

12.8.3.1. 서비스 시나리오

사용자 인증 및 전자서명과 관련하여 수신기에서 인증서 처리를 위한 인터페이스와 인증 모듈을 위한 상시 저장 공간을 제공한다. 지상파 방송사는 공통의 인증 처리와 관련한 모듈을 개발하여 수신기에 저장될 수 있도록 한다. 실제 인증 처리는 수신기에 저장된 인증 모듈과 인증서를 이용해서 애플리케이션에 비즈니스 로직을 포함하여 애플리케이션이 전송되는 방법으로 수신기에 다운로드 되어서 실행된다.

인증 모듈 저장 방법 : 수신기는 지상파 방송사가 제공하는 공통의 인증 모듈을 수신기에 저장할 수 있도록 상시 저장 공간을 제공한다. 상시 저장 공간에 인증 모듈이 없을 시에는 인증을 필요로 하는 애플리케이션이 이를 다운로드받아 저장하여 사용한다. 저장된 인증 모듈의 갱신은 네트워크 또는 방송파를 통해 다운로드받아 수신기의 상시 저장 공간에 저장한다.

12.8.3.2. 수신기 지원 사항

- USB 메모리 인터페이스 : 수신기의 기능 제약상 개인 인증서의 신규 발급 및 갱신 기능을 수신기에서 모두 구현하는 것은 어려우므로 개인 인증서의 신규 발급과 갱신 기능은 PC 를 사용하고 그 결과물인 개인 인증서를 사용할 수 있도록 수신기에서는 USB 메모리 인터페이스를 지원해야 한다. 이에 대한 자세한 내용은 부속서 0 이동식 저장 장치 접근 방식을 참조한다.

- 상시 저장 공간 : 수신기에서는 인증 모듈과 인증서 및 개인 키를 저장하기 위한 저장 공간을 제공해야 하며, 수신기 업체에 상관없이 단일화된 접근 방법을 제공해야 한다. 이때 사용되는 API 는 11.4.1.2 절 Persistent Storage API 를 따른다. 이에 대한 자세한 내용은 부속서 G 의 인증 모듈 저장/사용 및 부속서 G 의 이동식 저장 장치 접근 방식을 참조한다.

12.8.3.3. 방송사 지원 사항

커머스 서버 : 사용자 인증과 전자서명을 위해 방송사 커머스 서버에서는 수신기에서 처리되는 인증 모듈을 리턴 채널로 다운로드할 수 있어야 한다. 또한 수신기와 인터페이스하여 보안/인증과 관련된 로직을 처리해야 한다.

12.8.4. DVBClassloader

방송사에서 신뢰할 수 있는 채널로 전송된 애플리케이션이 DVBClassLoad 를 사용하여 interaction channel 을 통해 load 하여 실행하고자 할 경우, 다음의 조건을 모두 만족할 경우에는 수신기의 security manager 는 이를 허가한다.

- 방송사에서 신뢰할 수 있는 채널로 전송된 애플리케이션은 signed application 이어야 한다.
- 방송사에서 신뢰할 수 있는 채널로 전송된 애플리케이션에는 interaction channel 접근을 위한 적절한 permission request file이 있어야 한다.
- Interaction channel을 통해 다운로드되는 class 파일은 모두 ACAP signing framework에 따라 서명되어야 하며, 수신기에서 서명 검증되어야 한다.

수신기에서는 애플리케이션이 DVBClassLoader 를 사용 interaction channel 로 loading 하고자 하는 class 파일을 검증하기위해 해당 파일이 있는 디렉토리에 hashfile, signaturefile, certificates 파일을 함께 다운받아 서명 검증한다. 애플리케이션의 서명 및 서명 검증을 위해 요구되는 코드 사인용 인증서 발급 체계는 본 표준 12.8.6 절에서 정의하고 있는 바를 따른다.

여러 개의 class 파일이 동일 디렉토리상에 존재할 경우에도 hashfile 은 하나만 존재하며, 각각의 파일을 따로 digest 하여야 한다. 즉 acap.hashfile 내부에 검증하고자 하는 classfile 개수만큼 digest_count 가 있어야 하며, 각각의 digest 내부에 name_count 는 1 이 되어야 한다.

jar 파일에 있는 class 를 loading 할 경우에는 class 가 아닌 jar 파일에 대해 동일한 방법으로 검증하여야 한다.

signaturefile 과 certificates 파일은 각각 하나씩 존재하여야 하며, 재전송을 고려하여 acap 이외의 signaturefile 과 certificates 파일을 같은 디렉토리에 포함할 수도 있다. 즉 load 하고자하는 파일이 있는 디렉토리에는 acap.signaturefile.1, acap.certificates.1 파일이 각각 반드이 존재하여야 하고, 재전송을 고려했을 때는 선택적으로 ocap.signaturefile.1, ocap.certificates.1 파일과 같이 타 표준 파일이 추가로 존재할 수 있다.

12.8.5. PRF(Permission Request File)의 적용 범위

지상파로 전송된 애플리케이션은 trusted application 으로 간주될 수 있다. 그러나 Default Sandbox 를 제외하고, PRF 에 의해 명백히 요청되지 않은 permission 은 수신기에 의해 허가되지 않는다. 또한 수신기는, 그 정책에 따라, PRF 에 의해 요청된 모든 permission 에 대한 허가를 할 의무는 없다. Default sandbox 리소스 및 추가 리소스에 관한 정의는 MHP1.0.3('ETSI TS 101 812 V1.3.1')을 따른다.

12.8.6. 코드 사인 검증 인증 체계

방송사에서 interaction channel 을 통해 전송하는 애플리케이션의 서명 및 서명 검증을 위해 애플리케이션을 위한 코드 사인용 인증서 발급이 요구된다.

코드 사인용 인증서 발급을 위한 최상위 인증기관의 역할은 TTA 에서 담당하며, TTA 는 인증서 발급 정책, 인증서 발급 및 폐지, 인증서 관리 의무 등을 별도로 정의하여 그 역할을 수행한다.

12.8.6.1. 인증서 프로파일

TTA 는 다음과 같이 정의한 인증서 프로파일에 따라 코드 사인을 위한 인증서들을 발급하며, 프로파일의 상세한 정의는 부속서 I 를 따른다.

표 12.1 RootCA 인증서 프로파일 요약

#	Field 명	값
1	Version	V3
2	Subject Name	C=KR O=TTA OU=ACAP PKI CN=TTA ACAP Root CA
3	Signed By	Self-signed
4	Validity Period	30+ years
5	Modulus Length	2048 bits
6	Extensions	- KeyUsage [c,m](keyCertsign, cRL Sign) - subjectkeyidentifier [n,m]* - basicConstraints [c,m](cA=true)

* [c] = critical, [n] = non-critical, [m] = mandatory, [o] = optional

표 12.2 CA 인증서 프로파일

#	Field 명	값
1	Version	V3
2	Subject Name	C=KR O= TTA (ACAP PKI) CN= TTA CVC CA
3	Signed By	TTA ACAP Root CA
4	Validity Period	20 years
5	Modulus Length	2048 bits
6	Extensions	- KeyUsage [c,m](keyCertsign, cRL Sign) - subjectkeyidentifier [n,m] - authorityKeyIdentifier [n,m] - basicConstraints [c,m](cA=true, pathLenConstraint=0) - subjectAltName [n,o] (Directory Address)

표 12.3 애플리케이션 코드 사인용 인증서 프로파일

#	Field 명	값
1	Version	V3
2	Subject Name	C=<country>* O=<Company Name> CN=<Company Name> Application CVC
3	Signed By	TTA CVC CA
4	Validity Period	Up to 10 years
5	Modulus Length	1024, 1536 or 2048
6	Extensions	- extendedKeyUsage [c,m](id-kp-codeSigning) - subjectkeyidentifier [n,m] - authorityKeyIdentifier [n,m] - CrlDistributionPoint [c,o]

* <country> = KR 등의 국가명

**<organization_ID> : DVB Consortium에서 ID를 발급받아 인증서 신청을 함

각 방송사들을 애플리케이션 코드 사인용 인증서를 TTA 로부터 발급받아 리턴 채널을 통해 전송하는 애플리케이션을 코드 사인한다.

12.8.6.2. 인증서 폐지 목록(CRL) 프로파일의 전송

인증서 폐지 목록은 방송사의 지상파 데이터 방송 채널을 통해 수신기로 송출하거나 리턴 채널을 통해 수신기가 다운로드 받을 수 있도록 지원할 수 있다.

전송하는 인증서 폐지 목록 중 애플리케이션 인증서들에 대한 폐지 목록의 파일 이름은 다음과 같이 정의한다.

- acap.crl.<x>

여기에서 <x>는 각각의 crl 파일을 검증하기 위해 요구되는 crl 을 발급한 인증서의 <x> 부분과 대응되어야 한다.

전송하는 인증서 폐지 목록 중 root 인증서로 발급한 인증기관 인증서들에 대한 폐지 목록의 파일 이름은 다음과 같이 정의한다.

- acap.crl.root.<x>

여기에서 <x>는 단지 acap.crl.root 파일이 같은 디렉토리에 하나 이상 존재할 경우를 고려하여 식별자로서 붙여준다.

12.8.6.3. RootCA 인증서 관리 메시지 파일 이름 정의

국내 지상파 데이터 방송 코드 사인 인증서의 Root 인증서 관리 메시지의 파일 이름은 다음과 같이 정의한다.

- acap.rcmm

따라서, 최근의 acap.rcmm 을 수신하였을 경우, 수신기가 이름을 변경하여 저장해야 하는 이전의 rcmm 메시지 파일 이름은 acap.rcmm.<x>와 같이 정의하도록 하며, 여기서 x 는 1, 2, 3 과 같은 integer 숫자로 중복없이 1 부터 차례로 붙여야 한다.

13. 그래픽 참조 모델

ACAP 문서 'A/101A'의 '13 Graphics Reference Model'을 따라야 한다.

14. 시스템 통합

14.1. 리소스 참조 및 로케이터

ACAP 문서 'A/101A'의 '14.1 Resource Reference and Locators'를 따라야 한다.

14.1.1. ACAP URI scheme

ACAP 문서 'A/101A'의 '14.1.1 ACAP URI Scheme'을 따라야 한다.

14.2. 영구 로컬 저장 공간

ACAP 문서 'A/101A'의 '14.2 Persistent Local Storage'를 따라야 한다.

15. 최소 수신기 요구 사항

15.1. 일반

ACAP 문서 'A/101A'의 '15.1 General'을 따라야 한다.

15.2. 사용자 입력

ACAP 문서 'A/101A'의 '15.2 User Input'을 따라야 한다.

15.2.1. 리모콘 UI

리모콘의 칼라 버튼 순서는 R, G, Y, B(왼쪽 우선)로 한다.

15.3. 그래픽

ACAP 문서 'A/101A'의 '15.3 Graphics'를 따라야 한다.

15.3.1. 그래픽 해상도

수신기는 SD 의 경우 640x480 이상, HD 의 경우 960x540 이상의 해상도를 가지는 그래픽 평면을 제공해야 한다.

15.3.2. Color Model

수신기는 아래와 같이 총 32 bits 의 color model 을 지원하는 그래픽 평면을 제공해야 한다.

- Alpha : red : green : blue = 8 : 8 : 8 : 8

15.3.3. Translucency(alpha 값)

수신기는 비디오 위에 불투명하게 그래픽을 표현하기 위하여 총 8 bits(256 level) alpha blending 을 지원해야 한다.

16. 상세한 플랫폼 프로파일 정의

16.1. 일반 사항

ACAP 문서 'A/101A'의 '16.1 General'을 따라야 한다.

16.2. MHP 정의의 수정

16.2.1. 캐러셀

ACAP 문서 'A/101A'의 '16.2.1 Carousel'을 따라야 한다.

16.2.1.1. NSAP Address

ACAP 문서 'A/101A'의 '16.2.1.1 NSAP Address'를 따라야 한다.

16.2.1.2. Content Type and Timestamp Inheritance

ACAP 문서 'A/101A'의 '16.2.1.2 Content Type and Timestamp Inheritance'를 따라야 한다. 단, Service Gateway Object 에 Content Type Descriptor 가 없을 경우, default 값으로 ACAP ' A/101A ' 의 '16.2.1.2'에서 지정한 "application/java" 또는 "application/dvbj"를 사용하여야 한다.

16.2.1.3. Application transport over HTTP

ACAP 문서 'A/101A'의 '16.2.1.3 Application transport over HTTP'를 따라야 한다.

16.2.1.4. Time Stamp Descriptor

ACAP 문서 'A/101A'의 '16.2.1.4 Time Stamp Descriptor'를 따라야 한다.

16.2.1.5. Usage of Private Data for non-ACAP Extensions

ACAP 문서 'A/101A'의 '16.2.1.5 Usage of Private Data for non-ACAP Extensions'을 따라야 한다.

16.2.1.6. Data Broadcast Descriptor

ACAP 문서 'A/101A'의 '16.2.1.6 Data Broadcast Descriptor'를 따라야 한다.

16.2.2. Application Signaling

16.2.2.1. Application Content Types

ACAP 문서 'A/101A'의 '16.2.2.1 Application Content Types'을 따라야 한다.

16.2.2.2. Application Protocol ID

ACAP 문서 'A/101A'의 '16.2.2.2 Application Protocol ID'를 따라야 한다.

16.2.2.3. Signaling of Profiles and Versions Required by Applications

ACAP 문서 'A/101A'의 '16.2.2.3 Signaling of Profiles and Versions Required by Applications'을 따라야 한다.

16.2.2.4. ACAP-X Extensions

ACAP 문서 'A/101A'의 '16.2.2.4 ACAP-X Extensions'을 따라야 한다.

17. 정합

17.1. GEM과의 호환성

ACAP 문서 'A/101A'의 '17.1 Compliance with GEM'을 따라야 한다.

17.1.1. MHP 정의 수정

ACAP 문서 'A/101A'의 '17.1.1 Modifications to MHP Definitions of Functional Equivalents'를 따라야 한다.

18. 모니터 애플리케이션

ACAP 문서 'A/101A'의 '18 Monitor Application'를 따라야 한다.

부 속 서 A

Content Identification API

ACAP 문서 ‘A/101A’의 ‘Annex A1 PACKAGE ORG.ATSC.SI’을 따라야 한다.

부 속 서 B

Document Type Definitions(Normative)

B.1. SCOPE

ACAP 문서 'A/101A'의 'Annex B1'을 따라야 한다.

B.2. ACAP PERMISSION REQUEST FILE DOCUMENT TYPE

ACAP 문서 'A/101A'의 'Annex B2'를 따라야 한다.

B.3. ACAP-J FONT INDEX FILE DOCUMENT TYPE

ACAP 문서 'A/101A'의 'Annex B3'을 따라야 한다.

B.4. ACAP-X APPLICATION METADATA DOCUMENT TYPE

ACAP 문서 'A/101A'의 'Annex B4'를 따라야 한다.

B.5. ACAP-X MARKUP DOCUMENT TYPE

ACAP 문서 'A/101A'의 'Annex B5'를 따라야 한다.

부 속 서 C
Data Piping 프로토콜

데이터 방송 서비스를 제공함에 있어서 사용자 고유의 콘텐츠를 Data Piping 프로토콜을 이용하여 전송하는 것을 선택적으로 지원한다.

부 속 서 D
수신기의 표준 준수 시험을 위한 인터페이스
(Interface for Conformance Testing of a Receiver)

지상파 데이터 방송 수신기에 대한 표준 적합성 시험을 위해서는, 시험 애플리케이션 송출이 필요하고 수신기는 이를 수신하여 실행한 다음 실행 결과를 출력해야 한다. 일반적으로, 데이터 방송 표준 적합성 시험을 위해서는 수천 개에 달하는 시험 애플리케이션들을 송출하고 시험 결과를 수집하는 장치가 요구된다. 이와 같은 장치를 시험 서버라고 한다면, 수신기에서 실행하고 있는 시험 애플리케이션이 시험 서버와 약속된 프로토콜을 통해 시험 과정에서 필요한 데이터를 교환할 수 있도록 수신기와 시험 서버 간의 통신 방식 및 수신기와 시험 애플리케이션 간의 인터페이스 정의가 필요하고 수신기는 이를 지원하여야 한다.

D.1 절에서 수신기와 시험 서버 간의 통신 방식을 정의하고, D.2 절에서는 시험 애플리케이션과 시험 서버가 시험에 필요한 데이터를 주고받을 수 있도록 수신기가 제공하여야 하는 지상파 데이터 방송 테스트 API 에 대해 기술한다.

D.1. 수신기와 시험 서버 간의 통신 방식

지상파 데이터 방송 표준에서 정의한 바에 의하면, 수신기는 리턴 채널을 지원해야 하고, 리턴 채널 프로토콜들 중 하나로 TCP/IP 를 지원하도록 규정하고 있으므로, 수신기와 시험 서버 간의 통신 방식은 리턴 채널을 이용한 TCP/IP 소켓 인터페이스를 사용한다. 따라서, 지상파 데이터 방송 수신기는 표준 적합성 시험을 위해 아래와 같은 물리적 통신 인터페이스와 통신 프로토콜을 지원하여야 한다.

D.1.1. 물리적 통신 인터페이스

수신기는 지상파 데이터 방송 표준에서 리턴 채널의 Physical 및 Data-Link Layer 로서 권고하고 있는 Ethernet 인터페이스를 지원하고 RJ-45 커넥터를 지원하여야 한다.

D.1.2. 통신 프로토콜

수신기는 시험 서버와의 통신을 위해 지상파 데이터 방송 표준에서 기술하고 있는 리턴 채널의 TCP/IP 프로토콜을 지원하여야 한다.

D.2. 지상파 데이터 방송 테스트 API

지상파 데이터 방송 수신기에 대한 표준 적합성 시험을 위해 수신기에서 실행하는 시험 애플리케이션이 시험 서버와 시험에 필요한 데이터를 주고 받을 수 있도록 표준 적합성 시험용 API 인 org.acap.test.ACAPTest 클래스를 정의하고, 수신기는 이를 지원하여야 한다.

ACAPTest 클래스는 시험 애플리케이션과 시험 서버가 TCP/IP 프로토콜을 기반으로 메시지를 송수신할 수 있는 인터페이스를 제공하는 API 로서, 수신기와 시험 서버 간의 통신 채널을 설정하고 초기화하는 initialize() 메소드, 수신기에서 시험 서버로 메시지를 송신하는 sendMessage() 메소드, 그리고 시험 서버로부터 메시지를 수신하는 receiveMessage() 메소드 등으로 구성된다.

ACAPTest 클래스의 initialize()에 의해 수신기와 시험 서버 간의 TCP/IP 통신 채널이 설정되는데, 설정된 통신 채널은 수신기가 reboot 하기 전까지는 병행 실행하는 모든 시험 애플리케이션들이 공통으로 사용하는 구조이므로, 이를 고려하여 시험 애플리케이션을 작성하여야 한다.

ACAPTest 클래스의 sendMessage()와 receiveMessage()에 의해 전달되는 메시지는 일련의 바이트 열로 구성된 데이터로서, ACAPTest 클래스를 구현하는 수신기는 메시지의 내용이나 포맷을 수정하지 말아야 한다. 그러나, 송수신 메시지의 끝을 의미하는 MESSAGE_TERMINATION_BYTE 에 대해서는 ACAPTest 클래스 구현에서 처리하여야 한다. 즉, sendMessage()에서는 시험 서버로 메시지를 송신할 때 메시지 끝에 MESSAGE_TERMINATION_BYTE 를 추가하여 송신하여야 하며, receiveMessage()에서는 시험 서버로부터 수신한 메시지를 시험 애플리케이션에게 전달할 때 MESSAGE_TERMINATION_BYTE 를 제외해야 한다.

다음은 ACAPTest 클래스의 구현 예제이다. 이 예제는 단지 ACAPTest 클래스 구현을 위한 참조용으로 사용될 수 있으며, 어떠한 기술 지원이나 유지보수의 책임을 지지 않는다. 즉, 아래 코드 사용에 대한 책임은 전적으로 ACAPTest 클래스 개발자에게 있다.

```
/*
 * ACAPTest의 구현 예제로서, 아래의 코드 사용에 대한 모든 책임은 사용자에게 있다.
 */
package org.acap.test;

import java.io.*;
import java.net.*;
import java.security.*;

public class ACAPTest {
  /**
   * 시험 서버와 수신기 간에 byte array 형식의 메시지를 송수신하는 데 있어서,
   * 메시지의 마지막임을 표시하기 위한 값
   */
  public final static byte MESSAGE_TERMINATION_BYTE = (byte)0;

  /**
   * 시험 서버와 수신기 간에 송수신하는 메시지의 최대 바이트 수
   */
  public final static int MAX_MESSAGE_LENGTH = 1500;

  private static final int SOCKET_CREATE_RETRY_COUNT       = 10;
  private static final int SOCKET_CREATE_RETRY_WAIT_MILLIS = 1000;

  // Flag to check if initialize() method has ever been executed.
  private static boolean initializeCalled = false;
  private static boolean initSucceeded = false;

  private static BufferedInputStream bufferedInputStream = null;
  private static BufferedOutputStream bufferedOutputStream = null;

  /**
   * 객체 생성을 막기 위한 private constructor
   */
  private ACAPTest() {}

  /**
   * 이 메소드는 시험 애플리케이션에 의해 호출되어, 수신기가 시험 애플리케이션과
   * 시험 서버간의 메시지 송수신을 위한 TCP/IP 통신 채널을 초기화한다.
   *
   * @param ateIP            시험 서버의 IP V4 주소를 가리키는
```

```
     *                          "xxx.xxx.xxx.xxx" 형식의 문자열
     * @param atePort            메시지 송수신 채널로서 사용될 시험 서버의
     *                          TCP Port 번호
     */
    public static synchronized void initialize(final String ateIP, final int atePort) throws IOExcep
tion {
      if (initializeCalled) {
        System.err.println("ACAPTest.initialize() already called!.");
        return;
      }
      initializeCalled = true;

      Socket socket = null;
      try {
        socket = (Socket) AccessController.doPrivileged(
          new PrivilegedExceptionAction() {
            public Object run() throws Exception {
              int socketCreateTriesLeft = SOCKET_CREATE_RETRY_COUNT;
              Socket socket = null;
              while (socket == null) {
                try {
                  socket = new Socket(InetAddress.getByName(ateIP), atePort);
                }
                catch( IOException ioe ) {
                  if( --socketCreateTriesLeft < 0) {
                    throw ioe;
                  }
                  try {
                    Thread.sleep( SOCKET_CREATE_RETRY_WAIT_MILLIS );
                  }
                  catch( InterruptedException ignore ) {
                  }
                }
              }
              return socket;
            } // run
          }
        );
      }
      catch (PrivilegedActionException pae) {
        Exception e = pae.getException();
```

```
            System.err.println("ACAPTest.initialize got exception : e = " + e);
            if (e instanceof IOException) throw (IOException) e;
            else throw (RuntimeException) e;
        }

        bufferedInputStream = new BufferedInputStream(socket.getInputStream());
        bufferedOutputStream = new BufferedOutputStream(socket.getOutputStream());
        initSucceeded = true;
    }

    /**
     * byte array 형식으로 주어진 메시지를 TCP 소켓을 통하여 시험 서버로 송신한다.
     */
    public static synchronized void sendMessage(final byte[] rawMessage) throws IOException
{
        if (!initSucceeded) {
            throw new IOException("ACAPTest has not been successfully initialized.");
        }

        if (rawMessage == null) {
            throw new IllegalArgumentException("rawMessage is null ");
        }

        for (int i = 0; i < rawMessage.length; i++) {
            if (rawMessage[i] == MESSAGE_TERMINATION_BYTE) {
                throw new IllegalArgumentException("rawMessage contains "
                                    + "MESSAGE_TERMINATION_BYTE at " + i + "th byte!");
            }
        }

        try {
            AccessController.doPrivileged(
                new PrivilegedExceptionAction() {
                    public Object run() throws Exception {
                        bufferedOutputStream.write(rawMessage, 0, rawMessage.length);
                        bufferedOutputStream.write(MESSAGE_TERMINATION_BYTE);
                        bufferedOutputStream.flush();
                        return null;
                    } // run
                } // new PrivilegedAction
            ); // doPrivileged
```

```
    }
    catch (PrivilegedActionException pae) {
      Exception e = pae.getException();
      System.err.println("ACAPTest.sendMessage got " + e);
      if (e instanceof IOException) throw (IOException) e;
      else throw (RuntimeException) e;
    }
  }

  /**
   * 시험 서버로부터 byte array 형식의 메시지를 TCP 소켓을 통하여 수신한다.
   */
  public static synchronized byte[] receiveMessage() throws IOException {
    if (!initSucceeded) {
      throw new IOException("ACAPTest has not been successfully initialized.");
    }

    byte[] receivedMsg = null;
    try {
      receivedMsg = (byte[]) AccessController.doPrivileged(
        new PrivilegedExceptionAction() {
          public Object run() throws Exception {
            int count = 0;
            byte[] recvBuf = new byte[MAX_MESSAGE_LENGTH];

            while(true)  {
              byte rcvdByte = (byte)(bufferedInputStream.read() & 0xff);
              if (rcvdByte == MESSAGE_TERMINATION_BYTE)
                break;

              if (count < MAX_MESSAGE_LENGTH) {
                recvBuf[count]= rcvdByte;
                count++;
              }
              else {
                // Any bytes received beyond MAX_MESSAGE_LENGTH
                // should be discarded. !!
                System.err.println("receive : [WARN] Msg from ATE exceeds MAX_MESSAGE
_LENGTH! So, Discarded!!₩n");
              }
            }
```

```
                byte[] ret = new byte[count];
                System.arraycopy(recvBuf, 0, ret, 0, count);
                return ret;
              } // run
            } // new PrivilegedAction
          ); // doPrivileged
        }
        catch (PrivilegedActionException pae) {
          Exception e = pae.getException();
          System.err.println("ACAPTest.receiveMessage got " + e);
          if (e instanceof IOException) throw (IOException) e;
          else throw (RuntimeException) e;
        }
        return receivedMsg;
      }
    }
```

Class Summary	
Classes	
ACAPTest	이 클래스는 수신기와 시험 서버 간에 TCP/IP 프로토콜을 사용하는 간단한 형태의 메시지 송수신 방법을 제공한다.

org.acap.test
ACAPTest

Declaration
public class ACAPTest
java.lang.Object
  |
  +--org.acap.test.ACAPTest

Description

이 클래스는 수신기와 시험 서버 간에 TCP/IP 프로토콜을 사용하는 간단한 형태의 메시지 송수신 방법을 제공한다. TCP/IP 통신 채널을 제공하기 위한 물리적 수단에 대한 규정은 지상파 데이터 방송 표준의 본문에서 명시되어야 한다.

이 클래스를 사용하는 시험 애플리케이션은 반드시 initialize()를 먼저 호출하여 시험 서버와의 통신 채널을 초기화한 후 sendMessage()나 receiveMessage() 메소드를

사용하여야 한다. initialize() 호출을 통하여 통신 채널이 연결되면 수신기가 reboot 되기 전까지는 이 연결을 유지해야 한다.

시험 애플리케이션과 시험 서버와의 메시지 송수신은 공통된 하나의 통신 채널을 통하여 이루어져야 하며, 만일 둘 이상의 시험 애플리케이션이 동시에 통신 채널을 사용하고자 하는 경우에 각각의 메시지 송수신이 독립적이고 순차적으로 이루어지도록 보장되어야 한다. 시험 애플리케이션이 이 클래스가 제공하는 메소드의 기능을 사용하는데 있어서 어떤 Permission 이 요구될 경우, 그 메소드를 수행하는 동안에는 Permission 을 허용해 주어야 한다. 예를 들어, 시험 애플리케이션이 SocketPermission 을 갖고 있지 않더라도 Socket 사용이 허용되어야 한다.

Member Summary

Fields

static int MAX_MESSAGE_LENGTH
: 시험 서버와 수신기 간에 송수신하는 메시지의 최대 바이트 수

static byte MESSAGE_TERMINATION_BYTE
: 시험 서버와 수신기 간에 byte array 형식의 메시지를 송수신하는 데 있어서, 메시지의 마지막임을 표시하기 위한 값

Methods

static void initialize(java.lang.String ateIP, int atePort)
: 이 메소드는 시험 애플리케이션에 의해 호출되어, 수신기가 시험 애플리케이션과 시험 서버간의 메시지 송수신을 위한 통신 채널을 초기화하도록 한다.

static byte[] receiveMessage()
: 시험 서버로부터 byte array 형식의 메시지를 TCP 소켓을 통하여 수신한다.

static void sendMessage(byte[] rawMessage)
: byte array 형식으로 주어진 메시지를 TCP 소켓을 통하여 시험 서버로 송신한다.

Inherited Member Summary

Methods inherited from class java.lang.Object

clone(), equals(Object), finalize(), getClass(), hashCode(), notify(), notifyAll(), toString(), wait(), wait(), wait()

Fields

MESSAGE_TERMINATION_BYTE

public static final byte **MESSAGE_TERMINATION_BYTE**

시험 서버와 수신기 간에 byte array 형식의 메시지를 송수신하는 데 있어서, 메시지의 마지막임을 표시하기 위한 값

MAX_MESSAGE_LENGTH

public static final int **MAX_MESSAGE_LENGTH**

시험 서버와 수신기 간에 송수신하는 메시지의 최대 바이트 수

Methods

initialize(String, int)

public static void **initialize**(java.lang.String ateIP,
int atePort)
throws java.io.IOException

이 메소드는 시험 애플리케이션에 의해 호출되어, 수신기가 시험 애플리케이션과 시험 서버 간의 메시지 송수신을 위한 TCP/IP 통신 채널을 설정하고 초기화한다.

시험 애플리케이션이 이 메소드를 호출할 시점에는 이미 시험 서버가 시험 애플리케이션이 연결하고자 하는 TCP 포트에 대해 서버 소켓을 열고 대기하고 있는 것으로 가정한다. 만일 주어진 IP 주소와 TCP Port를 사용하여 시험 서버와의 소켓 연결에 실패하는 경우는 IOException을 발생시킨다.

이 메소드가 두 번 이상 호출될 경우, 첫 번째 호출에서만 통신 채널 설정 작업을 수행하고, 첫 번째 이후의 호출은 무시된다. 즉, 첫 번째 호출에서 초기화 작업이 실패하였더라도, 그 이후의 호출에서 통신 채널 초기화를 재시도하지 않는다.

initialize() 호출을 통하여 통신 채널이 연결되면 수신기가 reboot되기 전까지는 이 연결을 유지해야 한다.

Parameters:

ateIP - 시험 서버의 IP V4 주소를 가리키는 "xxx.xxx.xxx.xxx" 형식의 문자열

atePort - 메시지 송수신 채널로서 사용될 시험 서버의 TCP Port 번호

Throws:

java.io.IOException - 통신 채널 초기화 과정에 오류가 발생하는 경우

receiveMessage()

public static byte[] **receiveMessage**()

throws java.io.IOException

시험 서버로부터 byte array 형식의 메시지를 수신한다. 시험 서버로부터 수신되는 메시지는 수정되지 않고 그대로 반환 값으로 전달되어야 한다.

시험 서버로부터 수신되는 데이터는 MESSAGE_TERMINATION_BYTE를 수신하기 전까지 byte array로 축적하였다가 반환해야 하며, 만일 수신되는 데이터의 길이가 MAX_MESSAGE_LENGTH를 초과하는 경우에는 MAX_MESSAGE_LENGTH 이후부터 MESSAGE_TERMINATION_BYTE를 수신할 때까지의 데이터는 버려진다.

Returns:

시험 서버로부터 수신한 byte array 형식의 메시지. 최대 길이는 MAX_MESSAGE_LENGTH 이며 MESSAGE_TERMINATION_BYTE는 포함하지 않는다.

Throws:

java.io.IOException - initialize() 메소드가 호출되지 않았거나, initialize() 수행 시 통신 채널 초기화에 성공하지 못했거나, 메시지 수신 과정에서 오류가 발생하는 경우

sendMessage(byte[])

public static void **sendMessage**(byte[] rawMessage)

throws java.io.IOException

byte array 형식으로 주어진 메시지를 TCP 소켓을 통하여 시험 서버에 송신한다. 주어진 메시지는 수정되지 않고 그대로 시험 서버에 전달되어야 한다.

시험 애플리케이션이 인자로서 넘겨주는 byte array에는 임의의 데이터가 포함될 수 있으나, MAX_MESSAGE_LENGTH와 같거나 적은 길이여야 하고, MESSAGE_TERMINATION_BYTE를 포함하지 않아야 한다.

Parameters:

rawMessage - 시험 서버로 송신하고자 하는 메시지

Throws:

java.lang.IllegalArgumentException - rawMessage가 null이거나, MAX_MESSAGE_LENGTH 보다 큰 길이를 갖거나, MESSAGE_TERMINATION_BYTE를 포함하고 있는 경우

java.io.IOException - initialize() 메소드가 호출되지 않았거나, initialize() 수행 시 통신 채널 초기화에 성공하지 못했거나, 메시지 송신 과정에서 오류가 발생하는 경우

부 속 서 E

OCAP 1.0 OcapLocator API

'OC-SP-OCAP1.0.0-070814' OCAP 1.0.0 의 'Annex I OCAP 1.0 Net API OcapLocator API'를 따른다.

부 속 서 F

OCAP 1.0 Storage API

'OC-SP-OCAP1.0.0-070814' OCAP 1.0.0 의 'Annex V OCAP 1.0 Storage API'를 따른다.

부 속 서 G
이동식 저장 장치 접근 및 인증 모듈 저장/사용 방식
(Methods for Removable Storage Device Accessing and Certification Module Storing/Using)

G.1. 이동식 저장 장치 접근 방식

사용자의 인증서를 수신기에 전달하기 위하여 이동식 저장 장치를 사용하는데, 이동식 저장 장치에 접근하기 위한 API 는 11.4.1.2 절 Persistent Storage API 를 이용한다. 이동식 저장 장치를 위한 StorageProxy 의 StorageOption 은 DetachableStorageOption 이나 RemovableStorageOption 을 가지게 되며, 이동식 저장 장치의 모든 파일을 접근할 수 있도록 제공되는 LogicalStorageVolume 을 사용하여 인증서와 개인 키를 읽는다. 이 LogicalStorageVolume 은 하위 디렉토리에 대해 read 의 권한만 주어진다.

이동식 저장 장치에 접근하여 인증서와 개인 키 등의 데이터를 읽기 위해서는 보안 인증 모듈을 사용하는 애플리케이션이 permission request file 의 “file” element 가 “true”의 값을 가져야 한다.

인증서와 개인 키 등의 데이터를 읽기 위해 한글로 인코딩된 이름을 가지는 파일 및 디렉토리에 접근할 수 있어야 한다.

G.2. 인증 모듈 저장/사용 방식

인증 모듈의 저장을 위해 수신기는 상시 저장 공간을 제공하며, ACAP 애플리케이션이 인증 모듈의 저장/갱신/삭제를 주관한다. 상시 저장 공간을 접근하기 위한 API 는 11.4.1.2 절 Persistent Storage API 를 이용한다. 수신기는 이를 위해 기본적으로 하나의 LogicalStorageVolume 을 제공하며 이 LogicalStorageVolume 의 owner 는 수신기이다. 이 LogicalStoreVolume 은 지상파 방송국의 organization_id 를 가진 애플리케이션에 의해서만 read/write 권한이 주어진다.

애플리케이션이 인증 모듈의 목적으로 만들어진 LogicalStorageVolume 을 구분할 수 있는 방법이 제공되어야 하는데, StorageOption 이 DetachableStorageOption 이나 RemovableStorageOption 이 아닌 모든 StorageProxy 에서 얻어지는 모든 LogicalStorageVolume 에 대해 getPath() API 를 수행하여 나오는 결과의 하위 디렉토리

경로가 ‘OCAP_LSV/ACAP_SECURITY’ 이면 이 LogicalStorageVolume 을 보안 인증 모듈을 위한 것이라고 판단한다.

구해진 LogicalStorageVolume 에 APP_MODULE 과 CERT 디렉토리가 존재하는데 보안 인증 모듈은 APP_MODULE 디렉토리를 사용하고 이동식 저장 장치로부터 옮겨진 사용자 인증서/비밀 키는 CERT 디렉토리를 사용한다.

각 방송사의 애플리케이션이 보안 인증 모듈을 사용하기 위해서는 permission request file 에 “file” element 가 “true”의 값을 가져야 한다.

부 속 서 H
실시간 업데이트 데이터 처리 방법
(Real-time Updating Methods of Application's Data)

실시간으로 애플리케이션의 데이터를 업데이트 하는 방법에는 모듈 업데이트와 Stream Event 를 이용하는 방법이 있다.

H.1. 모듈 업데이트 이용

애플리케이션 데이터 파일 메시지를 포함하고 있는 캐러셀 모듈의 버전 변경을 통해 업데이트를 처리하는 방식이다. 파일 메시지의 업데이트를 모듈 데이터의 버전이 변경되었을 때 이 모듈 내에 포함되어 있는 메시지 object 가 업데이트 되었다는 것을 의미한다. 하지만 하나의 모듈 내에는 여러 개의 OC 메시지 object 가 포함될 수 있으며 모듈의 버전 변경이 모듈 내에 포함된 모든 파일 메시지 object 의 버전 변경을 의미하지는 않는다(MHP 1.0.3 B.5.1). 그러므로 모듈의 버전이 변경되면 해당 모듈에 포함된 모든 메시지 object 들이 업데이트 되었을 수 있다고 가정하고 업데이트에 필요한 처리하여야 한다.

이러한 문제를 극복하기 위해 Time Stamp Descriptor 를 이용할 수 있다(하지만 이 descriptor 의 본래 목적은 OC 에 포함된 파일의 버전을 체크하기 위한 것이 아니므로, 이 방식을 사용한 업데이트 확인은 미들웨어가 구현해야 하는 필수 사항이 아닌 참조 사항이다).

H.1.1. Time Stamp Descriptor 이용 방법

Time Stamp Descriptor 는 이를 포함하고 있는 object 의 가장 최근에 수정된 시간 정보를 담고 있다. 이 정보를 사용하여 모듈 업데이트가 일어 났을 때 해당 모듈에 포함된 object 들의 TimeStamp 가 변경되었는지를 확인하여 각각의 object 가 업데이트 되었는지를 알 수 있다.

H2. Stream Event 이용

Object Carousel 의 Stream Event 메시지 object 를 통해 앞으로 전송될 이벤트의 EventName 과 EventId 리스트를 미리 제공하고 이 정보를 바탕으로 애플리케이션이 필요한 특정 이벤트에 대해 subscribe 한다. 애플리케이션이 요청한 이벤트가 stream event descriptor 를 통해 전송되었을 때 미들웨어는 해당 descriptor 를 애플리케이션에 제공하며 애플리케이션은 이 정보를 참조하여 데이터를 업데이트 한다.

부 속 서 I
애플리케이션 코드 사인용 인증서 프로파일

애플리케이션을 위한 코드 사인용 인증서는 다음에 정의된 각 인증서 프로파일을 준용하여 발급하여야 한다.

각 필드의 지원 여부에 대한 의미는 다음과 같다.

- 생성 : 해당 필드를 삽입하여 인증서를 발급해야 함.
  - m : mandatory, 반드시 해당 필드를 삽입하여 인증서를 발급해야 함.
  - o : optional, 선택적으로 해당 필드를 인증서에 삽입하여 발급할 수 있음.
- 지원 : 인증서를 검증하는 수신기가 해당 필드를 해석하고 처리할 수 있도록 구현해야 함.
  - m : mandatory, 반드시 해당 필드를 해석하고 처리할 수 있어야 함.
  - o : optional, 해당 필드를 해석하고 처리하는 것을 선택적으로 구현할 수 있음.
- 확장 필드의 critical/non-critical에 대한 수신기의 처리.
  - critical : 수신기는 인증서를 검증하기 위해 인증서의 각 필드를 분석하다가 인지하지 못하는 critical 확장 필드를 만나면, 이 인증서를 reject 해야 한다.
  - non-critical : 수신기는 인증서를 검증하기 위해 인증서의 각 필드를 분석하다가 인지하지 못하는 critical 확장 필드를 만나더라도, 이를 무시하고 인증서 검증 프로세스를 진행할 수 있다.

I.1. Root CA 인증서 프로파일

- 기본 필드

#	Field명	ASN.1 type	Note	지원 여부	
				생성	지원
1	Version	INTEGER	0x2 (Version 3)	m	m
2	Serial Number	INTEGER	자동 할당	m	m
3	Issuer		(C=KR, O=TTA, OU=ACAP PKI, CN=TTA Root CA) - [KCAC.TS.DN] 준수 - C(Country)는 printableString, 그 이외의 속성값은 utf8String	m	m
	type	OID		m	m
	value	printableString or utf8String		m	m
4	Validity		30+ years	m	m
	notBefore	UTCTime		m	m
	notAfter	UTCTime		m	m
5	Subject		(C=KR, O=TTA, OU=ACAP PKI, CN=TTA Root CA) - [KCAC.TS.DN] 준수 - C(Country)는 printableString, 그 이외의 속성값은 utf8String	m	m
	type	OID		m	m
	value	printableString or utf8String		m	m
6	Subject Public Key Info			m	m
	algorithm	OID	전자서명인증체계 알고리즘 기술규격 준수	m	m
	subjectPublicKey	BIT STRING		m	m
7	Extensions	Extensions		m	m

● 확장 필드

#	Field명	ASN.1 type	Note	C	지원 여부	
					생성	지원
1	Authority Key Identifier		이 필드는 지원하지 않음	-	x	x
	KeyIdentifier	OCTET STRING				
	authorityCertIssuer	General Names				
	authorityCertSerialNumber	INTEGER				
2	Subject Key Identifier	OCTET STRING	subjctPublicKey의 160비트 해쉬값	n	m	m
3	Key Usage	BIT STRING	KeyCertSign, cRLSign	c	m	m
4	Certificate Policy		최상위인증기관 인증서 정책	n	o	m
	policyIdentifier	OID			o	m
	policyQualifiers				o	m
	PolicyQualifierId	OID			o	m
	Qualifier				o	m
	CPSuri	IA5String			o	m
	UserNotice				o	m
	NoticeReference	SEQUENCE			-	m
	ExplicitText	BMPString			o	m
5	Policy Mappings			n	o	m
	issuerDomainPolicy	OID				
	subjectDomainPolicy	OID				
6	Subject Alternative Names	otherName		n	o	m
7	Issuer Alternative Names	otherName	지원하지 않음	-	x	x
8	Basic Constraints			c	m	m
	cA	TRUE			m	m
	pathLenConstraint	INTEGER			o	m
9	Policy Constraints			n	o	m
	requireExplicitPolicy	INTEGER			o	m
	inhibitPolicyMapping	INTEGER			-	-
10	Name Constraints			n	o	m
	permittedSubtrees	GeneralSubtrees				
	excludedSubtrees	GeneralSubtrees				
11	Extended Key Usage	OID	지원하지 않음	-	x	x
12	CrlDistributionPoint			n	o	m

	distributionPoint	dISTRIBUTIONpOINTnAME			o	m
	reasons	ReasongFlags			o	m
	cRLIssuer	GeneralNames			o	m
13	Authority Information Access		지원하지 않음	-	x	x
	accessMethod	OID				
	accessLocation	GeneralNames				
14	Private key usage period			n	o	m
15	subject Directory attributes			n	o	m

I.2. CA 인증서 기본필드

- 기본 필드

#	Field명	ASN.1 type	Note	지원여부	
				생성	지원
1	Version	INTEGER	0x2 (Version 3)	m	m
2	Serial Number	INTEGER	자동 할당	m	m
3	Issuer		(C=KR, O=TTA, OU=ACAP PKI, CN=TTA Root CA) - [KCAC.TS.DN] 준수 - C(Country)는 printableString, 그 이외의 속성값은 utf8String	m	m
	type	OID		m	m
	value	printableString or utf8String		m	m
4	Validity		20 years	m	m
	notBefore	UTCTime		m	m
	notAfter	UTCTime		m	m
5	Subject		C=KR O=<Company Name*> CN=<Company Name*> CVC CA - [KCAC.TS.DN] 준수 - C(Country)는 printableString, 그 이외의 속성값은 utf8String	m	m
	type	OID		m	m
	value	printableString or utf8String		m	m
6	Subject Public Key Info			m	m
	algorithm	OID		m	m
	subjectPublicKey	BIT STRING		m	m
7	Extensions	Extensions		m	m

● 확장 필드

#	Field명	ASN.1 type	Note	C	지원여부	
					생성	지원
1	Authority Key Identifier			n	m	m
	KeyIdentifier	OCTET STRING			m	m
	authorityCertIssuer	General Names			m	m
	authorityCertSerialNumber	INTEGER			m	m
2	Subject Key Identifier	OCTET STRING	subjctPublicKey의 160비트 해쉬값	n	m	m
3	Key Usage	BIT STRING	KeyCertSign, cRLSign	c	m	m
4	Certificate Policy		인증기관 인증서 정책	n	o	m
	policyIdentifier	OID			o	m
	policyQualifiers				o	m
	PolicyQualifierId	OID			o	m
	Qualifier				o	m
	CPSuri	IA5String			o	m
	UserNotice				o	m
	NoticeReference	SEQUENCE			-	m
	ExplicitText	BMPString			o	m
5	Policy Mappings			-	-	-
	issuerDomainPolicy	OID				
	subjectDomainPolicy	OID				
6	Subject Alternative Names	otherName		n	o	m
7	Issuer Alternative Names	otherName		n	o	m
8	Basic Constraints		pathLenConstraint=0	c	m	m
	cA	TRUE			m	m
	pathLenConstraint	INTEGER			m	m
9	Policy Constraints			n	o	m
	requireExplicitPolicy	INTEGER			o	m
	inhibitPolicyMapping	INTEGER			-	-
10	Name Constraints			n	o	m
	permittedSubtrees	GeneralSubtrees				
	excludedSubtrees	GeneralSubtrees				
11	Extended Key Usage	OID	지원하지 않음	-	x	x
12	CrlDistributionPoint		ARL 획득 정보(URI)	n	o	m

	distributionPoint	DistributionPointName	ldap://hostname[:portnumber]/dn[?attribute]		o	m
	reasons	ReasongFlags			o	m
	cRLIssuer	GeneralNames			o	m
13	Authority Information Access		지원하지 않음	-	x	x
	accessMethod	OID				
	accessLocation	GeneralNames				
14	Private key usage period			n	o	m
15	subject Directory attributes			n	o	m

I.3. 애플리케이션 인증서 프로파일

- 기본 필드

#	Field명	ASN.1 type	Note	지원여부	
				생성	지원
1	Version	INTEGER	0x2 (Version 3)	m	m
2	Serial Number	INTEGER	자동 할당	m	m
3	Issuer		(C=KR, O=TTA (ACAP PKI), CN=TTA CVC CA) - [KCAC.TS.DN] 준수 - C(Country)는 printableString, 그 이외의 속성값은 utf8String	m	m
	type	OID		m	m
	value	printableString or utf8String		m	m
4	Validity		Up to 10 years	m	m
	notBefore	UTCTime		m	m
	notAfter	UTCTime		m	m
5	Subject		C=KR O=<Company Name*> CN=<Company Name*> Application CVC - [KCAC.TS.DN] 준수 - C(Country)는 printableString, 그 이외의 속성값은 utf8String	m	m
	type	OID		m	m
	value	printableString or utf8String		m	m
6	Subject Public Key Info			m	m
	algorithm	OID		m	m
	subjectPublicKey	BIT STRING		m	m
7	Extensions	Extensions		m	m

● 확장 필드

#	Field명	ASN.1 type	Note	C	지원여부	
					생성	지원
1	Authority Key Identifier		3가지 값을 모두 사용	n	m	m
	KeyIdentifier	OCTET STRING			m	m
	authorityCertIssuer	General Names			m	m
	authorityCertSerialNumber	INTEGER			m	m
2	Subject Key Identifier	OCTET STRING	subjctPublicKey의 160 비트 해쉬값	n	m	m
3	Key Usage	BIT STRING	Digital Signature, non-Repudiation	n	o	m
4	Certificate Policy		인증기관 인증서 정책	n	o	m
	policyIdentifier	OID			o	m
	policyQualifiers				o	m
	PolicyQualifierId	OID			o	m
	Qualifier				o	m
	CPSuri	IA5String			o	m
	UserNotice				o	m
	NoticeReference	SEQUENCE			o	m
	ExplicitText	BMPString			o	m
5	Policy Mappings			-	-	-
	issuerDomainPolicy	OID				
	subjectDomainPolicy	OID				
6	Subject Alternative Names	otherName		n	o	m
7	Issuer Alternative Names	otherName		n	o	m
8	Basic Constraints		지원하지 않음	-	x	x
	cA	FALSE				
	pathLenConstraint	INTEGER				
9	Policy Constraints			-	-	-
	requireExplicitPolicy	INTEGER				
	inhibitPolicyMapping	INTEGER				
10	Name Constraints			-	-	-
	permittedSubtrees	GeneralSubtrees				
	excludedSubtrees	GeneralSubtrees				
11	Extended Key Usage	OID	id-kp-codeSigning	c	m	m
12	CrlDistributionPoint		CRL 획득정보 (URI)	c	o	m

	distributionPoint	DistributionPointName	ldap://hostname[:port number]/dn[?attribute]		o	m
	reasons	ReasongFlags			–	–
	cRLIssuer	GeneralNames			o	m
13	Authority Information Access		온라인 폐기 목록의 접근 고려	n	o	m
	accessMethod	OID				
	accessLocation	GeneralNames				
14	Private key usage period			n	o	m
15	subject Directory attributes			n	o	m

부 속 서 J
지상파 데이터 방송 표준 API 리스트

본 부속서에서는 본 표준에서 정의하고 있는 API 리스트를 보여준다. 지상파 데이터 방송 애플리케이션은 본 부속서에서 정의하고 있는 API 범위에서 구현되어야 한다.

J.1. Java (PBP 1.1 기준)

애플리케이션은 PBP 1.1 API 중 아래 나열한 API 범위에서만 구현되어야 한다.

Package	Classes	
java.awt	ActiveEvent	Graphics2D
	Adjustable	GraphicsConfiguration
	AlphaComposite	GraphicsDevice
	AWTError	GraphicsEnvironment
	AWTEvent	GridBagConstraints
	AWTEventMulticaster	GridBagLayout
	AWTException	GridLayout
	AWTPermission	IllegalComponentStateException
	BorderLayout	Image
	CardLayout	Insets
	Color	ItemSelectable
	color.ColorSpace	LayoutManager
	Component	LayoutManager2
	Composite	MediaTracker
	Container	Point
	Cursor	Polygon
	Dimension	Rectangle
	EventQueue	Shape
	FlowLayout	SystemColor
	Font	Toolkit
	FontMetrics	Transparency
	Frame	Window
	Graphics	
java.awt.event	ActionEvent	ItemEvent
	ActionListener	ItemListener
	AdjustmentEvent	KeyAdapter
	AdjustmentListener	KeyEvent

Package	Classes	
	AWTEventListener	KeyListener
	ComponentAdapter	MouseAdapter
	ComponentEvent	MouseEvent
	ComponentListener	MouseListener
	ContainerAdapter	MouseMotionAdapter
	ContainerEvent	MouseMotionListener
	ContainerListener	PaintEvent
	FocusAdapter	TextEvent
	FocusEvent	TextListener
	FocusListener	WindowAdapter
	InputEvent	WindowEvent
	InvocationEvent	WindowListener
java.awt.image	AreaAveragingScaleFilter	ImageObserver
	BufferedImage	ImageProducer
	ColorModel	IndexColorModel
	CropImageFilter	MemoryImageSource
	DataBuffer	PixelGrabber
	DirectColorModel	RasterFormatException
	FilteredImageSource	ReplicateScaleFilter
	ImageConsumer	RGBImageFilter
	ImageFilter	
java.beans	VetoableChangeListener	VetoableChangeSupport
	PropertyChangeEvent	Beans
	PropertyChangeListener	PropertyVetoException
	PropertyChangeSupport	Visibility
java.lang	AbstractMethodError	NegativeArraySizeException
	ArithmeticException	NoClassDefFoundError
	ArrayIndexOutOfBoundsException	NoSuchFieldError
	ArrayStoreException	NoSuchFieldException
	Boolean	NoSuchMethodError
	Byte	NoSuchMethodException
	Character	NullPointerException
	Class	Number
	ClassCastException	NumberFormatException
	ClassCircularityError	Object
	ClassFormatError	OutOfMemoryError
	ClassLoader	Package
	ClassNotFoundException	Process
	Cloneable	Runnable
	CloneNotSupportedException	Runtime

Package	Classes	
	Comparable	RuntimeException
	Compiler	RuntimePermission
	Double	SecurityException
	Error	SecurityManager
	Exception	Short
	ExceptionInInitializerError	StackOverflowError
	Float	StrictMath
	IllegalAccessError	String
	IllegalAccessException	StringBuffer
	IllegalArgumentException	StringIndexOutOfBoundsExceptio
	IllegalMonitorStateException	n
	IllegalStateException	System
	IllegalThreadStateException	Thread
	IncompatibleClassChangeError	ThreadDeath
	IndexOutOfBoundsException	ThreadGroup
	InheritableThreadLocal	ThreadLocal
	InstantiationError	Throwable
	InstantiationException	UnknownError
	Integer	UnsatisfiedLinkError
	InternalError	UnsupportedClassVersionError
	InterruptedException	UnsupportedOperationException
	LinkageError	VerifyError
	Long	VirtualMachineError
	Math	Void
java.lang.ref	PhantomReference	SoftReference
	Reference	WeakReference
	ReferenceQueue	
java.lang.reflect	AccessibleObject	Member
	Array	Method
	Constructor	Modifier
	Field	Proxy
	InvocationHandler	ReflectPermission
	InvocationTargetException	UndeclaredThrowableException
java.io	BufferedInputStream	NotActiveException
	BufferedOutputStream	NotSerializableException
	BufferedReader	ObjectInput
	BufferedWriter	ObjectInputStream
	ByteArrayInputStream	ObjectInputValidation
	ByteArrayOutputStream	ObjectOutput

Package	Classes	
	CharArrayReader	ObjectOutputStream
	CharArrayWriter	ObjectStreamClass
	CharConversionException	ObjectStreamConstants
	DataInput	ObjectStreamException
	DataInputStream	ObjectStreamField
	DataOutput	OptionalDataException
	DataOutputStream	OutputStream
	EOFException	OutputStreamWriter
	Externalizable	PipedInputStream
	File	PipedOutputStream
	FileDescriptor	PipedReader
	FileFilter	PipedWriter
	FileInputStream	PrintStream
	FilenameFilter	PrintWriter
	FileNotFoundException	PushbackInputStream
	FileOutputStream	PushbackReader
	FilePermission	RandomAccessFile
	FileReader	Reader
	FileWriter	SequenceInputStream
	FilterInputStream	Serializable
	FilterOutputStream	SerializablePermission
	FilterReader	StreamCorruptedException
	FilterWriter	StreamTokenizer
	InputStream	StringReader
	InputStreamReader	StringWriter
	InterruptedIOException	SyncFailedException
	InvalidClassException	UnsupportedEncodingException
	InvalidObjectException	UTFDataFormatException
	IOException	WriteAbortedException
	LineNumberReader	Writer
java.math	BigInteger	
java.net	Authenticator	ProtocolException
	BindException	ServerSocket
	ConnectException	Socket
	ContentHandler	SocketException
	ContentHandlerFactory	SocketImpl
	DatagramPacket	SocketImplFactory
	DatagramSocket	SocketOptions
	DatagramSocketImpl	SocketPermission
	DatagramSocketImplFactory	UnknownHostException

Package	Classes	
	FileNameMap	UnknownServiceException
	HttpURLConnection	URL
	InetAddress	URLClassLoader
	JarURLConnection	URLConnection
	MalformedURLException	URLDecoder
	MulticastSocket	URLEncoder
	NetPermission	URLStreamHandler
	NoRouteToHostException	URLStreamHandlerFactory
	PasswordAuthentication	
java.rmi	AccessException	Remote
	AlreadyBoundException	RemoteException
	NotBoundException	UnexpectedException
	registry.Registry	
java.security	AccessControlContext	KeyStoreException
	AccessControlException	KeyStoreSpi
	AccessController	MessageDigest
	AlgorithmParameterGenerator	MessageDigestSpi
	AlgorithmParameterGeneratorSpi	NoSuchAlgorithmException
	AlgorithmParameters	NoSuchProviderException
	AlgorithmParametersSpi	Permission
	AllPermission	PermissionCollection
	BasicPermission	Permissions
	Certificate	Policy
	CodeSource	Principal
	DigestException	PrivateKey
	DigestInputStream	PrivilegedAction
	DigestOutputStream	PrivilegedActionException
	DomainCombiner	PrivilegedExceptionAction
	GeneralSecurityException	ProtectionDomain
	Guard	Provider
	GuardedObject	ProviderException
	Identity	PublicKey
	IdentityScope	SecureClassLoader
	InvalidAlgorithmParameterException	SecureRandom
	InvalidKeyException	SecureRandomSpi
	InvalidParameterException	Security
	Key	SecurityPermission
	KeyException	Signature
	KeyFactory	SignatureException
	KeyFactorySpi	SignatureSpi

Package	Classes	
	KeyManagementException	SignedObject
	KeyPair	Signer
	KeyPairGenerator	UnrecoverableKeyException
	KeyPairGeneratorSpi	UnresolvedPermission
	KeyStore	
java.security.acl	Acl	LastOwnerException
	AclEntry	NotOwnerException
	AclNotFoundException	Owner
	Group	Permission
java.security.cert	Certificate	CertificateParsingException
	CertificateEncodingException	CRL
	CertificateException	CRLException
	CertificateExpiredException	X509Certificate
	CertificateFactory	X509CRL
	CertificateFactorySpi	X509CRLEntry
	CertificateNotYetValidException	X509Extension
java.security.interfaces	DSAKey	RSAKey
	DSAKeyPairGenerator	RSAPrivateCrtKey
	DSAParams	RSAPrivateKey
	DSAPrivateKey	RSAPublicKey
	DSAPublicKey	
java.security.spec	AlgorithmParameterSpec	KeySpec
	DSAParameterSpec	PKCS8EncodedKeySpec
	DSAPrivateKeySpec	RSAKeyGenParameterSpec
	DSAPublicKeySpec	RSAPrivateCrtKeySpec
	EncodedKeySpec	RSAPrivateKeySpec
	InvalidKeySpecException	RSAPublicKeySpec
	InvalidParameterSpecException	X509EncodedKeySpec
java.text	Annotation	DecimalFormat
	AttributedCharacterIterator	DecimalFormatSymbols
	AttributedString	FieldPosition
	BreakIterator	Format
	CharacterIterator	MessageFormat
	ChoiceFormat	NumberFormat
	CollationElementIterator	ParseException
	CollationKey	ParsePosition
	Collator	RuleBasedCollator
	DateFormat	SimpleDateFormat
	DateFormatSymbols	StringCharacterIterator

Package	Classes	
java.util	AbstractCollection	ListResourceBundle
	AbstractList	Locale
	AbstractMap	Map
	AbstractSequentialList	MissingResourceException
	AbstractSet	NoSuchElementException
	ArrayList	Observable
	Arrays	Observer
	BitSet	Properties
	Calendar	PropertyPermission
	Collection	PropertyResourceBundle
	Collections	Random
	Comparator	ResourceBundle
	ConcurrentModificationException	Set
	Date	SimpleTimeZone
	Dictionary	SortedMap
	EmptyStackException	SortedSet
	Enumeration	Stack
	EventListener	StringTokenizer
	EventObject	Timer
	GregorianCalendar	TimerTask
	HashMap	TimeZone
	HashSet	TooManyListenersException
	Hashtable	TreeMap
	Iterator	TreeSet
	LinkedList	Vector
	List	WeakHashMap
	ListIterator	
java.util.jar	Attributes	JarInputStream
	JarEntry	JarOutputStream
	JarException	Manifest
	JarFile	
java.util.zip	Adler32	GZIPOutputStream
	CheckedInputStream	Inflater
	CheckedOutputStream	InflaterInputStream
	Checksum	ZipConstants
	CRC32	ZipEntry
	DataFormatException	ZipException
	Deflater	ZipFile
	DeflaterOutputStream	ZipInputStream
	GZIPInputStream	ZipOutputStream

Package	Classes	
javax.microedtion.io	Connection	HttpConnection
	ConnectionNotFoundException	InputConnection
	Connector	OutputConnection
	ContentConnection	StreamConnection
	Datagram	StreamConnectionNotifier
	DatagramConnection	
javax.microedition.xlet.ixc	IxcPermission	StubException
	IxcRegistry	
javax.microedition.xlet	UnavailableContainerException	XletContext
	Xlet	XletStateChangeException

J.2. JSSE 1.0.2

Package	Classes	
javax.net	ServerSocketFactory	SocketFactory
javax.net.ssl	HandshakeCompletedEvent	SSLServerSocketFactory
	HandshakeCompletedListener	SSLSession
	SSLException	SSLSessionBindingEvent
	SSLHandshakeException	SSLSessionBindingListener
	SSLKeyException	SSLSessionContext
	SSLPeerUnverifiedException	SSLSocket
	SSLProtocolException	SSLSocketFactory
	SSLServerSocket	
javax.security.cert	Certificate	CertificateNotYetValidException
	CertificateEncodingException	CertificateParsingException
	CertificateException	X509Certificate
	CertificateExpiredException	

J.3. JMF 1.0

Package	Classes	
javax.media	Clock	Manager
	ClockStartedError	MediaError
	ClockStoppedException	MediaEvent
	ConnectionErrorEvent	MediaException
	Control	MediaHandler
	Controller	MediaLocator
	ControllerClosedEvent	MediaProxy
	ControllerErrorEvent	MediaTimeSetEvent

Package	Classes	
	ControllerEvent	RateChangeEvent
	ControllerListener	RealizeCompleteEvent
	DataStarvedEvent	ResourceUnavailableEvent
	DeallocateEvent	RestartingEvent
	Duration	StartEvent
	DurationUpdateEvent	StopAtTimeEvent
	EndOfMediaEvent	StopByRequestEvent
	GainChangeEvent	StopEvent
	GainChangeListener	StopTimeChangeEvent
	GainControl	StopTimeSetError
	IncompatibleSourceException	Time
	IncompatibleTimeBaseException	TimeBase
	InternalErrorEvent	TransitionEvent
javax.media.protocol	ContentDescriptor	RateConfiguration
	Controls	RateConfigureable
	DataSource	RateRange
	Positionable	Seekable
	PullDataSource	SourceStream
	PullSourceStream	SourceTransferHandler
	PushDataSource	URLDataSource
	PushSourceStream	

J.4. Java TV 1.1

Package	Classes	
javax.tv.graphics	AlphaColor	TVContainer
javax.tv.locator	InvalidLocatorException	LocatorFactory
	Locator	MalformedLocatorException
javax.tv.media	AWTVideoSize	MediaSelectFailedEvent
	AWTVideoSizeControl	MediaSelectListener
	MediaSelectCARefusedEvent	MediaSelectPermission
	MediaSelectControl	MediaSelectSucceededEvent
	MediaSelectEvent	
javax.tv.net	InterfaceMap	
javax.tv.service.guide	ContentRatingAdvisory	ProgramScheduleChangeType
	ProgramEvent	ProgramScheduleEvent
	ProgramEventDescription	ProgramScheduleListener
	ProgramSchedule	
javax.tv.service.navigation	CAIdentification	ServiceDetails

Package	Classes	
	DeliverySystemType	ServiceDetailsSIChangeEvent
	FavoriteServicesName	ServiceFilter
	FilterNotSupportedException	ServiceIterator
	LocatorFilter	ServiceList
	PreferenceFilter	ServiceProviderInformation
	ServiceComponent	ServiceTypeFilter
	ServiceComponentChangeEvent	SIElementFilter
	ServiceComponentChangeListen	SortNotAvailableException
	er	StreamType
	ServiceDescription	
javax.tv.service.selection	AlternativeContentEvent	ServiceContext
	InsufficientResourcesException	ServiceContextDestroyedEvent
	InvalidServiceComponentExcepti	ServiceContextEvent
	on	ServiceContextException
	NormalContentEvent	ServiceContextFactory
	PresentationChangedEvent	ServiceContextListener
	PresentationTerminatedEvent	ServiceContextPermission
	SelectionFailedEvent	ServiceMediaHandler
	SelectPermission	
	ServiceContentHandler	
javax.tv.service	RatingDimension	SIChangeType
	ReadPermission	SIElement
	Service	SIException
	ServiceInformationType	SIManager
	ServiceMinorNumber	SIRequest
	ServiceNumber	SIRequestFailureType
	ServiceType	SIRequestor
	SIChangeEvent	SIRetrievable
	SIChangeListener	
javax.tv.service.transport	Network	Transport
	NetworkChangeEvent	TransportSIChangeEvent
	NetworkChangeListener	TransportStream
	NetworkCollection	TransportStreamChangeEvent
	ServiceDetailsChangeEvent	TransportStreamChangeListener
	ServiceDetailsChangeListener	TransportStreamCollection
javax.tv.util	TVTimer	TVTimerSpec
	TVTimerScheduleFailedExceptio	TVTimerWentOffEvent
	n	TVTimerWentOffListener
javax.tv.xlet	Xlet	XletStateChangeException

Package	Classes
	XletContext

J.5. DVB-MHP

본 절에서 나열한 API 리스트는 MHP 1.0.3('ETSI TS 101 812 V1.3. 1')에 정의하고 있는 API 리스트 중 본 표준에서 참조하고 있는 API 만 나타낸 것이다. 따라서, 본 표준에 만족하는 애플리케이션을 제작하기 위해서는 DVB-MHP API 리스트 중 본 절에서 나열한 API 범위에서만 사용하여야 한다.

Package	Classes	
org.dvb.application	AppsDatabase	AppStateChangeEvent
	AppsDatabaseFilter	AppsDatabaseEventListener
	AppAttributes	DVBJProxy
	AppsControlPermission	RunningApplicationsFilter
	AppIcon	AppStateChangeEventListener
	AppID	AppProxy
	LanguageNotAvailableException	AppsDatabaseEvent
	CurrentServiceFilter	IllegalProfileParameterException
org.dvb.io.ixc	IxcRegistry	
org.dvb.dsmcc	AsynchronousLoadingEvent	NotLoadedException
	AsynchronousLoadingEventListe	NPTDiscontinuityEvent
	ner	NPTListener
	DSMCCException	NPTPresentEvent
	DSMCCObject	NPTRate
	DSMCCStream	NPTRateChangeEvent
	DSMCCStreamEvent	NPTRemovedEvent
	IllegalObjectTypeException	NPTStatusEvent
	InsufficientResourcesEvent	ObjectChangeEvent
	InsufficientResourcesException	ObjectChangeEventListener
	InvalidAddressException	ServerDeliveryErrorEvent
	InvalidFormatEvent	ServerDeliveryException
	InvalidFormatException	ServiceDomain
	InvalidPathnameEvent	ServiceXFRErrorEvent
	InvalidPathNameException	ServiceXFRException
	LoadingAbortedEvent	ServiceXFRReference
	MPEGDeliveryErrorEvent	StreamEvent
	MPEGDeliveryException	StreamEventListener
	NotEntitledEvent	SuccessEvent

Package	Classes	
	NotEntitledException	UnknownEventException
	NothingToAbortException	
org.dvb.event	EventManager	UserEventAvailableEvent
	OverallRepository	UserEventListener
	RepositoryDescriptor	UserEventRepository
	UserEvent	UserEventUnavailableEvent
org.dvb.io.persistent	FileAccessPermissions	FileAttributes
org.dvb.lang	DVBClassLoader	
org.dvb.media	ActiveFormatDescriptionChange	PresentationChangedEvent
	dEvent	ServiceRemovedEvent
	AspectRatioChangedEvent	StopByResourceLossEvent
	BackgroundVideoPresentationC	VideoFormatControl
	ontrol	VideoFormatEvent
	DFCChangedEvent	VideoFormatListener
	DripFeedDataSource	VideoPresentationControl
	DripFeedPermission	VideoTransformation
	NoComponentSelectedEvent	
org.dvb.net	DatagramSocketBufferControl	
org.dvb.net.rc	ConnectionEstablishedEvent	IncompleteTargetException
	ConnectionFailedEvent	PermissionDeniedException
	ConnectionListener	RCInterface
	ConnectionParameters	RCInterfaceManager
	ConnectionRCEvent	RCInterfaceReleasedEvent
	ConnectionRCInterface	RCInterfaceReservedEvent
	ConnectionTerminatedEvent	RCPermission
org.dvb.ui	DVBAlphaComposite	FontFormatException
	DVBBufferedImage	FontNotAvailableException
	DVBColor	TestOpacity
	DVBGraphics	TextOverflowListener
	DVBRasterFormatException	UnsupportedDrawingOperationE
	DVBTextLayoutManager	xception
	FontFactory	
org.dvb.user	Facility	UserPreferenceChangeEvent
	GeneralPreference	UserPreferenceChangeListener
	Preference	UserPreferenceManager
	UnsupportedPreferenceExceptio	UserPreferencePermission
	n	

J.6. DAVIC 1.4.1p9

Package	Classes	
org.davic.media	AudioLanguageControl LanguageControl LanguageNotAvailableException MediaLocator MediaPresentedEvent MediaTimePositionChangedEvent	MediaTimePositionControl NotAuthorizedException NotAuthorizedMediaException ResourceReturnedEvent ResourceWithdrawnEvent
org.davic.mpeg	ApplicationOrigin ElementaryStream NotAuthorizedException NotAuthorizedInterface ObjectUnavailableException	ResourceException Service TransportStream TuningException
org.davic.mpeg.sections	ConnectionLostException EndOfFilteringEvent FilteringInterruptedException FilterResourceException FilterResourcesAvailableEvent ForcedDisconnectedEvent IllegalFilterDefinitionException IncompleteFilteringEvent InvalidSourceException NoDataAvailableException RingSectionFilter	Section SectionAvailableEvent SectionFilter SectionFilterEvent SectionFilterException SectionFilterGroup SectionFilterListener SimpleSectionFilter TableSectionFilter TimeOutEvent VersionChangeDetectedEvent
org.davic.net	InvalidLocatorException Locator	TransportDependentLocator
org.davic.net.tuning	DeliverySystemType IncorrectLocatorException NetworkInterface NetworkInterfaceController NetworkInterfaceEvent NetworkInterfaceException NetworkInterfaceListener NetworkInterfaceManager NetworkInterfaceReleasedEvent	NetworkInterfaceReservedEvent NetworkInterfaceTuningEvent NetworkInterfaceTuningOverEvent NoFreeInterfaceException NotOwnerException NotTunedException StreamNotFoundException StreamTable
org.davic.resources	ResourceClient ResourceProxy ResourceServer	ResourceStatusEvent ResourceStatusListener

J.7. HAVi 1.1

Package	Classes	
org.havi.ui.event	HActionEvent	HKeyEvent
	HActionListener	HKeyListener
	HAdjustmentEvent	HMouseCapabilities
	HAdjustmentListener	HRcCapabilities
	HBackgroundImageEvent	HRcEvent
	HBackgroundImageListener	HScreenConfigurationEvent
	HEventGroup	HScreenConfigurationListener
	HEventRepresentation	HScreenDeviceReleasedEvent
	HFocusEvent	HScreenDeviceReservedEvent
	HFocusListener	HScreenLocationModifiedEvent
	HItemEvent	HScreenLocationModifiedListener
	HItemListener	HTextEvent
	HKeyCapabilities	HTextListener
org.havi.ui	HActionable	HMultilineEntry
	HActionInputPreferred	HMultilineEntryLook
	HAdjustableLook	HNavigable
	HAdjustmentInputPreferred	HNavigationInputPreferred
	HAdjustmentValue	HNoInputPreferred
	HAnimateEffect	HOrientable
	HAnimateLook	HPermissionDeniedException
	HAnimation	HRange
	HBackgroundConfigTemplate	HRangeLook
	HBackgroundConfiguration	HRangeValue
	HBackgroundDevice	HScene
	HBackgroundImage	HSceneFactory
	HChangeData	HSceneTemplate
	HComponent	HScreen
	HComponentOrdering	HScreenConfigTemplate
	HConfigurationException	HScreenConfiguration
	HContainer	HScreenDevice
	HDefaultTextLayoutManager	HScreenDimension
	HEmulatedGraphicsConfiguratio	HScreenPoint
	n	HScreenRectangle
	HEmulatedGraphicsDevice	HSelectionInputPreferred
	HEventMulticaster	HSinglelineEntry
	HExtendedLook	HSinglelineEntryLook
	HFlatEffectMatte	HSound
	HFlatMatte	HState

Package	Classes	
	HFontCapabilities	HStaticAnimation
	HGraphicButton	HStaticIcon
	HGraphicLook	HStaticRange
	HGraphicsConfigTemplate	HStaticText
	HGraphicsConfiguration	HStillImageBackgroundConfiguration
	HGraphicsDevice	HSwitchable
	HIcon	HText
	HImageEffectMatte	HTextButton
	HImageHints	HTextLayoutManager
	HImageMatte	HTextLook
	HInvalidLookException	HTextValue
	HItemValue	HToggleButton
	HKeyboardInputPreferred	HToggleGroup
	HListElement	HUIException
	HListGroup	HVersion
	HListGroupLook	HVideoComponent
	HLook	HVideoConfigTemplate
	HMatte	HVideoConfiguration
	HMatteException	HVideoDevice
	HMatteLayer	HVisible

J.8. OCAP 1.0

아래 나열한 API 는 OCAP 1.0 에 정의된 API 로써 본 절에 기술된 API 만 OCAP 에서 참조하여 사용하도록 정하고 있다.

Package	Classes	
org.ocap.storage	StorageManager	DetachableStorageOption
	StorageManagerListener	StorageOption
	StorageManagerEvent	StorageProxy
	LogicalStorageVolume	ExtendedFileAccessPermissions
	RemovableStorageOption	
org.ocap.media	ClosedCaptioningListener	ClosedCaptioningControl
	ClosedCaptioningEvent	
org.ocap.net	OcapLocator	

J.9. ACAP

Package	Classes
org.acap.test	ACAPTest

부 록 I

관련 문헌

- ANSI/SCTE 90-1, 'SCTE Application Platform Standard OCAP 1.0 Profile', 2005.
- ATSC, A/90, 'ATSC Data Broadcasting Standard', 2003. 7. 20.
- Cablelabs, OC-SP-OCAP1.0.0-070814, 'OpenCableTM Application Platform Specifications OCAP 1.0 Profile'
- ETSI TS 101 812 V1.3.1, 'DVB Multimedia Home Platform 1.0.3', 2003.06.
- ETSI TS 102 812 V1.2.1, 'DVB Multimedia Home Platform 1.1.1', 2003.06.
- ETSI TS 102 819 V1.4.1, 'DVB Globally Executable MHP version 1.0.3', 2008. 05.
- JSR-217, 'Personal Basis Profile 1.1', http://www.jcp.org/
- JSR-218, 'Connected Device Configuration 1.1', http://www.jcp.org/
- JSR-927, 'JavaTM TV API 1.1', http://www.jcp.org/
- TTAK.KO-07.0014/R2, '지상파 디지털 TV방송 송수신 정합표준', 2009.06.
- TTAS.KO-12.0004/R1, '128비트 블록암호알고리즘 SEED', 2005.12.

방송통신표준

지상파 데이터 방송
(Standard for Terrestrial Data Broadcasting)

발행인 : 미래창조과학부 장관

발행처 : 미래창조과학부 국립전파연구원

140-848, 서울 용산구 원효로41길 29

발행일 : 2013.12.

국립전파연구원 고시 제 2013-20호