

**FED5.0 Packager(JAVA)**

**개발자 매뉴얼**

|  |  |  |  |
| --- | --- | --- | --- |
| **버전** | **내용** | **작성자** | **날짜** |
| **1.0** | 최초 작성 | 김지홍 | 2016.04.05 |
| **1.1** | Stream 관련 인터페이스, FSS 설정 관련 인터페이스 추가 | 김지홍 | 2016.09.30 |
| **1.2** | 권한관련, 서버전송 관련 인터페이스 설명 추가 | 김지홍 | 2017.01.03 |
| **1.3** | SetUsageLogInfo 추가 | 김지홍 | 2017.08.04 |
|  |  |  |  |
|  |  |  |  |

목 차

[Specification of WorkPackager 14](#_Toc489617659)

[DoPackaging 17](#_Toc489617660)

[Description 17](#_Toc489617661)

[Syntax 17](#_Toc489617662)

[Parameters 17](#_Toc489617663)

[Return Values 17](#_Toc489617664)

[Example 18](#_Toc489617665)

[DoPackagingFsn2 19](#_Toc489617666)

[Description 19](#_Toc489617667)

[Syntax 19](#_Toc489617668)

[Parameters 19](#_Toc489617669)

[Return Values 20](#_Toc489617670)

[Example 20](#_Toc489617671)

[DoExtract 21](#_Toc489617672)

[Description 21](#_Toc489617673)

[Syntax 21](#_Toc489617674)

[Parameters 21](#_Toc489617675)

[Return Values 21](#_Toc489617676)

[Example 22](#_Toc489617677)

[DoExtract\_s 23](#_Toc489617678)

[Description 23](#_Toc489617679)

[Syntax 23](#_Toc489617680)

[Parameters 23](#_Toc489617681)

[Return Values 23](#_Toc489617682)

[Example 24](#_Toc489617683)

[DoExtract\_c 25](#_Toc489617684)

[Description 25](#_Toc489617685)

[Syntax 25](#_Toc489617686)

[Parameters 25](#_Toc489617687)

[Return Values 25](#_Toc489617688)

[Example 26](#_Toc489617689)

[GetFileType 27](#_Toc489617690)

[Description 27](#_Toc489617691)

[Syntax 27](#_Toc489617692)

[Parameters 27](#_Toc489617693)

[Return Values 27](#_Toc489617694)

[Example 27](#_Toc489617695)

[IsPackageFile 28](#_Toc489617696)

[Description 28](#_Toc489617697)

[Syntax 28](#_Toc489617698)

[Parameters 28](#_Toc489617699)

[Return Values 28](#_Toc489617700)

[Example 28](#_Toc489617701)

[IsSupportFile 29](#_Toc489617702)

[Description 29](#_Toc489617703)

[Syntax 29](#_Toc489617704)

[Parameters 29](#_Toc489617705)

[Return Values 29](#_Toc489617706)

[Example 30](#_Toc489617707)

[GetFileHeader 31](#_Toc489617708)

[Description 31](#_Toc489617709)

[Syntax 31](#_Toc489617710)

[Parameters 31](#_Toc489617711)

[Return Values 31](#_Toc489617712)

[Example 32](#_Toc489617713)

[getLastErrorStr 33](#_Toc489617714)

[Description 33](#_Toc489617715)

[Syntax 33](#_Toc489617716)

[Parameters 33](#_Toc489617717)

[Return Values 33](#_Toc489617718)

[getLastErrorNum 34](#_Toc489617719)

[Description 34](#_Toc489617720)

[Syntax 34](#_Toc489617721)

[Parameters 34](#_Toc489617722)

[Return Values 34](#_Toc489617723)

[setCharset 35](#_Toc489617724)

[Description 35](#_Toc489617725)

[Syntax 35](#_Toc489617726)

[Parameters 35](#_Toc489617727)

[Return Values 35](#_Toc489617728)

[Example 35](#_Toc489617729)

[setPackagerType 36](#_Toc489617730)

[Description 36](#_Toc489617731)

[Syntax 36](#_Toc489617732)

[Parameters 36](#_Toc489617733)

[Return Values 36](#_Toc489617734)

[getPackagerType 37](#_Toc489617735)

[Description 37](#_Toc489617736)

[Syntax 37](#_Toc489617737)

[Parameters 37](#_Toc489617738)

[Return Values 37](#_Toc489617739)

[setOverWriteFlag 38](#_Toc489617740)

[Description 38](#_Toc489617741)

[Syntax 38](#_Toc489617742)

[Parameters 38](#_Toc489617743)

[Return Values 38](#_Toc489617744)

[setIsTemp 39](#_Toc489617745)

[Description 39](#_Toc489617746)

[Syntax 39](#_Toc489617747)

[Parameters 39](#_Toc489617748)

[Return Values 39](#_Toc489617749)

[getContainerFileName 40](#_Toc489617750)

[Description 40](#_Toc489617751)

[Syntax 40](#_Toc489617752)

[Parameters 40](#_Toc489617753)

[Return Values 40](#_Toc489617754)

[AddRightsFsn 41](#_Toc489617755)

[Description 41](#_Toc489617756)

[Syntax 41](#_Toc489617757)

[Parameters 41](#_Toc489617758)

[Return Values 41](#_Toc489617759)

[<참고> 41](#_Toc489617760)

[Example 42](#_Toc489617761)

[DeleteRightsFsn 42](#_Toc489617762)

[Description 42](#_Toc489617763)

[Syntax 43](#_Toc489617764)

[Parameters 43](#_Toc489617765)

[Return Values 43](#_Toc489617766)

[Example 43](#_Toc489617767)

[DeleteAllRightsFsn 45](#_Toc489617768)

[Description 45](#_Toc489617769)

[Syntax 45](#_Toc489617770)

[Parameters 45](#_Toc489617771)

[Return Values 45](#_Toc489617772)

[Example 46](#_Toc489617773)

[ReadRightsFsn 47](#_Toc489617774)

[Description 47](#_Toc489617775)

[Syntax 47](#_Toc489617776)

[Parameters 47](#_Toc489617777)

[Return Values 47](#_Toc489617778)

[<참고> 47](#_Toc489617779)

[Example 48](#_Toc489617780)

[ReadAllRightsFsn 49](#_Toc489617781)

[Description 49](#_Toc489617782)

[Syntax 49](#_Toc489617783)

[Parameters 49](#_Toc489617784)

[Return Values 49](#_Toc489617785)

[<참고> 49](#_Toc489617786)

[Example 50](#_Toc489617787)

[RevokeUsage4Fse 51](#_Toc489617788)

[Description 51](#_Toc489617789)

[Syntax 51](#_Toc489617790)

[Parameters 51](#_Toc489617791)

[Return Values 51](#_Toc489617792)

[Example 51](#_Toc489617793)

[RestoreUsage4Fse 52](#_Toc489617794)

[Description 52](#_Toc489617795)

[Syntax 52](#_Toc489617796)

[Parameters 52](#_Toc489617797)

[Return Values 52](#_Toc489617798)

[Example 52](#_Toc489617799)

[RegisterUsage4Fse 53](#_Toc489617800)

[Description 53](#_Toc489617801)

[Syntax 53](#_Toc489617802)

[Parameters 53](#_Toc489617803)

[Return Values 53](#_Toc489617804)

[Example 53](#_Toc489617805)

[RevokeUsageReaders4Fse 54](#_Toc489617806)

[Description 54](#_Toc489617807)

[Syntax 54](#_Toc489617808)

[Parameters 54](#_Toc489617809)

[Return Values 54](#_Toc489617810)

[Example 54](#_Toc489617811)

[RestoreUsageReaders4Fse 55](#_Toc489617812)

[Description 55](#_Toc489617813)

[Syntax 55](#_Toc489617814)

[Parameters 55](#_Toc489617815)

[Return Values 55](#_Toc489617816)

[Example 55](#_Toc489617817)

[SetUsageLogInfo 56](#_Toc489617818)

[Description 56](#_Toc489617819)

[Syntax 56](#_Toc489617820)

[Parameter 56](#_Toc489617821)

[Return Value 56](#_Toc489617822)

[Example 56](#_Toc489617823)

[Specification of WPackager 57](#_Toc489617824)

[setSender 59](#_Toc489617825)

[Description 59](#_Toc489617826)

[Syntax 59](#_Toc489617827)

[Parameters 59](#_Toc489617828)

[Return Values 59](#_Toc489617829)

[setReader 60](#_Toc489617830)

[Description 60](#_Toc489617831)

[Syntax 60](#_Toc489617832)

[Parameters 60](#_Toc489617833)

[Return Values 60](#_Toc489617834)

[addReader 61](#_Toc489617835)

[Description 61](#_Toc489617836)

[Syntax 61](#_Toc489617837)

[Parameters 61](#_Toc489617838)

[Return Values 61](#_Toc489617839)

[setCharset 62](#_Toc489617840)

[Description 62](#_Toc489617841)

[Syntax 62](#_Toc489617842)

[Parameters 62](#_Toc489617843)

[Return Values 62](#_Toc489617844)

[setPackagerType 63](#_Toc489617845)

[Description 63](#_Toc489617846)

[Syntax 63](#_Toc489617847)

[Parameters 63](#_Toc489617848)

[Return Values 63](#_Toc489617849)

[setOverWriteFlag 64](#_Toc489617850)

[Description 64](#_Toc489617851)

[Syntax 64](#_Toc489617852)

[Parameters 64](#_Toc489617853)

[Return Values 64](#_Toc489617854)

[getContentID 65](#_Toc489617855)

[Description 65](#_Toc489617856)

[Syntax 65](#_Toc489617857)

[Parameters 65](#_Toc489617858)

[Return Values 65](#_Toc489617859)

[setOptionLicenseType 66](#_Toc489617860)

[Description 66](#_Toc489617861)

[Syntax 66](#_Toc489617862)

[Parameters 66](#_Toc489617863)

[Return Values 66](#_Toc489617864)

[setOptionStartDate 67](#_Toc489617865)

[Description 67](#_Toc489617866)

[Syntax 67](#_Toc489617867)

[Parameters 67](#_Toc489617868)

[Return Values 67](#_Toc489617869)

[setOptionEndDate 68](#_Toc489617870)

[Description 68](#_Toc489617871)

[Syntax 68](#_Toc489617872)

[Parameters 68](#_Toc489617873)

[Return Values 68](#_Toc489617874)

[setOptionView 69](#_Toc489617875)

[Description 69](#_Toc489617876)

[Syntax 69](#_Toc489617877)

[Parameters 69](#_Toc489617878)

[Return Values 69](#_Toc489617879)

[setOptionPrint 70](#_Toc489617880)

[Description 70](#_Toc489617881)

[Syntax 70](#_Toc489617882)

[Parameters 70](#_Toc489617883)

[Return Values 70](#_Toc489617884)

[setOptionPrintScreen 71](#_Toc489617885)

[Description 71](#_Toc489617886)

[Syntax 71](#_Toc489617887)

[Parameters 71](#_Toc489617888)

[Return Values 71](#_Toc489617889)

[setOptionSave 72](#_Toc489617890)

[Description 72](#_Toc489617891)

[Syntax 72](#_Toc489617892)

[Parameters 72](#_Toc489617893)

[Return Values 72](#_Toc489617894)

[setOptionSecureSave 73](#_Toc489617895)

[Description 73](#_Toc489617896)

[Syntax 73](#_Toc489617897)

[Parameters 73](#_Toc489617898)

[Return Values 73](#_Toc489617899)

[setOptionCount 74](#_Toc489617900)

[Description 74](#_Toc489617901)

[Syntax 74](#_Toc489617902)

[Parameters 74](#_Toc489617903)

[Return Values 74](#_Toc489617904)

[setOptionPcCount 75](#_Toc489617905)

[Description 75](#_Toc489617906)

[Syntax 75](#_Toc489617907)

[Parameters 75](#_Toc489617908)

[Return Values 75](#_Toc489617909)

[setOptionConfirmMail 76](#_Toc489617910)

[Description 76](#_Toc489617911)

[Syntax 76](#_Toc489617912)

[Parameters 76](#_Toc489617913)

[Return Values 76](#_Toc489617914)

[setOptionConfirmMail4In 77](#_Toc489617915)

[Description 77](#_Toc489617916)

[Syntax 77](#_Toc489617917)

[Parameters 77](#_Toc489617918)

[Return Values 77](#_Toc489617919)

[setOptionTransfer 78](#_Toc489617920)

[Description 78](#_Toc489617921)

[Syntax 78](#_Toc489617922)

[Parameters 78](#_Toc489617923)

[Return Values 78](#_Toc489617924)

[setOptionPeriod 79](#_Toc489617925)

[Description 79](#_Toc489617926)

[Syntax 79](#_Toc489617927)

[Parameters 79](#_Toc489617928)

[Return Values 79](#_Toc489617929)

[setOptionAuthentication 80](#_Toc489617930)

[Description 80](#_Toc489617931)

[Syntax 80](#_Toc489617932)

[Parameters 80](#_Toc489617933)

[Return Values 80](#_Toc489617934)

[setOptionStr 81](#_Toc489617935)

[Description 81](#_Toc489617936)

[Syntax 81](#_Toc489617937)

[Parameters 81](#_Toc489617938)

[Return Values 81](#_Toc489617939)

[setIsTemp 83](#_Toc489617940)

[Description 83](#_Toc489617941)

[Syntax 83](#_Toc489617942)

[Parameters 83](#_Toc489617943)

[Return Values 83](#_Toc489617944)

[setSenderRight 84](#_Toc489617945)

[Description 84](#_Toc489617946)

[Syntax 84](#_Toc489617947)

[Parameters 84](#_Toc489617948)

[Return Values 84](#_Toc489617949)

[setFssStateOn 85](#_Toc489617950)

[Description 85](#_Toc489617951)

[Syntax 85](#_Toc489617952)

[Return Values 85](#_Toc489617953)

[DoPackagingFSE 86](#_Toc489617954)

[Description 86](#_Toc489617955)

[Syntax 86](#_Toc489617956)

[Parameters 86](#_Toc489617957)

[Return Values 86](#_Toc489617958)

[DoPackagingFSE2 87](#_Toc489617959)

[Description 87](#_Toc489617960)

[Syntax 87](#_Toc489617961)

[Parameters 87](#_Toc489617962)

[Return Values 87](#_Toc489617963)

[getContainerFileName 88](#_Toc489617964)

[Description 88](#_Toc489617965)

[Syntax 88](#_Toc489617966)

[Parameters 88](#_Toc489617967)

[Return Values 88](#_Toc489617968)

[getContainerFilePathName 89](#_Toc489617969)

[Description 89](#_Toc489617970)

[Syntax 89](#_Toc489617971)

[Parameters 89](#_Toc489617972)

[Return Values 89](#_Toc489617973)

[getLastErrorStr 90](#_Toc489617974)

[Description 90](#_Toc489617975)

[Syntax 90](#_Toc489617976)

[Parameters 90](#_Toc489617977)

[Return Values 90](#_Toc489617978)

[getLastErrorNum 91](#_Toc489617979)

[Description 91](#_Toc489617980)

[Syntax 91](#_Toc489617981)

[Parameters 91](#_Toc489617982)

[Return Values 91](#_Toc489617983)

[[SAMPLE CODE] 92](#_Toc489617984)

[Specification of StreamPackager 93](#_Toc489617985)

[SetHomeDir 94](#_Toc489617986)

[Description 94](#_Toc489617987)

[Syntax 94](#_Toc489617988)

[Parameters 94](#_Toc489617989)

[Return Values 94](#_Toc489617990)

[SetServerID 95](#_Toc489617991)

[Description 95](#_Toc489617992)

[Syntax 95](#_Toc489617993)

[Parameters 95](#_Toc489617994)

[Return Values 95](#_Toc489617995)

[StrmPackagingFSDInit 96](#_Toc489617996)

[Description 96](#_Toc489617997)

[Syntax 96](#_Toc489617998)

[Return Values 96](#_Toc489617999)

[StrmPackagingFSNInit 97](#_Toc489618000)

[Description 97](#_Toc489618001)

[Syntax 97](#_Toc489618002)

[Return Values 97](#_Toc489618003)

[StrmPackagingUpdate 98](#_Toc489618004)

[Description 98](#_Toc489618005)

[Syntax 98](#_Toc489618006)

[Parameters 98](#_Toc489618007)

[Return Values 98](#_Toc489618008)

[StrmPackagingFinal 99](#_Toc489618009)

[Description 99](#_Toc489618010)

[Syntax 99](#_Toc489618011)

[Parameters 99](#_Toc489618012)

[Return Values 99](#_Toc489618013)

[StrmExtractInit() 100](#_Toc489618014)

[Description 100](#_Toc489618015)

[Syntax 100](#_Toc489618016)

[Return Values 100](#_Toc489618017)

[StrmExtractUpdate 101](#_Toc489618018)

[Description 101](#_Toc489618019)

[Syntax 101](#_Toc489618020)

[Parameters 101](#_Toc489618021)

[Return Values 101](#_Toc489618022)

[StrmExtractFinal 102](#_Toc489618023)

[Description 102](#_Toc489618024)

[Syntax 102](#_Toc489618025)

[Parameters 102](#_Toc489618026)

[Return Values 102](#_Toc489618027)

[setStrmHeader 103](#_Toc489618028)

[Description 103](#_Toc489618029)

[Syntax 103](#_Toc489618030)

[Parameters 103](#_Toc489618031)

[Return Values 103](#_Toc489618032)

[[SAMPLE CODE] 104](#_Toc489618033)

[Specification of Version 112](#_Toc489618034)

[More Information 113](#_Toc489618035)

[Error Code 참조 (FED4.0, FED5.0 공통) 113](#_Toc489618036)

[Error Code 참조 (FED5.0 전용) 113](#_Toc489618037)

[FSE Option String 참조 114](#_Toc489618038)

# Specification of WorkPackager

* **CLASS**: com.fasoo.fcwpkg.packager.WorkPackager
* **Methods**

|  |  |
| --- | --- |
| 암호화  Method | public boolean DoPackaging(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile,  String oContentName, String oUserID,  String oUserName, String oEtc1,  [String oEtc2,][String oEtc3,]  [String oEtc4,][String oEtc5]) |
| public boolean DoPackagingFsn2(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile, String oContentName, String oUserID, String oUserName, String WriterID,  String WriterNm, String WriterID,  String WriterNm, String WdeptID,  String WdeptNm, String OwnerID,  String OwnerNm, String OdeptID,  String OdeptNm, String Level) |
| public boolean DoPackagingNx(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile, String oContentName, String oUserID, String oUserName, String WriterID,  String WriterNm, String WriterID,  String WriterNm, String WdeptID,  String WdeptNm, String OwnerID,  String OwnerNm, String OdeptID,  String OdeptNm, String Level) |
| 복호화  Method | public boolean DoExtract(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile) |
| public boolean DoExtract\_s(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile,  String oUserid, String oPurpose,  String oDummy1, String oDummy2,  String oDummy3, String oDummy4,  String oDummy5) |
| public boolean DoExtract\_s2(String oHomeDir, String oServerID, String oSourceFile, String oTargetFile,  String oUserId, String oPurpose,  boolean saveflag, String oDum1,  String oDum2, String oDum3, String oDum4, String oDum5) |
| public boolean DoExtract\_c(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile,  String oPurpose) |
| 권한관련  Method | public boolean AddRightsFsn(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile,  String oTargettype, String oTargetid,  String oTargetname, String oRights) |
| public boolean DeleteRightsFsn(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile,  String oTargettype, String oTargetid) |
| public boolean DeleteAllRightsFsn(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile) |
| public boolean ReadRightsFsn(String oEncFile, String oTargettype,  String oTargetid) |
| public boolean ReadAllRightsFsn(String oEncFile) |
| public String GetFseReaderAndOption(String sfile) |
| public Vector GetFseReaderAndOption2(String sfile) |
| 유틸  Method | public int GetFileType(String SourceFile) |
| * public boolean IsPackageFile(String oSourceFile) |
| public boolean IsSupportFile(String oHomeDir, String oServerID,  String oSourceFile) |
| public Hashtable GetFileHeader(String sfile) |
| 주요  Property | public String getLastErrorStr() |
| public int getLastErrorNum() |
| public void setCharset(String sCharset) |
| public void setPackagerType(int d) |
| public int getPackagerType() |
| public void setOverWriteFlag(boolean overWriteFlag) |
| public void setIsTemp(boolean Flag) |
| public String getContainerFileName() |
| public String getContainerFilePathName() |
| 서버전송  Property,  Method | public void SetREQUrl(String szurl) |
| public void SetREQSubUrl(String szurl) |
| public void SetREQPort(int nport) |
| public void SetREQProtocol(String szprotocol) |
| public boolean RevokeUsage4Fse(String oHomeDir, String oServerID,  String oFile) |
| public boolean RestoreUsage4Fse(String oHomeDir, String oServerID,  String oFile) |
| public boolean RegisterUsage4Fse(String oHomeDir, String oServerID,  String oFile) |
| public boolean RevokeUsageReaders4Fse(String oHomeDir, String oServerID,  String oFile, String Readers) |
| public boolean RevokeUsageReaders4Fse(String oHomeDir, String oServerID,  String oFile, String Readers) |
| public void setUsageLogInfo(boolean bLogging, String userId,  String userName, String DeptId, String DeptName,  String PositionId, String PositionName) |

##### DoPackaging

###### Description

암호화 작업을 진행합니다.

###### Syntax

public boolean DoPackaging(String oHomeDir,   
String oServerID,   
String oSourceFile,   
String oTargetFile,   
String oContentName,   
String oUserID,   
String oUserName,   
String oEtc1,   
[String oEtc2,]   
[String oEtc3,]   
[String oEtc4,]   
[String oEtc5])

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 암호화 하고자 하는 원본 파일 |
| String oTargetFile | 암호화 후 생성 하고자 하는 파일 |
| String oContentName | 원본 파일명 |
| String oUserID | 생성자(등록자) ID |
| String oUserName | 생성자(등록자) 이름 |
| String oEtc1 | 확장(일반적으로 시스템 명) |
| String oEtc2 | 확장(일반적으로 ACL 구분) [옵션] |
| String oEtc3 | 확장(프로그램 ID) [옵션] |
| String oEtc4 | 확장 [옵션] |
| String oEtc5 | 확장 [옵션] |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(), getLastErrorNum()를 참조한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*.*

*.*

*.*

*WorkPackager oWorkPackager = new WorkPackager();*

*oWorkPackager.setOverWriteFlag(false);*

*boolean bret = oWorkPackager.DoPackaging("/Workpackager/fsdinit",*

*"000000000000001E",*

*"/Workpackager/contents/test.doc",*

"/Workpackager/target/test.doc",

*"test.doc",*

*"admin",*

*"관리자",*

*"etc1",*

*"etc2");*

*if (!bret) {*

*System.out.println("암호화 중 오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+oWorkPackager.getLastErrorStr());*

*}*

*.*

*.*

*.*

##### DoPackagingFsn2

###### Description

Fsn 형식의 암호화 작업을 진행합니다.

###### Syntax

public boolean DoPackagingFsn2(String oHomeDir,   
String oServerID,   
String oSourceFile,   
String oTargetFile,   
String oContentName,   
String oUserID,   
String oUserName,   
String oWriterID,   
String oWriterName,  
String oWriterDeptCode,

String oWriterDeptName,

String oOwnerID,

String oOwnerName,

String oOwnerDeptCode,

String oOwnerDeptName,

String SecurityLevel);

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 암호화를 하고자 하는 원본 파일 |
| String oTargetFile | 암호화 후 생성 하고자 하는 파일 |
| String oContentName | 원본 파일명 |
| String oUserID | 생성자(등록자) ID |
| String oUserName | 생성자(등록자) 이름 |
| String oWriterID | 작성자 ID |
| String oWriterName | 작성자 Name |
| String oWriterDeptCode | 작성자 부서코드 |
| String oWriterDeptName | 작성자 부서명 |
| String oOwnerID | 소유자 ID |
| String oOwnerName | 소유자 Name |
| String oOwnerDeptCode | 소유자 부서코드 |
| String oOwnerDeptName | 소유자 부서명 |
| String SecurityLevel | 보안등급 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(),getLastErrorNum()를 참조 한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*.*

*.*

*.*

*WorkPackager oWorkPackager = new WorkPackager();*

*oWorkPackager.setOverWriteFlag(false);*

*boolean bret = oWorkPackager.DoPackagingFsn2(“Workpackager/fsdinit",*

*"000000000000001E",*

*"/Workpackager/contents/test.doc",*

"/Workpackager/target/test.doc",

*"test.doc",*

*"admin",*

*"관리자",*

*"test", “아무개", "001", 팀\_1",*

*"test", “아무개", "001", 팀\_1",*

*“1”);*

*if (!bret) {*

*System.out.println("암호화 중 오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+oWorkPackager.getLastErrorStr());*

*}*

##### DoExtract

###### Description

복호화 작업을 진행합니다.

###### Syntax

public boolean DoExtract(String oHomeDir,   
String oServerID,   
String oSourceFile,   
String oTargetFile)

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 복호화 하고자 하는 원본 파일 |
| String oTargetFile | 복호화 후 생성 하고자 하는 파일 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(), getLastErrorNum()를 참조한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class DoExtract {*

*public static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*boolean bret = oWorkPackager.DoExtract("/Workpackager/fsdinit", "000000000000001E",*

*"/Workpackager/encfile/test.doc",*

*"/Workpackager/decfile/test.doc" );*

*if (bret) {*

*System.out.println("복호화가 정상적으로 수행 됐습니다.");*

*} else {*

*System.out.println("복호화 중 오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+  
oWorkPackager.getLastErrorStr());*

*}*

*}*

*}*

##### DoExtract\_s

###### Description

특정유저의 특정권한이 있을 경우 복호화 작업을 진행합니다.

(서버모듈로 연동 시 사용합니다. 권한 판단하는 서버페이지가 있어야 함)

###### Syntax

public boolean DoExtract\_s(String oHomeDir,   
String oServerID,   
String oSourceFile,   
String oTargetFile,

String oUserId,

String oPurpose,

String oDummy1,

String oDummy2,

String oDummy3,

String oDummy4,

String oDummy5)

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 복호화 하고자 하는 원본 파일 |
| String oTargetFile | 복호화 후 생성 하고자 하는 파일 |
| String oUserId | 복호화 하고자 하는 유저 아이디 |
| String oPurpose | 해당 권한이 있을 경우만 복호화 |
| String oDummy1 | 사이트 별 권한체크 커스터마이징 위한 dummy1 |
| String oDummy2 | 사이트 별 권한체크 커스터마이징 위한 dummy2 |
| String oDummy3 | 사이트 별 권한체크 커스터마이징 위한 dummy3 |
| String oDummy4 | 사이트 별 권한체크 커스터마이징 위한 dummy4 |
| String oDummy5 | 사이트 별 권한체크 커스터마이징 위한 dummy5 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(), getLastErrorNum()를 참조한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class DoExtract {*

*public static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*boolean bret = oWorkPackager.DoExtract\_s("/Workpackager/fsdinit", "000000000000001E",*

*"/Workpackager/encfile/test.doc",*

*"/Workpackager/decfile/test.doc",*

*“admin”, “VIEW”, “”,””,””,””,”” );*

*if (bret) {*

*System.out.println("복호화가 정상적으로 수행 됐습니다.");*

*} else {*

*System.out.println("복호화 중 오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+  
oWorkPackager.getLastErrorStr());*

*}*

*}*

*}*

##### DoExtract\_c

###### Description

특정권한이 있을 경우 복호화 작업을 진행합니다.

(클라이언트모듈로 연동 시 사용합니다. DRM클라이언트가 설치되어 있어야 작동)

###### Syntax

public boolean DoExtract\_c(String oHomeDir,   
String oServerID,   
String oSourceFile,   
String oTargetFile,

String oPurpose)

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 복호화를 하고자 하는 원본 파일 |
| String oTargetFile | 복호화 후 생성 하고자 하는 파일 |
| String oPurpose | 해당권한이 있을 경우만 복호화 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(), getLastErrorNum()를 참조한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class DoExtract {*

*public static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*boolean bret = oWorkPackager.DoExtract\_c("/Workpackager/fsdinit", "000000000000001E",*

*"/Workpackager/encfile/test.doc",*

*"/Workpackager/decfile/test.doc",*

*“VIEW” );*

*if (bret) {*

*System.out.println("복호화가 정상적으로 수행 됐습니다.");*

*} else {*

*System.out.println("복호화 중 오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+  
oWorkPackager.getLastErrorStr());*

*}*

*}*

*}*

##### GetFileType

###### Description

암호화된 파일인지 체크 합니다.

###### Syntax

public int GetFileType(String oSourceFile)

###### Parameters

|  |  |
| --- | --- |
| String oSourceFile | 체크 하고자 하는 파일 |

###### Return Values

|  |  |
| --- | --- |
| int pVal | 파일 정보를 int 형으로 Return |

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class GetFileType {*

*public static void main(String[] args) {*

*WorkPackager workpackager = new WorkPackager();*

*int i = 0;*

*System.out.println("20: 파일을 찾을 수 없습니다.");*

*System.out.println("21: 파일 사이즈가 0 입니다.");*

*System.out.println("22: 파일을 읽을 수 없습니다.");*

*System.out.println("29: 암호화 파일이 아닙니다.");*

*System.out.println("26: FSD 파일입니다.");*

*System.out.println("105: Wrapsody 파일입니다.");*

*System.out.println("101: MarkAny 파일입니다.");*

*System.out.println("104: INCAPS 파일입니다.");*

*System.out.println("103: FSN 파일입니다.");*

*System.out.println("106: NX 파일입니다.");*

*System.out.println("107: NASCA 파일입니다.");*

*i = workpackager.GetFileType("/Workpackager/content/fsd.doc");*

*System.out.println("fsd.doc : " + i);*

*i = workpackager.GetFileType("/Workpackager/encfile/wrapsody.doc");*

*System.out.println("wrapsody.doc : " + i);*

*}*

*}*

##### IsPackageFile

###### Description

암호화된 파일인지 체크 합니다.

###### Syntax

public boolean IsPackageFile(String oSourceFile)

###### Parameters

|  |  |
| --- | --- |
| String oSourceFile | 체크 하고자 하는 파일 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | 0: 일반 파일 또는 에러 1: 암호화 파일 |

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class IsPackageFile {*

*public static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*boolean nret = false;*

*System.out.println("");*

*nret = oWorkPackager.IsPackageFile("/Workpackager/content/fsd.doc");*

*System.out.println("fsd.doc : "+nret);*

*nret = oWorkPackager.IsPackageFile("/Workpackager/content/일반파일.doc");*

*System.out.println("일반파일.doc : "+nret);*

*}*

*}*

##### IsSupportFile

###### Description

지원 가능한 파일인지 체크합니다.

###### Syntax

public boolean IsSupportFile(String oHomeDir,   
String oServerID,   
String oSourceFile)

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 점검 하고자 하는 파일 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True : 지원가능 False : 지원 불가능 또는 에러 |

동작에 실패하였는지는 getLastErrorStr(),getLastErrorNum()를 참조 한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class IsSupportFile {*

*public static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*boolean nret = false;*

*System.out.println("");*

*nret = oWorkPackager.IsSupportFile( "/Workpackager/fsdinit",*

*"000000000000001E",*

*"fsd.doc");*

*if (oWorkPackager.getLastErrorNum()==0) {*

*System.out.println("fsd.doc : "+nret);*

*} else {*

*System.out.println(" 오류 정보..");*

*System.out.println("["+ oWorkPackager.getLastErrorNum()+"]   
"+oWorkPackager.getLastErrorStr());*

*}*

*nret = oWorkPackager.IsSupportFile( "/Workpackager/fsdinit",*

*"000000000000001E",*

*"fsd.aaa");*

*if (oWorkPackager.getLastErrorNum()==0) {*

*System.out.println("fsd.aaa : "+nret);*

*} else {*

*System.out.println(" 오류 정보..");*

*System.out.println("["+ oWorkPackager.getLastErrorNum()+"]   
"+oWorkPackager.getLastErrorStr());*

*}*

*}*

*}*

##### GetFileHeader

###### Description

암호화된 파일의 헤더 정보를 읽어 온다.

###### Syntax

public Hashtable GetFileHeader(String sfile)

암호화되지 않은 문서일 경우는 빈 Hashtable 로 Return

###### Parameters

|  |  |
| --- | --- |
| String sfile | 점검 하고자 하는 파일 |

###### Return Values

|  |  |
| --- | --- |
| Hashtable hVal | 헤더 정보 |

동작에 실패하였는지는 getLastErrorStr(),getLastErrorNum()를 참조 한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*import java.util.\*;*

*import java.io.\*;*

*public class GetFileHeader {*

*public static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*Hashtable hTable = null;*

*try {*

*hTable = oWorkPackager.GetFileHeader("/Workpackager/encfile/test.doc");*

*if (oWorkPackager.getLastErrorNum()!=0) {*

*System.out.println(" 오류 정보..");*

*System.out.println("["+ oWorkPackager.getLastErrorNum()+"]   
"+oWorkPackager.getLastErrorStr());*

*}*

*for(Enumeration e = hTable.keys(); e.hasMoreElements(); )*

*{*

*String msgName = (String)e.nextElement();*

*String msgValue = (String)hTable.get(msgName);*

*System.out.println(msgName + "=" + msgValue);*

*}*

*} catch (Exception e) {*

*e.printStackTrace();*

*}*

*}*

*}*

##### getLastErrorStr

###### Description

마지막 실행한 메소드에서 발생한 에러 내용을 확인 한다.

###### Syntax

public String getLastErrorStr()

에러가 없으면 “” 로 Return

###### Parameters

|  |  |
| --- | --- |
| *없음* |  |

###### Return Values

|  |  |
| --- | --- |
| String sVal | 에러 문자열 |

##### getLastErrorNum

###### Description

마지막 실행한 메소드에서 발생한 에러 번호를 확인 한다.

###### Syntax

public int getLastErrorNum()

에러가 없으면 0으로 Return

###### Parameters

|  |  |
| --- | --- |
| *없음* |  |

###### Return Values

|  |  |
| --- | --- |
| int nVal | 에러번호 |

*※ 부록 에러 코드 참조*

##### setCharset

###### Description

문자열의 charset을 설정합니다.

###### Syntax

public void setCharset(String sCharset)

각 메소드에 들어 가는 문자열의 Charset 을 정의 합니다.

(Default 로 KSC5601 입니다.)

###### Parameters

|  |  |
| --- | --- |
| String sCharset | charset |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*.*

*.*

*.*

*WorkPackager oWorkPackager = new WorkPackager();*

*oWorkPackager.setCharset(“UTF-8”);*

*or*

*oWorkPackager.setCharset(“eucKR”);*

*or*

*oWorkPackager.setCharset(“KSC5601”);*

*………*

##### setPackagerType

###### Description

파일 암호화 시 암호화 Type 를 정의 합니다. .

###### Syntax

public void setPackagerType(int d)

|  |  |  |  |
| --- | --- | --- | --- |
| mime\_type.ini | 번호 | TYPE | static 변수 |
| 지원 | 2 | OLE | WorkPackager.PACKAGE\_OLE |
| **3** | **CMH** | **WorkPackager.PACKAGE\_CMH [Default]** |
| 무시 | 4 | OLE | WorkPackager.PACKAGE\_NOCHECK\_OLE |
| 5 | CMH | WorkPackager.PACKAGE\_NOCHECK\_CMH |

###### Parameters

|  |  |
| --- | --- |
| int d | Type |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### getPackagerType

###### Description

현재 설정된 암호화 Type을 읽어 온다..

###### Syntax

public int getPackagerType()

###### Parameters

|  |  |
| --- | --- |
| *없음* |  |

###### Return Values

|  |  |
| --- | --- |
| int d | 암호화 Type |

##### setOverWriteFlag

###### Description

파일 암호화 시 target 파일이 이미 존재 할 경우 덮어 쓸지 아니면 새로운 이름으로 생성 할지를 설정 합니다.

###### Syntax

public void setOverWriteFlag(boolean overWriteFlag)

새로운 이름은 기존 파일명\_[n] 으로 생성됨.

###### Parameters

|  |  |
| --- | --- |
| boolean overWriteFlag | true : 덮어씀 false : 새로운 이름으로 생성 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setIsTemp

###### Description

파일 암복호화시 target파일의 PATH정보를 윈도우 temporary 폴더로 자동설정

(윈도우만 가능, Vista의 폴더보안 적용 시에 사용하기 위함..)

###### Syntax

public void setIsTemp (boolean Flag)

###### Parameters

|  |  |
| --- | --- |
| boolean Flag | true : temporary 폴더에 문서생성  false : 사용자가 지정한 target폴더에 문서생성 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### getContainerFileName

###### Description

암호화 성공 시 생성된 파일의 파일명을 받는다.

###### Syntax

public String getContainerFileName()

setOverWriteFlag 를 false로 설정 후 암호화를 하면 target 으로 지정된 파일명과 다르게 생성될 수 있으므로 암호화 후에는 반드시 getContainerFileName 로 파일명을 받아서 처리 해야 함.

###### Parameters

|  |  |
| --- | --- |
| *없음* |  |

###### Return Values

|  |  |
| --- | --- |
| String sVal | 파일명 문자열 |
|  |  |

##### AddRightsFsn

###### Description

Fsn2.0 형식 문서의 특정유저(혹은 부서)의 권한정보를 추가&수정한다.

###### Syntax

public boolean AddRightsFsn ( String oHomeDir,   
String oServerID,   
String oSourceFile,   
String oTargetFile,   
String oTargettype,   
String oTargetid,   
String oTargetname,   
String oRights);

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 권한수정 하고자 하는 암호화 원본 파일 |
| String oTargetFile | 권한수정 후 생성하고자 하는 파일 |
| String oTargettype | 권한수정대상 타입(“user”/”group”) |
| String oTargetid | 권한수정대상 ID (user/dept id) |
| String oTargetname | 권한수정대상 이름 (user/dept name) |
| String oRights | 권한정보 <아래참고> |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(),getLastErrorNum()를 참조 한다.

###### <참고>

<<입력 예>>

*“VIEW@-1@2004-01-01 12:00:00@2008-01-01 12:00:00,SECURE\_SAVE@-1@0@0”*

=> [PURPOSE STRING] 포맷

[PURPOSE]@[LIMITCOUNT]@[STARTDATE]@[ENDDATE],…

|  |  |
| --- | --- |
| 파라메타 | 설명 |
| PURPOSE | DRM권한 |
| LIMITCOUNT | 사용횟수제한(여기선 무조건 -1) |
| STARTDATE | 라이선스 유효기간(시작일) :YYYY-MM-DD HH:MM:SS |
| ENDDATE | 라이선스 유효기간(종료일) :YYYY-MM-DD HH:MM:SS |

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*.*

*WorkPackager oWorkPackager = new WorkPackager();*

*boolean nret = oWorkPackager.AddRightsFsn("./fsdinit",*

*"0000000000000906",*

*"./target/가나\_org.ppt",*

*"./target/가나\_edit.ppt",*

*"user",*

*"test",*

*"테스트",*

*"VIEW@-1@2004-04-01 12:00:00@2008-12-31 12:00:00,SECURE\_SAVE@-1@0@0,SAVE@-1@0@0");*

*if (oWorkPackager.getLastErrorNum()==0)*

*{*

*System.out.println(" 권한수정 성공.");*

*}else{*

*System.out.println(" 권한수정 중 오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+oWorkPackager.getLastErrorStr());*

##### DeleteRightsFsn

###### Description

Fsn2.0 형식 문서의 특정유저(혹은 부서)의 권한정보를 삭제한다.

###### Syntax

public boolean DeleteRightsFsn ( String oHomeDir,   
String oServerID,   
String oSourceFile,   
String oTargetFile,   
String oTargettype,   
String oTargetid);

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 권한삭제 하고자 하는 암호화 원본 파일 |
| String oTargetFile | 권한삭제 후 생성하고자 하는 파일 |
| String oTargettype | 권한삭제 대상 타입(“user”/”group”) |
| String oTargetid | 권한삭제 대상 ID (user/dept id) |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(),getLastErrorNum()를 참조 한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*.*

*WorkPackager oWorkPackager = new WorkPackager();*

*boolean nret = oWorkPackager.* *DeleteRightsFsn ("./fsdinit",*

*"0000000000000906",*

*"./target/가나\_org.ppt",*

*"./target/가나\_edit.ppt",*

*"user",*

*"test");*

*if (oWorkPackager.getLastErrorNum()==0)*

*{*

*System.out.println(" 권한삭제 성공.");*

*}else{*

*System.out.println(" 권한삭제 중 오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+oWorkPackager.getLastErrorStr());*

*}*

##### DeleteAllRightsFsn

###### Description

Fsn2.0 형식 문서의 모든 권한정보를 삭제한다.

###### Syntax

public boolean DeleteAllRightsFsn ( String oHomeDir,   
String oServerID,   
String oSourceFile,   
String oTargetFile);

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 권한삭제 하고자 하는 암호화 원본 파일 |
| String oTargetFile | 권한삭제 후 생성하고자 하는 파일 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(),getLastErrorNum()를 참조 한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*.*

*WorkPackager oWorkPackager = new WorkPackager();*

*boolean nret = oWorkPackager.* *DeleteAllRightsFsn ("./fsdinit",*

*"0000000000000906",*

*"./target/가나\_org.ppt",*

*"./target/가나\_edit.ppt",);*

*if (oWorkPackager.getLastErrorNum()==0)*

*{*

*System.out.println(" 권한삭제 성공.");*

*}else{*

*System.out.println(" 권한삭제 중 오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+oWorkPackager.getLastErrorStr());*

*}*

##### ReadRightsFsn

###### Description

Fsn2.0 형식 문서의 특정유저(혹은 부서)의 권한정보를 조회한다.

###### Syntax

public String ReadRightsFsn ( String oSourceFile,   
String oTargettype,   
String oTargetid);

###### Parameters

|  |  |
| --- | --- |
| String oSourceFile | 권한조회 하고자 하는 암호화 원본 파일 |
| String oTargettype | 권한조회 대상 타입(“user”/”group”) |
| String oTargetid | 권한조회 대상 ID (user/dept id) |

###### Return Values

|  |  |
| --- | --- |
| String pVal | 권한 String 혹은 NULL <아래참고> |

###### <참고>

<<출력결과>>

*“VIEW@-1@2004-01-01 12:00:00@2008-01-01 12:00:00,SECURE\_SAVE@-1@0@0”*

=> [PURPOSE STRING] 포맷

[PURPOSE]@[LIMITCOUNT]@[STARTDATE]@[ENDDATE],…

|  |  |
| --- | --- |
| 파라메타 | 설명 |
| PURPOSE | DRM권한 |
| LIMITCOUNT | 사용횟수제한(여기선 무조건 -1) |
| STARTDATE | 라이선스 유효기간(시작일) :YYYY-MM-DD HH:MM:SS |
| ENDDATE | 라이선스 유효기간(종료일) :YYYY-MM-DD HH:MM:SS |

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*.*

*WorkPackager oWorkPackager = new WorkPackager();*

*String rights = oWorkPackager.* *ReadRightsFsn ("./target/가나\_org.ppt", "user", "test");*

*if (rights != null )*

*{*

*System.out.println(" 권한조회 성공." + rights);*

*}*

##### ReadAllRightsFsn

###### Description

Fsn2.0 형식 문서의 모든 권한정보를 조회한다.

###### Syntax

public String ReadAllRightsFsn ( String oSourceFile);

###### Parameters

|  |  |
| --- | --- |
| String oSourceFile | 권한조회 하고자 하는 암호화 원본 파일 |

###### Return Values

|  |  |
| --- | --- |
| String pVal | 권한String 혹은 NULL <아래참고> |

###### <참고>

<<출력결과>>

*“group,client,Client팀&VIEW@-1@0@0|user,patriot317,김덕진(pa*

*triot317)&VIEW@-1@0@0|user,nuyij,최지연&VIEW@-1@0@0,SECURE\_SAVE@-1@0@0”*

=> 전체포맷

[TARGET STRING]&[ PURPOSE STRING]|…

=> [TARGET STRING] 포맷

[TargetType],[TargetID],[TargetName]

|  |  |
| --- | --- |
| 파라메타 | 설명 |
| TargetType | “user” or “group” |
| TargetID | 유저/부서 ID |
| TargetName | 유저/부서 Name |

=> [PURPOSE STRING] 포맷

[PURPOSE]@[LIMITCOUNT]@[STARTDATE]@[ENDDATE],…

|  |  |
| --- | --- |
| 파라메타 | 설명 |
| PURPOSE | DRM권한 |
| LIMITCOUNT | 사용횟수제한(여기선 무조건 -1) |
| STARTDATE | 라이선스 유효기간(시작일) :YYYY-MM-DD HH:MM:SS |
| ENDDATE | 라이선스 유효기간(종료일) :YYYY-MM-DD HH:MM:SS |

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*.*

*WorkPackager oWorkPackager = new WorkPackager();*

*String rights = oWorkPackager.* *ReadAllRightsFsn ("./target/가나\_org.ppt");*

*if (rights != null )*

*{*

*System.out.println(" 권한조회 성공." + rights);*

*}*

##### RevokeUsage4Fse

###### Description

FSE암호화문서를 폐기한다. 해당 API는 서버로 문서의 폐기정보를 전송하므로

SetREQUrl/SetREQSubUrl/SetREQPort/SetREQProtocol을 호출한 후에 사용한다.

###### Syntax

public boolean RevokeUsage4Fse (String oHomeDir, String oServerID, String oFile);

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oFile | 폐기하고자 하는 FSE암호화 파일 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(), getLastErrorNum()를 참조한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class RevokeUsage4Fse {*

*pu blic static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*oWorkPackager.* *SetREQProtocol(“http”);*

*oWorkPackager.SetREQUrl(“drm.fasoo.com”);*

*oWorkPackager.SetREQPort(8080);*

*oWorkPackager.SetREQSubUrl (“/drmone/fse/fs/docmanage/docmanage\_service.do”);*

*boolean bret = oWorkPackager.RevokeUsage4Fse("/Workpackager/fsdinit", "000000000000001E",*

*"/Workpackager/encfile/test.doc" );*

*if (bret) { System.out.println("정상적으로 수행 됐습니다.");}*

*else { System.out.println("오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+ oWorkPackager.getLastErrorStr());}*

*}*

*}*

##### RestoreUsage4Fse

###### Description

폐기된 FSE암호화문서를 복원한다. 해당 API는 서버로 문서의 복원정보를 전송하

므로 SetREQUrl/SetREQSubUrl/SetREQPort/SetREQProtocol을 호출한 후에 사용한다.

###### Syntax

public boolean RestoreUsage4Fse (String oHomeDir, String oServerID, String oFile);

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oFile | 복원하고자 하는 FSE암호화 파일 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(), getLastErrorNum()를 참조한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class RestoreUsage4Fse {*

*pu blic static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*oWorkPackager.* *SetREQProtocol(“http”);*

*oWorkPackager.SetREQUrl(“drm.fasoo.com”);*

*oWorkPackager.SetREQPort(8080);*

*oWorkPackager.SetREQSubUrl (“/drmone/fse/fs/docmanage/docmanage\_service.do”);*

*boolean bret = oWorkPackager.RestoreUsage4Fse("/Workpackager/fsdinit", "000000000000001E",*

*"/Workpackager/encfile/test.doc" );*

*if (bret) { System.out.println("정상적으로 수행 됐습니다.");}*

*else { System.out.println("오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+ oWorkPackager.getLastErrorStr());}*

*}*

*}*

##### RegisterUsage4Fse

###### Description

FSE문서를 패키져 API연동하여 암호화 하는 경우 해당 api를 사용하여 서버에 암

호화문서정보를 등록한다. 해당 API는 서버로 문서의 복원정보를 전송하므로

SetREQUrl/SetREQSubUrl/SetREQPort/SetREQProtocol을 호출한 후에 사용한다.

###### Syntax

public boolean RegisterUsage4Fse(String oHomeDir, String oServerID, String oFile);

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oFile | 등록하고자 하는 FSE암호화 파일 |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(), getLastErrorNum()를 참조한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class RegisterUsage4Fse {*

*pu blic static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*oWorkPackager.* *SetREQProtocol(“http”);*

*oWorkPackager.SetREQUrl(“drm.fasoo.com”);*

*oWorkPackager.SetREQPort(8080);*

*oWorkPackager.SetREQSubUrl (“/drmone/fse/fs/docmanage/docmanage\_service.do”);*

*boolean bret = oWorkPackager.RegisterUsage4Fse ("/Workpackager/fsdinit", "000000000000001E",*

*"/Workpackager/encfile/test.doc" );*

*if (bret) { System.out.println("정상적으로 수행 됐습니다.");}*

*else { System.out.println("오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+ oWorkPackager.getLastErrorStr());}*

*}*

*}*

##### RevokeUsageReaders4Fse

###### Description

입력된 수신자에 대한 FSE암호화문서를 권한을 폐기한다. 권한을 해당 API는

서버로 문서의 폐기정보를 전송하므로 SetREQUrl/SetREQSubUrl/SetREQPort/

SetREQProtocol을 호출한 후에 사용한다.

###### Syntax

public boolean RevokeUsageReaders4Fse (String oHomeDir, String oServerID,  
 String oFile, String Readers);

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oFile | 폐기하고자 하는 FSE암호화 파일 |
| String Readers | 폐기하고자 하는 수신자 이메일 리스트 (“|”로 구분) |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(), getLastErrorNum()를 참조한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class RevokeUsageReaders4Fse {*

*pu blic static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*oWorkPackager.* *SetREQProtocol(“http”);*

*oWorkPackager.SetREQUrl(“drm.fasoo.com”);*

*oWorkPackager.SetREQPort(8080);*

*oWorkPackager.SetREQSubUrl (“/drmone/fse/fs/docmanage/docmanage\_service.do”);*

*boolean bret = oWorkPackager.RevokeUsageReaders4Fse("/Workpackager/fsdinit", "000000000000001E",*

*"/Workpackager/encfile/test.doc", "yoonki.lee@fasoo.com|kyusoo@fasoo.com" );*

*if (bret) { System.out.println("정상적으로 수행 됐습니다.");}*

*else { System.out.println("오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+ oWorkPackager.getLastErrorStr());}*

*}*

*}*

##### RestoreUsageReaders4Fse

###### Description

입력된 수신자에 대한 FSE암호화문서를 권한을 복원한다. 권한을 해당 API는

서버로 문서의 복원정보를 전송하므로 SetREQUrl/SetREQSubUrl/SetREQPort/

SetREQProtocol을 호출한 후에 사용한다.

###### Syntax

public boolean RestoreUsageReaders4Fse(String oHomeDir, String oServerID,  
 String oFile, String Readers);

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oFile | 복원하고자 하는 FSE암호화 파일 |
| String Readers | 복원하고자 하는 수신자 이메일 리스트 (“|”로 구분) |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(), getLastErrorNum()를 참조한다.

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class RestoreUsageReaders4Fse {*

*pu blic static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*oWorkPackager.* *SetREQProtocol(“http”);*

*oWorkPackager.SetREQUrl(“drm.fasoo.com”);*

*oWorkPackager.SetREQPort(8080);*

*oWorkPackager.SetREQSubUrl (“/drmone/fse/fs/docmanage/docmanage\_service.do”);*

*boolean bret = oWorkPackager.RestoreUsageReaders4Fse ("/Workpackager/fsdinit", "000000000000001E",*

*"/Workpackager/encfile/test.doc", "yoonki.lee@fasoo.com|kyusoo@fasoo.com" );*

*if (bret) { System.out.println("정상적으로 수행 됐습니다.");}*

*else { System.out.println("오류입니다.");*

*System.out.println(" 오류 정보..");*

*System.out.println(" ["+ oWorkPackager.getLastErrorNum()+"] "+ oWorkPackager.getLastErrorStr());}*

*}*

*}*

##### SetUsageLogInfo

###### Description

암호화나 복호화를 할 경우 서버로 사용자 로그를 전송하고자 할 때 사용

###### Syntax

public void setUsageLogInfo(boolean bLogging, String userId,

String userName, String DeptId, String DeptName,

String PositionId, String PositionName)

###### Parameter

|  |  |
| --- | --- |
| boolean bLogging | 로그 전송 여부 설정 |
| String userId | 사용자 아이디 |
| String userName | 사용자 이름 |
| String DeptId | 부서 아이디 |
| String DeptName | 부서 이름 |
| String PositionId | 직급 아이디 |
| String PositionName | 직급 이름 |

###### Return Value

없음

###### Example

*import com.fasoo.fcwpkg.packager.\*;*

*public class DoExtract {*

*public static void main(String[] args) {*

*WorkPackager oWorkPackager = new WorkPackager();*

*oWorkPackager. setUsageLogInfo(true, "userId", "userName", "ddept73", "ddeptN73",*

*"posdId", "posdName", "systemId", "systemName");*

*boolean bret = oWorkPackager.DoExtract\_c("/Workpackager/fsdinit", "000000000000001E",*

*"/Workpackager/encfile/test.doc",*

*"/Workpackager/decfile/test.doc",*

*“VIEW” );*

*}*

*}*

# Specification of WPackager

* **CLASS** : com.fasoo.fcwpkg.packager.WPackager
* **Methods**

|  |  |
| --- | --- |
| Method | public void setSender(String sender) |
| public void setReader(String reader) |
| public void addReader(String reader) |
| public void setCharset(String sCharset) |
| public void setPackagerType(int d) |
| public void setOverWriteFlag(boolean overWriteFlag) |
| public String getContentID() |
| public void setOptionLicenseType(String str) |
| public void setOptionStartDate(String str) |
| public void setOptionEndDate(String str) |
| public void setOptionView(int ntype) |
| public void setOptionPrint(int ntype) |
| public void setOptionPrintScreen(int ntype) |
| public void setOptionSave(int ntype) |
| public void setOptionSecureSave(int ntype) |
| public void setOptionCount(int ntype) |
| public void setOptionPcCount(int ntype) |
| public void setOptionConfirmMail(int ntype) |
| public void setOptionConfirmMail4In(int ntype) |
| public void setOptionTransfer(int ntype) |
| public void setOptionPeriod(int ntype) |
| public void setOptionAuthentication(int ntype) |
| public void setOptionStr(String str) |
| public void setIsTemp(int ntype) |
| public void setSenderRight(int ntype) |
| public void setFssStateOn() |
| public boolean DoPackagingFSE(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile,  String oContentName, String oUserID,  String oUserName, String oEtc1,  [String oEtc2,][String oEtc3,]  [String oEtc4,][String oEtc5]) |
| public boolean DoPackagingFSE2(String oHomeDir, String oServerID,  String oSourceFile, String oTargetFile,  String oContentName, String oUserID,  String oUserName, String oEtc1,  [String oEtc2,][String oEtc3,] [String oEtc4,][String oEtc5]) |
| public String getContainerFileName() |
| public String getContainerFilePathName() |
| public String getLastErrorStr() |
| public int getLastErrorNum() |

##### setSender

###### Description

발신자 이메일 주소를 지정합니다.

###### Syntax

public void setSender(String sender)

###### Parameters

|  |  |
| --- | --- |
| String sender | 발신자 이메일 주소 |

###### Return Values

|  |  |
| --- | --- |
| void |  |

##### setReader

###### Description

수신자 이메일 리스트를 새로 지정합니다. 수신자가 여러 명인 경우 한번에 여러 명의 수신자 이메일을 설정할 수 있습니다. 수신자 리스트는 “|” 구분자로 여러 개의 이메일 주소를 문자열로 만듭니다.

###### Syntax

public void setReader(String reader)

###### Parameters

|  |  |
| --- | --- |
| String reader | 수신자 이메일 주소 |

###### Return Values

|  |  |
| --- | --- |
| void |  |

##### addReader

###### Description

기존 수신자 리스트에 수신자를 추가합니다. 수신자가 여러 명일 때 연속적으로 해당 함수를 호출하거나 “|”로 구분하여 여러 명의 수신자를 추가할 수 있습니다.

###### Syntax

public void setReader(String reader)

###### Parameters

|  |  |
| --- | --- |
| String reader | 수신자 이메일 주소 |

###### Return Values

|  |  |
| --- | --- |
| void |  |

##### setCharset

###### Description

각 메소드에 들어 가는 문자열의 Charset 을 정의 합니다.

###### Syntax

public void setCharset(String sCharset)

###### Parameters

|  |  |
| --- | --- |
| String sCharset | Character Set(ex. “UTF-8”) |

###### Return Values

|  |  |
| --- | --- |
| void |  |

##### setPackagerType

###### Description

파일 암호화 시 암호화 Type 를 정의 합니다. .

###### Syntax

public void setPackagerType(int d)

|  |  |  |  |
| --- | --- | --- | --- |
| mime\_type.ini | 번호 | TYPE | static 변수 |
| 지원 | 2 | OLE | WorkPackager.PACKAGE\_OLE |
| **3** | **CMH** | **WorkPackager.PACKAGE\_CMH [Default]** |
| 무시 | 4 | OLE | WorkPackager.PACKAGE\_NOCHECK\_OLE |
| 5 | CMH | WorkPackager.PACKAGE\_NOCHECK\_CMH |

###### Parameters

|  |  |
| --- | --- |
| int d | Type |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOverWriteFlag

###### Description

파일 암호화 시 target 파일이 이미 존재 할 경우 덮어 쓸지 아니면 새로운 이름으로 생성 할지를 설정 합니다.

###### Syntax

public void setOverWriteFlag(boolean overWriteFlag)

새로운 이름은 기존 파일명\_[n] 으로 생성됨.

###### Parameters

|  |  |
| --- | --- |
| boolean overWriteFlag | true : 덮어씀 false : 새로운 이름으로 생성 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### getContentID

###### Description

패키징된 문서의 콘텐트 ID를 얻어옵니다.

###### Syntax

public String getContentID()

###### Parameters

없음

###### Return Values

|  |  |
| --- | --- |
|  |  |
| String | 콘텐트 ID |

##### setOptionLicenseType

###### Description

패키징된 파일에 대한 열람 시 서버 접속 여부(파일에 대한권한 라이선스 획득방법)를 지정합니다. 기본값은 “on-line”으로 설정되어 있습니다.

###### Syntax

public void setOptionLicenseType(String str)

###### Parameters

|  |  |
| --- | --- |
| String str | “ONLINE” : 온라인 발급, “OFFLINE” : 오프라인 라이선스 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionStartDate

###### Description

위 StartDate 속성은 유효기간 중 시작일을 지정할 수 있습니다. 유효기간은 Period 속성을 사용할 수도 있지만 Period 속성은 패키징 시점부터의 유효기간을 정하는 반면 이 속성은 시작일부터 종료일을 명시적으로 기입하기 위한 속성입니다. StartDate 속성은 반듯이 GMT 값으로 입력되어야 하며 다음과 같은 형식입니다.

“YYYY-MM-DD HH:mm:ss”

StartDate 속성이 활성화 되면 Period 속성은 무시됩니다..

###### Syntax

public void setOptionStartDate(String str)

###### Parameters

|  |  |
| --- | --- |
| String str | 문서 라이선스 시작일(ex. “2010-01-29 23:59:59”) |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionEndDate

###### Description

위 EndDate 속성은 유효기간 중 종료일을 지정할 수 있습니다. 유효기간은 Period 속성을 사용할 수도 있지만 Period 속성은 패키징 시점부터의 유효기간을 정하는 반면 이 속성은 시작일부터 종료일을 명시적으로 기입하기 위한 속성입니다.

EndDate 속성은 반듯이 GMT 값으로 입력되어야 하며 다음과 같은 형식입니다.

“YYYY-MM-DD HH:mm:ss”

EndDate 속성이 활성화 되면 Period 속성은 무시됩니다.

###### Syntax

public void setOptionEndDate(String str)

###### Parameters

|  |  |
| --- | --- |
| String str | 문서 라이선스 종료일(ex. “2010-01-29 23:59:59”) |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionView

###### Description

열람 권한을 설정합니다.

###### Syntax

public void setOptionView(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 권한 없음, 1 : 권한 있음 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionPrint

###### Description

인쇄 권한을 설정합니다.

###### Syntax

public void setOptionPrint(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 권한 없음, 1 : 권한 있음 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionPrintScreen

###### Description

화면 캡쳐 권한을 설정합니다.

###### Syntax

public void setOptionPrintScreen(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 권한 없음, 1 : 권한 있음 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionSave

###### Description

패키징된 파일에 대한 암호화해제 권한을 지정합니다.

###### Syntax

public void setOptionSave(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 권한 없음, 1 : 권한 있음 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionSecureSave

###### Description

패키징된 파일에 대한 암호화저장 권한을 지정합니다.

###### Syntax

public void setOptionSecureSave(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 권한 없음, 1 : 권한 있음 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionCount

###### Description

열람 횟수를 지정합니다.

###### Syntax

public void setOptionCount(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | -1 : 제한 없음, 0~n : 해당회수만큼 열람 가능 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionPcCount

###### Description

패키징된 파일을 열람할 수 있는 물리적인 PC 대수를 지정합니다.

###### Syntax

public void setOptionPcCount(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | -1 : 제한 없음, 0~n : 해당회수만큼 열람 가능 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionConfirmMail

###### Description

속성은 패키징된 파일에 대한 열람내역을 메일로 통보 받을 수 있는 기능을 설정합니다.

###### Syntax

public void setOptionConfirmMail(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 메일 통보 받지 않음  1 : 한번만 받음  2: 매번 받음 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionConfirmMail4In

###### Description

ConfirmMail과 동일하나, 사내와 사외 수신자를 구분하는 경우에 특별한 용도로 사용됩니다.

###### Syntax

public void setOptionConfirmMail4In(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 메일 통보 받지 않음  1 : 한번만 받음  2: 매번 받음 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionTransfer

###### Description

패키징된 파일을 수신자가 권한을 재 지정하여 사용할 수 있도록 하는 특별한 권한을 말합니다.

###### Syntax

public void setOptionTransfer(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 권한 없음, 1 : 권한 있음 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionPeriod

###### Description

패키징된 파일에 대한 열람 기간을 지정합니다.

###### Syntax

public void setOptionPeriod(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | -1 : 제한 없음, 0~n : 지정된 날짜만큼 열람 가능 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionAuthentication

###### Description

이메일 인증에 방법을 지정합니다.

1로 지정되는 경우 패키징된 파일을 수신한 사용자는 최초 PC에 대해서 이메일을 통한 사용자 인증 과정이 필요하지 않게 됩니다.

###### Syntax

public void setOptionAuthentication(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 1 : 처음 한번 자동 인증, 2 : 이메일 인증 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setOptionStr

###### Description

패키징 속성을 개별적으로 설정하지 않고 하나의 문자열 스트링으로 지정할 수 있는 방법을 제공합니다.

###### Syntax

public void setOptionStr(String str)

###### Parameters

|  |  |
| --- | --- |
| String str | 아래 표 참조. |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 권한 명 | 문자열 | 값 | 예 |
| 열람 권한 | can\_view | 0,1 | can\_view=1 |
| 인쇄 권한 | can\_print | 0,1 | can\_print=1 |
| 편집 권한 | can\_securesave | 0,1 | can\_save=1 |
| 암호화해제 권한 | can\_save | 0,1 | can\_securesave=1 |
| 화면 캡쳐 권한 | can\_printscreen | 0, 1 | can\_printscreen=1 |
| PC 대수 제한 | pc\_count | -1,1,2,3~ | pc\_count=-1  // -1 : 무제한, 1,2,3… |
| 유효 기간 제한 | valid\_days | -1,1,2,3~365 | valid\_days=-1  // -1 : 무제한, 1,2,3… |
| 열람 횟수 제한 | use\_count | -1,1,2,3~ | use\_count=-1  // -1 : 무제한, 1,2,3… |
| 개봉 확인 메일 | confirm\_mail | 0,1,2 | confirm\_mail=0  //외부 사용자에 대한 개봉 확인 메일 옵션  //0 : 확인 안 함, 1: 한번만 확인  2 : 매번 확인 |
| 개봉 확인 메일  - 사이트 커스터마이징 | confirm\_mail\_4in | 0,1,2 | confirm\_mail\_4in=0  //사이트 커스터마이징 속성  //0 : 확인 안 함, 1: 한번만 확인  2 : 매번 확인 |
| 이메일 인증 사용 | authentication | 0,1,2 | Authentication=2  0 : 인증 하지 않음  1 : 최초PC 자동인증  2 : 매번 인증 |
| 유효기간 지정 시작일 | start\_date | 2008-07-17 00:00:00 | start\_date=2008-07-17 00:00:00 |
| 유효기간 지정 종료일 | end\_date | 2008-09-30 23:59:59 | end\_date=2008-09-30 23:59:59 |
| 문서 사용 형태 | License\_type | ONLINE  OFFLINE | // ONLINE: 매번 서버 접속  // OFFLINE : 처음 한번 서버 접속 |

<주의>

- 열람 권한은 반드시 있어야 합니다. 이 권한이 지정되어 있지 않으면 다른 권한이 유효할 수 없습니다.

- 권한 문자열과 권한 문자열 사이는 “|” 구분자로 구분합니다.

- valid\_days와 start\_date , end\_date 는 혼용하여 사용할 수 없습니다.

##### setIsTemp

###### Description

Window Temp 폴더를 Target Directory로 사용하고자 할 때 설정합니다.

###### Syntax

public void setIsTemp(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 사용 안 함, 1 : 사용 함 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setSenderRight

###### Description

발신자에게 수신자와 똑같은 권한을 설정할 것인지 여부를 설정합니다. 기본값으로 미 설정 입니다.

###### Syntax

public void setSenderRight(int ntype)

###### Parameters

|  |  |
| --- | --- |
| Int ntype | 0 : 설정 안 함, 1 : 설정 함 |

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### setFssStateOn

###### Description

FSS가 설치되어 있지 않는 상태에서는 FSE 문서를 열람할수 없도록 설정

###### Syntax

public void setFssStateOn()

###### Return Values

|  |  |
| --- | --- |
| *없음* |  |

##### DoPackagingFSE

###### Description

발신자에게 수신자와 똑같은 권한을 설정할 것인지 여부를 설정합니다. 기본값으로 미 설정 입니다.

###### Syntax

public boolean DoPackagingFSE(String oHomeDir, String oServerID,   
String oSourceFile, String oTargetFile,   
String oContentName, String oUserID,   
String oUserName, String oEtc1,

[String oEtc2,][String oEtc3,]

[String oEtc4,][String oEtc5])

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 암호화를 하고자 하는 원본 파일 |
| String oTargetFile | 암호화 후 생성 하고자 하는 파일 |
| String oContentName | 원본 파일명 |
| String oUserID | 생성자(등록자) ID |
| String oUserName | 생성자(등록자) 이름 |
| String oEtc1 | 확장 |
| String oEtc2 | 확장 [옵션] |
| String oEtc3 | 확장 [옵션] |
| String oEtc4 | 확장 [옵션] |
| String oEtc5 | 확장 [옵션] |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(),getLastErrorNum()를 참조 한다.

##### DoPackagingFSE2

###### Description

DoPackageringFSE와 동일함. FED4.0 과의 인터페이스 호환을 위하여 남겨둠.

###### Syntax

public boolean DoPackagingFSE2(String oHomeDir, String oServerID,   
String oSourceFile, String oTargetFile,   
String oContentName, String oUserID,   
String oUserName, String oEtc1,

[String oEtc2,][String oEtc3,]

[String oEtc4,][String oEtc5])

###### Parameters

|  |  |
| --- | --- |
| String oHomeDir | FSD 홈디렉토리 |
| String oServerID | Server ID |
| String oSourceFile | 암호화를 하고자 하는 원본 파일 |
| String oTargetFile | 암호화 후 생성 하고자 하는 파일 |
| String oContentName | 원본 파일명 |
| String oUserID | 생성자(등록자) ID |
| String oUserName | 생성자(등록자) 이름 |
| String oEtc1 | 확장 |
| String oEtc2 | 확장 [옵션] |
| String oEtc3 | 확장 [옵션] |
| String oEtc4 | 확장 [옵션] |
| String oEtc5 | 확장 [옵션] |

###### Return Values

|  |  |
| --- | --- |
| boolean pVal | True 또는 false |

동작에 실패하였을 때 false return. 에러 내용은 getLastErrorStr(),getLastErrorNum()를 참조 한다.

##### getContainerFileName

###### Description

암호화 성공 시 생성된 파일의 파일명을 받는다.

###### Syntax

public String getContainerFileName()

setOverWriteFlag 를 false로 설정 후 암호화를 하면 target 으로 지정된 파일명과 다르게 생성될 수 있으므로 암호화 후에는 반드시 getContainerFileName 로 파일명을 받아서 처리 해야 함.

###### Parameters

|  |  |
| --- | --- |
| *없음* |  |

###### Return Values

|  |  |
| --- | --- |
| String sVal | 파일명 문자열 |

##### getContainerFilePathName

###### Description

암호화 성공 시 생성된 파일의 파일명(Full Path)을 받는다.

###### Syntax

public String getContainerFilePathName()

###### Parameters

|  |  |
| --- | --- |
| *없음* |  |

###### Return Values

|  |  |
| --- | --- |
| String sVal | 파일명 문자열 |

##### getLastErrorStr

###### Description

마지막 실행한 메소드에서 발생한 에러 내용을 확인 한다.

###### Syntax

public String getLastErrorStr()

에러가 없으면 “” 로 Return

###### Parameters

|  |  |
| --- | --- |
| *없음* |  |

###### Return Values

|  |  |
| --- | --- |
| String sVal | 에러 문자열 |

##### getLastErrorNum

###### Description

마지막 실행한 메소드에서 발생한 에러 번호를 확인 한다.

###### Syntax

public int getLastErrorNum()

에러가 없으면 0으로 Return

###### Parameters

|  |  |
| --- | --- |
| *없음* |  |

###### Return Values

|  |  |
| --- | --- |
| int nVal | 에러번호 |

*※ 부록 에러 코드 참조*

##### [SAMPLE CODE]

|  |
| --- |
| WPackager oEPackager = new WPackager();  oEPackager.setOverWriteFlag(false);  oEPackager.setOptionView(1);  oEPackager.setOptionPrint(1);  oEPackager.setOptionSecureSave(1);  oEPackager.setOptionSave(1);  oEPackager.setOptionPrintScreen(1);  oEPackager.setOptionCount(10);  oEPackager.setOptionPcCount(20);  oEPackager.setOptionConfirmMail(1);  oEPackager.setOptionConfirmMail4In(1);  oEPackager.setOptionTransfer(1);  oEPackager.setOptionPeriod(30);  oEPackager.setOptionStartDate("2010-01-01 00:00:00");  oEPackager.setOptionEndDate("2010-12-31 23:59:59");  oEPackager.setOptionAuthentication(1);  oEPackager.setOptionLicenseType("on-line");  oEPackager.setOptionStr("license\_type=on-line|can\_view=1|can\_print=1|can\_save=1");  oEPackager.setSender("yoonki.lee@gmail.com");  oEPackager.setReader("yoonki.lee@fasoo.com|kyusoo@fasoo.com");  oEPackager.addReader("tyolee83@daum.net|");  oEPackager.setIsTemp(1);  oEPackager.setSenderRight(1);    boolean bret = oEPackager.DoPackagingFSE("./fsdinit", "DRM0000000000001", "./content/테스트.doc","./target/테스트.doc", "테스트.doc", "yklee", "yoonki lee", "etc1");  if (!bret) {  WriteLOG(outp, " [" + oEPackager.getLastErrorNum() + "] "+ oEPackager.getLastErrorStr());  return;  } else {  WriteLOG(outp, "패키징 완료 ["+ oEPackager.getContainerFilePathName() + "]");  } |

# Specification of StreamPackager

* **CLASS**: com.fasoo.fcwpkg.packager.StreamPackager
* **Methods**

|  |  |
| --- | --- |
| Method | public int SetHomeDir(String szpath) |
| public int SetServerID(String szid) |
| public byte[] StrmPackagingFSDInit() |
| public byte[] StrmPackagingFSNInit() |
| public byte[] StrmPackagingUpdate(byte[] inbuf, int insz) |
| public byte[] StrmPackagingFinal(byte[] inbuf, int insz) |
| public int StrmExtractInit() |
| public byte[] StrmExtractUpdate(byte[] inbuf, int insz) |
| public byte[] StrmExtractFinal(byte[] inbuf, int insz) |
| public int setStrmHeader(byte[] inbuf) |
|  |
|  |
|  |

##### SetHomeDir

###### Description

conf 파일이 위치하는 디렉토리 설정

###### Syntax

public int SetHomeDir

###### Parameters

|  |  |
| --- | --- |
| *String szpath* | 디렉토리 경로 |

###### Return Values

|  |  |
| --- | --- |
| int | 에러 번호 |

##### SetServerID

###### Description

ServerID 설정

###### Syntax

public int SetServerID

###### Parameters

|  |  |
| --- | --- |
| *String szid* | ServerID |

###### Return Values

|  |  |
| --- | --- |
| int | 오류 번호 |

##### StrmPackagingFSDInit

###### Description

Streaming으로 데이터를 전달받아 FSD 타입으로 암호화하기 위한 초기화.

###### Syntax

public byte[] StrmPackagingFSDInit

###### Return Values

|  |  |
| --- | --- |
| byte[] | 암호화 문서의 헤더 |

##### StrmPackagingFSNInit

###### Description

Streaming으로 데이터를 전달받아 FSN 타입으로 암호화하기 위한 초기화.

###### Syntax

public byte[] StrmPackagingFSDInit

###### Return Values

|  |  |
| --- | --- |
| byte[] | 암호화 문서의 헤더 |

##### StrmPackagingUpdate

###### Description

데이터 암호화 업데이트

###### Syntax

public byte[] StrmPackagingUpdate

###### Parameters

|  |  |
| --- | --- |
| *byte[] inbuf* | 암호화하고자 하는 데이터 |
| *int insz* | 암호화하고자 하는 데이터 길이 |

###### Return Values

|  |  |
| --- | --- |
| byte[] | 암호화된 데이터 |

##### StrmPackagingFinal

###### Description

최종 암호화 결과 반환 및 정리

###### Syntax

public byte[] StrmPackagingFinal

###### Parameters

|  |  |
| --- | --- |
| *byte[] inbuf* | 암호화하고자 하는 데이터 |
| *int insz* | 암호화하고자 하는 데이터 길이 |

###### Return Values

|  |  |
| --- | --- |
| byte[] | 암호화된 데이터 |

##### StrmExtractInit()

###### Description

Streaming으로 암호화된 데이터를 전달받아 복호화하기 위한 초기화.

###### Syntax

public int StrmExtractInit

###### Return Values

|  |  |
| --- | --- |
| int | 에러번호 |

##### StrmExtractUpdate

###### Description

데이터 복호화 업데이트

###### Syntax

public byte[] StrmExtractUpdate

###### Parameters

|  |  |
| --- | --- |
| *byte[] inbuf* | 복호화하고자 하는 데이터 |
| *int insz* | 복호화하고자 하는 데이터 길이 |

###### Return Values

|  |  |
| --- | --- |
| byte[] | 복호화된 데이터 |

##### StrmExtractFinal

###### Description

최종 복호화 결과 반환 및 정리

###### Syntax

public byte[] StrmExtractFinal

###### Parameters

|  |  |
| --- | --- |
| *byte[] inbuf* | 복호화하고자 하는 데이터 |
| *int insz* | 복호화하고자 하는 데이터 길이 |

###### Return Values

|  |  |
| --- | --- |
| byte[] | 복호화된 데이터 |

##### setStrmHeader

###### Description

streaming으로 암호화하고자 할 경우에 암호화 문서의 헤더 설정

###### Syntax

public int setStrmHeader

###### Parameters

|  |  |
| --- | --- |
| *byte[] inbuf* | 헤더 |

###### Return Values

|  |  |
| --- | --- |
| int | 파일명 문자열 |

##### [SAMPLE CODE]

* 암호문서의 헤더를 가져와 스트리밍으로 암호화하고자 하는 문서의 헤더로 설정하여 암호화하고 복호화하는 샘플

|  |
| --- |
| package com.fasoo.test;  import java.io.\*;  import com.fasoo.fcwpkg.packager.\*;  public class Test  {  public static void main(String[] args)  {  FileEncryptFSN("C:/init/content/docx.txt", "C:/init/target/aaa\_strm\_fsn\_java.txt");  FileDecryptFSN("C:/init/target/aaa\_strm\_fsn\_java.txt", "C:/init/target/aaa\_dec\_fsn\_java.txt");    FileEncryptFSD("C:/init/content/docx.txt", "C:/init/target/aaa\_strm\_fsd\_java.txt");  FileDecryptFSD("C:/init/target/aaa\_strm\_fsd\_java.txt", "C:/init/target/aaa\_dec\_fsd\_java.txt");  }    // Stream Test  public static void FileEncryptFSN(String oriFilePath, String encFilePath)  {  byte[] buffer = new byte[1024];  byte[] outbuf = null;    FileInputStream fis;  FileOutputStream fos;    FasooPackagerJNI jni = new FasooPackagerJNI();  byte[] byteHeader = jni.GetHeaderByte("C:/init/target/aaa\_fsn.txt");  if (byteHeader != null)  System.out.println("GetHeaderByte success.");  else  System.out.println("GetHeaderByte fail.");    try  {  fis = new FileInputStream(oriFilePath);  fos = new FileOutputStream(encFilePath);    BufferedInputStream bis = new BufferedInputStream(fis);  BufferedOutputStream bos = new BufferedOutputStream(fos);    StreamPackager strm = new StreamPackager();  strm.CreateObject();  strm.setStrmHeader(byteHeader);  strm.SetHomeDir("C:/init/fsdinit");  strm.SetServerID("FED5000000000908");  byte[] header = strm.StrmPackagingFSDInit();  bos.write(header, 0, header.length);    int numRead;  while((numRead = bis.read(buffer, 0, buffer.length)) > 0)  {  outbuf = strm.StrmPackagingUpdate(buffer, numRead);  if (outbuf != null)  bos.write(outbuf, 0, outbuf.length);  }    outbuf = strm.StrmPackagingFinal(buffer, numRead);  if (outbuf != null)  bos.write(outbuf, 0, outbuf.length);    bis.close();  bos.close();  fis.close();  fos.close();    strm.ClearAll();  }  catch(IOException e)  {  System.out.println(e.getMessage());  }  }    public static void FileDecryptFSN(String oriFilePath, String encFilePath)  {  byte[] buffer = new byte[1024];  byte[] outbuf = null;    FileInputStream fis;  FileOutputStream fos;    try  {  fis = new FileInputStream(oriFilePath);  fos = new FileOutputStream(encFilePath);    BufferedInputStream bis = new BufferedInputStream(fis);  BufferedOutputStream bos = new BufferedOutputStream(fos);    StreamPackager strm = new StreamPackager();  strm.CreateObject();  strm.SetHomeDir("C:/init/fsdinit");  strm.SetServerID("FED5000000000908");  strm.StrmExtractInit();    int numRead;  while((numRead = bis.read(buffer, 0, buffer.length)) > 0)  {  outbuf = strm.StrmExtractUpdate(buffer, numRead);  if (outbuf != null)  bos.write(outbuf, 0, outbuf.length);  }    outbuf = strm.StrmExtractFinal(buffer, numRead);  if (outbuf != null)  bos.write(outbuf, 0, outbuf.length);    bis.close();  bos.close();  fis.close();  fos.close();    strm.ClearAll();  }  catch(IOException e)  {  System.out.println(e.getMessage());  }  }  public static void FileEncryptFSD(String oriFilePath, String encFilePath)  {  byte[] buffer = new byte[1024];  byte[] outbuf = null;    FileInputStream fis;  FileOutputStream fos;    FasooPackagerJNI jni = new FasooPackagerJNI();  byte[] byteHeader = jni.GetHeaderByte("C:/init/target/aaa\_fsd.txt");  if (byteHeader != null)  System.out.println("GetHeaderByte success.");  else  System.out.println("GetHeaderByte fail.");    try  {  fis = new FileInputStream(oriFilePath);  fos = new FileOutputStream(encFilePath);    BufferedInputStream bis = new BufferedInputStream(fis);  BufferedOutputStream bos = new BufferedOutputStream(fos);    StreamPackager strm = new StreamPackager();  strm.CreateObject();  strm.setStrmHeader(byteHeader);  strm.SetHomeDir("C:/init/fsdinit");  strm.SetServerID("FED5000000000910");  byte[] header = strm.StrmPackagingFSDInit();  if (header != null)  bos.write(header, 0, header.length);    int numRead;  while((numRead = bis.read(buffer, 0, buffer.length)) > 0)  {  outbuf = strm.StrmPackagingUpdate(buffer, numRead);  if (outbuf != null)  bos.write(outbuf, 0, outbuf.length);  }    outbuf = strm.StrmPackagingFinal(buffer, numRead);  if (outbuf != null)  bos.write(outbuf, 0, outbuf.length);    bis.close();  bos.close();  fis.close();  fos.close();    strm.ClearAll();  }  catch(IOException e)  {  System.out.println(e.getMessage());  }  }    public static void FileDecryptFSD(String oriFilePath, String encFilePath)  {  byte[] buffer = new byte[1024];  byte[] outbuf = null;    FileInputStream fis;  FileOutputStream fos;    try  {  fis = new FileInputStream(oriFilePath);  fos = new FileOutputStream(encFilePath);    BufferedInputStream bis = new BufferedInputStream(fis);  BufferedOutputStream bos = new BufferedOutputStream(fos);    StreamPackager strm = new StreamPackager();  strm.CreateObject();  strm.SetHomeDir("C:/init/fsdinit");  strm.SetServerID("FED5000000000910");  strm.StrmExtractInit();    int numRead;  while((numRead = bis.read(buffer, 0, buffer.length)) > 0)  {  outbuf = strm.StrmExtractUpdate(buffer, numRead);  if (outbuf != null)  bos.write(outbuf, 0, outbuf.length);  }    outbuf = strm.StrmExtractFinal(buffer, numRead);  if (outbuf != null)  bos.write(outbuf, 0, outbuf.length);    bis.close();  bos.close();  fis.close();  fos.close();    strm.ClearAll();  }  catch(IOException e)  {  System.out.println(e.getMessage());  }  }  } |

# Specification of Version

* **CLASS** : com.fasoo.fcwpkg.info.Version
* **Methods**

|  |  |
| --- | --- |
| public String GetVersion() | Jni / jar파일의 버전을 리턴한다.. |
| public String GetFasooPackagerVersion() | 패키져JNI dll,so의 버전을 리턴한다. |
| public String GetCryptoJNIVersion() | 크립토JNI dll,so의 버전을 리턴한다. |
| public String GetCryptoUtilJNIVersion() | 크립토UtilJNI dll,so의 버전을 리턴한다. |
| public String GetFasooUtilJNIVersion() | UtilJNI dll,so의 버전을 리턴한다. |

* **Sample**

*import com.fasoo.fcwpkg.info.\*;*

*public class GetVersion {*

*public static void main(String[] args) {*

*Version obj = new Version();*

*System.out.println("jar VER = [" + obj.GetVersion() + "]");*

*System.out.println("Packager VER = [" + obj.GetFasooPackagerVersion() + "]");*

*System.out.println("Ctypto VER = [" + obj.GetCryptoJNIVersion() + "]");*

*System.out.println("CryptoUtil VER = [" + obj.GetCryptoUtilJNIVersion() + "]");*

*System.out.println("FasooUtil VER = [" + obj.GetFasooUtilJNIVersion() + "]");*

*obj = null;*

*}*

*}*

# More Information

## **Error Code 참조 (FED4.0, FED5.0 공통)**

|  |  |  |
| --- | --- | --- |
| **CODE** | **Define** | **Description** |
| 3 | E\_SERVERID\_NOT\_EXIST | server id not present... |
| 4 | E\_INVALID\_PARAM | invalid function parameter(s)... |
| 16 | E\_REQ\_INFO\_NOT\_EXIST | user not set HomeDir or required property |
| 11 | E\_INVALID\_FILE\_EXT\_NSPRT | not supported source file extension |
| 20 | E\_FILE\_NOT\_EXIST | file not exists... |
| 21 | E\_FILE\_SIZE\_ZERO | file size is zero... |
| 22 | E\_FILE\_OPEN\_FAIL | file open fail... is file exist? |
| 23 | E\_FILE\_READ\_FAIL | read file fail... |
| 26 | E\_PACKAGED\_FILE | source file is packaged. repackaging is denied... |
| 27 | E\_OLD\_OLE\_FILE | source file is unsupported ole type. packaging is denied... |
| 29 | E\_NOT\_PACKAGED\_FILE | source file is not packaged. unpackaging is denied... |
| 40 | E\_CHECK\_CERTIFICATE\_FAIL | certificate is not existing, or invalid certificate... |
| 41 | E\_COCREATE\_INST\_FAIL | initialize COM object fail... |
| 51 | E\_DECODING\_FAIL | crypto decoding function fails... |
| 54 | E\_SYMMDEC\_FAIL | crypto symmetric decryption fails... |
| 70 | E\_MEMALLOCFAIL | memory allocation fails... |
| 72 | E\_NOT\_INITIALIZE | initialize function must call before this function call |

## **Error Code 참조 (FED5.0 전용)**

|  |  |  |
| --- | --- | --- |
| **CODE** | **Define** | **Description** |
| 1150 | SC\_FILE\_FED5\_PACKAGED | FED5.0 FSD 문서 |
| 1151 | SC\_FILE\_FED5\_FSN\_PACKAGED | FED5.0 FSN 문서 |
| 1152 | SC\_FILE\_FED5\_FSE\_PACKAGED | FED5.0 FSE 문서 |
| 1153 | SC\_FILE\_FED5\_NX\_PACKAGED | FED5.0 NX 문서 |
| 1166 | SC\_NOT\_SUPPORT\_RIGHT | 문서 타입 변환시 해당 문서에 ‘CONVERSION’ 또는 ‘SAVE’ 권한이 없을 경우 |
| 1167 | SC\_DISABLED\_LOGGING | 로그 전송 설정이 되어 있지 않거나 인증서가 설정되어 있지 않을 경우 |
| 1176 | SC\_NOT\_FED\_VERSION | conf 파일의 [PACKAGER] 필드에 VERSION 값이 존재하지 않을 경우 (FED4.0 conf 파일에는 해당값이 존재하지 않음) |
| 1178 | SC\_FAILED\_GET\_RIGHTS | 문서 복호화시 서버나 클라이언트에 복호화 권한 확인시 권한이 없거나 권한 확인에 실패할 경우 |
| 1180 | SC\_NOT\_SUPPORT\_ALGORITHM | 지원하지 않는 알고리즘 |
| 1201 | SC\_FAILED\_SECTION\_FIND | 문서 헤더에서 섹션을 찾는데 실패할 경우 |
| 1205 | SC\_FAILED\_HMAC\_COMPARE | 문서 헤더의 HMAC 확인에 실패할 경우 (문서 헤더 위변조 여부 확인) |
| 1214 | SC\_FAILED\_GET\_DOMAINKEY | 인증서에서 도메인키를 가져오는데 실패할 경우 |
| 1216 | SC\_DISCARD\_APPCERT | 폐기된 인증서를 호출할 경우 |
| 1217 | SC\_DISCARD\_DKCOUNT | 폐기된 도메인키 카운트를 적용할 경우 |
| 1500 | SC\_FAILED\_CERT\_SIGN | cert에 실패할 경우 |
| 1501 | SC\_FAILED\_GET\_KEY | 키를 가져오는데 실패할 경우 |
| 1502 | SC\_FAILED\_PFX\_LOAD | 인증서 로드에 실패할 경우 |
| 1503 | SC\_FAILED\_GET\_APPKEY | 응용서버 인증서 디코드에 실패할 경우 |

## **FSE Option String 참조**

|  |  |  |  |
| --- | --- | --- | --- |
| 권한 명 | 문자열 | 값 | 예 |
| 열람 권한 | can\_view | 0,1 | can\_view=1 |
| 인쇄 권한 | can\_print | 0,1 | can\_print=1 |
| 편집 권한 | can\_securesave | 0,1 | can\_save=1 |
| 암호화해제 권한 | can\_save | 0,1 | can\_securesave=1 |
| 화면 캡쳐 권한 | can\_printscreen | 0, 1 | can\_printscreen=1 |
| PC 대수 제한 | pc\_count | -1,1,2,3~ | pc\_count=-1  // -1 : 무제한, 1,2,3… |
| 유효 기간 제한 | valid\_days | -1,1,2,3~365 | valid\_days=-1  // -1 : 무제한, 1,2,3… |
| 열람 횟수 제한 | use\_count | -1,1,2,3~ | use\_count=-1  // -1 : 무제한, 1,2,3… |
| 개봉 확인 메일 | confirm\_mail | 0,1,2 | confirm\_mail=0  //외부 사용자에 대한 개봉 확인 메일 옵션  //0 : 확인 안 함, 1: 한번만 확인  2 : 매번 확인 |
| 개봉 확인 메일  - 사이트 커스터마이징 | confirm\_mail\_4in | 0,1,2 | confirm\_mail\_4in=0  //사이트 커스터마이징 속성  //0 : 확인 안 함, 1: 한번만 확인  2 : 매번 확인 |
| 이메일 인증 사용 | authentication | 0,1,2 | Authentication=2  0 : 인증 하지 않음  1 : 최초PC 자동인증  2 : 매번 인증 |
| 유효기간 지정 시작일 | start\_date | 2008-07-17 00:00:00 | start\_date=2008-07-17 00:00:00 |
| 유효기간 지정 종료일 | end\_date | 2008-09-30 23:59:59 | end\_date=2008-09-30 23:59:59 |
| 문서 사용 형태 | License\_type | On-line  Off-line | // on-line : 매번 서버 접속  // off-line : 처음 한번 서버 접속 |

<주의>

- 권한 문자열과 권한 문자열 사이는 “|” 구분자로 구분합니다.