```
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS
        AlgorithmIdentifier
        FROM MOJCMPSuspReq { 1 2 392 100300 1 4 21 }

        SubjectPublicKeyInfo, Name, RDNSequence
        FROM MOJCorpCertExplicit { 1 2 392 100300 1 4 1 };


CertTemplate ::= SEQUENCE {
    serialNumber [1] INTEGER,
    issuer       [3] Name }

EncryptedValue ::= SEQUENCE {
    keyAlg        [3] AlgorithmIdentifier,
    encValue       BIT STRING }

CertId ::= SEQUENCE {
    issuer           GeneralName,
    serialNumber     INTEGER }

GeneralName ::= CHOICE {
        directoryName               [4] Name  }

ReasonFlags ::= BIT STRING {
        certificateHold         (6) }

END
```

付録５　電子証明書に係る証明及びその請求の方式（送受信電文の ASN.１構造とオブジェクト識別子）

```
MOJOCSP { 1 2 392 100300 1 4 31 }

DEFINITIONS EXPLICIT TAGS ::=
BEGIN

IMPORTS

          Certificate
          FROM MOJCorpCertExplicit { 1 2 392 100300 1 4 1 };


OCSPRequest ::=  SEQUENCE {
    tbsRequest                  TBSRequest  }

TBSRequest ::= SEQUENCE {
    version             [0] EXPLICIT Version DEFAULT v1,
    requestList             SEQUENCE OF Request,
    requestExtensions   [2] EXPLICIT Extensions  -- nonce
}

Version ::= INTEGER  { v1(0) }

Request ::= SEQUENCE {
    reqCert                     CertID,
    singleRequestExtensions     [0] EXPLICIT Extensions OPTIONAL  -- confirmationTime
}

CertID ::= SEQUENCE {
    hashAlgorithm       AlgorithmIdentifier,
    issuerNameHash      OCTET STRING, -- Hash of Issuer's DN
    issuerKeyHash       OCTET STRING, -- Hash of Issuers public key
    serialNumber        CertificateSerialNumber }

CertificateSerialNumber ::= INTEGER

OCSPResponse ::= SEQUENCE {
   responseStatus         OCSPResponseStatus,
   responseBytes          [0] EXPLICIT ResponseBytes OPTIONAL }

OCSPResponseStatus ::= ENUMERATED {
    successful            (0),      --Response has valid confirmations
    malformedRequest      (1)       --Illegal confirmation request
}

ResponseBytes ::=       SEQUENCE {
    responseType   RESPONSE.&Type({SupportedResponses}),
    response       RESPONSE.&Value({SupportedResponses}{ @responceType })
}

RESPONSE ::= CLASS {
     &Type    OBJECT IDENTIFIER UNIQUE,
     &Value
}
WITH SYNTAX { SYNTAX &Value IDENTIFIED BY &Type }

SupportedResponses RESPONSE ::= { ..., -- extensible
                                  basicOCSPResponse }
```