```
    &Type
}
WITH SYNTAX { SYNTAX &Type IDENTIFIED BY &id }

InfoSet INFORMATION-ID ::= { genmInfoReqContent |
                             genpInfoResContent
}

genmInfoReqContent INFORMATION-ID ::= {
  SYNTAX         GenmInfoReqContent
  IDENTIFIED BY  id-registeredcert-mg-genminforeq
}

genpInfoResContent INFORMATION-ID ::= {
  SYNTAX         GenmInfoReqContent
  IDENTIFIED BY  id-registeredcert-mg-genpinfores
}

PKIMessage ::= SEQUENCE {
      header           PKIHeader,
      body             PKIBody,
      protection   [0] PKIProtection OPTIONAL,
      extraCerts   [1] SEQUENCE SIZE (1..MAX) OF Certificate OPTIONAL
}

PKIHeader ::= SEQUENCE {
      pvno                INTEGER     { ietf-version2 (1) },
      sender              GeneralName,
      recipient           GeneralName,
      protectionAlg   [1] AlgorithmIdentifier     OPTIONAL,
      senderKID       [2] KeyIdentifier           OPTIONAL,
      transactionID   [4] OCTET STRING,
      senderNonce     [5] OCTET STRING,
      recipNonce      [6] OCTET STRING            OPTIONAL
}

PKIBody ::= CHOICE {       -- message-specific body elements
      ir      [0]  CertReqMessages,        --Initialization Request
      ip      [1]  CertRepMessage,         --Initialization Response
      genm    [21] GenMsgContent,          --General Message
      genp    [22] GenRepContent,          --General Response
      error   [23] ErrorMsgContent         --Error Message
}

PKIProtection ::= BIT STRING

ProtectedPart ::= SEQUENCE {
      header    PKIHeader,
      body      PKIBody
}

KeyIdentifier ::= OCTET STRING

PKIStatus ::= INTEGER {
      granted                (0),
      -- you got exactly what you asked for
      rejection              (2)
      -- you don't get it, more information elsewhere in the message
}

PKIStatusInfo ::= SEQUENCE {
      status        PKIStatus
}

CertRepMessage ::= SEQUENCE {
      response         SEQUENCE OF CertResponse
}

CertResponse ::= SEQUENCE {
      certReqId           INTEGER,
      status              PKIStatusInfo,
      certifiedKeyPair    CertifiedKeyPair
}

CertifiedKeyPair ::= SEQUENCE {
      certOrEncCert       CertOrEncCert
}

CertOrEncCert ::= CHOICE {
      encryptedCert   [1] EncryptedValue
}

GenMsgContent ::= SEQUENCE OF InfoTypeAndValue

GenRepContent ::= SEQUENCE OF InfoTypeAndValue

ErrorMsgContent ::= SEQUENCE {
      pKIStatusInfo          PKIStatusInfo
}

id-registeredcert OBJECT IDENTIFIER ::= { 1 2 392 100300 1 }
```