```
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

--EXPORTS ALL --

IMPORTS
        Name, CertificateSerialNumber, DirectoryString, EXTENSION
        FROM MOJCorpCertExplicit { 1 2 392 100300 1 4 1 };


authorityKeyIdentifier EXTENSION ::= {
        SYNTAX          AuthorityKeyIdentifier
        IDENTIFIED BY   id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier               [0] KeyIdentifier,
    authorityCertIssuer         [1] GeneralNames,
    authorityCertSerialNumber   [2] CertificateSerialNumber }

KeyIdentifier ::= OCTET STRING

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
        directoryName               [4] Name,
        uniformResourceIdentifier   [6] IA5String
}

subjectKeyIdentifier EXTENSION ::= {
        SYNTAX          SubjectKeyIdentifier
        IDENTIFIED BY   id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier

keyUsage EXTENSION ::= {
        SYNTAX KeyUsage
        IDENTIFIED BY id-ce-keyUsage }

KeyUsage ::= BIT STRING {
        digitalSignature     (0),
        keyEncipherment      (2),
        dataEncipherment     (3),
        keyCertSign          (5),
        cRLSign              (6)
}
```

```
privateKeyUsagePeriod EXTENSION ::= {
        SYNTAX  PrivateKeyUsagePeriod
        IDENTIFIED BY { id-ce-privateKeyUsagePeriod } }

PrivateKeyUsagePeriod ::= SEQUENCE {
        notBefore       [0]     GeneralizedTime,
        notAfter        [1]     GeneralizedTime }

certificatePolicies EXTENSION ::= {
        SYNTAX  CertificatePoliciesSyntax
        IDENTIFIED BY id-ce-certificatePolicies }

CertificatePoliciesSyntax ::=
                SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
        policyIdentifier   CertPolicyId,
        policyQualifiers   SEQUENCE SIZE (1..MAX) OF PolicyQualifierInfo }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
        policyQualifierId       CERT-POLICY-QUALIFIER.&id
                                    ({SupportedPolicyQualifiers}),
        qualifier               CERT-POLICY-QUALIFIER.&Qualifier
                                    ({SupportedPolicyQualifiers}
                                    {@policyQualifierId}) }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { noticeToUser }

CERT-POLICY-QUALIFIER ::= CLASS {
        &id             OBJECT IDENTIFIER UNIQUE,
        &Qualifier  }
WITH SYNTAX {
        POLICY-QUALIFIER-ID     &id
        QUALIFIER-TYPE          &Qualifier }

noticeToUser CERT-POLICY-QUALIFIER ::= {
    POLICY-QUALIFIER-ID id-qt-unotice  QUALIFIER-TYPE   UserNotice }

UserNotice ::= SEQUENCE {
    noticeRef       NoticeReference,
    explicitText    DisplayText }

NoticeReference ::= SEQUENCE {
```