-- 9 ASN.1 DEFINITIONS
--
-- This section describes INAP ASN.1 definition in SDF-SDF interface. All the definitions
-- are defined by subtyping X.500 related definitions or by using them. The modules in this
-- section are only for information, and the syntax is not assured.
--
-- 9.1 INAP DEFINED BY SUBTYPED X.500 RELATED DEFINITION
--
-- This clause doesn't specify the protocol definition, This functional subset of X.500 is used
-- in this IN capability set. The following module is only for information, and the syntax is
-- not assured.

```
IN-CS3-SDF-SDF-Operations
        {ccitt recommendation q 1238 modules(1) in-cs3-sdf-sdf-ops-args(18) version1(0)}

DEFINITIONS ::=

BEGIN

IMPORTS

ds-UsefulDefinitions, ros-InformationObjects, scf-sdf-Additional-Definitions
        FROM IN-CS3-object-identifiers
                { ccitt recommendation q 1238 modules(1) in-cs3-object-identifiers(0) version1(0) }

directoryShadowAbstractService, disp, distributedOperations, enhancedSecurity
        FROM  UsefulDefinitions ds-UsefulDefinitions

coordinateShadowUpdateArgument, coordinateShadowUpdateResult,
updateShadowArgument, updateShadowResult, requestShadowUpdateArgument,
requestShadowUpdateResult, shadowError, refreshInformation, shadowingAgreementInfo
        FROM DirectoryShadowAbstractService directoryShadowAbstractService

id-opcode-coordinateShadowUpdate, id-opcode-updateShadow,  id-opcode-RequestShadowUpdate
        FROM DirectoryInformationShadowProtocol disp

chainingArguments, chainingResults, referenceType, dsaReferral, continuationReference, accessPoint,
masterOrShadowAccessPoint, accessPointInformation
        FROM DistributedOperations distributedOperations

OPTIONALLY-PROTECTED{}, DIRQOP
        FROM EnhancedSecurity enhancedSecurity

OPERATION
        FROM Remote-Operations-Information-Objects ros-InformationObjects

execute, in-AddEntry, in-DirectoryBind, in-RemoveEntry, in-Search, in-ModifyEntry
        FROM IN-CS3-SCF-SDF-Additional-Definitions scf-sdf-Additional-Definitions
;

-- IN X.500 DISP definition
IN-ShadowingAgreementInfo ::= ShadowingAgreementInfo (
        WITH COMPONENTS {
                ...,
                master                  ABSENT,
                secondaryShadows        ABSENT})
-- shadowSubject specifies the subtree, entries and attributes to shadow. The components of
-- UnitOfReplicationare defined in section 9.2 of Recommendation X.525.
-- updateMode specifies when updates of a shadowed area are scheduled to occur. The components of
-- updateMode are defined in section 9.3 of Recommendation X.525.
```

```
-- master contains the access point of the DSA containing the mastered area. "As this information is
-- already known by the DSA it is not required for IN."
-- secondaryShadows permits secondary shadow information to be subsequently supplied to the shadow
-- supplier.
-- The secondary shadows are ignored in the IN context (assumption 5), then this component should not
-- be included.


        in-DSAShadowBind OPERATION ::= in-DirectoryBind
-- An IN-DSAShadowBind operation is used at the beginning of a period of providing shadows.

        in-CoordinateShadowUpdate  OPERATION ::= {
                ARGUMENT        IN-CoordinateShadowUpdateArgument
                RESULT          IN-CoordinateShadowUpdateResult
                ERRORS          {shadowError}
                CODE            id-opcode-coordinateShadowUpdate}
-- The IN-CoordinateShadowUpdate operation is used by the shadow supplier to indicate the shadowing
-- agreement for which it intends to send updates.

        IN-CoordinateShadowUpdateArgument ::= CoordinateShadowUpdateArgument (
                WITH COMPONENTS {
                        ...,
                        updateStrategy      (standard:{total | incremental})})
        IN-CoordinateShadowUpdateResult     ::=     CoordinateShadowUpdateResult(
                WITH COMPONENTS {
                        ...,
                        null            PRESENT})
-- The various parameters have the meanings defined below:
-- a)       The agreementID argument identifies the shadowing agreement.
-- b)       The lastUpdate argument indicates the shadow supplier's understanding of the time at which
--          the last update for this agreement was sent and is the time as provided by the shadow supplier
--          DSA. This argument may only be omitted in the first instance of either a
--          IN-CoordinateShadowUpdate or IN-RequestShadowUpdate operation for a particular
 --         shadowing agreement
-- c)       The updateStrategy argument identifies the update strategy the shadow supplier intends to use
--          for this update. For IN CS-3, a total or incremental replacement strategy should be used.
--          The "NoChanges" option will not be used.
-- d)       The securityParameters argument is defined in 7.10 of ITU-T Rec. X.511 | ISO/IEC 9594-3.


        in-UpdateShadow  OPERATION ::= {
                ARGUMENT        IN-UpdateShadowArgument
                RESULT          IN-UpdateShadowResult
                ERRORS          {shadowError}
                CODE            id-opcode-updateShadow}
-- An IN-UpdateShadow operation is invoked by the shadow supplier to send updates to the shadow
-- consumer for a unit of replication. Prior to this operation being initiated, a
-- IN-CoordinateShadowUpdate or IN-RequestShadowUpdate operation must have been successfully
-- completed for the identified shadowing agreement.

        IN-UpdateShadowArgument ::= UpdateShadowArgument (
                WITH COMPONENTS {
                        ...,
                        updatedInfo         (IN-RefreshInformation)})
        IN-UpdateShadowResult  ::=      UpdateShadowUpdateResult(
                WITH COMPONENTS {
                        ...,
                        null            PRESENT})
-- The various parameters have the meanings as defined below:
-- a)       The agreementID identifies the shadowing agreement that has been established.
-- b)       The updateTime argument is supplied by the shadow supplier. This time is used during the
--          next IN-CoordinateShadowUpdate or IN-RequestShadowUpdate to ensure that the shadow
--          supplier and shadow consumer have a common view of the shadowed information.
-- c)       The updateWindow argument, when present, indicates the next window during which the
```

```
--          shadow supplier expects to send an update.
-- d)       The updatedInfo argument provides the information required by the shadow consumer to
--          update its shadowed information. The semantics of the information conveyed in this parameter
--          shall result in the shadow consumer reflecting the changes supplied.
-- e)       The securityParameters argument is defined in 7.10 of ITU-T Rec. X.511 | ISO/IEC 9594-3.

        IN-RefreshInformation ::= RefreshInformation (
                WITH COMPONENTS {
                        ...,
                        otherStrategy               ABSENT})
-- The various parameters have the meanings as defined below:
-- a)       noRefresh indicates that there have been no changes to the shadowed information from the
--          previous instance to the present. This may be used where an IN-UpdateShadow operation
--          must be supplied at a certain interval defined in the shadowing agreement (updateMode), but
--          no modification has actually occurred.
-- b)       total provides a new instance of the shadowed information. The incremental strategy should
--          be preferably used because it saves signalling.
-- c)       incremental provides, instead of a complete replacement of the shadowed information, only
--          the changes which have occurred to that shadowed information between lastUpdate in the
--          most recent IN-CoordinateShadowUpdate (or IN-RequestShadowUpdate) request and
--          updateTime in the current IN-UpdateShadow request (or IN-RequestShadowUpdate
--          response).
-- d)       otherStrategy provides the ability to send updates by mechanisms outside the scope of the
--          Directory Specification. For IN CS-3, either a total or incremental strategy should be used.
--          Should the request succeed, a result will be returned, although no information will be
--          conveyd with it. Should the request fail, a shadowError shall be reported. Circumstances
--          under which the particular shadow problems will be returned are defined in X.525 Section
--          11.3.3.


        in-RequestShadowUpdate  OPERATION ::= {
                ARGUMENT        IN-RequestShadowUpdateArgument
                RESULT          IN-RequestShadowUpdateResult
                ERRORS          {shadowError}
                CODE            id-opcode-RequestShadowUpdate}
-- An IN-RequestShadowUpdate operation is used by the shadow consumer to request updates from the
-- shadow supplier.

        IN-RequestShadowUpdateArgument ::= RequestShadowUpdateArgument (
                WITH COMPONENTS {
                        ...,
                        requestedStrategy   (standard:{incremental | total})})
        IN-RequestShadowUpdateResult ::= RequestShadowUpdateResult(
                WITH COMPONENTS {
                        ...,
                        null            PRESENT})
-- The various parameters have the meanings as defined below:
-- a)       The agreementID identifies the shadowing agreement.
-- b)       The lastUpdate argument is the time provided by the shadow supplier in the most recent
--          successful update. This argument may only be omitted in the first instance of either a
--          IN-CoordinateShadowUpdate or IN-RequestShadowUpdate operation for a particular
--          shadowing agreement.
-- c)       The requestedStrategy argument identifies the type of update being requested by the shadow
--          consumer. The shadow consumer may request either an incremental or a total update from
--          the shadow supplier.
-- d)       The securityParameters argument is defined in 7.10 of ITU-T Rec. X.511 | ISO/IEC 9594-3.


-- IN X.500 DSP definition

        IN-ChainingArguments ::= ChainingArguments (
                WITH COMPONENTS {
                        ...,
```

```
                aliasDereferenced       ABSENT,
                aliasedRDNs             ABSENT,
                returnCrossRefs         ABSENT,
                info                    ABSENT,
                timeLimit               ABSENT,
                excludeShadows          ABSENT,
                nameResolveOnMaster     ABSENT})
-- The IN-ChainingArguments are present in each chained operation, to convey to a DSA the
-- informatio needed to successfully perform its part of the overall task:


    IN-ChainingResults ::= ChainingResults (
        WITH COMPONENTS {
                ...,
                info                ABSENT,
                crossReferences     ABSENT })
-- The ChainingResults are present in the result of each operation and provide feedback to
-- the DSA which invoked the operation.


    IN-ReferenceType ::= ReferenceType (1|2|4|5|6|7|8)
-- A ReferenceType value indicates one of the various kinds of reference defined in ITU-T
-- Rec. X.501 | ISO/IEC 9594-2.
-- Value (3)(cross reference) is not applicable for IN CS-3 as direct references are assumed.

    IN-AccessPoint ::= AccessPoint (
        WITH COMPONENTS {
                ...,
                protocolInformation     ABSENT})
-- An AccessPoint value identifies a particular point at which access to the Directory, specifically to a
-- DSA, can occur. The access point has a Name, that of the DSA concerned,
-- and a PresentationAddress,
-- to be used in SS7 signalling to that DSA.

-- The address contains the network address of the DSA in the SS7.

    IN-MasterOrShadowAccessPoint ::= MasterOrShadowAccessPoint (
        WITH COMPONENTS {
                ...,
                COMPONENTS OF IN-AccessPoint})
-- A MasterOrShadowAccessPoint value identifies an access point to the Directory.
-- The category, either master or shadow, of the access point is dependent upon whether it points to a
-- naming context or commonly useable replicated area.

    IN-MasterAndShadowAccessPoints      ::= MasterOrShadowAccessPoint
-- A MasterAndShadowAccessPoints value identifies a set of access points to the Directory, i.e., a set of
-- related DSAs. These access points share the property that each refers to a DSA holding entry
-- information from a common naming context (or a common set of naming contexts mastered in one
-- DSA when the value is a value of the nonSpecificKnowledge attribute.
-- A MasterAndShadowAccessPoints value indicates the category of each AccessPoint value it contains.
-- The access point of the master DSA of the naming context need not be included in the set.

-- An AccessPointInformation value identifies one or more access points to the Directory.
    IN-AccessPointInformation ::= AccessPointInformation (
        WITH COMPONENTS {
                ...,
                COMPONENTS OF IN-MasterOrShadowAccessPoint })


    IN-ContinuationReference        ::=     ContinuationReference (
        WITH COMPONENTS {
                ...,
                aliasedRDNs             ABSENT,
                rdnsResolved            ABSENT,
```

```
                referenceType           (IN-ReferenceType),
                accessPoints            SET OF (IN-AccessPoint))
-- A ContinuationReference describes how the performance of all or part of an operation can be
-- continued at a different DSA or DSAs. It is typically returned as a referral when the DSA involved is
-- unable or unwilling to propagate the request itself.
-- The various components have the meanings as defined below:
-- a)   The targetObject name indicates the name which is proposed to be used in continuing the
--      operation. This might be different from the targetObject name received on the incoming request
--      if, for example, an alias has been dereferenced, or the base object in a search has been located.
-- b)   The aliasedRDNs component indicates how many (if any) of the RDNs in the target object
--      name have been produced by dereferencing an alias. Since alias entries in IN are just a
--      means to provide an alternative name for an object and therefore should be dereferenced
--      when needed, there is no need for this indicator.
-- c)   The operationProgress indicates the amount of name resolution which has been achieved, and
--      which will govern the further performance of the operation by the DSAs named, should the
--      DSA or DUA receiving the ContinuationReference wish to follow it up.
-- d)   The rdnsResolved component value (which need only be present if some of the RDNs in the
--      name have not been the subject of full name resolution, but have been assumed to be correct
--      from a cross reference) indicates how many RDNs have actually been resolved, using internal
--      references only. Since direct knowledge references are assumed, this parameter is deemed not
--      applicable for IN CS-3.
-- e)   The referenceType component indicates what type of knowledge was used in generating this
--      continuation.
-- f)   The accessPoints component indicates the access points which are to be contacted to achieve
--      this continuation. Only where non-specific subordinate references are involved can there be
--      more than one AccessPointInformation item.
-- g)   The entryOnly component is set to TRUE if the original operation was a search, with the
--      subset argument set to oneLevel, and an alias entry was encountered as an immediate
--      subordinate of the baseObject. The DSA which successfully performs name resolution on the
--      targetObject name, shall perform object evaluation -- on only the named entry. Since alias
--      entries in IN are just a means to provide an alternative name for an object and therefore
--      should be dereferenced when needed, there is no need for this indicator.
-- h)   The exclusions component identifies a set of subordinate naming contexts that should not be
--      explored by the receiving DSA.
-- i)   The returnToDUA element is optionally supplied when the DSA creating the continuation
--      reference wishes to indicate that it is unwilling to return information via an intermediate DSA
--      (e.g., for security reasons), and wishes to indicate that information may be directly available
--      via an operation over DAP between the originating DUA and the DSA. When returnToDUA is
--      set to TRUE, referenceType may be set to self. This element may be used in IN for support of
--      the shadowing agreement established between network operators (e.g., SDF_v to SDF_h Modify
--      may fail based upon access control restrictions).
-- j)   The nameResolveOnMaster element is optionally supplied when the DSA creating the
--      continuation reference has encountered NSSRs. Since direct knowledge references are
--      assumed, this parameter is deemed not applicable for IN CS-3.

    in-DSABind OPERATION ::= in-DirectoryBind
-- An IN-DSABind operation is used to begin of a period of cooperation between two DSAs providing
-- the Directory service.
-- A DSA, having received an operation from a DUA, may elect to construct a chained form of that
-- operation to propagate to another DSA. For IN CS-3 a DSA, having received a chained form of an
-- operation, must either process the operation or if the originating DSA is in another network, chain it
-- to another DSA within the same network as the receiving DSA. The DSA invoking a chained form of
-- an operation may optionally sign the argument of the operation; the DSA performing the operation,
--if so requested, may sign the result of the operation.
-- The chained form of an operation is specified using the parameterized type IN-chained {}.
    in-Chained { OPERATION : operation } OPERATION      ::= {
        ARGUMENT        OPTIONALLY-PROTECTED { SET {
            in-chainedArgument      IN-ChainingArguments,
            argument            [0] operation.&ArgumentType },
            DIRQOP.&dspChainedOp-QOP@dirqop}
        RESULT      OPTIONALLY-PROTECTED { SET {
            in-chainedResult        IN-ChainingResults,
```

```
            result              [0]     operation.&ResultType },
            DIRQOP.&dspChainedOp-QOP@dirqop }
        ERRORS      { operation.&Errors EXCEPT (referral | in-DSAReferral) }
        CODE        operation.&code }

    chainedExecute OPERATION ::= in-Chained { execute }

    in-ChainedSearch OPERATION ::= in-Chained { in-Search }

    in-ChainedAddEntry OPERATION ::= in-Chained { in-AddEntry }

    in-ChainedRemoveEntry OPERATION ::= in-Chained { in-RemoveEntry }

    in-ChainedModifyEntry OPERATION ::= in-Chained { in-ModifyEntry }
-- Errors in DSP

    in-DSAReferral ::= dsaReferral(WITH COMPONENTS{
        ...,
        reference     (IN-ContinuationReference),
        contextPrefix         ABSENT})

-- The IN-DSAReferral error is generated by the DSA when, for whatever reason, it doesn't wish to
-- continue performing an operation by chaining the operation to another DSA. For this CS, DSAs may
-- not chain operations incoming from another DSA unless the DSA is in another network.
-- The various parameter have the meanings as described below:
-- a) The IN-ContinueReference contains the information needed by the invoker to propagate
-- an appropriate further request, perhaps to another DSA.
-- b) If the returnCrossRefs component of the ChainingArguments for this operation had the value
-- TRUE, and the referral is being based upon a subordinate or cross-reference, then the contextPrefix
-- parameter may optionnally be included. The administrative authority of any DSA will decide which
-- knowledge references, if any, can be returned in this manner (the others, for example, may be
-- confidential to that DSA). Since direct knowledge references are assumed for IN-CS3, this parameter
-- is not applicable.

-- IN DISP/DSP common Operation definition

-- inUnbind  OPERATION
-- This operation is described in Q.1238.1.
-- The INUnbind operation replaces the X.525 dSAShadowUnbind operation and the X.518 dSAUnbind
-- operation to provide class 4 operation behaviour for unbind procedures.
-- This operation is also used to unbind the TFCBind.

-- trafficFlowControl OPERATION

-- This operation is described in Q.1238.4.

-- tfcBind OPERATION

-- This operation is described in Q.1238.4.

END

-- The table below lists operation timer value range. The definitive timer value may
-- be network specific and has to be defined by the network operator.
--
--          Table 2/Q.1238.5 – Operation timer values
--          Operation                       Timer value range
--          - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
--          IN-ChainedAddEntry              medium
```

```
--              IN-ChainedRemoveEntry           medium
--              IN-ChainedModifyEntry           medium
--              ChainedExecute                  medium
--              IN-ChainedSearch                medium
--              IN-UpdateShadow                 medium
--              IN-CoordinateShadowUpdate       medium
--              IN-RequestShadowUpdate          medium
--              TrafficFlowControl              medium
--                                              medium :   1 - 60 second
```

-- 9.2 CONTRACTS, PACKAGES, AND ABSTRACT SYNTAXES

```
IN-CS3-SDF-SDF-Protocol
        {ccitt recommendation q 1238 modules(1) in-cs3-sdf-sdf-pkgs-contracts-acs(19) version1(0)}

DEFINITIONS ::=

BEGIN

IMPORTS

ds-UsefulDefinitions, ros-InformationObjects, ros-genericPDUs, tc-Messages,
tc-NotationExtensions,
common-classes, scf-sdf-Operations, scf-sdf-Protocol,
id-contract-indsp, id-contract-shadowConsumer, id-contract-shadowSupplier,
id-package-dspConnection, id-package-inchainedModify, id-package-inchainedSearch,
id-package-chainedExecute, id-package-dispConnection, id-package-shadowConsumer,
id-package-shadowSupplier,
id-as-indirectorySystemAS, id-as-indirectoryDSABindingAS, id-as-indirectoryShadowAS,
id-as-indsaShadowBindingAS,
id-ac-indirectorySystemAC, id-ac-inShadowSupplierInitiatedAC,
id-ac-inShadowConsumerInitiatedAC,
id-ac-inShadowSupplierInitiatedWith3seAC,id-ac-inShadowConsumerInitiatedWith3seAC,
id-ac-indirectorySystemWith3seAC
        FROM IN-CS3-object-identifiers
                {ccitt recommendation q 1238 modules(1) in-cs3-object-identifiers(0) version1(0)}

distributedOperations, directoryShadowAbstractService, protocolObjectIdentifiers
        FROM UsefulDefinitions ds-UsefulDefinitions

id-se-threewayse
       FROM ProtocolObjectIdentifiers protocolObjectIdentifiers

CONTRACT, OPERATION-PACKAGE, CONNECTION-PACKAGE, OPERATION
        FROM Remote-Operations-Information-Objects ros-InformationObjects

Bind{}, Unbind{}
        FROM Remote-Operations-Generic-ROS-PDUs ros-genericPDUs

TCMessage {}
        FROM TCAPMessages tc-Messages

APPLICATION-CONTEXT, dialogue-abstract-syntax
        FROM TC-Notation-Extensions tc-NotationExtensions

dSABind, chainedSearch, chainedAddEntry, chainedRemoveEntry, chainedModifyEntry, chained{}
        FROM DistributedOperations distributedOperationsdSAShadowBind, coordinateShadowUpdate,

updateShadow, requestShadowUpdate
        FROM DirectoryShadowAbstractService directoryShadowAbstractService
```

```
execute        FROM IN-CS3-SCF-SDF-Operations scf-sdf-Operations

inUnbind
        FROM IN-CS3-common-classes common-classes

inSESEAbstractSyntax
        FROM IN-CS3-SCF-SDF-Protocol scf-sdf-Protocol
;

-- DSP Contract and Packages

indspContract  CONTRACT ::= {
        CONNECTION                  dspConnectionPackage
        INITIATOR CONSUMER OF       { inChainedModifyPackage | inChainedSearchPackage |
                                      chainedExecutePackage }
        ID                          id-contract-indsp}
-- In the indspContract either DSA may assume the role of initiator and invoke the operations of the
-- contract.
-- When a pair of DSAs from different open systems interact, this association contract is realised as an
-- SS7 application layer protocol, referred to as the IN Directory System Protocol (DSP). The definition
-- of this protocol in terms of an SS7 application context is provided in this module.

dspConnectionPackage  CONNECTION-PACKAGE ::= {
        BIND        dSABind
        UNBIND      inUnbind
        ID          id-package-dspConnection}
-- This connection package is identical to the dapConnectionPackage in Q.1238.4.

inChainedModifyPackage  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES  {chainedAddEntry | chainedRemoveEntry |
                           chainedModifyEntry}
        ID                id-package-inchainedModify}

inChainedSearchPackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES  {chainedSearch}
        ID                id-package-inchainedSearch}

chainedExecutePackage OPERATION-PACKAGE ::= {
        CONSUMER INVOKES  { chainedExecute }
        ID                id-package-chainedExecute}

chainedExecute OPERATION ::= chained { execute }

-- DISP Contracts and Packages

shadowConsumerContract  CONTRACT ::= {
        CONNECTION                  dispConnectionPackage
        INITIATOR CONSUMER OF       {shadowConsumerPackage}
        ID                          id-contract-shadowConsumer}

shadowSupplierContract  CONTRACT ::= {
        CONNECTION                  dispConnectionPackage
        RESPONDER CONSUMER OF       {shadowSupplierPackage}
        ID                          id-contract-shadowSupplier}

-- Since the shadow consumer is the initiator of the ShadowConsumerContract, it assumes the role of
-- consumer of the shadowConsumerPackage. This means that the shadow consumer invokes the
-- IN-RequestShadowUpdate operation and that the shadow supplier invokes the IN-UpdateShadow
-- operation.
-- Since the shadow supplier is the initiator of the shadowSupplierContract, it assumes the role of
-- supplier of the shadowSupplierPackage. This means that the shadow supplier invokes the
-- operations of the contract.
-- The SS7 realisation of the two forms of Shadow Abstract Service, referred to as the IN
```

```
-- Directory Information Shadowing Protocol (DISP) are defined in terms of several SS7
-- application contexts provided in this module.

dispConnectionPackage  CONNECTION-PACKAGE ::= {
        BIND        dSAShadowBind
        UNBIND      inUnbind
        ID          id-package-dispConnection}
-- This connection package is identical to the dapConnectionPackage in Q.1238.4.

shadowConsumerPackage  OPERATION-PACKAGE ::= {
        CONSUMER INVOKES  {requestShadowUpdate}
        SUPPLIER INVOKES  {updateShadow}
        ID                id-package-shadowConsumer}
shadowSupplierPackage  OPERATION-PACKAGE ::= {
        SUPPLIER INVOKES  {coordinateShadowUpdate | updateShadow}
        ID                id-package-shadowSupplier}

-- DSP Abstract-Syntaxes

-- The Directory ASEs that realise the DSP operation packages specified in the above share the
-- following single abstract syntax.
inDirectorySystemAbstractSyntax  ABSTRACT-SYNTAX ::= {
        BasicDSP-PDUs
        IDENTIFIED BY       id-as-indirectorySystemAS}

BasicDSP-PDUs ::= TCMessage {{DSP-Invokable},{DSP-Returnable}}

DSP-Invokable OPERATION ::= {chainedAddEntry | chainedRemoveEntry |
                             chainedModifyEntry | chainedSearch | chainedExecute }

DSP-Returnable OPERATION ::= {chainedAddEntry | chainedRemoveEntry |
                             chainedModifyEntry | chainedSearch | chainedExecute }

-- The realisation of the DSP connection package specified in the above uses a separate abstract syntax
-- as follows.
inDirectoryDSABindingAbstractSyntax  ABSTRACT-SYNTAX ::= {
        DSABinding-PDUs
        IDENTIFIED BY       id-as-indirectoryDSABindingAS}

DSABinding-PDUs ::= CHOICE {
        bind        Bind {dSABind},
        unbind      Unbind {inUnbind}}

-- DISP Abstaract-Syntaxes

-- The Directory ASEs that realise the DISP operation packages specified in the above share the
-- following abstract syntax.
inDirectoryShadowAbstractSyntax ABSTRACT-SYNTAX ::= {
        BasicDISP-PDUs
        IDENTIFIED BY       id-as-indirectoryShadowAS}

BasicDISP-PDUs ::= TCMessage {{DISP-Invokable},{DISP-Returnable}}

DISP-Invokable OPERATION ::={requestShadowUpdate | updateShadow |
                             coordinateShadowUpdate}

DISP-Returnable OPERATION ::={requestShadowUpdate | updateShadow |
                             coordinateShadowUpdate}

-- The realisation of the DISP connection package specified in the above uses a separate abstract
-- syntax as follows.
inDirectoryDSAShadowBindingAbstractSyntax  ABSTRACT-SYNTAX ::= {
        DISPBinding-PDUs
```

```
        IDENTIFIED BY           id-as-indsaShadowBindingAS}

DISPBinding-PDUs ::= CHOICE {
        bind            Bind {dSAShadowBind},
        unbind          Unbind {inUnbind}}

-- DSP Application Contexts

inDirectorySystemAC APPLICATION-CONTEXT ::=     {
        CONTRACT                        indspContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        inDirectorySystemAbstractSyntax |
                                        inDirectoryDSABindingAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-indirectorySystemAC}
-- The inDirectorySystemAC is used for the realisation of indspContract.

inDirectorySystemWith3seAC APPLICATION-CONTEXT ::=    {
        CONTRACT                        indspContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASES                 {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        inDirectorySystemAbstractSyntax |
                                        inDirectoryDSABindingAbstractSyntax |
                                        inSESEAbstractSyntax }
        APPLICATION CONTEXT NAME        id-ac-indirectorySystemWith3seAC}
-- The inDirectorySystemWith3seAC is used for the realisation of indspContract when 3-way
-- authentication is required.

-- DISP Application Contexts

inShadowSupplierInitiatedAC APPLICATION-CONTEXT ::= {
        CONTRACT                        shadowSupplierContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        inDirectoryShadowAbstractSyntax |
                                        inDirectoryDSAShadowBindingAbstractSyntax}
        APPLICATION CONTEXT NAME        id-ac-inShadowSupplierInitiatedAC}
-- The inShadowSupplierInitiatedAC is used for the realisation of shadowSupplierContract.

inShadowSupplierInitiatedWith3seAC APPLICATION-CONTEXT ::= {
        CONTRACT                        shadowSupplierContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ADDITIONAL ASES                 {id-se-threewayse}
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                          inDirectoryShadowAbstractSyntax |
                                          inDirectoryDSAShadowBindingAbstractSyntax |
                                          inSESEAbstractSyntax }
        APPLICATION CONTEXT NAME        id-ac-inShadowSupplierInitiatedWith3seAC}
-- The inShadowSupplierInitiatedWith3seAC is used for the realisation of shadowSupplierContract
-- when 3-way authentication is required.

inShadowConsumerInitiatedAC APPLICATION-CONTEXT ::= {
        CONTRACT                        shadowConsumerContract
        DIALOGUE MODE                   structured
        TERMINATION                     basic
        ABSTRACT SYNTAXES               {dialogue-abstract-syntax |
                                        inDirectoryShadowAbstractSyntax |
                                        inDirectoryDSAShadowBindingAbstractSyntax}
```

```
        APPLICATION CONTEXT NAME      id-ac-inShadowConsumerInitiatedAC}
-- The inShadowConsumerInitiatedAC is used for the realisation of shadowConsumerContract.

inShadowConsumerInitiatedWith3seAC APPLICATION-CONTEXT ::= {
        CONTRACT                      shadowConsumerContract
        DIALOGUE MODE                 structured
        TERMINATION                   basic
        ADDITIONAL ASES               {id-se-threewayse}
        ABSTRACT SYNTAXES             {dialogue-abstract-syntax |
                                       inDirectoryShadowAbstractSyntax |
                                       inDirectoryDSAShadowBindingAbstractSyntax |
                                       inSESEAbstractSyntax }
        APPLICATION CONTEXT NAME      id-ac-inShadowConsumerInitiatedWith3seAC}
-- The inShadowConsumerInitiatedWith3seAC is used for the realisation of
--shadowConsumerContract when 3-way authentication is required.


-- DSP/DISP Common Application Context

-- trafficFlowControlAC APPLICATION-CONTEXT
-- This is defined in Q.1238.4.
-- Contracts, packages, and abstract syntaxes used in the trafficFlowControlAC is also shown in Q.1238.4.

END
```