

```

--
-- 12  ASN.1 DEFINITION
--
-- 12.1  IN CS-3 TYPES
--
-- 12.1.1          DATA TYPES

IN-CS3-scf-srf-datatypes {ccitt recommendation q 1238 modules(1) in-cs3-scf-srf-datatypes(10) version1(0)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

    tc-Messages,
    common-classes,
    common-datatypes,
    ssf-scf-classes,
    scf-srf-classes,
    ssf-scf-datatypes,
    ros-InformationObjects

FROM IN-CS3-object-identifiers
    { ccitt recommendation q 1238 modules(1) in-cs3-object-identifiers(0) version1(0) }

    EXTENSION,
    SupportedExtensions { },
    COMMON-BOUNDS

FROM IN-CS3-common-classes common-classes

    Integer4
FROM IN-CS3-common-datatypes common-datatypes

    Code
FROM Remoteoperations-Information-Objects ros-InformationObjects

    Digits {},
    DisplayInformation {},
    SDSSinformation {}

FROM IN-CS3-SSF-SCF-datatypes ssf-scf-datatypes

    SCF-SSF-BOUNDS
FROM IN-CS3-SSF-SCF-classes ssf-scf-classes

    SCF-SRF-BOUNDS

FROM IN-CS3-scf-srf-classes scf-srf-classes
;

CollectedDigits ::= SEQUENCE {
    minimumNbOfDigits      [0] INTEGER (1..127)          DEFAULT 1,
    maximumNbOfDigits      [1] INTEGER (1..127),
    endOfReplyDigit        [2] OCTET STRING (SIZE (1..2))  OPTIONAL,
    cancelDigit             [3] OCTET STRING (SIZE (1..2))  OPTIONAL,
    startDigit              [4] OCTET STRING (SIZE (1..2))  OPTIONAL,
    firstDigitTimeout       [5] INTEGER (1..127)           OPTIONAL,
    interDigitTimeout       [6] INTEGER (1..127)           OPTIONAL,
    errorTreatment          [7] ErrorTreatment             DEFAULT reportErrorToScf,
    interruptableAnnInd     [8] BOOLEAN                   DEFAULT TRUE,

```

```

        voiceInformation      [9] BOOLEAN      DEFAULT FALSE,
        voiceBack            [10] BOOLEAN     DEFAULT FALSE,
        detectModem          [11] BOOLEAN     DEFAULT FALSE,
    ...
}
-- The use of voiceBack is network operator specific.
-- The endOfReplyDigit, cancelDigit, and startDigit parameters have been designated as OCTET STRING,
-- and are to be encoded as BCD, one digit per octet only, contained
-- in the four least significant bits of each OCTET. The usage is service dependent.

CollectedInfo ::= CHOICE {
    collectedDigits          [0] CollectedDigits,
    iA5Information          [1] BOOLEAN,
    detectModem              [2] BOOLEAN
}

ElementaryMessageID ::= Integer4

ErrorTreatment ::= ENUMERATED {
    reportErrorToScf(0),
    help(1),
    repeatPrompt(2)
}
-- reportErrorToScf means returning the "ImproperCallerResponse" error in the event of an error
-- condition during collection of user info.

GapOnResource ::= Code

InbandInfo { SCF-SSF-BOUNDS : bound2, SCF-SRF-BOUNDS : bound3 } ::= SEQUENCE {
    messageID                [0] MessageID {bound2, bound3},
    numberOfRepetitions       [1] INTEGER (1..127)          OPTIONAL,
    duration                  [2] INTEGER (0..32767)         OPTIONAL,
    interval                  [3] INTEGER (0.. 32767)        OPTIONAL,
    preferredLanguage         [4] Language                  OPTIONAL,
    ...
}
-- Interval is the time in seconds between each repeated announcement. Duration is the total
-- amount of time in seconds, including repetitions and intervals.
-- The end of announcement is either the end of duration or numberOfRepetitions, whatever comes first.
-- duration with value 0 indicates infinite duration

InformationToRecord { SCF-SRF-BOUNDS : bound3 } ::= SEQUENCE {
    messageID                [0] ElementaryMessageID        OPTIONAL,
    messageDeletionTimeOut    [1] INTEGER (1..3600)          OPTIONAL,
    --Time units = hours
    timeToRecord              [3] INTEGER (0..bound3.&maxRecordingTime)
    OPTIONAL,
    --Time units = seconds
    controlDigits             [4] SEQUENCE {
        endOfRecordingDigit    [0] OCTET STRING (SIZE(1..2))  OPTIONAL,
        cancelDigit            [1] OCTET STRING (SIZE(1..2))  OPTIONAL,
        replayDigit            [2] OCTET STRING (SIZE(1..2))  OPTIONAL,
        restartRecordingDigit   [3] OCTET STRING (SIZE(1..2))  OPTIONAL,
        restartAllowed          [4] BOOLEAN                   DEFAULT FALSE,
        replayAllowed           [5] BOOLEAN                   DEFAULT FALSE,
        ...
    },
    ...
}

InformationToSend { COMMON-BOUNDS : bound1, SCF-SSF-BOUNDS : bound2, SCF-SRF-BOUNDS :
bound3 } ::= CHOICE {

```

inbandInfo	[0] InbandInfo {bound1, bound3},
tone	[1] Tone,
displayInformation	[2] DisplayInformation {bound3},
sDSSinformation	[3] SDSSinformation {bound2}
}	

Language ::= PrintableString (SIZE (3)) -- ISO 639 codes only;

MailBoxID { SCF-SRF-BOUNDS : bound3} ::= OCTET STRING (SIZE(
bound3.&minMailBoxIDLength..bound3.&maxMailBoxIDLength))

Media ::= ENUMERATED {
voiceMail (0),
faxGroup3 (1),
faxGroup4 (2)
}

MessageID { SCF-SSF-BOUNDS : bound2, SCF-SRF-BOUNDS : bound3} ::= CHOICE {
elementaryMessageID [0] Integer4,
text [1] SEQUENCE {
messageContent [0] IA5String (SIZE (
bound3.&minMessageContentLength..
bound3.&maxMessageContentLength)),
attributes [1] OCTET STRING (SIZE (
bound3.&minAttributesLength..
bound3.&maxAttributesLength))
OPTIONAL,
...
},
elementaryMessageIDs [29] SEQUENCE SIZE (1.. bound3.&numOfMessageIDs) OF
Integer4,
variableMessage [30] SEQUENCE {
elementaryMessageID [0] Integer4,
variableParts [1] SEQUENCE SIZE
(1.. bound3.&maxVariableParts) OF VariablePart {bound2},
...
}
}

-- OPTIONAL denotes network operator specific use.

ReceivedStatus ::=ENUMERATED {
messageComplete (0),
messageInterrupted (1),
messageTimeOut (2)
}

RecordedMessageID ::= Integer4

SRFGapCriteria { SCF-SSF-BOUNDS: bound2} ::= CHOICE {
iPAddressValue [1] Digits {bound2},
gapOnResource [2] GapOnResource,
iPAddressAndresource [3] SEQUENCE {
iPAddressValue [1] Digits {bound2},
gapOnResource [2] GapOnResource,
...
}
}

Tone ::= SEQUENCE {
toneID [0] Integer4,
duration [1] Integer4 OPTIONAL,
...
}

-- The duration specifies the length of the tone in seconds, value 0 indicates infinite duration.

```
VariablePart { SCF-SSF-BOUNDS : bound2} ::= CHOICE {  
    integer          [0] Integer4,  
    number           [1] Digits { bound2},           -- Generic digits  
    time             [2] OCTET STRING (SIZE(2)),      -- HH:MM, BCD coded  
    date             [3] OCTET STRING (SIZE(3)),      -- YYMMDD, BCD coded  
    price            [4] OCTET STRING (SIZE(4))  
}
```

-- Indicates the variable part of the message.

-- BCD coded variable parts are encoded as described in the examples below.

-- For example, time = 12:15 would be encoded as:

```
--      Bits          HGFE          DCBA  
--      leading octet  2          1  
--      5      1
```

-- date = 1993 September 30th would be encoded as:

```
--      Bits          HGFE          DCBA  
--      leading octet  3          9  
--      9      0  
--      0      3
```

-- For a system operating when or after this Recommendation is released, the 2-digit value

-- representing a Year shall be interpreted as follows:

-- - If the two-digits value is 00 through 49 inclusive, it shall be interpreted as representing
-- year 2000 through 2049.

-- - If the two-digits value is 50 through 99 inclusive, it shall be interpreted as representing
-- year 1950 through 1999.

END

-- 12.1.2 **CLASSES**

IN-CS3-scf-srf-classes {ccitt recommendation q 1238 modules(1) in-cs3-scf-srf-classes (11) version1(0)}

DEFINITIONS ::=

BEGIN

IMPORTS

```
id-package-emptyConnection,  
id-rosObject-srf,  
id-rosObject-ssf,  
ros-InformationObjects,  
ros-UsefulDefinitions,  
scf-srf-Protocol,  
scf-srf-datatypes,  
ssf-scf-datatypes
```

**FROM IN-CS3-object-identifiers{ ccitt recommendation q 1238 modules(1) in-cs3-object-identifiers(0)
version1(0) }**

```
ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, Code, OPERATION,  
CONNECTION-PACKAGE
```

**FROM Remote-Operations-Information-Objects ros-InformationObjects
emptyBind, emptyUnbind**

FROM Remote-Operations-Useful-Definitions ros-UsefulDefinitions

;

UIScript ::= CLASS {

```
    &SpecificInfo  
    &Result
```

```
OPTIONAL,  
OPTIONAL,
```

```

        &id
    }

    WITH SYNTAX {
        [WITH-SPECIFICINFO      &SpecificInfo]
        [WITH-RESULT            &Result]
        IDENTIFIED BY           &id
    }
    -- SpecificInfo is used for ScriptRun and ScriptInformation and ScriptClose
    -- Result is used for ScriptEvent

    firstScript UISCRIPT ::=
    {
        IDENTIFIED BY local:1
    }
    -- firstScript is just an example.

    SupportedUIScripts UISCRIPT ::= {firstScript , ...
    --full set of User Interaction script
    }
    -- SupportedUIScripts is the full set of User Interaction scripts.

    SCF-SRF-BOUNDS ::= CLASS
    {
        &minAttributesLength      INTEGER                OPTIONAL,
        &maxAttributesLength      INTEGER                OPTIONAL,
        &minMailBoxIDLength       INTEGER                OPTIONAL,
        &maxMailBoxIDLength       INTEGER                OPTIONAL,
        &minMessageContentLength  INTEGER                OPTIONAL,
        &maxMessageContentLength  INTEGER                OPTIONAL,
        &minReceivedInformationLength  INTEGER            OPTIONAL,
        &maxReceivedInformationLength  INTEGER            OPTIONAL,
        &maxRecordingTime         INTEGER                OPTIONAL,
        &numOfMessageIDs          INTEGER                OPTIONAL,
        &maxRecordedMessageUnits  INTEGER                OPTIONAL,
        &maxVariableParts        INTEGER                OPTIONAL

    }

    WITH SYNTAX
    {
        MINIMUM-FOR-ATTRIBUTES      &minAttributesLength
        MAXIMUM-FOR-ATTRIBUTES      &maxAttributesLength
        MINIMUM-FOR-MAIL-BOX-ID     &minMailBoxIDLength
        MAXIMUM-FOR-MAIL-BOX-ID     &maxMailBoxIDLength
        MINIMUM-FOR-MESSAGE-CONTENT &minMessageContentLength
        MAXIMUM-FOR-MESSAGE-CONTENT &maxMessageContentLength
        MINIMUM-FOR-RECEIVED-INFORMATION &minReceivedInformationLength
        MAXIMUM-FOR-RECEIVED-INFORMATION &maxReceivedInformationLength
        MAXIMUM-FOR-RECORDING-TIME  &maxRecordingTime
        NUM-OF-MESSAGE-IDS          &numOfMessageIDs
        MAXIMUM-FOR-RECORDED-MESSAGE-UNITS &maxRecordedMessageUnits
        NUM-OF-VARIABLE-PARTS      &maxVariableParts
    }

    -- The following instance of the parameter bound is just an example

    networkSpecificSCFSRFBoundSet SCF-SRF-BOUNDS ::=
    {

        MINIMUM-FOR-ATTRIBUTES      1
        MAXIMUM-FOR-ATTRIBUTES      5
        MINIMUM-FOR-MAIL-BOX-ID     1
    }

```

```

    MAXIMUM-FOR-MAIL-BOX-ID 5
    MINIMUM-FOR-MESSAGE-CONTENT 1
    MAXIMUM-FOR-MESSAGE-CONTENT 5
    MINIMUM-FOR-RECEIVED-INFORMATION 1
    MAXIMUM-FOR-RECEIVED-INFORMATION 5
    MAXIMUM-FOR-RECORDING-TIME 5
    MAXIMUM-FOR-RECORDED-MESSAGE-UNITS 5
    NUM-OF-MESSAGE-IDS 2
    NUM-OF-VARIABLE-PARTS 5 -- must be 5 or
-- greater.
}
END

-- 12.2 OPERATIONS AND ARGUMENTS

IN-CS3-scf-srf-ops-args {ccitt recommendation q 1238 modules(1) in-cs3-scf-srf-ops-args (12) version1(0)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

IMPORTS

    ros-InformationObjects,
    operationcodes,
    common-datatypes,
    errortypes,
    common-classes,
    scf-srf-classes,
    ssf-scf-classes,
    ssf-scf-datatypes,
    scf-srf-datatypes

FROM IN-CS3-object-identifiers
{ccitt recommendation q 1238 modules(1) in-cs3-object-identifiers(0) version1(0)}

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

    opcode-playAnnouncement,
    opcode-promptAndCollectUserInformation,
    opcode-promptAndReceiveMessage,
    opcode-scriptClose,
    opcode-scriptEvent,
    opcode-scriptInformation,
    opcode-scriptRun,
    opcode-specializedResourceReport,
    opcode-activityTest,
    opcode-srfCallGap

FROM IN-CS3-scf-srf-operationcodes operationcodes

    SCF-SSF-BOUNDS

FROM IN-CS3-SSF-SCF-classes ssf-scf-classes

    CallSegmentID {},
    Digits {},
    GenericNumber {},
    LegID,
    ControlType,
    GapIndicators

```

FROM IN-CS3-SSF-SCF-datatypes ssf-scf-datatypes

**InformationToSend {},
CollectedInfo ,
MailBoxID {},
InformationToRecord {},
Media,
ReceivedStatus,
RecordedMessageID,
SRFGapCriteria {}**

FROM IN-CS3-scf-srf-datatypes scf-srf-datatypes

**canceled,
improperCallerResponse,
missingParameter,
parameterOutOfRange,
systemFailure,
taskRefused,
unavailableResource,
unexpectedComponentSequence,
unexpectedDataValue,
unexpectedParameter,
unknownLegID**

FROM IN-CS3-errortypes errortypes

**UISCRIPT,
SupportedUIScripts,
SCF-SRF-BOUNDS**

FROM IN-CS3-scf-srf-classes scf-srf-classes

Extensions

FROM IN-CS3-common-datatypes common-datatypes

COMMON-BOUNDS

FROM IN-CS3-common-classes common-classes

;

B1::=COMMON-BOUNDS

B2::=SCF-SSF-BOUNDS

B3::=SCF-SRF-BOUNDS

-- activityTest OPERATION

-- This operation is described in Recommendation Q.1238.2.

-- assistRequestInstruction OPERATION

-- This operation is described in Recommendation Q.1238.2.

--cancel OPERATION

-- This operation is described in Recommendation Q.1238.2.

--connectToResource OPERATION

-- This operation is described in Recommendation Q.1238.2.

--disconnectForwardConnection OPERATION

-- This operation is described in Recommendation Q.1238.2.

-- disconnectForwardConnectionWithArgument OPERATION

-- This operation is described in Recommendation Q.1238.2.

-- establishTemporaryConnection OPERATION

-- This operation is described in Recommendation Q.1238.2.

```
playAnnouncement { B1 : bound1, B2 : bound2, B3 : bound3 } OPERATION ::= {
    ARGUMENT          PlayAnnouncementArg { bound1,bound2,bound3}
    RETURN RESULT      FALSE
    ERRORS             {canceled |
                        missingParameter |
                        parameterOutOfRange |
                        systemFailure |
                        taskRefused |
                        unexpectedComponentSequence |
                        unexpectedDataValue |
                        unexpectedParameter |
                        unavailableResource |
                        unknownLegID
                       }
    LINKED             {specializedResourceReport}
    ALWAYS RESPONDS    FALSE
    CODE               opcode-playAnnouncement
}
```

-- Direction: SCF -> SRF, Timer: T_{pa}

-- This operation is to be used after Establish Temporary Connection (assist procedure with a second SSP)
-- or a Connect to Resource (no assist) operation. It may be used for inband interaction with an analogue user,
-- or for interaction with an ISDN user. In the former case, the SRF is usually collocated with the SSF for
-- standard tones (congestion tone ...) or standard announcements. In the latter case, the SRF is always
-- collocated with the SSF in the switch. Any error is returned to the SCF. The timer associated with this
-- operation must be of a sufficient duration to allow its linked operation to be correctly correlated.

```
PlayAnnouncementArg {B1 : bound1, B2 : bound2, B3 : bound3} ::= SEQUENCE {
    informationToSend      [0] InformationToSend { bound1, bound2, bound3},
    disconnectFromIPForbidden [1] BOOLEAN                      DEFAULT TRUE,
    requestAnnouncementComplete [2] BOOLEAN                    DEFAULT TRUE,
    extensions             [3] Extensions {bound1}             OPTIONAL,
    connectedParty         CHOICE {
        legID [4] LegID,
        callSegmentID [5] CallSegmentID {bound2}
    }
    ...
}
```

```
promptAndCollectUserInfo { B1 : bound1, B2 : bound2, B3 : bound3 } OPERATION ::= {
    ARGUMENT          PromptAndCollectUserInfoArg {bound1,bound2,bound3}
    RESULT            ReceivedInformationArg {bound1,bound2,bound3}
    ERRORS            {canceled |
                    improperCallerResponse |
                    missingParameter |
                    parameterOutOfRange |
                    systemFailure |
                    taskRefused |
                    unexpectedComponentSequence |
                    unavailableResource |
                    unexpectedDataValue |
                    unexpectedParameter
                   }
    CODE              opcode-promptAndCollectUserInfo
}
```



```

    }
-- Direction: SCF -> SRF, Timer: Tpc
-- This operation is used to interact with a user to collect information.

PromptAndCollectUserInformationArg { B1 : bound1, B2 : bound2, B3 : bound3 } ::= SEQUENCE {
    collectedInfo                [0] CollectedInfo,
    disconnectFromIPForbidden    [1] BOOLEAN                                DEFAULT TRUE,
    informationToSend            [2] InformationToSend { bound1, bound2, bound3}
    OPTIONAL,
    extensions                   [3] Extensions {bound1}                    OPTIONAL,
    callSegmentID               [4] CallSegmentID {bound2}                  OPTIONAL,
    ...
}

ReceivedInformationArg { B1 : bound1, B2 : bound2, B3 : bound3 } ::= CHOICE {
    digitsResponse               [0] Digits {bound2},
    iA5Response                  [1] IA5String,
    modemdetected                [2] BOOLEAN
}

promptAndReceiveMessage { B1 : bound1, B2 : bound2, B3 : bound3 } OPERATION ::= {
    ARGUMENT                     PromptAndReceiveMessageArg{bound1,bound2,bound3}
    RESULT                       MessageReceivedArg { bound1,bound2,bound3}
    ERRORS                       {canceled |
                                improperCallerResponse |
                                missingParameter |
                                parameterOutOfRange |
                                taskRefused |
                                systemFailure |
                                unavailableResource |
                                unexpectedComponentSequence |
                                unexpectedDataValue |
                                unexpectedParameter
                                }
    CODE                         opcode-promptAndReceiveMessage
}
-- Direction: SCF -> SRF, Timer: Tprm
-- Used to prompt a user to store a message

PromptAndReceiveMessageArg { B1 : bound1, B2 : bound2, B3 : bound3 } ::= SEQUENCE {
    disconnectFromIPForbidden    [0] BOOLEAN                                DEFAULT TRUE,
    informationToSend            [1] InformationToSend { bound1, bound2, bound3}
    OPTIONAL,
    extensions                   [3] Extensions {bound1}                    OPTIONAL,
    subscriberID                 [4] GenericNumber {bound2}                OPTIONAL,
    mailBoxID                    [5] MailBoxID {bound3}                    OPTIONAL,
    informationToRecord           [6] InformationToRecord {bound3},
    media                        [7] Media                                DEFAULT voiceMail,
    callSegmentID               [8] CallSegmentID {bound2}                  OPTIONAL,
    ...
}

MessageReceivedArg { B1 : bound1, B2 : bound2, B3 : bound3 } ::= SEQUENCE {
    receivedStatus               [0] ReceivedStatus,
    recordedMessageID            [1] RecordedMessageID                    OPTIONAL,
    recordedMessageUnits         [2] INTEGER(1..bound3.&maxRecordedMessageUnits)
    OPTIONAL,
    extensions                   [3] Extensions {bound1}                    OPTIONAL,
    ...
}

-- reportUTSI OPERATION

```

-- This operation is described in Recommendation Q.1238.2.

-- requestReportUTSI OPERATION

-- This operation is described in Recommendation Q.1238.2.

-- sendSTUI OPERATION

-- This operation is described in Recommendation Q.1238.2.

```
scriptClose { B1 : bound1, B2 : bound2 } OPERATION ::= {
  ARGUMENT          ScriptCloseArg { bound1,bound2 }
  RETURN RESULT     FALSE
  ERRORS            {
                    systemFailure |
                    missingParameter |
                    taskRefused |
                    unavailableResource |
                    unexpectedComponentSequence |
                    unexpectedDataValue |
                    unexpectedParameter
                    }
  ALWAYS RESPONDS   FALSE
  CODE              opcode-scriptClose
}
```

-- Direction:SCF-> SRF. Timer : T_{CI}

-- This operation is issued by the SCF to deallocate the resources used to perform the
-- instance of the "User Interaction" script: the context is released.

```
ScriptCloseArg { B1 : bound1, B2 : bound2 } ::= SEQUENCE {
  uIScriptId          UISCRIPT.&id({SupportedUIScripts }),
  uIScriptSpecificInfo [0] UISCRIPT.&SpecificInfo({SupportedUIScripts }{@uIScriptId})
                                OPTIONAL,
  extensions          [1] Extensions {bound1}                                OPTIONAL,
  callSegmentID       [2] CallSegmentID {bound2}                            OPTIONAL,
  ...
}
```

```
scriptEvent { B1 : bound1, B2 : bound2 } OPERATION ::= {
  ARGUMENT          ScriptEventArg { bound1, bound2 }
  RETURN RESULT     FALSE
  ALWAYS RESPONDS   FALSE
  CODE              opcode-scriptEvent
}
```

-- Direction:SRF-> SCF. Timer : T_{Re}

-- This operation is issued by the SRF to return information to the SCF on the results of the
-- execution of the instance of User Interaction script.

```
ScriptEventArg { B1 : bound1, B2 : bound2 } ::= SEQUENCE {
  uIScriptId          UISCRIPT.&id({SupportedUIScripts }),
  uIScriptResult       [0] UISCRIPT.&Result({SupportedUIScripts }{@uIScriptId})
                                OPTIONAL,
  extensions          [1] Extensions {bound1}                                OPTIONAL,
  callSegmentID       [2] CallSegmentID {bound2}                            OPTIONAL,
  lastEventIndicator  [3] BOOLEAN                                           DEFAULT FALSE,
  ...
}
```

```
scriptInformation { B1 : bound1, B2 : bound2 } OPERATION ::= {
  ARGUMENT          ScriptInformationArg { bound1, bound2 }
  RETURN RESULT     FALSE
  ERRORS            {

```

```

        systemFailure |
        missingParameter |
        taskRefused |
        unavailableResource |
        unexpectedComponentSequence |
        unexpectedDataValue |
        unexpectedParameter
    }
    ALWAYS RESPONDS
    CODE
}
-- Direction:SCF-> SRF. Timer :Tinf

ScriptInformationArg { B1 : bound1, B2 : bound2 } ::= SEQUENCE {
    uIScriptId          UISCRIPT.&id({SupportedUIScripts }),
    uIScriptSpecificInfo [0] UISCRIPT.&SpecificInfo({SupportedUIScripts }{@uIScriptId})
    OPTIONAL,
    extensions          [1] Extensions {bound1}          OPTIONAL,
    callSegmentID       [2] CallSegmentID {bound2}        OPTIONAL,
    ...
}

scriptRun { B1 : bound1, B2 : bound2 } OPERATION ::= {
    ARGUMENT          ScriptRunArg { bound1, bound2 }
    RETURN RESULT     FALSE
    ERRORS            {
        systemFailure |
        missingParameter |
        taskRefused |
        unavailableResource |
        unexpectedComponentSequence |
        unexpectedDataValue |
        unexpectedParameter
    }
    ALWAYS RESPONDS
    CODE              FALSE
    opcode-scriptRun
}
-- Direction:SCF-> SRF. Timer :Tru
-- This operation is issued by the SCF to allocate the necessary resources to perform the
-- instance of the "User Interaction" script and then to activate this "User Interaction" script
-- instance. A context is partially defined for it if necessary.

ScriptRunArg { B1 : bound1, B2 : bound2 } ::= SEQUENCE {
    uIScriptId          UISCRIPT.&id({SupportedUIScripts }),
    uIScriptSpecificInfo [0] UISCRIPT.&SpecificInfo({SupportedUIScripts }{@uIScriptId})
    OPTIONAL,
    extensions          [1] Extensions {bound1}          OPTIONAL,
    disconnectFromIPForbidden [2] BOOLEAN                DEFAULT TRUE,
    callSegmentID       [3] CallSegmentID {bound2}        OPTIONAL,
    ...
}

specializedResourceReport OPERATION ::= {
    ARGUMENT          SpecializedResourceReportArg
    RETURN RESULT     FALSE
    ALWAYS RESPONDS
    CODE              FALSE
    opcode-specializedResourceReport
}
-- Direction: SRF -> SCF, Timer: TSrr
-- This operation is used as the response to a PlayAnnouncement operation when the announcement completed
-- report indication is set.

SpecializedResourceReportArg ::= NULL

```

```

SRFCallGap { B1 : bound1, B2 : bound2, B3 : bound3} OPERATION ::= {
    ARGUMENT                SRFCallGapArg {bound1, bound2,bound3}
    RETURN RESULT          FALSE
    ALWAYS RESPONDS        FALSE
    CODE                    opcode-srfCallGap
}
-- Direction: SRF -> SCF, Timer:  $T_{cg}$ 
-- This operation is used to request the SCF to reduce the rate at which specific service requests are sent to
-- the SRF.

```

```

SRFCallGapArg { B1 : bound1, B2 : bound2, B3 : bound3} ::= SEQUENCE {
    sRfGapCriteria          [0] SRFGapCriteria {bound1},
    gapIndicators          [1] GapIndicators,
    controlType            [2] ControlType                OPTIONAL,
    ...,
    ...,
extensions                [30] Extensions {bound1}        OPTIONAL
}

```

END

-- The following value ranges do apply for operation specific timers in INAP:

```

-- short:      1-10 seconds
-- medium:     1-60 seconds
-- long:       1 second - 30 minutes

```

-- The table below lists all operation timers and the value range for each timer.
-- The definitive value for each operation timer may be network specific and has
-- to be defined by the network operator.

--
-- **TABLE 6-1/Q.1238.3 – OPERATION TIMERS AND THEIR VALUE RANGE**

Operation Name	Timer	value range
ScriptClose	T_{cl}	Short
ScriptInformation	T_{inf}	Short
PlayAnnouncement	T_{pa}	Long
PromptAndCollectUserInformation	T_{pc}	Long
PromptAndReceiveMessage	T_{prm}	Long
ScriptEvent	T_{re}	Short
ScriptRun	T_{ru}	Long
SpecializedResourceReport	T_{srr}	Short
SRFActivitytest	T_{at}	Short
SRFCallGap	T_{cg}	Short

-- **12.3 PACKAGE, CONTRACTS AND APPLICATION CONTEXTS**

-- **12.3.1 ASN.1 MODULES**

IN-CS3-SCF-SRF-pkgs-contracts-acs {ccitt recommendation q 1238 modules(1) in-cs3-scf-srf-pkgs-contracts-acs(13) version1(0)}

DEFINITIONS ::=

BEGIN

-- This module describes the operation-packages, contracts and application-contexts used
-- over the SCF-SRF interface.

IMPORTS

id-package-specializedResourceControl,
 id-ac-srf-scfAC,
 id-contract-srf-scf,
 id-package-srf-scfCancel,
 id-package-scriptControl,
 id-package-messageControl,
 id-package-srfManagement,
 id-package-activityTest,
 id-as-basic-srf-scf,
 common-classes, ros-InformationObjects, tc-Messages, tc-NotationExtensions,
 scf-srf-Operations, ssf-scf-Operations, ssf-scf-classes, scf-srf-classes, ssf-scf-Protocol

FROM IN-CS3-object-identifiers {ccitt recommendation q 1238 modules(1) in-cs3-object-identifiers (0)
 version1(0)}

SCF-SSF-BOUNDS

FROM IN-CS3-SSF-SCF-classes ssf-scf-classes

COMMON-BOUNDS

FROM IN-CS3-common-classes common-classes

SCF-SRF-BOUNDS ,
 networkSpecificSCFSRFBoundSet

FROM IN-CS3-scf-srf-classes scf-srf-classes

ROS-OBJECT-CLASS, CONTRACT, OPERATION-PACKAGE, OPERATION

FROM Remote-Operations-Information-Objects ros-InformationObjects

TCMessage {}
 FROM TCAPMessages tc-Messages

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

playAnnouncement {},
 promptAndReceiveMessage {},
 promptAndCollectUserInformation {},
 scriptClose {},
 scriptEvent {},
 scriptInformation {},
 scriptRun {},
 specializedResourceReport,
 sRFCallGap {}

FROM IN-CS3-SCF-SRF-ops-args scf-srf-Operations

cancel {},
 assistRequestInstructions {},
 activityTest

FROM IN-CS3-SSF-SCF-ops-args ssf-scf-Operations

srf-scfActivationOfAssistPackage {},
 activityTestPackage

FROM IN-CS3-SSF-SCF-pkgs-contracts-acs ssf-scf-Protocol
 ;

B1::=COMMON-BOUNDS
B2::=SCF-SSF-BOUNDS
B3::=SCF-SRF-BOUNDS

```

networkSpecificBoundSet COMMON-BOUNDS ::=
{
    NUM-OF-EXTENSIONS          1
}

networkSpecificSSFSCFBoundSet SCF-SSF-BOUNDS ::=
{
    MINIMUM-FOR-DIGITS          1
    MAXIMUM-FOR-DIGITS          5
    MINIMUM-FOR-DISPLAY         1
    MAXIMUM-FOR-DISPLAY         5
    MINIMUM-FOR-GENERIC-NUMBER  1
    MAXIMUM-FOR-GENERIC-NUMBER  5
    MINIMUM-FOR-SCF-ID          1
    MAXIMUM-FOR-SCF-ID          5
    MINIMUM-FOR-SDSS-INFORMATION 1
    MAXIMUM-FOR-SDSS-INFORMATION 5
    NUM-OF-GENERIC-NUMBERS      2
}

-- Application Contexts --

srf-scf-ac APPLICATION-CONTEXT ::= {
    CONTRACT          srf-scf-contract
    DIALOGUE MODE      structured
    TERMINATION        basic
    ABSTRACT SYNTAXES  {dialogue-abstract-syntax |
                        srf-scf-abstract-syntax }
    APPLICATION CONTEXT NAME id-ac-srf-scfAC}

-- Contracts --

srf-scf-contract CONTRACT ::= {
    INITIATOR CONSUMER OF {srf-scfActivationOfAssistPackage {networkSpecificBoundSet,
networkSpecificSSFSCFBoundSet }}
                        srfManagementPackage {networkSpecificBoundSet,
networkSpecificSSFSCFBoundSet, networkSpecificSCFSRFBoundSet }
    }
    RESPONDER CONSUMER OF {specializedResourceControlPackage {networkSpecificBoundSet,
networkSpecificSSFSCFBoundSet, networkSpecificSCFSRFBoundSet }}
                        srf-scfCancelPackage {networkSpecificBoundSet,
networkSpecificSSFSCFBoundSet, networkSpecificSCFSRFBoundSet }}
                        scriptControlPackage {networkSpecificBoundSet,
networkSpecificSSFSCFBoundSet }}
                        activityTestPackage |
                        messageControlPackage {networkSpecificBoundSet,
networkSpecificSSFSCFBoundSet, networkSpecificSCFSRFBoundSet }}
    ID                  id-contract-srf-scf }

-- specializedResourceControl package --

specializedResourceControlPackage { B1 : bound1, B2 : bound2, B3 : bound3 } OPERATION-PACKAGE ::= {
    CONSUMER INVOKES {playAnnouncement { bound1,bound2,bound3} |
promptAndCollectUserInformation { bound1,bound2,bound3}
                    }
    SUPPLIER INVOKES {specializedResourceReport}
    ID               id-package-specializedResourceControl}

-- srf-scfCancel package --

srf-scfCancelPackage { B1 : bound1, B2 : bound2, B3 : bound3 } OPERATION-PACKAGE ::= {
    CONSUMER INVOKES {cancel {bound2 }}
    ID               id-package-srf-scfCancel}

-- scriptControl package --

```

```

scriptControlPackage { B1 : bound1, B2 : bound2 } OPERATION-PACKAGE ::= {
    CONSUMER INVOKES      { scriptClose {bound1, bound2}| scriptRun {bound1, bound2} |
                           scriptInformation { bound1,bound2 }}
    SUPPLIER INVOKES      { scriptEvent { bound1,bound2 }}
    ID                    id-package-scriptControl}

-- messageControl package

messageControlPackage { B1 : bound1, B2 : bound2, B3 : bound3 } OPERATION-PACKAGE ::= {
    CONSUMER INVOKES      {promptAndReceiveMessage { bound1,bound2,bound3}}
    ID                    id-package-messageControl
}

-- SRF management package

srfManagementPackage { B1 : bound1, B2 : bound2, B3 : bound3 } OPERATION-PACKAGE ::= {
    CONSUMER INVOKES      {sRFCallGap { bound1,bound2,bound3}}
    ID                    id-package-srfManagement
}

-- Abstract Syntaxes --

srf-scf-abstract-syntax ABSTRACT-SYNTAX ::= {
    BASIC-SRF-SCF-PDUs
    IDENTIFIED BY          id-as-basic-srf-scf}

BASIC-SRF-SCF-PDUs ::= TCMMessage {{SRF-SCF-Invokable},{SRF-SCF-Returnable} }

SRF-SCF-Invokable OPERATION ::= {
    assistRequestInstructions {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    cancel {networkSpecificSSFSCFBoundSet}|
    playAnnouncement {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
                      networkSpecificSCFSRFBoundSet }|
    promptAndCollectUserInfo {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
                              networkSpecificSCFSRFBoundSet }|
    scriptClose {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptEvent {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptInformation {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptRun {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    specializedResourceReport |
    promptAndReceiveMessage {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
                              networkSpecificSCFSRFBoundSet }|
    sRFCallGap {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
                networkSpecificSCFSRFBoundSet }|
    activityTest
}

SRF-SCF-Returnable OPERATION ::= {
    assistRequestInstructions {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    cancel {networkSpecificSSFSCFBoundSet}|
    playAnnouncement {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
                      networkSpecificSCFSRFBoundSet }|
    promptAndCollectUserInfo {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
                              networkSpecificSCFSRFBoundSet }|
    scriptClose {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptInformation {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    scriptRun {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet }|
    promptAndReceiveMessage {networkSpecificBoundSet, networkSpecificSSFSCFBoundSet,
                              networkSpecificSCFSRFBoundSet }|
    activityTest
}

END

```