

10 *CSTA DATA TYPES* §

This section describes the data types used by the functions and messages defined for the TSAPI. The data type are divided into two categories: CSTA Data Type which are associated with telephony functions and Interface Data Types which are associated with the API itself and the NetWare Telephony Services client/server interface.

Device Identifiers §

```
typedef char          Nulltype;typedef char          DeviceID_t[64];

typedef enum ConnectionID_Device_t {
    STATIC_ID = 0,
    DYNAMIC_ID = 1
} ConnectionID_Device_t;

typedef struct ConnectionID_t {
    long          callID;
    DeviceID_t    deviceID;
    ConnectionID_Device_t devIDType;
} ConnectionID_t;
```

Basic Call Control Confirmation Events XE "Basic Call Control Confirmation Events"§

CSTAAAlternateCallConfEvent structures

```
XE "CSTAAAlternateCallConfEvent structures"§
typedef struct CSTAAAlternateCallConfEvent_t {           Nulltype           null;}
CSTAAAlternateCallConfEvent_t;
```

CSTAAAnswerCallConfEvent structures

```
typedef struct CSTAAAnswerCallConfEvent_t {           Nulltype           null;}
CSTAAAnswerCallConfEvent_t;typedef enum Feature_t { FT_CAMPON = 0, FT_CALLBACK =
1, FT_INTRUDE = 2} Feature_t;
```

CSTACallCompletionConfEvent structures XE "CSTACallCompletionConfEvent structures"§

```
typedef struct CSTACallCompletionConfEvent_t {           Nulltype           null;}
CSTACallCompletionConfEvent_t;
```

```
CSTAClearCallConfEvent structuresXE
"CSTAClearCallConfEvent structures"§
```

```
typedef struct CSTAClearCallConfEvent_t { Nulltype
null;}
CSTAClearCallConfEvent_t;CSTAClearConnectionConfE
vent structuresXE "CSTAClearConnectionConfEvent
structures"§
```

```
typedef struct CSTAClearConnectionConfEvent_t
{ Nulltype null;}
CSTAClearConnectionConfEvent_t;CSTAConferenceCon
fEvent structuresXE "CSTAConferenceConfEvent
structures"§
```

```
typedef struct Connection_t { ConnectionID_t party;
DeviceID_t staticDevice; /* NULL for not present */}
Connection_t;typedef struct ConnectionList { int
count; Connection_t *connection;}
ConnectionList;typedef struct
CSTAConferenceCallConfEvent_t { ConnectionID_t
activeCall; ConnectionList connList;}
CSTAConferenceCallConfEvent_t;CSTAConsultationCall
ConfEvent structuresXE
"CSTAConsultationCallConfEvent structures"§
```

```
typedef struct CSTAConsultationCallConfEvent_t { ConnectionID_t newCall;}
CSTAConsultationCallConfEvent_t;
```

```
CSTADeflectCallConfEvent structuresXE
"CSTADeflectCallConfEvent structures"§
```

```
typedef struct CSTADeflectCallConfEvent_t { Nulltype null;} CSTADeflectCallConfEvent_t;
```

**CSTAGroupPickupCallConfEvent structuresXE
"CSTAGroupPickupCallConfEvent structures"§**

**typedef struct CSTAGroupPickupCallConfEvent_t
{ Nulltype null;}
CSTAGroupPickupCallConfEvent_t;CSTAHoldCallConfE
vent structuresXE "CSTAHoldCallConfEvent structures"§**

**typedef struct CSTAHoldCallConfEvent_t { Nulltype
null;}
CSTAHoldCallConfEvent_t;CSTAMakeCallConfEvent
structuresXE "CSTAMakeCallConfEvent structures"§**

**typedef struct CSTAMakeCallConfEvent_t
{ ConnectionID_t newCall;}
CSTAMakeCallConfEvent_t;CSTAMakePredictiveCallCo
nfEvent structuresXE
"CSTAMakePredictiveCallConfEvent structures"§**

typedef enum AllocationState_t { AS_CALL_DELIVERED = 0, AS_CALL_ESTABLISHED =
1} AllocationState_t;typedef struct CSTAMakePredictiveCallConfEvent_t { ConnectionID_t
newCall;} CSTAMakePredictiveCallConfEvent_t;

CSTAPickupCallConfEvent structuresXE
"CSTAPickupCallConfEvent structures"§

```
typedef struct CSTAPickupCallConfEvent_t { Nulltype  
null;}  
CSTAPickupCallConfEvent_t;CSTARReconnectCallConfEv  
ent structuresXE "CSTARReconnectCallConfEvent  
structures"§
```

```
typedef struct CSTARReconnectCallConfEvent_t {}  
CSTARReconnectCallConfEvent_t;CSTARRetrieveCallConf  
Event structuresXE "CSTARRetrieveCallConfEvent  
structures"§
```

```
typedef struct CSTARRetrieveCallConfEvent_t { Nulltype  
null;}  
CSTARRetrieveCallConfEvent_t;CSTATransferCallConfEv  
ent structures
```

```
XE "CSTATransferCallConfEvent structures"§  
typedef struct CSTATransferCallConfEvent_t { ConnectionID_t resultingCall; ConnectionList  
connList;} CSTATransferCallConfEvent_t;
```

CSTAUniversalFailureEvent structuresXE
"CSTAUniversalFailureEvent structures"§

```

typedef enum CSTAUniversalFailure_t {
    GENERIC_UNSPECIFIED = 0,
    GENERIC_OPERATION = 1,
    REQUEST_INCOMPATIBLE_WITH_OBJECT = 2,
    VALUE_OUT_OF_RANGE = 3,
    OBJECT_NOT_KNOWN = 4,
    INVALID_CALLING_DEVICE = 5,
    INVALID_CALLED_DEVICE = 6,
    INVALID_FORWARDING_DESTINATION = 7,
    PRIVILEGE_VIOLATION_ON_SPECIFIED_DEVICE = 8,
    PRIVILEGE_VIOLATION_ON_CALLED_DEVICE = 9,
    PRIVILEGE_VIOLATION_ON_CALLING_DEVICE = 10,
    INVALID_CSTA_CALL_IDENTIFIER = 11,
    INVALID_CSTA_DEVICE_IDENTIFIER = 12,
    INVALID_CSTA_CONNECTION_IDENTIFIER = 13,
    INVALID_DESTINATION = 14,
    INVALID_FEATURE = 15,
    INVALID_ALLOCATION_STATE = 16,
    INVALID_CROSS_REF_ID = 17,
    INVALID_OBJECT_TYPE = 18,
    SECURITY_VIOLATION = 19,
    GENERIC_STATE_INCOMPATIBILITY = 21,
    INVALID_OBJECT_STATE = 22,
    INVALID_CONNECTION_ID = 23,
    NO_ACTIVE_CALL = 24,
    NO_HELD_CALL = 25,
    NO_CALL_TO_CLEAR = 26,
    NO_CONNECTION_TO_CLEAR = 27,
    NO_CALL_TO_ANSWER = 28,
    NO_CALL_TO_COMPLETE = 29,
    GENERIC_SYSTEM_RESOURCE_AVAILABILITY = 31,
    SERVICE_BUSY = 32,
    RESOURCE_BUSY = 33,
    RESOURCE_OUT_OF_SERVICE = 34,
    NETWORK_BUSY = 35,
    NETWORK_OUT_OF_SERVICE = 36,
    OVERALL_MONITOR_LIMIT_EXCEEDED = 37,
    CONFERENCE_MEMBER_LIMIT_EXCEEDED = 38,
    GENERIC_SUBSCRIBED_RESOURCE_AVAILABILITY = 41,
    OBJECT_MONITOR_LIMIT_EXCEEDED = 42,
    EXTERNAL_TRUNK_LIMIT_EXCEEDED = 43,
    OUTSTANDING_REQUEST_LIMIT_EXCEEDED = 44,
    GENERIC_PERFORMANCE_MANAGEMENT = 51,
    PERFORMANCE_LIMIT_EXCEEDED = 52,
    SEQUENCE_NUMBER_VIOLATED = 61,
    TIME_STAMP_VIOLATED = 62,
    PAC_VIOLATED = 63,
    SEAL_VIOLATED = 64}
CSTAUniversalFailure_t;typedef struct CSTAUniversalFailureConfEvent_t {
    CSTAUniversalFailure_t
    error;}
CSTAUniversalFailureConfEvent_t;Telephony Supplementary Confirmation
EventsXE "Telephony Supplementary Confirmation Events"§

```

```
CSTASetMsgeWaitingConfEvent          structuresXE
"CSTASetMsgeWaitingConfEvent structures"§
```

```
typedef struct CSTASetMwiConfEvent_t {      Nulltype
null;}
CSTASetMwiConfEvent_t;CSTASetDoNotDisturbConfEvent
structuresXE "CSTASetDoNotDisturbConfEvent
structures"§
```

```
typedef struct CSTASetDndConfEvent_t {      Nulltype
null;}
CSTASetDndConfEvent_t;CSTASetForwardingConfEvent
structuresXE "CSTASetForwardingConfEvent
structures"§
```

```
typedef enum ForwardingType_t {   FWD_IMMEDIATE =
0,      FWD_BUSY = 1,      FWD_NO_ANS = 2,
FWD_BUSY_INT = 3,      FWD_BUSY_EXT = 4,
FWD_NO_ANS_INT = 5,      FWD_NO_ANS_EXT = 6}
ForwardingType_t;typedef struct ForwardingInfo_t
{      ForwardingType_t forwardingType;      Boolean
forwardingOn; DeviceID_t forwardDN; /* NULL for
not present */} ForwardingInfo_t;typedef struct
CSTASetFwdConfEvent_t {      Nulltype      null;}
CSTASetFwdConfEvent_t;CSTASetAgentStateConfEvent
structuresXE "STASetAgentStateConfEvent structures"§
```

```
typedef enum AgentMode_t {
AM_LOG_IN = 0,
AM_LOG_OUT = 1,
AM_NOT_READY = 2,
AM_READY = 3,
AM_WORK_NOT_READY = 4,
AM_WORK_READY = 5
} AgentMode_t;
```

10-8 CSTA Data Types


```

typedef char    AgentID_t[32];

typedef DeviceID_t  AgentGroup_t;

typedef char    AgentPassword_t[32];

typedef struct CSTASetAgentStateConfEvent_t {
    Nulltype    null;
} CSTASetAgentStateConfEvent_t;

```

CSTAQueryMsgWaitingIndConfEvent structures XE "CSTAQueryMsgWaitingIndConfEvent structures"§

```

typedef struct CSTAQueryMwiConfEvent_t { Boolean
messages;}
CSTAQueryMwiConfEvent_t;CSTAQueryDoNotDisturbCo
nfEvent structures

```

```

XE "CSTAQueryDoNotDisturbConfEvent structures"§typedef
struct CSTAQueryDndConfEvent_t { Boolean
doNotDisturb;}
CSTAQueryDndConfEvent_t;CSTAQueryForwardingConf
Event structuresXE "CSTAQueryForwardingConfEvent
structures"§

```

```

typedef enum ForwardingType_t {
    FWD_IMMEDIATE = 0,
    FWD_BUSY = 1,
    FWD_NO_ANS = 2,
    FWD_BUSY_INT = 3,
    FWD_BUSY_EXT = 4,
    FWD_NO_ANS_INT = 5,
    FWD_NO_ANS_EXT = 6
} ForwardingType_t;

typedef struct ForwardingInfo_t {
    ForwardingType_t forwardingType;
    Boolean forwardingOn;
    DeviceID_t forwardDN;
} ForwardingInfo_t;

typedef struct ListForwardParameters_t {
    short count;
    ForwardingInfo_t param[7];
}

```

```

} ListForwardParameters_t;

typedef struct CSTAQueryFwdConfEvent_t {
    ListForwardParameters_t forward;
} CSTAQueryFwdConfEvent_t;

```

CSTAQueryAgentStateConfEvent structure
"CSTAQueryAgentStateConfEvent structure"

```

typedef enum AgentState_t {
    AG_NOT_READY = 0,
    AG_NULL = 1,
    AG_READY = 2,
    AG_WORK_NOT_READY = 3,
    AG_WORK_READY = 4}
AgentState_t; typedef
CSTAQueryAgentStateConfEvent_t {
    AgentState_t
agentState;}
CSTAQueryAgentStateConfEvent_t; CSTAQueryLastNum
berConfEvent
"CSTAQueryLastNumberConfEvent structures"

```

```

typedef struct CSTAQueryLastNumberConfEvent_t
{
    DeviceID_t
lastNumber;}
CSTAQueryLastNumberConfEvent_t; CSTAQueryDeviceI
nfoConfEvent structures

```

```

XE "CSTAQueryDeviceInfoConfEvent structures"
typedef enum DeviceType_t
{
    DT_STATION = 0,
    DT_LINE = 1,
    DT_BUTTON = 2,
    DT_ACD = 3,
    DT_TRUNK = 4,
    DT_OPERATOR = 5,
    DT_STATION_GROUP = 16,
    DT_LINE_GROUP = 17,
    DT_BUTTON_GROUP = 18,
    DT_ACD_GROUP = 19,
    DT_TRUNK_GROUP = 20,
    DT_OPERATOR_GROUP = 21,
    DT_OTHER = 255}
DeviceType_t; typedef unsigned char
DeviceClass_t; #define
DC_VOICE 0x80 #define
DC_DATA 0x40 #define
DC_IMAGE 0x20 #define
DC_OTHER 0x10 typedef struct CSTAQueryDeviceInfoConfEvent_t
{
    DeviceID_t device;
    DeviceType_t deviceType;
    DeviceClass_t deviceClass;}
CSTAQueryDeviceInfoConfEvent_t;

```

Status Reporting Confirmation EventsXE "Status Reporting Confirmation Events"§

cstaMonitorDevice structures

```
typedef long CSTAMonitorCrossRefID_t;typedef CSTAObject_t
CSTAMonitorObject_t;typedef unsigned short CSTACallFilter_t;
CF_CALL_CLEARED 0x8000 CF_CONFERENCED 0x4000
CF_CONNECTION_CLEARED 0x2000 CF_DELIVERED 0x1000
CF_DIVERTED 0x0800 CF_ESTABLISHED 0x0400 CF_FAILED 0x0200
CF_HELD 0x0100 CF_NETWORK_REACHED 0x0080
CF_ORIGINATED 0x0040 CF_QUEUED 0x0020 CF_RETRIEVED
0x0010 CF_SERVICE_INITIATED 0x0008 CF_TRANSFERRED
0x0004typedef unsigned char CSTAFeatureFilter_t; FF_CALL_INFORMATION 0x80
FF_DO_NOT_DISTURB 0x40 FF_FORWARDING 0x20
FF_MESSAGE_WAITING 0x10typedef unsigned char CSTAAgentFilter_t;
AF_LOGGED_ON 0x80 AF_LOGGED_OFF 0x40 AF_NOT_READY
0x20 AF_READY 0x10 AF_WORK_NOT_READY 0x08
AF_WORK_READY 0x04typedef unsigned char CSTAMaintenanceFilter_t;
MF_BACK_IN_SERVICE 0x80 MF_OUT_OF_SERVICE 0x40typedef struct
CSTAMonitorFilter_t { CSTACallFilter_t call; CSTAFeatureFilter_t feature;
CSTAAgentFilter_t agent; CSTAMaintenanceFilter_t maintenance; Boolean private;}
CSTAMonitorFilter_t;
```

```
typedef enum CSTAMonitorType_t { MT_CALL = 0,
MT_DEVICE = 1} CSTAMonitorType_t;cstaMonitorCall
structuresXE "cstaMonitorCall structures"§
```

See **cstaMonitorDevice**
structures.cstaMonitorCallsViaDevice structureXE
"staMonitorCallsViaDevice structure"§s

See **cstaMonitorDevice**
structures.CSTAMonitorConfEvent structuresXE
"CSTAMonitorConfEvent structures"§

```

typedef struct CSTAMonitorStartConfEvent_t
{
    CSTAMonitorCrossRefID_t monitorCrossRefID;
    CSTAMonitorFilter_t monitorFilter;}
CSTAMonitorStartConfEvent_t;CSTACHangeMonitorFilterConfEvent
structuresXE
"CSTACHangeMonitorFilterConfEvent structures"§

```

```

typedef struct CSTACHangeMonitorFilterConfEvent_t
{
    CSTAMonitorFilter_t filterList;}
CSTACHangeMonitorFilterConfEvent_t;CSTAMonitorStopConfEvent
structuresXE "CSTAMonitorStopConfEvent
structures"§

```

```

typedef struct CSTAMonitorStopConfEvent_t { Nulltype null;} CSTAMonitorStopConfEvent_t;

```

```

CSTAMonitorStopEvent structuresXE
"CSTAMonitorStopEvent structures"§

```

```

typedef struct CSTAMonitorStopEvent_t { InvokeID_t invokeID;}
CSTAMonitorStopEvent_t

```

Call Event ReportsXE "Call Event Reports"§

Call Event Report data structuresXE "Call Event Report data structures"§

```
typedef enum LocalConnectionState_t { CS_NULL = 0,
CS_INITIATE = 1, CS_ALERTING = 2, CS_CONNECT =
3, CS_HOLD = 4, CS_QUEUED = 5, CS_FAIL = 6}
LocalConnectionState_t;typedef
enum
CSTAEventCause_t { ACTIVE_MONITOR = 1,
ALTERNATE = 2, BUSY = 3, CALL_BACK = 4,
CALL_CANCELLED = 5, CALL_FORWARD_ALWAYS =
6, CALL_FORWARD_BUSY = 7,
CALL_FORWARD_NO_ANSWER = 8, CALL_FORWARD
= 9, CALL_NOT_ANSWERED = 10, CALL_PICKUP =
11, CAMP_ON = 12, DEST_NOT_OBTAINABLE = 13,
DO_NOT_DISTURB = 14,
INCOMPATIBLE_DESTINATION = 15,
INVALID_ACCOUNT_CODE = 16, KEY_CONFERENCE =
17, LOCKOUT = 18, MAINTENANCE = 19,
NETWORK_CONGESTION = 20,
NETWORK_NOT_OBTAINABLE = 21, NEW_CALL = 22,
NO_AVAILABLE_AGENTS = 23, OVERRIDE = 24,
PARK = 25, OVERFLOW = 26, RECALL = 27,
REDIRECTED = 28, REORDER_TONE = 29,
RESOURCES_NOT_AVAILABLE = 30,
SILENT_MONITOR = 31, TRANSFER = 32,
TRUNKS_BUSY = 33, VOICE_UNIT_INITIATOR = 34}
CSTAEventCause_t;CSTACallClearedEvent structuresXE
"CSTACallClearedEvent structures"§
```

```
typedef          struct          CSTACallClearedEvent_t
{
          ConnectionID_t          clearedCall;
LocalConnectionState_t          localConnectionInfo;
CSTAEventCause_t          cause;}
CSTACallClearedEvent_t;CSTAConferencedEvent
structuresXE "CSTAConferencedEvent structures"$
```

```

typedef enum DeviceIDType_t {  DEVICE_IDENTIFIER =
0,                               IMPLICIT_PUBLIC      =    20,
EXPLICIT_PUBLIC_UNKNOWN          =                    30,
EXPLICIT_PUBLIC_INTERNATIONAL    =                    31,
EXPLICIT_PUBLIC_NATIONAL         =                    32,
EXPLICIT_PUBLIC_NETWORK_SPECIFIC =                    33,
EXPLICIT_PUBLIC_SUBSCRIBER       =                    34,
EXPLICIT_PUBLIC_ABBREVIATED      =                    35,
IMPLICIT_PRIVATE                 =                    40,
EXPLICIT_PRIVATE_UNKNOWN         =                    50,
EXPLICIT_PRIVATE_LEVEL3_REGIONAL_NUMBER = 51,
EXPLICIT_PRIVATE_LEVEL2_REGIONAL_NUMBER = 52,
EXPLICIT_PRIVATE_LEVEL1_REGIONAL_NUMBER = 53,
EXPLICIT_PRIVATE_PTN_SPECIFIC_NUMBER  = 54,
EXPLICIT_PRIVATE_LOCAL_NUMBER        = 55,
EXPLICIT_PRIVATE_ABBREVIATED         = 56,
OTHER_PLAN = 60} DeviceIDType_t;typedef enum
DeviceIDStatus_t {  PROVIDED = 0,  NOT_KNOWN = 1,
NOT_REQUIRED = 2} DeviceIDStatus_t;typedef struct
ExtendedDeviceID_t {      DeviceID_t      deviceID;
DeviceIDType_t  deviceIDType;      DeviceIDStatus_t
deviceIDStatus;}      ExtendedDeviceID_t;typedef struct
ExtendedDeviceID_t SubjectDeviceID_t;typedef struct
CSTAConferencedEvent_t {      ConnectionID_t
primaryOldCall;      ConnectionID_t  secondaryOldCall;
SubjectDeviceID_t confController;      SubjectDeviceID_t
addedParty;      ConnectionList  conferenceConnections;
LocalConnectionState_t      localConnectionInfo;
CSTAEventCause_t      cause;}
CSTAConferencedEvent_t;CSTAConnectionClearedEven
t      structuresXE      "CSTAConnectionClearedEvent
structures"§

```

```

typedef      struct      CSTAConnectionClearedEvent_t
{
      ConnectionID_t      droppedConnection;
      SubjectDeviceID_t      releasingDevice;
      LocalConnectionState_t      localConnectionInfo;
      CSTAEventCause_t      cause;}
CSTAConnectionClearedEvent_t;CSTADeliveredEvent
structuresXE "CSTADeliveredEvent structures"§

```

```

typedef ExtendedDeviceID_t CallingDeviceID_t;typedef
ExtendedDeviceID_t      CalledDeviceID_t;typedef
ExtendedDeviceID_t      RedirectionDeviceID_t;typedef
struct CSTADeliveredEvent_t {      ConnectionID_t
connection;      SubjectDeviceID_t alertingDevice;
CallingDeviceID_t callingDevice;      CalledDeviceID_t
calledDevice;      RedirectionDeviceID_t
lastRedirectionDevice;      LocalConnectionState_t
localConnectionInfo;      CSTAEventCause_t cause;}
CSTADeliveredEvent_t;CSTADivertedEvent structuresXE
"CSTADivertedEvent structures"§

```

```

typedef struct CSTADivertedEvent_t {      ConnectionID_t
connection;      SubjectDeviceID_t divertingDevice;
CalledDeviceID_t      newDestination;
LocalConnectionState_t      localConnectionInfo;
CSTAEventCause_t      cause;}
CSTADivertedEvent_t;CSTAEstablishedEvent
structuresXE "CSTAEstablishedEvent structures"§

```



```

typedef          struct          CSTAEstablishedEvent_t
{
    ConnectionID_t          establishedConnection;
    SubjectDeviceID_t answeringDevice;  CallingDeviceID_t
callingDevice;              CalledDeviceID_t  calledDevice;
    RedirectionDeviceID_t          lastRedirectionDevice;
    LocalConnectionState_t          localConnectionInfo;
    CSTAEventCause_t          cause;}
CSTAEstablishedEvent_t;CSTAFailedEvent structuresXE
"CSTAFailedEvent structures"§

```

```

typedef struct CSTAFailedEvent_t {    ConnectionID_t
failedConnection;    SubjectDeviceID_t failingDevice;
CalledDeviceID_t calledDevice;    LocalConnectionState_t
localConnectionInfo;    CSTAEventCause_t cause;}
CSTAFailedEvent_t;CSTAHeldEvent    structuresXE
"CSTAHeldEvent structures"§

```

```

typedef struct CSTAHeldEvent_t {    ConnectionID_t
heldConnection;    SubjectDeviceID_t holdingDevice;
LocalConnectionState_t          localConnectionInfo;
CSTAEventCause_t          cause;}
CSTAHeldEvent_t;CSTANetworkReachedEvent
structuresXE " CSTANetworkReachedEvent structures"§

```

```

typedef          struct          CSTANetworkReachedEvent_t
{
    ConnectionID_t connection;    SubjectDeviceID_t
trunkUsed;              CalledDeviceID_t  calledDevice;
    LocalConnectionState_t          localConnectionInfo;
    CSTAEventCause_t          cause;}
CSTANetworkReachedEvent_t;CSTAOrginatedEvent
structures

```

```

XE "CSTAOriginatedEvent structures"§typedef struct
CSTAOriginatedEvent_t {      ConnectionID_t
originatedConnection;      SubjectDeviceID_t
callingDevice;      CalledDeviceID_t calledDevice;
LocalConnectionState_t      localConnectionInfo;
CSTAEventCause_t      cause;}
CSTAOriginatedEvent_t;CSTAQueuedEvent
structuresXE "CSTAQueuedEvent structures"§

```

```

typedef struct CSTAQueuedEvent_t {      ConnectionID_t
queuedConnection;      SubjectDeviceID_t queue;
CallingDeviceID_t callingDevice;      CalledDeviceID_t
calledDevice;      RedirectionDeviceID_t
lastRedirectionDevice;      int      numberQueued;
LocalConnectionState_t      localConnectionInfo;
CSTAEventCause_t      cause;}
CSTAQueuedEvent_t;CSTARetrievedEvent structuresXE
" CSTARetrievedEvent structures"§

```

```

typedef struct CSTARetrievedEvent_t {      ConnectionID_t
retrievedConnection;      SubjectDeviceID_t
retrievingDevice;      LocalConnectionState_t
localConnectionInfo;      CSTAEventCause_t cause;}
CSTARetrievedEvent_t;CSTAServiceInitiatedEvent
structuresXE " CSTAServiceInitiatedEvent structures"§

```

```

typedef      struct      CSTAServiceInitiatedEvent_t
{
      ConnectionID_t      initiatedConnection;
      LocalConnectionState_t      localConnectionInfo;
      CSTAEventCause_t      cause;}
CSTAServiceInitiatedEvent_t;CSTATransferedEvent
structuresXE "CSTATransferedEvent structures"§

```

```

typedef struct CSTATransferedEvent_t { ConnectionID_t primaryOldCall;
ConnectionID_t      secondaryOldCall;      SubjectDeviceID_t
transferringDevice;      SubjectDeviceID_t      transferredDevice;
ConnectionList      transferredConnections;      LocalConnectionState_t
localConnectionInfo;      CSTAEventCause_t      cause;}
CSTATransferedEvent_t;Feature Event ReportsXE "Feature Event
Reports"§

```

```

CSTACallInformationEvent      structuresXE
"CSTACallInformationEvent structures"§

```

```

typedef char      AccountInfo_t[32];
typedef char      AuthCode_t[32];
typedef struct CSTACallInformationEvent_t {
      ConnectionID_t connection;
      SubjectDeviceID_t device;
      AccountInfo_t accountInfo;
      AuthCode_t      authorisationCode;
} CSTACallInformationEvent_t;

```

CSTADoNotDisturbEvent **structuresXE**
"CSTADoNotDisturbEvent structures"§

```
typedef struct CSTADoNotDisturbEvent_t
{
    SubjectDeviceID_t device;
    Boolean doNotDisturbOn;}
CSTADoNotDisturbEvent_t;CSTAForwardingEvent
structuresXE "CSTAForwardingEvent structures"§
```

```
typedef struct CSTAForwardingEvent_t {
    SubjectDeviceID_t device;
    ForwardingInfo_t forwardingInformation;}
CSTAForwardingEvent_t;typedef struct CSTAMessageWaitingEvent_t
{
    CalledDeviceID_t deviceForMessage;
    SubjectDeviceID_t invokingDevice;
    Boolean messageWaitingOn;}
CSTAMessageWaitingEvent_t;Agent Status Report EventsXE "Agent
Status Report Events"§
```

CSTALoggedOnEvent **structuresXE**
"CSTALoggedOnEvent structures"§

```
typedef struct CSTALoggedOnEvent_t
{
    SubjectDeviceID_t agentDevice;
    AgentID_t agentID;
    AgentGroup_t agentGroup;
    AgentPassword_t password;}
CSTALoggedOnEvent_t;CSTALoggedOffEvent
structuresXE "CSTALoggedOffEvent structures"§
```

```

typedef          struct          CSTALoggedOffEvent_t
{          SubjectDeviceID_t agentDevice;          AgentID_t
agentID;          AgentGroup_t          agentGroup;}
CSTALoggedOffEvent_t;CSTANotReadyEvent
structuresXE "CSTANotReadyEvent structures"§

```

```

typedef          struct          CSTANotReadyEvent_t
{          SubjectDeviceID_t agentDevice;          AgentID_t
agentID;}          CSTANotReadyEvent_t;CSTARReadyEvent
structuresXE "CSTARReadyEvent structures"§

```

```

typedef struct CSTARReadyEvent_t {          SubjectDeviceID_t
agentDevice;          AgentID_t          agentID;}
CSTARReadyEvent_t;CSTAWorkNotReadyEvent
structuresXE "CSTAWorkNotReadyEvent structures"§

```

```

typedef struct CSTAWorkNotReadyEvent_t {          SubjectDeviceID_t agentDevice;          AgentID_t
agentID;} CSTAWorkNotReadyEvent_t;

```

```

CSTAWorkReadyEvent          structuresXE
"CSTAWorkReadyEvent structures"§

```

```

typedef struct CSTAWorkReadyEvent_t {          SubjectDeviceID_t agentDevice;          AgentID_t
agentID;} CSTAWorkReadyEvent_t;

```

```

Snapshot ServicesXE "Snapshot Services"§

```

```

CSTASnapshotDeviceConfEvent          structuresXE
"CSTASnapshotDeviceConfEvent structures"§

```

```

typedef struct CSTASnapshotDeviceData_t {          int          count;          struct
CSTASnapshotDeviceResponseInfo_t          *info;}          CSTASnapshotDeviceData_t;typedef struct
CSTASnapshotDeviceConfEvent_t {          CSTASnapshotDeviceData_t          snapshotData;}
CSTASnapshotDeviceConfEvent_t;

```

CSTASnapshotCallConfEvent structuresXE **"CSTASnapshotCallConfEvent structures"§**

```
typedef struct CSTASnapshotCallData_t {      int          count;      struct
CSTASnapshotCallResponseInfo_t      *info;}      CSTASnapshotCallData_t;typedef      struct
CSTASnapshotCallConfEvent_t      {      CSTASnapshotCallData_t      snapshotData;}
CSTASnapshotCallConfEvent_t;
```

CSTASnapshotDeviceConfEvent structuresXE **"CSTASnapshotDeviceConfEvent structures"§**

```
typedef enum CSTASimpleCallState_t {      CALL_NULL = 0,      CALL_PENDING = 1,
CALL_ORIGINATED = 3,      CALL_DELIVERED = 35,      CALL_DELIVERED_HELD = 36,
CALL_RECEIVED = 50,      CALL_ESTABLISHED = 51,      CALL_ESTABLISHED_HELD = 52,
CALL_RECEIVED_ON_HOLD = 66,      CALL_ESTABLISHED_ON_HOLD = 67,
CALL_QUEUED = 83,      CALL_QUEUED_HELD = 84,      CALL_FAILED = 99,
CALL_FAILED_HELD = 100} CSTASimpleCallState_t;
typedef struct CSTACallState_t {
    int          count;
    LocalConnectionState_t *state;
} CSTACallState_t;

/* Used to take a CSTACallState_t which contains only two
* LocalConnectionState_t and match them to the set of
* CSTASimpleCallState_t above.
*/#define SIMPLE_CALL_STATE(ccs)      (ccs.stat[0] + (ccs.state[1] << 4))
```

```

typedef struct CSTASnapshotDeviceResponseInfo_t { ConnectionID_t
callIdentifier; CSTACallState_t callState;}
CSTASnapshotDeviceResponseInfo_t;typedef struct
CSTASnapshotCallResponseInfo_t { SubjectDeviceID_t deviceOnCall;
ConnectionID_t callIdentifier; LocalConnectionState_t
localConnectionState;} CSTASnapshotResponseInfoEvent_t;Computing
Function ServicesXE "Computing Function Services"§

```

cstaRouteRegisterReq structuresXE
"cstaRouteRegisterReq structures"§

```

typedef struct CSTARouteRegisterReq_t {
DeviceID_t routingDevice;
} CSTARouteRegisterReq_t;

```

cstaRouteRegisterReqConfEvent structuresXE
"cstaRouteRegisterReqConfEvent structures"§

```

typedef long RegisterReqID_t;
typedef struct {
RegisterReqID_t registerReqID;} CSTARouteRegisterReqConfEvent_t;

```

cstaRouteRegisterCancel structuresXE
"cstaRouteRegisterCancel structures"§

```

typedef struct CSTARouteRegisterCancel_t {
RegisterRegID_t routingRegID;} CSTARouteRegisterCancel_t;

```

cstaRouteRegisterCancelConfEvent structuresXE
"cstaRouteRegisterCancelConfEvent structures"§

```

typedef struct {
RegisterRegID_t routingRegID;} CSTARouteRegisterCancelConfEvent_t;

```

cstaRouteRequestEvent structuresXE
"cstaRouteRequestEvent structures"§

```

typedef enum SelectValue_t { SV_NORMAL = 0, SV_LEAST_COST = 1, SV_EMERGENCY
= 2, SV_ACD = 3, SV_USER_DEFINED = 4} SelectValue_t;typedef struct SetUpValues_t
{ int length; unsigned char *value;} SetUpValues_t;

```

```

typedef struct CSTARouteRequestEvent_t { RegisterReqID_t registerReqID;
RoutingCrossRefID_t routingCrossRefID; DeviceID_t currentRoute;
DeviceID_t callingDevice; ConnectionID_t routedCall; SelectValue_t
routeSelAlgorithm; Boolean priority; SetupValues_t
setupInformation;
} CSTARouteRequestEvent_t;

```

cstaRouteSelect structuresXE "cstaRouteSelect structures"§

```

typedef int RetryValue_t;#define noListAvailable -1#define
noCountAvailable -2typedef struct CSTARouteSelect_t {
RegisterReqID_t registerReqID;
RoutingCrossRefID_t routingCrossRefID; DeviceID_t routeSelected;
RetryValue_t remainRetry; SetupValues_t setupInformation; Boolean
routeUsedReq;} CSTARouteSelect_t;

```

cstaReRouteEvent structuresXE "cstaReRouteEvent structures"§

```

typedef struct CSTAReRouteEvent_t {
RegisterReqID_t registerReqID;
RoutingCrossRefID_t routingCrossRefID;} CSTAReRouteEvent_t;

```

cstaRouteUsedEvent structuresXE "cstaRouteUsedEvent structures"§

```

typedef struct CSTARouteUsedEvent_t {
RegisterReqID_t registerReqID;
RoutingCrossRefID_t routingCrossRefID; DeviceID_t routeUsed;
DeviceID_t callingDevice; Boolean domain;}
CSTARouteUsedEvent_t;

```

cstaRouteEndEvent structuresXE "cstaRouteEndEvent structures"§

```

typedef struct CSTARouteEndEvent_t {
RegisterReqID_t registerReqID;
RoutingCrossRefID_t routingCrossRefID; CSTAUniversalFailure_t
errorValue;} CSTARouteEndEvent_t;

```


cstaRouteEnd structuresXE "cstaRouteEnd structures"§

```
typedef struct CSTARouteEnd_t {  
    RegisterReqID_t registerReqID;
```

**RoutingCrossRefID_t routingCrossRefID;
CSTAUniversalFailure_t errorValue;} CSTARouteEnd_t;Escape Services**

**XE "Escape Services"§cstaEscapeService structuresXE
"cstaEscapeService structures"§**

```
typedef struct CSTAEscapeService_t { Nulltype null;} CSTAEscapeService_t;
```

**CSTAEscapeServiceConfEvent structuresXE
"CSTAEscapeServiceConfEvent structures"§**

```
typedef struct CSTAEscapeServiceConfEvent_t { Nulltype null;}  
CSTAEscapeServiceConfEvent_t;
```

PrivateEvent structuresXE "PrivateEvent structures"§

```
typedef struct CSTAPrivateEvent_t { int length;  
    unsigned char *data;}  
CSTAPrivateEvent_t;PrivateStatusEvent structuresXE "  
PrivateStatusEvent structures"§
```

```
typedef struct CSTAPrivateStatusEvent_t {  
    Nulltype null;  
}
```

**cstaPrivateStatusEvent structuresXE
"cstaPrivateStatusEvent structures"§**

```
typedef struct CSTAEscapeServiceEventConf_t  
{  
    UniversalFailure_t error;  
};
```

CSTAEscapeServiceEvent structuresXE
"CSTAEscapeServiceEvent structures"§

```
typedef struct CSTAEscapeServiceEvent_t {  
    Nulltype null;  
};
```

cstaEscapeServiceConf structuresXE
"cstaEscapeServiceConf structures"§

```
typedef struct cstaEscapeServiceConf_t {  
    CSTAUniversalFailure_t error;  
};
```

cstaSendPrivateEvent structuresXE
"cstaSendPrivateEvent structures"§

```
typedef struct cstaSendPrivateEvent_t {  
    Nulltype null;  
};
```

Maintenance ServicesXE "Maintenance Services"§

CSTABackInServiceEvent structuresXE
"CSTABackInServiceEvent structures"§

```
typedef struct CSTABackInServiceEvent_t { DeviceID_t device; CSTAEventCause_t cause;}  
CSTABackInServiceEvent_t;
```

CSTAOutOfServiceEvent structures XE
"CSTAOutOfServiceEvent structures "§

```
typedef struct CSTAOutOfServiceEvent_t { DeviceID_t device; CSTAEventCause_t cause;}  
CSTAOutOfServiceEvent_t;
```

cstaSysStatReq structuresXE **"cstaSysStatReq structures"§**

```
typedef struct CSTASysStatReq_t (  
    Nulltype null;  
) CSTASysStatReq_t
```

CSTASysStatReqConfEvent structures XE **"CSTASysStatReqConfEvent structures"§**

```
typedef enum SystemStatus_t { SS_INITIALIZING = 0, SS_ENABLED = 1, SS_NORMAL = 2, SS_MESSAGES_LOST = 3, SS_DISABLED = 4, SS_OVERLOAD_IMMINENT = 5, SS_OVERLOAD_REACHED = 6, SS_OVERLOAD_RELIEVED = 7} SystemStatus_t;
typedef struct CSTASysStatReqConfEvent_t (
    SystemStatus_t          systemStatus;
) CSTASysStatReqConfEvent_t
```

cstaSysStatStart structures XE **"cstaSysStatStart structures"§**

```
typedef unsigned char SystemStatusFilter_t;
#define SS_Initializing          0x800
#define SS_Enabled              0x400
#define SS_Normal               0x200
#define SS_MessageLost         0x100
#define SS_Disabled            0x080
#define SS_OverloadImminent    0x040
#define SS_OverloadReached     0x020
#define SS_OverloadRelieved    0x010
typedef struct CSTASysStatStart_t (
    SystemStatusFilter_t statusFilter;
) CSTASysStatStart_t;
```

CSTASysStatStartConfEvent structures XE **"CSTASysStatStartConfEvent structures"§**

```
typedef struct CSTASysStatStartConfEvent_t (
    SystemStatusFilter_t systemFilter;
) CSTASysStatStartConfEvent_t
```

cstaSysStatStop structures XE **"cstaSysStatStop structures"§**

```
typedef struct CSTASysStatStop_t (
    Nulltype          null;
) CSTASysStatStop_t;
```

CSTASysStatStopConfEvent structures XE **"CSTASysStatStopConfEvent structures"§**

```
typedef struct CSTASysStatStopConfEvent_t (
    Nulltype          null;
) CSTASysStatStopConfEvent_t
```

cstaChangeSysStatFilter structures XE
"cstaChangeSysStatFilter structures"§

```
typedef struct CSTAChangeSysStatFilter_t (  
    SystemStatusFilter_t statusFilter;  
) CSTAChangeSysStatFilter_t;
```

CSTAChangeSysStatFilterConfEvent structuresXE
"CSTAChangeSysStatFilterConfEvent structures"§

```
typedef struct CSTAChangeSysStatFilerConfEvent_t  
{  
    SystemStatusFilter_t statusFilterSelected;  
    SystemStatusFilter_t statusFilterActive;  
} CSTAChangeSysStatFilterConfEvent_t;
```

CSTASysStatEvent structuresXE "CSTASysStatEvent
structures"§

```
typedef struct CSTASysStatEvent_t {  
    SystemStatus_t systemStatus;  
};
```

CSTASysStatReqEvent structuresXE "
CSTASysStatReqEvent structures"§

```
typedef struct CSTASysStatReqEvent_t {  
    Nulltype null;  
} CSTASysStatReqEvent_t;
```

cstaSysStatReqConf structures XE "cstaSysStatReqConf
structures"§

```
typedef struct CSTASysStatReqConf_t {  
    SystemStatus_t systemStatus;  
} CSTASysStatReqConf_t;
```

cstaSysStatEventSend structuresXE
"cstaSysStatEventSend structures"§

```
typedef struct CSTASysStatEventSend_t {  
    SystemStatus_t systemStatus;  
} CSTASysStatEventSend_t;
```

CSTA Control ServicesXE "CSTA Control Services"§

cstaGetAPICaps structures XE "cstaGetAPICaps structures"§

```
typedef struct CSTAGetAPICaps_t {  
    Nulltype null;  
} CSTAGetAPICaps_t;
```

CSTAGetAPICapsConfEvent structuresXE "CSTAGetAPICapsConfEvent structures"§

```
typedef struct CSTAGetAPICapsConfEvent_t {  
    int alternateCall;  
    int answerCall;  
    int callCompletion;  
    int clearCall;  
    int clearConnection;  
    int conferenceCall;  
    int consultationCall;  
    int deflectCall;  
    int pickupCall;  
    int groupPickupCall;  
    int holdCall;  
    int makeCall;  
    int makePredictiveCall;  
    int queryMwi;  
    int queryDnd;  
    int queryFwd;  
    int queryAgentState;  
    int queryLastNumber;  
    int queryDeviceInfo;  
    int reconnectCall;  
    int retrieveCall;  
    int setMwi;  
    int setDnd;  
    int setFwd;  
    int setAgentState;  
    int transferCall;  
    int eventReport;  
    int callClearedEvent;  
    int conferencedEvent;  
    int connectionClearedEvent;  
    int deliveredEvent;  
    int divertedEvent;  
    int establishedEvent;  
    int failedEvent;  
    int heldEvent;  
    int networkReachedEvent;  
    int originatedEvent;  
    int queuedEvent;  
    int retrievedEvent;
```

```

int     serviceInitiatedEvent;
int     transferredEvent;
int     callInformationEvent;
int     doNotDisturbEvent;
int     forwardingEvent;
int     messageWaitingEvent;
int     loggedOnEvent;
int     loggedOffEvent;
int     notReadyEvent;
int     readyEvent;
int     workNotReadyEvent;
int     workReadyEvent;
int     backInServiceEvent;
int     outOfServiceEvent;
int     privateEvent;
int     routeRequestEvent;
int     reRoute;
int     routeSelect;
int     routeUsedEvent;
int     routeEndEvent;
int     monitorDevice;
int     monitorCall;
int     monitorCallsViaDevice;
int     changeMonitorFilter;
int     monitorStop;
int     monitorEnded;
int     snapshotDeviceReq;
int     snapshotCallReq;
int     escapeService;
int     privateStatusEvent;
int     escapeServiceEvent;
int     escapeServiceConf;
int     sendPrivateEvent;
int     sysStatReq;
int     sysStatStart;
int     sysStatStop;
int     changeSysStatFilter;
int     sysStatReqEvent;
int     sysStatReqConf;
int     sysStatEvent;
} CSTAGetAPICapsConfEvent_t;

```

cstaGetDeviceList structuresXE "cstaGetDeviceList structures"§

```

typedef enum CSTALevel_t {
    CSTA_LEVEL1 = 1,
    CSTA_LEVEL2 = 2,
    CSTA_LEVEL3 = 3,
    CSTA_LEVEL4 = 4,
    CSTA_LEVEL5 = 5,
    CSTA_LEVEL6 = 6
} CSTALevel_t;

```

CSTAGetDeviceListConfEvent structuresXE "CSTAGetDeviceListConfEvent structures"§

```
typedef struct CSTAGetDeviceListConfEvent_t
{
    Level_t          acsLevelReq;
    int              totalDevices;
    DeviceID_t      *deviceIDs;
} CSTAGetDeviceListConfEvent_t;
```

CSTA Event StructuresXE "CSTA Event Structures"§

CSTA event typesXE "CSTA event types"§

```
#define CSTAREQUEST          3
#define CSTAUNSOLICITED     4
#define CSTACONFIRMATION    5
```

#define CSTAEVENTREPORT6CSTA Request Event structureXE "CSTA Request Event structure"§

```
typedef struct
{
    InvokeID_t      invokeID;
    union
    {
        CSTARouteRequestEvent_t      routeRequest;
        CSTARouteUsedEvent_t         routeUsed;
        CSTARouteEndEvent_t          routeEnd;
        CSTAEscapeSvcReqEvent_t      escapeSvcRequest;
        CSTASysStatReqEvent_t        sysStatRequest;
    } u;
} CSTARequestEvent;
```

CSTA Event Report structureXE "CSTA Event Report structure"§

```
typedef struct
{
    union
    {
        CSTARouteRegisterAbortEvent_t registerAbort;
        CSTARouteUsedEvent_t          routeUsed;
        CSTARouteEndEvent_t           routeEnd;
        CSTAPrivateEvent_t            privateEvent;
        CSTASysStatEvent_t            sysStat;
        CSTASysStatEndedEvent_t       sysStatEnded;
    } u;
} CSTAEventReport;
```

CSTA Unsolicited Event structureXE "CSTA Unsolicited Event structure"§

```

typedef struct
{
    CSTAMonitorCrossRefID_t    monitorCrossRefId;
    union
    {
        CSTACallClearedEvent_t    callCleared;
        CSTAConferencedEvent_t    conferenced;
        CSTAConnectionClearedEvent_t connectionCleared;
        CSTADeliveredEvent_t    delivered;
        CSTADivertedEvent_t    diverted;
        CSTAEstablishedEvent_t    established;
        CSTAFailedEvent_t        failed;
        CSTAHeldEvent_t          held;
        CSTANetworkReachedEvent_t networkReached;
        CSTAOriginatedEvent_t    originated;
        CSTAQueuedEvent_t        queued;
        CSTARetrievedEvent_t     retrieved;
        CSTAServiceInitiatedEvent_t serviceInitiated;
        CSTATransferredEvent_t    transferred;
        CSTACallInformationEvent_t callInformation;
        CSTADoNotDisturbEvent_t    doNotDisturb;
        CSTAForwardingEvent_t      forwarding;
        CSTAMessageWaitingEvent_t messageWaiting;
        CSTALoggedOnEvent_t        loggedOn;
        CSTALoggedOffEvent_t       loggedOff;
        CSTANotReadyEvent_t       notReady;
        CSTAReadyEvent_t          ready;
        CSTAWorkNotReadyEvent_t    workNotReady;
        CSTAWorkReadyEvent_t       workReady;
        CSTABackInServiceEvent_t    backInService;
        CSTAOutOfServiceEvent_t     outOfService;
        CSTAPrivateStatusEvent_t    privateStatus;
        CSTAMonitorEndedEvent_t     monitorEnded;
    } u;
} CSTAUnsolicitedEvent;

```


CSTA Confirmation Event structureXE "CSTA Confirmation Event structure"§

```

typedef struct {
    InvokeID_t          invokeID;
    union {
        CSTAAlternateCallConfEvent_t          alternateCall;
        CSTAAnswerCallConfEvent_t            answerCall;
        CSTACallCompletionConfEvent_t        callCompletion;
        CSTAClearCallConfEvent_t            clearCall;
        CSTAClearConnectionConfEvent_t      clearConnection;
        CSTAConferenceCallConfEvent_t       conferenceCall;
        CSTAConsultationCallConfEvent_t     consultationCall;
        CSTADeflectCallConfEvent_t         deflectCall;
        CSTAPickupCallConfEvent_t          pickupCall;
        CSTAGroupPickupCallConfEvent_t      groupPickupCall;
        CSTAHoldCallConfEvent_t            holdCall;
        CSTAMakeCallConfEvent_t            makeCall;
        CSTAMakePredictiveCallConfEvent_t   makePredictiveCall;
        CSTAQueryMwiConfEvent_t            queryMwi;
        CSTAQueryDndConfEvent_t           queryDnd;
        CSTAQueryFwdConfEvent_t           queryFwd;
        CSTAQueryAgentStateConfEvent_t     queryAgentState;
        CSTAQueryLastNumberConfEvent_t     queryLastNumber;
        CSTAQueryDeviceInfoConfEvent_t     queryDeviceInfo;
        CSTAReconnectCallConfEvent_t       reconnectCall;
        CSTARetrieveCallConfEvent_t        retrieveCall;
        CSTASetMwiConfEvent_t              setMwi;
        CSTASetDndConfEvent_t              setDnd;
        CSTASetFwdConfEvent_t              setFwd;
        CSTASetAgentStateConfEvent_t       setAgentState;
        CSTATransferCallConfEvent_t        transferCall;
        CSTAUniversalFailureConfEvent_t    universalFailure;
        CSTAMonitorConfEvent_t             monitorStart;
        CSTAChangeMonitorFilterConfEvent_t changeMonitorFilter;
        CSTAMonitorStopConfEvent_t        monitorStop;
        CSTASnapshotDeviceConfEvent_t     snapshotDevice;
        CSTASnapshotCallConfEvent_t       snapshotCall;
        CSTARouteRegisterReqConfEvent_t   routeRegister;
        CSTARouteRegisterCancelConfEvent_t routeCancel;
        CSTAEscapeSvcConfEvent_t          escapeService;
        CSTASysStatReqConfEvent_t         sysStatReq;
        CSTASysStatStartConfEvent_t       sysStatStart;
        CSTASysStatStopConfEvent_t       sysStatStop;
        CSTAChangeSysStatFilterConfEvent_t changeSysStatFilter;
    } u;
} CSTAConfirmationEvent;

```

CSTA Event_t structureXE "CSTA Event_t structure"§

```

typedef struct
{
    ACSEventHeader_t  eventHeader;
    union
    {
        ACSUnsolicitedEvent acsUnsolicited;
    }
}

```

```
        ACSConfirmationEvent    acsConfirmation;
        CSTARequestEvent    cstaRequest;
        CSTAUnsolicitedEvent    cstaUnsolicited;
        CSTAConfirmationEvent    cstaConfirmation;
        CSTAEventReport    cstaEventReport;
    } event;
    char    heap[CSTA_MAX_HEAP];
} CSTAEvent_t;
```

10-34 CSTA Data Types