CAP

Table of Contents

The (Customized Applications for Mobile network Enhanced Logic) Application Part (CAP) protocol is used by network operators to provide their subscribers with operator specific services even when roaming outside the HPLMN. provides services such as prepaid roaming services, fraud control, special numbers and closed user groups (e.g., office extension numbers that work everywhere). CAP has been defined in four versions (phases), each of which has an accompanying specification that builds upon the previous phase. This application layer provides a standardized set of services. uses the services of the SS7network, specifically the Signaling Connection Control Part (SCCP) and the Transaction Capabilities Application Part (TCAP)

For better understanding of this chapter you must be familiar with the specifications defined in 3GPP TS 22.078 (service aspects) and 3GPP TS 23.078 (technical realization).

Restcomm jSS7 CAP has full implementation for Phase 1 and Phase 2. The list of implemented operations can be found at CAPMessagesImplemented.ods. You can find the list of implemented operations here: CAPMessagesImplemented.ods. The document uses color coding to represent the status of the implementation:

Green

Fully Implemented

Red

Interface available (but not yet implemented)

Yellow

Implementation in progress

Blue

Implemeted only for CAP V2 and not available for CAP V3 and V4

Restcomm jSS7 Any contribution to implement other messages are welcome. We will provide you with all the help that you may require initially.

jSS7 CAP

The org.mobicents.protocols.ss7.cap.api.CAPStackinterface defines the methods required to

represent CAP Protocol Stack. CAPStack exposes `org.mobicents.protocols.ss7.cap.api.CAPProvider` that interacts directly with CAPStack. This interface defines the methods that will be used by any registered CAP-User application. CAP-User application must implement interfaces to listen for CAP messages and for dialogue and component handling primitives. One interface is `org.mobicents.protocols.ss7.cap.api.CAPDialogListener` (listening associated with dialog events). Others are one or more interfaces for listening to messages associated with component events based on `org.mobicents.protocols.ss7.cap.api.CAPServiceListener` interface.

Each CAP-User interested in listening to messages specific to a CAP Service must implement the specific `CAPServiceListener`.

- CAP-User interested only in circuit switched call operations implements `org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CAPServiceCircuitSwitchedCallListener`
- CAP-User interested only in GPRS operations implements `org.mobicents.protocols.ss7.cap.api.service.gprs.CAPServiceGprsListener`
- CAP-User interested only in SMS operations implements `org.mobicents.protocols.ss7.cap.api.service.sms.CAPServiceSmsListener`

CAP-User interested in all the services must implement all the service listener classes.

The `org.mobicents.protocols.ss7.cap.CAPStackImpl` is a concrete implementation of `CAPStack`. The CAP-User application gets access to `CAPProvider` by doing a JNDI look-up as explained in [_design_overview_ss7_service].

```
String providerJndiName = "java:/mobicents/ss7/cap";
this.capProvider = ((CAPProvider) ctx.lookup(providerJndiName));
...
```

The CAP-User application should register the concrete implementation of `CAPDialogListener` with `CAPProvider` to listen for incoming CAP Dialog and CAP Primitive messages. The CAP-User application should register the concrete implementation of `CAPServiceListener` with corresponding `CAPServiceBase ` to listen for incoming CAP Service messages. Following CAPServiceBase are exposed by [class]`CAPProvider`.

- For circuit switched call service `org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CAPServiceCircuitSwitchedCall`
- For GPRS service `org.mobicents.protocols.ss7.cap.api.service.gprs.CAPServiceGprs`
- For SMS service `org.mobicents.protocols.ss7.cap.api.service.sms.CAPServiceSms`

```
public class CAPExample implements CAPDialogListener,
CAPServiceCircuitSwitchedCallListener {
        ....
        capProvider.addCAPDialogListener(this);
        capProvider.getCAPServiceCircuitSwitchedCall().addCAPServiceListener(this);
        ....
}
```

Before any CAP specific service can be used, the corresponding service should be activated as shown below:

```
        ....
        // Make the circuitSwitchedCall service activated
        capProvider.getCAPServiceCircuitSwitchedCall().acivate();
        ....
```

The CAP-User Application leverages:

- `MAPParameterFactory` to create instances of `IMSI`, `ISDNAddressString`and many other MAP primitives that are used by CAP services.
- `CAPParameterFactory` to create instances of CAP primitives that are used by CAP services.
- `ISUPParameterFactory` to create instances of ISUP primitives that are used by CAP services.
- `INAPPParameterFactory` to create instances of INAP primitives that are used by CAP services.
- `CAPErrorMessageFactory` to create instances of CAP error messages like `CAPErrorMessageSystemFailure`.

```
        MapParameterFactory mapParamFact = capProvider.getMapServiceFactory();
        CapParameterFactory capParamFact = capProvider.getCapServiceFactory();
        ISDNAddressString msisdn = mapParamFact.createISDNAddressString(
                AddressNature.international_number, NumberingPlan.ISDN,
                "31628838002");
        BCSMEvent ev = capParamFact.createBCSMEvent(EventTypeBCSM.routeSelectFailure,
                MonitorMode.notifyAndContinue, null, null, false);
```

jSS7 Sending a CAP request (InitialDPRequest as an example)

For sending a CAP request you must do the following at the SSF side:

- Create a new CAP Dialog

```
// First create Dialog
SccpAddress origAddress = createLocalAddress();
SccpAddress destAddress = createRemoteAddress();
CAPApplicationContext acn = CAPApplicationContext.CapV2_gsmSSF_to_gsmSCF;
currentCapDialog = capProvider.getCAPServiceCircuitSwitchedCall().
    createNewDialog(acn, origAddress, remoteAddress);
```

- Add an Invoke component (InitialDPRequest message)

```
int serviceKey = 1;

CalledPartyNumber calledPartyNumber = client.getCAPProvider().
    getISUPParameterFactory().createCalledPartyNumber();
calledPartyNumber.setAddress("552348762");
calledPartyNumber.setNatureOfAddresIndicator(NAINumber._NAI_INTERNATIONAL_NUMBER);
calledPartyNumber.setNumberingPlanIndicator(CalledPartyNumber._NPI_ISDN);
calledPartyNumber.setInternalNetworkNumberIndicator(CalledPartyNumber._INN_ROUTING_ALL
OWED);
CalledPartyNumberCap calledPartyNumberCap = client.getCAPProvider().
    getCAPParameterFactory().createCalledPartyNumberCap(calledPartyNumber);

CallingPartyNumber callingPartyNumber = client.getCAPProvider().
    getISUPParameterFactory().createCallingPartyNumber();
callingPartyNumber.setAddress("55998223");
callingPartyNumber.setNatureOfAddresIndicator(NAINumber._NAI_INTERNATIONAL_NUMBER);
callingPartyNumber.setNumberingPlanIndicator(CalledPartyNumber._NPI_ISDN);
callingPartyNumber.setAddressRepresentationREstrictedIndicator(CallingPartyNumber._APR
I_ALLOWED);
callingPartyNumber.setScreeningIndicator(CallingPartyNumber._SI_NETWORK_PROVIDED);
CallingPartyNumberCap callingPartyNumberCap = client.getCAPProvider().
    getCAPParameterFactory().createCallingPartyNumberCap(callingPartyNumber);

LocationNumber locationNumber = client.getCAPProvider().getISUPParameterFactory().
    createLocationNumber();
locationNumber.setAddress("55200001");
locationNumber.setNatureOfAddresIndicator(NAINumber._NAI_INTERNATIONAL_NUMBER);
locationNumber.setNumberingPlanIndicator(LocationNumber._NPI_ISDN);
locationNumber.setAddressRepresentationRestrictedIndicator(LocationNumber._APRI_ALLOWE
D);
locationNumber.setScreeningIndicator(LocationNumber._SI_NETWORK_PROVIDED);
locationNumber.setInternalNetworkNumberIndicator(LocationNumber._INN_ROUTING_ALLOWED);
LocationNumberCap locationNumberCap = client.getCAPProvider().
getCAPParameterFactory().
    createLocationNumberCap(locationNumber);

ISDNAddressString vlrNumber = client.getCAPProvider().getMAPParameterFactory()
        .createISDNAddressString(AddressNature.international_number,
        NumberingPlan.ISDN, "552000002");
LocationInformation locationInformation = client.getCAPProvider
().getMAPParameterFactory()
        .createLocationInformation(10, null, vlrNumber, null,
        null, null, null, vlrNumber, null, false, false, null, null);

currentCapDialog.addInitialDPRequest(serviceKey, calledPartyNumber,
callingPartyNumber,
        null, null, null, locationNumber, null, null, null, null, null,
        eventTypeBCSM, null, null, null, null, null, null, null, false,
        null, null, locationInformation, null, null, null, null, null, false, null);
```

- Send a TC-Begin message to the server peer

```
// This will initiate the TC-BEGIN with INVOKE component
currentCapDialog.send();
```

- Wait for a response and other instructions from the SCF

At the SCF side when the TC-Begin message is received, the following sequence of events occur:

```
void CAPDialogListener.onDialogRequest(CAPDialog capDialog,
    CAPGprsReferenceNumber capGprsReferenceNumber);
```

This is the request for CAP Dialog processing. A CAP-User can reject the Dialog by invoking `capDialog.refuse()` method.

Then the incoming primitives corresponding events (one or more) occur. In this case it is

```
void CAPServiceCircuitSwitchedCallListener.onInitialDPRequest(InitialDPRequest ind);
```

When processing component-dependant messages you can add response components or add other Invoke requests. In this case it is RequestReportBCSMEventRequest as an example:

```
ArrayList<BCSMEvent> bcsmEventList = new ArrayList<BCSMEvent>();
BCSMEvent ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(
    EventTypeBCSM.routeSelectFailure, MonitorMode.notifyAndContinue,
    null, null, false);
bcsmEventList.add(ev);
ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(EventTypeBCSM
.oCalledPartyBusy,
    MonitorMode.interrupted, null, null, false);
bcsmEventList.add(ev);
ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(EventTypeBCSM.
oNoAnswer,
    MonitorMode.interrupted, null, null, false);
bcsmEventList.add(ev);
ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(EventTypeBCSM.oAnswer,
    MonitorMode.notifyAndContinue, null, null, false);
bcsmEventList.add(ev);
LegID legId = this.capProvider.getINAPParameterFactory().createLegID(true, LegType
.leg1);
ev = this.capProvider.getCAPParameterFactory()
        .createBCSMEvent(EventTypeBCSM.oDisconnect, MonitorMode.notifyAndContinue,
legId, null, false);
bcsmEventList.add(ev);
legId = this.capProvider.getINAPParameterFactory().createLegID(true, LegType.leg2);
ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(EventTypeBCSM
.oDisconnect,
    MonitorMode.interrupted, legId, null, false);
bcsmEventList.add(ev);
ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(EventTypeBCSM.oAbandon,
    MonitorMode.notifyAndContinue, null, null, false);
bcsmEventList.add(ev);
currentCapDialog.addRequestReportBCSMEventRequest(bcsmEventList, null);

currentCapDialog.send();
```

If preparing the response takes time, you should return the control and prepare the answer in a separate thread.

If error or reject primitives are included into a TCAP message the following events occur:

```
public void onErrorComponent(CAPDialog capDialog, Long invokeId, CAPErrorMessage
capErrorMessage);
public void onProviderErrorComponent(CAPDialog capDialog, Long invokeId,
    CAPComponentErrorReason providerError);
public void onRejectComponent(CAPDialog capDialog, Long invokeId, Problem problem);
```

After all incoming components have been processed, the event `onDialogDelimiter(CAPDialog capDialog);` event is invoked (or `onDialogClose(CAPDialog capDialog)` in TC-END case). If all response components have been prepared you can tell the Stack to send response:

- `capDialog.close(false);` - to send TC-END
- `capDialog.send();` - to send TC-CONTINUE
- `capDialog.close(true);` - sends TC-END without any components (prearrangedEnd)

Instead of the methods `send()` and `close(boolean prearrangedEnd)`, you can invoke `sendDelayed()` or `closeDelayed(boolean prearrangedEnd)`. These methods are the same as `send()` and `close(boolean prearrangedEnd)` methods, but when invoking them from events of parsing incoming components real sending and dialog closing will occur only when all incoming component events and onDialogDelimiter() or onDialogClose() would be processed. If all response components have been prepared you should return the control and send a response when all components are ready.

In case of an error, you can terminate a CAP dialog in any method by invoking:

- `capDialog.abort(CAPUserAbortReason abortReason);` - sends TC-U-ABORT primitive

If there are no local actions or no response from a remote size for a long time, timeouts occur and the following methods are invoked:

- `CAPDialogListener.onDialogTimeout(CAPDialog capDialog);`
- `CAPServiceListener.onInvokeTimeout(CAPDialog capDialog, Long invokeId);`

In the `onDialogTimeout()` method you can invoke `capDialog.keepAlive();` to prevent a Dialog from closing. For preventing an Invoke timeout you should invoke `resetInvokeTimer(Long invokeId);` (before `onInvokeTimeout()` occurs).

jSS7 CAP Usage

```
package org.mobicents.protocols.ss7.cap;

import org.mobicents.protocols.ss7.cap.api.EsiBcsm.OAnswerSpecificInfo;
import org.mobicents.protocols.ss7.cap.api.EsiBcsm.ODisconnectSpecificInfo;
import org.mobicents.protocols.ss7.cap.api.isup.CalledPartyNumberCap;
import org.mobicents.protocols.ss7.cap.api.isup.CallingPartyNumberCap;
import org.mobicents.protocols.ss7.cap.api.isup.CauseCap;
import org.mobicents.protocols.ss7.cap.api.isup.LocationNumberCap;
import org.mobicents.protocols.ss7.cap.api.primitives.EventTypeBCSM;
import org.mobicents.protocols.ss7.cap.api.primitives.ReceivingSideID;
import org.mobicents.protocols.ss7.inap.api.primitives.LegType;
import org.mobicents.protocols.ss7.inap.api.primitives.MiscCallInfo;
import org.mobicents.protocols.ss7.inap.api.primitives.MiscCallInfoMessageType;
import org.mobicents.protocols.ss7.indicator.RoutingIndicator;
import org.mobicents.protocols.ss7.isup.message.parameter.CalledPartyNumber;
import org.mobicents.protocols.ss7.isup.message.parameter.CallingPartyNumber;
import org.mobicents.protocols.ss7.isup.message.parameter.CauseIndicators;
import org.mobicents.protocols.ss7.isup.message.parameter.LocationNumber;
import org.mobicents.protocols.ss7.isup.message.parameter.NAINumber;
import org.mobicents.protocols.ss7.map.api.primitives.AddressNature;
import org.mobicents.protocols.ss7.map.api.primitives.ISDNAddressString;
```

```
import org.mobicents.protocols.ss7.map.api.primitives.NumberingPlan;
import
org.mobicents.protocols.ss7.map.api.service.mobility.subscriberInformation.LocationInf
ormation;
import org.mobicents.protocols.ss7.sccp.parameter.SccpAddress;

public class Example {

    private static SccpAddress createLocalAddress() {
        return new SccpAddress(RoutingIndicator.ROUTING_BASED_ON_DPC_AND_SSN, 1, null,
146);
    }

    private static SccpAddress createRemoteAddress() {
        return new SccpAddress(RoutingIndicator.ROUTING_BASED_ON_DPC_AND_SSN, 2, null,
146);
    }


    public static void startCallSsf() throws Exception {
        CallSsfExample client = new CallSsfExample();

        client.start();

        // starting a call
        SccpAddress origAddress = createLocalAddress();
        SccpAddress destAddress = createRemoteAddress();

        int serviceKey = 1;

        CalledPartyNumber calledPartyNumber = client.getCAPProvider
().getISUPParameterFactory()
            .createCalledPartyNumber();
        calledPartyNumber.setAddress("552348762");
        calledPartyNumber.setNatureOfAddresIndicator(NAINumber
._NAI_INTERNATIONAL_NUMBER);
        calledPartyNumber.setNumberingPlanIndicator(CalledPartyNumber._NPI_ISDN);
        calledPartyNumber.setInternalNetworkNumberIndicator(CalledPartyNumber
._INN_ROUTING_ALLOWED);
        CalledPartyNumberCap calledPartyNumberCap = client.getCAPProvider()
            .getCAPParameterFactory().createCalledPartyNumberCap(calledPartyNumber);

        CallingPartyNumber callingPartyNumber = client.getCAPProvider
().getISUPParameterFactory()
            .createCallingPartyNumber();
        callingPartyNumber.setAddress("55998223");
        callingPartyNumber.setNatureOfAddresIndicator(NAINumber
._NAI_INTERNATIONAL_NUMBER);
        callingPartyNumber.setNumberingPlanIndicator(CalledPartyNumber._NPI_ISDN);
        callingPartyNumber.setAddressRepresentationREstrictedIndicator
(CallingPartyNumber._APRI_ALLOWED);
```

```
        callingPartyNumber.setScreeningIndicator(CallingPartyNumber
._SI_NETWORK_PROVIDED);
        CallingPartyNumberCap callingPartyNumberCap = client.getCAPProvider()
            .getCAPParameterFactory().createCallingPartyNumberCap(callingPartyNumber);

        LocationNumber locationNumber = client.getCAPProvider
().getISUPParameterFactory()
        .createLocationNumber();
        locationNumber.setAddress("55200001");
        locationNumber.setNatureOfAddresIndicator(NAINumber.
_NAI_INTERNATIONAL_NUMBER);
        locationNumber.setNumberingPlanIndicator(LocationNumber._NPI_ISDN);
        locationNumber.setAddressRepresentationRestrictedIndicator(LocationNumber
._APRI_ALLOWED);
        locationNumber.setScreeningIndicator(LocationNumber._SI_NETWORK_PROVIDED);
        locationNumber.setInternalNetworkNumberIndicator(LocationNumber
._INN_ROUTING_ALLOWED);
        LocationNumberCap locationNumberCap = client.getCAPProvider
().getCAPParameterFactory()
            .createLocationNumberCap(locationNumber);

        ISDNAddressString vlrNumber = client.getCAPProvider().getMAPParameterFactory()
                .createISDNAddressString(AddressNature.international_number,
NumberingPlan.ISDN,
                        "552000002");
        LocationInformation locationInformation = client.getCAPProvider
().getMAPParameterFactory()
                .createLocationInformation(10, null, vlrNumber, null, null, null,
null,
                        vlrNumber, null, false, false, null, null);

        client.sendInitialDP(origAddress, destAddress, serviceKey,
calledPartyNumberCap,
                callingPartyNumberCap, locationNumberCap, EventTypeBCSM.collectedInfo,
                locationInformation);

        // sending oAnswer in 5 sec
        Thread.sleep(5000);
        OAnswerSpecificInfo oAnswerSpecificInfo = client.getCAPProvider
().getCAPParameterFactory()
                .createOAnswerSpecificInfo(null, false, false, null, null, null);
        ReceivingSideID legID = client.getCAPProvider().getCAPParameterFactory()
            .createReceivingSideID(LegType.leg2);
        MiscCallInfo miscCallInfo = client.getCAPProvider().getINAPParameterFactory()
            .createMiscCallInfo(MiscCallInfoMessageType.notification, null);
        client.sendEventReportBCSM_OAnswer(oAnswerSpecificInfo, legID, miscCallInfo);

        // sending oDisconnect in 20 sec
        Thread.sleep(20000);
        CauseIndicators causeIndicators = client.getCAPProvider
().getISUPParameterFactory()
```

```
            .createCauseIndicators();
        causeIndicators.setLocation(CauseIndicators._LOCATION_USER);
        causeIndicators.setCodingStandard(CauseIndicators._CODING_STANDARD_ITUT);
        causeIndicators.setCauseValue(CauseIndicators._CV_ALL_CLEAR);
        CauseCap releaseCause = client.getCAPProvider().getCAPParameterFactory()
            .createCauseCap(causeIndicators);
        ODisconnectSpecificInfo oDisconnectSpecificInfo = client.getCAPProvider()
            .getCAPParameterFactory().createODisconnectSpecificInfo(releaseCause);
        legID = client.getCAPProvider().getCAPParameterFactory().
createReceivingSideID(LegType.leg1);
        miscCallInfo = client.getCAPProvider().getINAPParameterFactory()
            .createMiscCallInfo(MiscCallInfoMessageType.notification, null);
        client.sendEventReportBCSM_ODisconnect(oDisconnectSpecificInfo, legID,
miscCallInfo);

        // wait for answer
        Thread.sleep(600000);

        client.stop();
    }

    public static void startCallScf() throws Exception {
        CallScfExample server = new CallScfExample();

        server.start();

        // wait for a request
        Thread.sleep(600000);

        server.stop();
    }
}
```

```
package org.mobicents.protocols.ss7.cap;

import javax.naming.InitialContext;
import javax.naming.NamingException;

import org.mobicents.protocols.ss7.cap.api.CAPApplicationContext;
import org.mobicents.protocols.ss7.cap.api.CAPDialog;
import org.mobicents.protocols.ss7.cap.api.CAPDialogListener;
import org.mobicents.protocols.ss7.cap.api.CAPException;
import org.mobicents.protocols.ss7.cap.api.CAPMessage;
import org.mobicents.protocols.ss7.cap.api.CAPParameterFactory;
import org.mobicents.protocols.ss7.cap.api.CAPProvider;
import org.mobicents.protocols.ss7.cap.api.CAPStack;
import org.mobicents.protocols.ss7.cap.api.EsiBcsm.OAnswerSpecificInfo;
import org.mobicents.protocols.ss7.cap.api.EsiBcsm.ODisconnectSpecificInfo;
import org.mobicents.protocols.ss7.cap.api.dialog.CAPGeneralAbortReason;
import org.mobicents.protocols.ss7.cap.api.dialog.CAPGprsReferenceNumber;
```

```
import org.mobicents.protocols.ss7.cap.api.dialog.CAPNoticeProblemDiagnostic;
import org.mobicents.protocols.ss7.cap.api.dialog.CAPUserAbortReason;
import org.mobicents.protocols.ss7.cap.api.errors.CAPErrorMessage;
import org.mobicents.protocols.ss7.cap.api.isup.CalledPartyNumberCap;
import org.mobicents.protocols.ss7.cap.api.isup.CallingPartyNumberCap;
import org.mobicents.protocols.ss7.cap.api.isup.LocationNumberCap;
import org.mobicents.protocols.ss7.cap.api.primitives.EventTypeBCSM;
import org.mobicents.protocols.ss7.cap.api.primitives.ReceivingSideID;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ActivityTestRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ActivityTestResponse;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ApplyChargingReportReq
uest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ApplyChargingRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.AssistRequestInstructi
onsRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CAPDialogCircuitSwitch
edCall;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CAPServiceCircuitSwitc
hedCallListener;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CallInformationReportR
equest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CallInformationRequest
Request;
import org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CancelRequest;
import org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ConnectRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ConnectToResourceReque
st;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ContinueRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.DisconnectForwardConne
ctionRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.EstablishTemporaryConn
ectionRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.EventReportBCSMRequest
;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.FurnishChargingInforma
tionRequest;
```

```
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.InitialDPRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.PlayAnnouncementReques
t;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.PromptAndCollectUserIn
formationRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.PromptAndCollectUserIn
formationResponse;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ReleaseCallRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.RequestReportBCSMEvent
Request;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ResetTimerRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.SendChargingInformatio
nRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.SpecializedResourceRep
ortRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.primitive.DestinationR
outingAddress;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.primitive.EventSpecifi
cInformationBCSM;
import org.mobicents.protocols.ss7.inap.api.primitives.MiscCallInfo;
import org.mobicents.protocols.ss7.map.api.MAPProvider;
import
org.mobicents.protocols.ss7.map.api.service.mobility.subscriberInformation.LocationInf
ormation;
import org.mobicents.protocols.ss7.sccp.SccpProvider;
import org.mobicents.protocols.ss7.sccp.parameter.SccpAddress;
import org.mobicents.protocols.ss7.tcap.asn.comp.PAbortCauseType;
import org.mobicents.protocols.ss7.tcap.asn.comp.Problem;

public class CallSsfExample implements CAPDialogListener,
CAPServiceCircuitSwitchedCallListener {

    private CAPProvider capProvider;
    private CAPParameterFactory paramFact;
    private CAPDialogCircuitSwitchedCall currentCapDialog;
    private CallContent cc;

    public CallSsfExample() throws NamingException {
        InitialContext ctx = new InitialContext();
        try {
```

```
            String providerJndiName = "java:/mobicents/ss7/cap";
            this.capProvider = ((CAPProvider) ctx.lookup(providerJndiName));
        } finally {
            ctx.close();
        }

        paramFact = capProvider.getCAPParameterFactory();

        capProvider.addCAPDialogListener(this);
        capProvider.getCAPServiceCircuitSwitchedCall().addCAPServiceListener(this);
    }

    public CAPProvider getCAPProvider() {
        return capProvider;
    }

    public void start() {
        // Make the circuitSwitchedCall service activated
        capProvider.getCAPServiceCircuitSwitchedCall().acivate();

        currentCapDialog = null;
    }

    public void stop() {
         capProvider.getCAPServiceCircuitSwitchedCall().deactivate();
    }

    public void sendInitialDP(SccpAddress origAddress, SccpAddress remoteAddress,
            int serviceKey, CalledPartyNumberCap calledPartyNumber,
            CallingPartyNumberCap callingPartyNumber, LocationNumberCap
locationNumber,
            EventTypeBCSM eventTypeBCSM, LocationInformation locationInformation)
            throws CAPException {
        // First create Dialog
        CAPApplicationContext acn = CAPApplicationContext.CapV2_gsmSSF_to_gsmSCF;
        currentCapDialog = capProvider.getCAPServiceCircuitSwitchedCall
().createNewDialog(
                acn, origAddress, remoteAddress);

        currentCapDialog.addInitialDPRequest(serviceKey, calledPartyNumber,
                callingPartyNumber, null, null, null, locationNumber, null, null,
                null, null, null,
                eventTypeBCSM, null, null, null, null, null, null, null, false,
                null, null, locationInformation, null, null, null, null, null, false,
null);
        // This will initiate the TC-BEGIN with INVOKE component
        currentCapDialog.send();

        this.cc.step = Step.initialDPSent;
        this.cc.calledPartyNumber = calledPartyNumber;
        this.cc.callingPartyNumber = callingPartyNumber;
```

```
    }

    public void sendEventReportBCSM_OAnswer(OAnswerSpecificInfo oAnswerSpecificInfo,
            ReceivingSideID legID, MiscCallInfo miscCallInfo) throws CAPException {
        if (currentCapDialog != null && this.cc != null) {
            EventSpecificInformationBCSM eventSpecificInformationBCSM =
                this.capProvider.getCAPParameterFactory
().createEventSpecificInformationBCSM(
                        oAnswerSpecificInfo);
            currentCapDialog.addEventReportBCSMRequest(EventTypeBCSM.oAnswer,
                    eventSpecificInformationBCSM, legID, miscCallInfo, null);
            currentCapDialog.send();
            this.cc.step = Step.answered;
        }
    }

    public void sendEventReportBCSM_ODisconnect(ODisconnectSpecificInfo
oDisconnectSpecificInfo,
            ReceivingSideID legID, MiscCallInfo miscCallInfo) throws CAPException {
        if (currentCapDialog != null && this.cc != null) {
            EventSpecificInformationBCSM eventSpecificInformationBCSM =
                this.capProvider.getCAPParameterFactory
().createEventSpecificInformationBCSM(
                        oDisconnectSpecificInfo);
            currentCapDialog.addEventReportBCSMRequest(EventTypeBCSM.oDisconnect,
                    eventSpecificInformationBCSM, legID, miscCallInfo, null);
            currentCapDialog.send();
            this.cc.step = Step.disconnected;
        }
    }

    @Override
    public void onRequestReportBCSMEventRequest(RequestReportBCSMEventRequest ind) {
        if (currentCapDialog != null && this.cc != null && this.cc.step != Step
.disconnected) {
            this.cc.requestReportBCSMEventRequest = ind;

            // initiating BCSM events processing
        }
        ind.getCAPDialog().processInvokeWithoutAnswer(ind.getInvokeId());
    }

    @Override
    public void onActivityTestRequest(ActivityTestRequest ind) {
        if (currentCapDialog != null && this.cc != null && this.cc.step != Step
.disconnected) {
            this.cc.activityTestInvokeId = ind.getInvokeId();
        }
    }
```

```
    @Override
    public void onActivityTestResponse(ActivityTestResponse ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onContinueRequest(ContinueRequest ind) {
        this.cc.step = Step.callAllowed;
        ind.getCAPDialog().processInvokeWithoutAnswer(ind.getInvokeId());
        // sending Continue to use the original calledPartyAddress
    }

    @Override
    public void onConnectRequest(ConnectRequest ind) {
        this.cc.step = Step.callAllowed;
        this.cc.destinationRoutingAddress = ind.getDestinationRoutingAddress();
        ind.getCAPDialog().processInvokeWithoutAnswer(ind.getInvokeId());
        // sending Connect to force routing the call to a new  number
    }

    @Override
    public void onDialogTimeout(CAPDialog capDialog) {
        if (currentCapDialog != null && this.cc != null && this.cc.step != Step
.disconnected) {
            // if the call is still up - keep the sialog alive
            currentCapDialog.keepAlive();
        }
    }

    @Override
    public void onDialogDelimiter(CAPDialog capDialog) {
        if (currentCapDialog != null && this.cc != null && this.cc.step != Step
.disconnected) {
            if (this.cc.activityTestInvokeId != null) {
                try {
                    currentCapDialog.addActivityTestResponse(this.cc
.activityTestInvokeId);
                    this.cc.activityTestInvokeId = null;
                    currentCapDialog.send();
                } catch (CAPException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
        }
    }

    @Override
    public void onErrorComponent(CAPDialog capDialog, Long invokeId, CAPErrorMessage
capErrorMessage) {
```

```
        // TODO Auto-generated method stub

    }

    @Override
    public void onRejectComponent(CAPDialog capDialog, Long invokeId, Problem problem,
            boolean isLocalOriginated) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onInvokeTimeout(CAPDialog capDialog, Long invokeId) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onCAPMessage(CAPMessage capMessage) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onInitialDPRequest(InitialDPRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onApplyChargingRequest(ApplyChargingRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onEventReportBCSMRequest(EventReportBCSMRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onApplyChargingReportRequest(ApplyChargingReportRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onReleaseCallRequest(ReleaseCallRequest ind) {
        // TODO Auto-generated method stub
```

```
    }

    @Override
    public void onCallInformationRequestRequest(CallInformationRequestRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onCallInformationReportRequest(CallInformationReportRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onAssistRequestInstructionsRequest(AssistRequestInstructionsRequest
ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onEstablishTemporaryConnectionRequest
(EstablishTemporaryConnectionRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDisconnectForwardConnectionRequest
(DisconnectForwardConnectionRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onConnectToResourceRequest(ConnectToResourceRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onResetTimerRequest(ResetTimerRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onFurnishChargingInformationRequest(FurnishChargingInformationRequest
ind) {
        // TODO Auto-generated method stub
```

```
    }

    @Override
    public void onSendChargingInformationRequest(SendChargingInformationRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onSpecializedResourceReportRequest(SpecializedResourceReportRequest
ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onPlayAnnouncementRequest(PlayAnnouncementRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onPromptAndCollectUserInformationRequest
(PromptAndCollectUserInformationRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onPromptAndCollectUserInformationResponse
(PromptAndCollectUserInformationResponse ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onCancelRequest(CancelRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogRequest(CAPDialog capDialog, CAPGprsReferenceNumber
capGprsReferenceNumber) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogAccept(CAPDialog capDialog, CAPGprsReferenceNumber
```

```
capGprsReferenceNumber) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogUserAbort(CAPDialog capDialog, CAPGeneralAbortReason
generalReason,
            CAPUserAbortReason userReason) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogProviderAbort(CAPDialog capDialog, PAbortCauseType abortCause)
{
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogClose(CAPDialog capDialog) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogRelease(CAPDialog capDialog) {
        this.currentCapDialog = null;
        this.cc = null;
    }

    @Override
    public void onDialogNotice(CAPDialog capDialog, CAPNoticeProblemDiagnostic
noticeProblemDiagnostic) {
        // TODO Auto-generated method stub

    }

    private enum Step {
        initialDPSent,
        callAllowed,
        answered,
        disconnected;
    }

    private class CallContent {
        public Step step;
        public Long activityTestInvokeId;

        public CalledPartyNumberCap calledPartyNumber;
```

```
        public CallingPartyNumberCap callingPartyNumber;
        public RequestReportBCSMEventRequest requestReportBCSMEventRequest;
        public DestinationRoutingAddress destinationRoutingAddress;
    }

}
```

```
package org.mobicents.protocols.ss7.cap;

import java.util.ArrayList;

import javax.naming.InitialContext;
import javax.naming.NamingException;

import org.mobicents.protocols.ss7.cap.api.CAPDialog;
import org.mobicents.protocols.ss7.cap.api.CAPDialogListener;
import org.mobicents.protocols.ss7.cap.api.CAPException;
import org.mobicents.protocols.ss7.cap.api.CAPMessage;
import org.mobicents.protocols.ss7.cap.api.CAPParameterFactory;
import org.mobicents.protocols.ss7.cap.api.CAPProvider;
import org.mobicents.protocols.ss7.cap.api.dialog.CAPGeneralAbortReason;
import org.mobicents.protocols.ss7.cap.api.dialog.CAPGprsReferenceNumber;
import org.mobicents.protocols.ss7.cap.api.dialog.CAPNoticeProblemDiagnostic;
import org.mobicents.protocols.ss7.cap.api.dialog.CAPUserAbortReason;
import org.mobicents.protocols.ss7.cap.api.errors.CAPErrorMessage;
import org.mobicents.protocols.ss7.cap.api.isup.CalledPartyNumberCap;
import org.mobicents.protocols.ss7.cap.api.primitives.BCSMEvent;
import org.mobicents.protocols.ss7.cap.api.primitives.EventTypeBCSM;
import org.mobicents.protocols.ss7.cap.api.primitives.MonitorMode;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ActivityTestRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ActivityTestResponse;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ApplyChargingReportReq
uest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ApplyChargingRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.AssistRequestInstructi
onsRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CAPDialogCircuitSwitch
edCall;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CAPServiceCircuitSwitc
hedCallListener;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CallInformationReportR
equest;
```

```
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CallInformationRequest
Request;
import org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.CancelRequest;
import org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ConnectRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ConnectToResourceReque
st;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ContinueRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.DisconnectForwardConne
ctionRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.EstablishTemporaryConn
ectionRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.EventReportBCSMRequest
;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.FurnishChargingInforma
tionRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.InitialDPRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.PlayAnnouncementReques
t;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.PromptAndCollectUserIn
formationRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.PromptAndCollectUserIn
formationResponse;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ReleaseCallRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.RequestReportBCSMEvent
Request;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.ResetTimerRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.SendChargingInformatio
nRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.SpecializedResourceRep
ortRequest;
import
org.mobicents.protocols.ss7.cap.api.service.circuitSwitchedCall.primitive.DestinationR
outingAddress;
import org.mobicents.protocols.ss7.inap.api.primitives.LegID;
import org.mobicents.protocols.ss7.inap.api.primitives.LegType;
```

```
import org.mobicents.protocols.ss7.isup.message.parameter.CalledPartyNumber;
import org.mobicents.protocols.ss7.isup.message.parameter.NAINumber;
import org.mobicents.protocols.ss7.tcap.asn.comp.PAbortCauseType;
import org.mobicents.protocols.ss7.tcap.asn.comp.Problem;

public class CallScfExample implements CAPDialogListener,
CAPServiceCircuitSwitchedCallListener {

    private CAPProvider capProvider;
    private CAPParameterFactory paramFact;
    private CAPDialogCircuitSwitchedCall currentCapDialog;
    private CallContent cc;

    public CallScfExample() throws NamingException {
        InitialContext ctx = new InitialContext();
        try {
            String providerJndiName = "java:/mobicents/ss7/cap";
            this.capProvider = ((CAPProvider) ctx.lookup(providerJndiName));
        } finally {
            ctx.close();
        }
        paramFact = capProvider.getCAPParameterFactory();

        capProvider.addCAPDialogListener(this);
        capProvider.getCAPServiceCircuitSwitchedCall().addCAPServiceListener(this);
    }

    public CAPProvider getCAPProvider() {
        return capProvider;
    }

    public void start() {
        // Make the circuitSwitchedCall service activated
        capProvider.getCAPServiceCircuitSwitchedCall().acivate();

        currentCapDialog = null;
    }

    public void stop() {
        capProvider.getCAPServiceCircuitSwitchedCall().deactivate();
    }

    @Override
    public void onInitialDPRequest(InitialDPRequest ind) {
        this.cc = new CallContent();
        this.cc.idp = ind;
        this.cc.step = Step.initialDPRecieved;

        ind.getCAPDialog().processInvokeWithoutAnswer(ind.getInvokeId());
    }
```

```
    @Override
    public void onEventReportBCSMRequest(EventReportBCSMRequest ind) {
        if (this.cc != null) {
            this.cc.eventList.add(ind);

            switch (ind.getEventTypeBCSM()) {
            case oAnswer:
                this.cc.step = Step.answered;
                break;
            case oDisconnect:
                this.cc.step = Step.disconnected;
                break;
            }
        }

        ind.getCAPDialog().processInvokeWithoutAnswer(ind.getInvokeId());
    }

    @Override
    public void onDialogDelimiter(CAPDialog capDialog) {
        try {
            if (this.cc != null) {
                switch (this.cc.step) {
                case initialDPRecieved:
                    // informing SSF of BCSM events processing
                    ArrayList<BCSMEvent> bcsmEventList = new ArrayList<BCSMEvent>();
                    BCSMEvent ev = this.capProvider.getCAPParameterFactory
().createBCSMEvent(
                            EventTypeBCSM.routeSelectFailure, MonitorMode
.notifyAndContinue,
                            null, null, false);
                    bcsmEventList.add(ev);
                    ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(
                            EventTypeBCSM.oCalledPartyBusy, MonitorMode.interrupted,
null,
                            null, false);
                    bcsmEventList.add(ev);
                    ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(
                            EventTypeBCSM.oNoAnswer, MonitorMode.interrupted, null,
null, false);
                    bcsmEventList.add(ev);
                    ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(
                            EventTypeBCSM.oAnswer, MonitorMode.notifyAndContinue,
null,
                            null, false);
                    bcsmEventList.add(ev);
                    LegID legId = this.capProvider.getINAPParameterFactory
().createLegID(
                            true, LegType.leg1);
                    ev = this.capProvider.getCAPParameterFactory()
                            .createBCSMEvent(EventTypeBCSM.oDisconnect,
```

```
                                    MonitorMode.notifyAndContinue, legId, null,
false);
                    bcsmEventList.add(ev);
                    legId = this.capProvider.getINAPParameterFactory().createLegID
(true,
                            LegType.leg2);
                    ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(
                            EventTypeBCSM.oDisconnect, MonitorMode.interrupted, legId,
                            null, false);
                    bcsmEventList.add(ev);
                    ev = this.capProvider.getCAPParameterFactory().createBCSMEvent(
                            EventTypeBCSM.oAbandon, MonitorMode.notifyAndContinue,
null,
                            null, false);
                    bcsmEventList.add(ev);
                    currentCapDialog.addRequestReportBCSMEventRequest(bcsmEventList,
null);

                    // calculating here a new called party number if it is needed
                    String newNumber = "22123124";
                    if (newNumber != null) {
                        // sending Connect to force routing the call to a new  number
                        ArrayList<CalledPartyNumberCap> calledPartyNumber =
                            new ArrayList<CalledPartyNumberCap>();
                        CalledPartyNumber cpn = this.capProvider
.getISUPParameterFactory()
                            .createCalledPartyNumber();
                        cpn.setAddress("5599999988");
                        cpn.setNatureOfAddresIndicator(NAINumber
._NAI_INTERNATIONAL_NUMBER);
                        cpn.setNumberingPlanIndicator(CalledPartyNumber._NPI_ISDN);
                        cpn.setInternalNetworkNumberIndicator(
                                CalledPartyNumber._INN_ROUTING_ALLOWED);
                        CalledPartyNumberCap cpnc = this.capProvider
.getCAPParameterFactory()
                        .createCalledPartyNumberCap(cpn);
                        calledPartyNumber.add(cpnc);
                        DestinationRoutingAddress destinationRoutingAddress = this
.capProvider
                            .getCAPParameterFactory().createDestinationRoutingAddress(
                                calledPartyNumber);
                        currentCapDialog.addConnectRequest(destinationRoutingAddress,
null,
                                null, null, null, null, null, null, null, null, null,
                                null, null,
                                false, false, false, null, false);
                    } else {
                        // sending Continue to use the original calledPartyAddress
                        currentCapDialog.addContinueRequest();
                    }
```

```
                    currentCapDialog.send();
                    break;

                case disconnected:
                    // the call is terminated - close dialog
                    currentCapDialog.close(false);
                    break;
                }
            }
        } catch (CAPException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    @Override
    public void onDialogTimeout(CAPDialog capDialog) {
        if (currentCapDialog != null && this.cc != null && this.cc.step != Step
.disconnected
                && this.cc.activityTestInvokeId == null) {
            // check the SSF if the call is still alive
            currentCapDialog.keepAlive();
            try {
                this.cc.activityTestInvokeId = currentCapDialog.
addActivityTestRequest();
                currentCapDialog.send();
            } catch (CAPException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
        }
    }

    @Override
    public void onActivityTestResponse(ActivityTestResponse ind) {
        if (currentCapDialog != null && this.cc != null) {
            this.cc.activityTestInvokeId = null;
        }
    }

    @Override
    public void onInvokeTimeout(CAPDialog capDialog, Long invokeId) {
        if (currentCapDialog != null && this.cc != null) {
            if (this.cc.activityTestInvokeId == invokeId) { // activityTest failure
                try {
                    currentCapDialog.close(true);
                } catch (CAPException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
```

```
        }
    }

    @Override
    public void onErrorComponent(CAPDialog capDialog, Long invokeId,
            CAPErrorMessage capErrorMessage) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onRejectComponent(CAPDialog capDialog, Long invokeId, Problem problem,
            boolean isLocalOriginated) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onCAPMessage(CAPMessage capMessage) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onRequestReportBCSMEventRequest(RequestReportBCSMEventRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onApplyChargingRequest(ApplyChargingRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onContinueRequest(ContinueRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onApplyChargingReportRequest(ApplyChargingReportRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onReleaseCallRequest(ReleaseCallRequest ind) {
        // TODO Auto-generated method stub
```

```
    }

    @Override
    public void onConnectRequest(ConnectRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onCallInformationRequestRequest(CallInformationRequestRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onCallInformationReportRequest(CallInformationReportRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onActivityTestRequest(ActivityTestRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onAssistRequestInstructionsRequest(AssistRequestInstructionsRequest
ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onEstablishTemporaryConnectionRequest
(EstablishTemporaryConnectionRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDisconnectForwardConnectionRequest
(DisconnectForwardConnectionRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onConnectToResourceRequest(ConnectToResourceRequest ind) {
        // TODO Auto-generated method stub
```

```
    }

    @Override
    public void onResetTimerRequest(ResetTimerRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onFurnishChargingInformationRequest(FurnishChargingInformationRequest
ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onSendChargingInformationRequest(SendChargingInformationRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onSpecializedResourceReportRequest(SpecializedResourceReportRequest
ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onPlayAnnouncementRequest(PlayAnnouncementRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onPromptAndCollectUserInformationRequest
(PromptAndCollectUserInformationRequest ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onPromptAndCollectUserInformationResponse
(PromptAndCollectUserInformationResponse ind) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onCancelRequest(CancelRequest ind) {
        // TODO Auto-generated method stub
```

```
    }

    @Override
    public void onDialogRequest(CAPDialog capDialog, CAPGprsReferenceNumber
capGprsReferenceNumber) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogAccept(CAPDialog capDialog, CAPGprsReferenceNumber
capGprsReferenceNumber) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogUserAbort(CAPDialog capDialog, CAPGeneralAbortReason
generalReason,
            CAPUserAbortReason userReason) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogProviderAbort(CAPDialog capDialog, PAbortCauseType abortCause)
{
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogClose(CAPDialog capDialog) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onDialogRelease(CAPDialog capDialog) {
        this.currentCapDialog = null;
        this.cc = null;
    }

    @Override
    public void onDialogNotice(CAPDialog capDialog,
            CAPNoticeProblemDiagnostic noticeProblemDiagnostic) {
        // TODO Auto-generated method stub

    }
```

```
    private enum Step {
        initialDPRecieved,
        answered,
        disconnected;
    }

    private class CallContent {
        public Step step;
        public InitialDPRequest idp;
        public ArrayList<EventReportBCSMRequest> eventList = new ArrayList
<EventReportBCSMRequest>();
        public Long activityTestInvokeId;
    }
}
```