

Extending Adempiere:

Dialing out via Asterisk

Dirk Niemeyer
action 42 GmbH

File Bearbeiten Ansicht Gehe Extras Fenster Hilfe

Kunde

Auftrag

Orderline

Kundennotiz

Anruf

Anrufzeit 0900

IS [redacted]

Deliverytour 20110830-202-DI-Brockmann

Geschäftspartner 243283

Telefon 0 [redacted] / 0 [redacted] oder Mobil: [redacted]

Adresseintrag [redacted]

Kontakt Einkauf Kein Kontakt Einkauf für diesen Kunden vorhanden

Intern

Notiz TS Kunde ohne Geräte

Reklamation nichts

Gerätebestellung nichts

OPK A C H T U N G !! 1 OffenePosten!

Status

Anrufstatus nicht bestellt Auftragshinweis nichts

Telefonverbindung starten

Auftrag erfassen

Wochentag 5

Lieferung

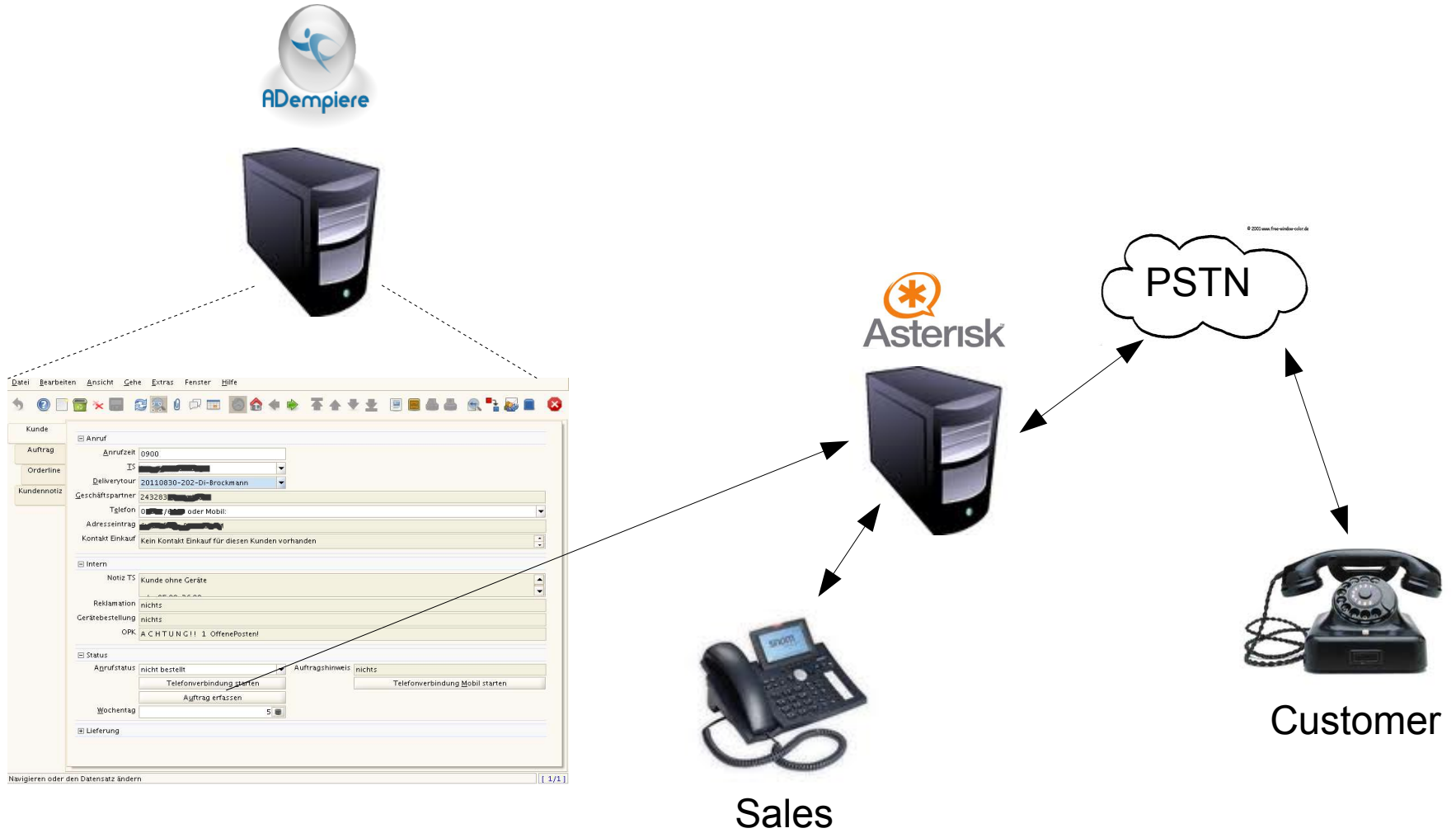
ProduktID 101 Name/Status Weihnachtswurst + Küchentü Menge



Navigieren oder den D

Eingegeben								Abzufragen			
ProdID	Name	Menge	Verfügbar	Rabatt	Preis	Brutto	Vol.	Status	ProdID	Name	0-30
1500	Musterware Korn an Korn	3	0,0	0,0	0,00	0,00	1,50	OK	312	Apfelkuchen 6 Streifen	2
									314	Schlemmer Butterkuchen 6 Streifen	1

Barzahler Kartons 3 gewährter Rabatt 0,00 Netto 0,00 Brutto 0,00 Derz. Volumen 158,50 Max. Volumen 320



- Adempiere
- Asterisk (V1.4)
- Asterisk Manager Interface (AMI)
- Asterisk-Java library (V0.3.1)
- IP-Phone
- Telephony-Card (ISDN,...)

Manager Actions

Output from the [CLI](#) command **show manager commands**:
(For Asterisk 1.4 and greater, use **manager show commands**)

- [AbsoluteTimeout](#): Set Absolute Timeout (**privilege: call,all**)
- [ChangeMonitor](#): Change monitoring filename of a channel (**privilege: call,all**)
- [Command](#): Execute Command (**privilege: command,all**)
- [Events](#): Control Event Flow
- [ExtensionState](#): Check Extension State (**privilege: call,all**)
- [GetVar](#): Gets a Channel Variable
- [Hangup](#): Hangup Channel
- [IAXpeers](#): List IAX Peers
- [ListCommands](#): List available commands
- [Logoff](#): Logoff Manager
- [MailboxCount](#): Check Mailbox Count
- [MailboxStatus](#): Check Mailbox Status
- [Monitor](#): Monitor a channel
- [Originate](#): Originate a call
- [ParkedCalls](#): List parked calls
- [Ping](#): Ping
- [QueueAdd](#): Queue a call
- [QueueRemove](#): Remove a call from a queue
- [Queues](#): Queues
- [QueueStatus](#): Queue status
- [Redirect](#): Redirect (**privilege: call,all**)
- [SetCDRUserField](#): Set CDR User Field
- [SetVar](#): Set Channel Variable
- [SIPpeers](#): List SIP Peers
- [Status](#): Status (**privilege: call,all**)
- [StopMonitor](#): Stop monitoring a channel (**privilege: call,all**)
- [ZapDialOffhook](#): Dial over Zap channel while offhook
- [ZapDNDOff](#): Toggle Zap channel Do Not Disturb status OFF
- [ZapDNDOn](#): Toggle Zap channel Do Not Disturb status ON
- [ZapHangup](#): Hangup Zap Channel
- [ZapTransfer](#): Transfer Zap Channel
- [ZapShowChannels](#): Show Zap Channels

Asterisk Manager API Action Originate

Action: Originate

Parameters:

- **Channel**: Channel on which to originate the call (The same as you specify in the Dial application command)
- **Context**: Context to use on connect (must use Exten & Priority with it)
- **Exten**: Extension to use on connect (must use Context & Priority with it)
- **Priority**: Priority to use on connect (must use Context & Exten with it)
- **Timeout**: Timeout (in milliseconds) for the originating connection to happen(defaults to 30000 milliseconds)
- **CallerID**: CallerID to use for the call
- **Variable**: Channels variables to set (max 32). Variables will be set for both channels (local and connected).
- **Account**: Account code for the call
- **Application**: [Application](#) to use on connect (use Data for parameters)
- **Data**: Data if Application parameter is used
- **Async**: For the origination to be asynchronous (allows multiple calls to be generated without waiting for a response)
- **ActionID**: The request identifier. It allows you to identify the response to this request. You may use a number or a string. Useful when you make several simultaneous requests.

```
; Asterisk Call Management support
```

```
; By default asterisk will listen on localhost only.
```

```
[general]
```

```
enabled = yes
```

```
port = 5038
```

```
bindaddr = 127.0.0.1
```

```
; No access is allowed by default.
```

```
; To set a password, create a file in /etc/asterisk/manager
```

```
; use creative permission games to allow other services to
```

```
; files
```

```
#include "manager.d/*.conf"
```

manager.conf

```
[manager]
```

```
secret = mysecret
```

```
;deny=0.0.0.0/0.0.0.0
```

```
;permit=209.16.236.73/255.255.255.0
```

```
permit=0.0.0.0/0.0.0.0
```

```
;
```

```
; If the device connected via this user accepts input slowly,  
the timeout for writes to it can be increased to keep it  
from being disconnected (value is in milliseconds)
```

```
;
```

```
; writetimeout = 100
```

```
;
```

```
; Authorization for various classes
```

```
;read = system,call,log,verbose,command,agent,user,config
```

```
;write = system,call,log,verbose,command,agent,user,config
```

```
read = system,call,log,verbose,command,agent,user
```

```
write = system,call,log,verbose,command,agent,user
```

```
[3000]
```

```
username=3000
```

```
type=friend
```

```
secret=inspirosecret
```

```
record_out=Adhoc
```

```
record_in=Adhoc
```

```
qualify=no
```

```
port=5060
```

```
nat=never
```

```
mailbox=3000@device
```

```
host=dynamic
```

```
dtmfmode=rfc2833
```

```
context=from-internal
```

```
canreinvite=no
```

```
callerid=device <3000>
```

sip.conf

```
;exten => 6245, hint, SIP/Grandstream1&SIP/Xlite1, Joe Schmoe ; Channel hints fo  
;exten => 6245, 1, Dial(SIP/Grandstream1,20,rt) ; permit transfer  
;exten => 6245, n(dial), Dial(${HINT},20,rtT) ; Use hint as listed  
;exten => 6245, n, Voicemail(6245,u) ; Voicemail (unavailable)  
;exten => 6245, s+1, Hangup ; s+1, same as n  
;exten => 6245, dial+101, Voicemail(6245,b) ; Voicemail (busy)  
;exten => 6361, 1, Dial(IAX2/JaneDoe,,rm) ; ring without time limit  
;exten => 6389, 1, Dial(MGCP/aaln/1@192.168.0.14)  
;exten => 6390, 1, Dial(JINGLE/caller/callee) ; Dial via jingle using labels  
;exten => 6391, 1, Dial(JINGLE/asterisk@digium.com/mogorman@astjab.org) ; Dial v  
;exten => 6394, 1, Dial(Local/6275/n) ; this will dial ${MARK}
```

extensions.conf

User gardenusr GardenUser SuperUser@GardenWorld.HQ [debian{debian-xe-adempiere}]

File Edit View Go Tools Window Help

Client GardenWorld Organization HQ

Description

☒ Active

AstHost 10.1.1.100

AstUser gardenuser AstPass gardenpass

AstChan SIP/1234 AstContext outbound

User Contact

User Roles

User Substitute

Org Assignment

Org Access

User Mail

Queries

LDAP Access

XX_Asterisk_Data

Navigate or Update record 1/1

```
5 import org.asteriskjava.manager.AuthenticationFailedException;
6 import org.asteriskjava.manager.ManagerConnection;
7 import org.asteriskjava.manager.ManagerConnectionFactory;
8 import org.asteriskjava.manager.TimeoutException;
9 import org.asteriskjava.manager.action.OriginateAction;
10 import org.asteriskjava.manager.response.ManagerResponse;
11
12 public class AsteriskManager {
13
14     private ManagerConnection managerConnection;
15
16     public AsteriskManager(String server,String user, String pw) throws IOException
17     {
18         ManagerConnectionFactory factory = new ManagerConnectionFactory(
19             server, user, pw); // e.g. "192.168.1.11", "manager", "mysecret");
20
21         this.managerConnection = factory.createManagerConnection();
22     }
23
24     public String run(String astChannel, String astContext, String astExten, Integer astPriority, Long astTimeout) throws
25         TimeoutException
26     {
27         OriginateAction originateAction;
28         ManagerResponse originateResponse;
29
30         originateAction = new OriginateAction();
31         originateAction.setChannel(astChannel);
32         originateAction.setContext(astContext);
33         originateAction.setExten(astExten);
34         originateAction.setPriority(astPriority);
35         originateAction.setTimeout(astTimeout);
36
37         // connect to Asterisk and log in
38         managerConnection.login();
39
40         // send the originate action and wait for a maximum of 30 seconds for Asterisk
41         // to send a reply
42         originateResponse = managerConnection.sendAction(originateAction, 30000);
43
44         // print out whether the originate succeeded or not
45         String result = originateResponse.getResponse();
46
47         // and finally log off and disconnect
48         managerConnection.logoff();
49
50         return(result);
51     }
52 }
```



```

* Generate Orders from ToCall-List
public class CallAsterisk extends SvrProcess
{
    protected void prepare()
    {
        protected String doIt() throws Exception
        {
            // Search for the Asterisk Settings
            String sql = "SELECT XX_Asterisk_Data_ID FROM XX_Asterisk_Data " +
                " WHERE AD_User_ID=" + adUserId
                + " AND AD_Client_ID=" + Env.getAD_Client_ID(Env.getCtx());
            PreparedStatement pstmt = null;
            try {
                pstmt = DB.prepareStatement(sql, null);
                ResultSet rs = pstmt.executeQuery ();
                if (rs.next()) {
                    xx_asterisk_data_id = rs.getInt(1);
                }
                rs.close ();
                pstmt.close ();
                pstmt = null;
            }
            catch (SQLException e)
            {
                log.log(Level.SEVERE, sql, e);
            }

            log.log(Level.FINE, "CallAsterisk - adUserId=" + adUserId);
            log.log(Level.FINE, "CallAsterisk - astExten=" + astExten);
            X_XX_Asterisk_Data astData = new X_XX_Asterisk_Data(Env.getCtx(), xx_asterisk_data_id, null);
            astChannel = astData.getAstChan();
            astContext = astData.getAstContext();
            log.log(Level.FINE, "CallAsterisk - astChannel=" + astChannel + ", astContext=" + astContext);

            log.log(Level.FINER, "CallAsterisk - Calling AsteriskManager with AstHost=" + astData.getAstHost() + " AstUser=" + astData.getAstUser());
            AsteriskManager asteriskManager = new AsteriskManager(astData.getAstHost(), astData.getAstUser(), astData.getAstPass());

            log.log(Level.FINER, "CallAsterisk - Calling AsteriskManager run");
            String result = asteriskManager.run(astChannel, astContext, astExten, astPriority, astTimeout);

            return "@Ergebnis@ = " + result;
        }
    }
}

```

- Asterisk
 - <http://www.asterisk.org>
- Asterik-Java
 - <http://asterisk-java.org>
- Asterisk Manage API
 - <http://www.voip-info.org/wiki/view/Asterisk+manager+API>
- Adempiere Wiki
 - <http://www.adempiere.com/AsteriskJava>



action 42
Zeitgerechte DV-Lösungen

e-mail: info@action42.de