

# РВХ приложения на сервере Communicate Pro. Разбираем простейший пример — приложение для записи контактного телефона клиента

Основным преимуществом CommunicatePro перед другими системами является один серверный язык для всех модулей — звонков, почты, календарей. То есть мы можем в одной программе принять звонок, проиграть голосовое меню, принять DTMF и сформировать на основе этого DTMF письма, SMS, календарные события.

В качестве примера напишем программу для записи контактного телефона клиента.

Программа должна будет:

1. Проиграть стартовое меню с выбором специалистов по фамилии.
2. В рабочее время у специалиста звонит телефон, в нерабочее включается программа «Мы вам перезвоним».
3. Предлагаем ввести контактный телефон. После ввода отправляем письмо пользователю, которому звонил клиент.

*Для полного понимания материала в этой статье, необходимо иметь представление об основных интерфейсах сервера — WebAdmin, WebUser и Pronto. С ними и с процессом установки бесплатной версии можно ознакомиться в [этой статье](#)*

Для начала немного теории.

## Сигналы

В CommunicatePro это сервер обеспечивающий коммуникации в «реальном времени» (далее по тексту real-time) — IM, SMS, голосовая связь. Сигнал это элементарная задача выполняемая сервером для real-time коммуникаций, они используются всеми модулями обеспечивающими real-time связь — SIP, XMPP, SMPP, XIMSS для передачи информации о начале, разрыве, и изменения состояния соединения.

Для определения адресатов во всех этих протоколах используются AOR (address of registration), которые по формату соответствуют email адресу, поэтому на сервере AOR для сигналов совпадает с email адресом и является просто полным именем учетной записи.

Также как и для писем для сигналов доступен набор настроек, позволяющих изменять адресатов — таблица роутинга, forworders, пользовательские правила для входящих сигналов.

## Запуск программ.Язык CG/PL

CommunicatePro умеет выполнять программы на простом и мощном языке CommuniGate Programming Language (CG/PL). Для каждого модуля и функциональности есть набор встроженных в язык функций. Поддерживается два стиля:

```

//
// A simple CG/PL application
// Basic style
//
entry Main is
  myName = "Jim" + " " + "Smith";
  if length(myName) > 10 then
    myName = Substring(myName,0,8) + "..";
  end if;
end;

/*
 * A simple CG/PL application
 * C style
 */
entry Main {
  myName = "Jim" + " " + "Smith";
  if(myName.length() > 10) {
    myName = myName.substring(0,8) + "..";
  }
}

```

PBX программы так же пишутся на CG/PL и организованы в окружение/среду — набор файлов из CG/PL программ, вспомогательных файлов(например аудио) и папок для поддержки различных языков.

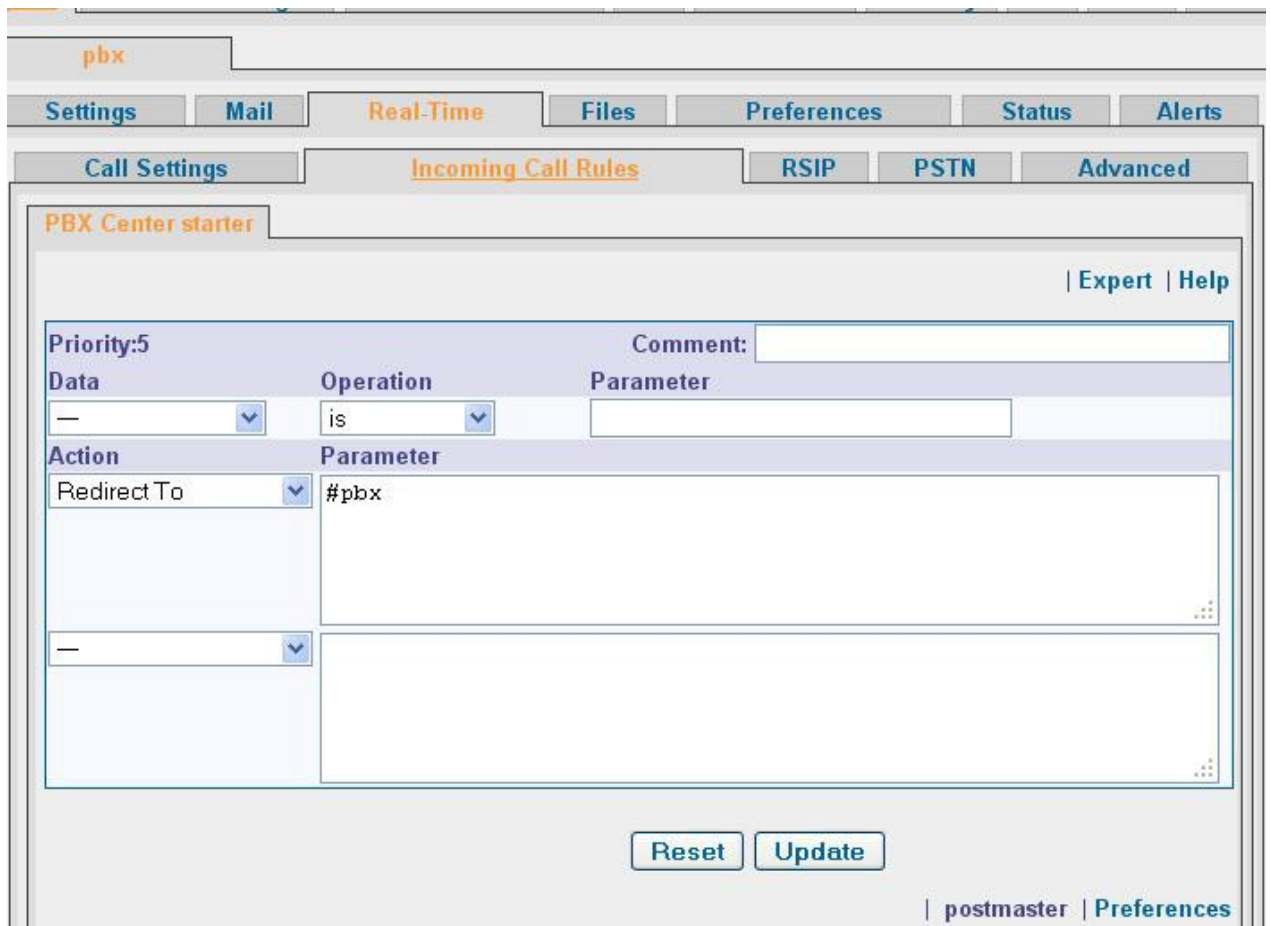
Существует общесерверная PBX среда (страница Users->PBX в WebAdmin интерфейсе) и доменные на каждый домен (Users->Domains->[Domain]->PBX). При этом когда PBX программа, запущенная от имени некоторого аккаунта, запрашивает файл из среды, поиск в начале ведется в доменной среде, а потом в серверной.

Самый простой способ запустить CG/PL программу это отправить сигнал на адрес формата: pbxAppName#account@domain.com, при этом сервер запустит задачу исполняющую программу записанную в файле pbxAppName.sprg (этот файл должен находиться либо в PBX окружении домена domain.com, либо в серверном окружении) от имени аккаунта account@domain.com (сигнал при этом будет передан в только что созданную задачу).

### **Пользователь pbx**

Этот пользователь предназначен для функционирования большинства PBX программ установленных по умолчанию на сервере.

У этого пользователя изначально уже есть одно real-time правило:

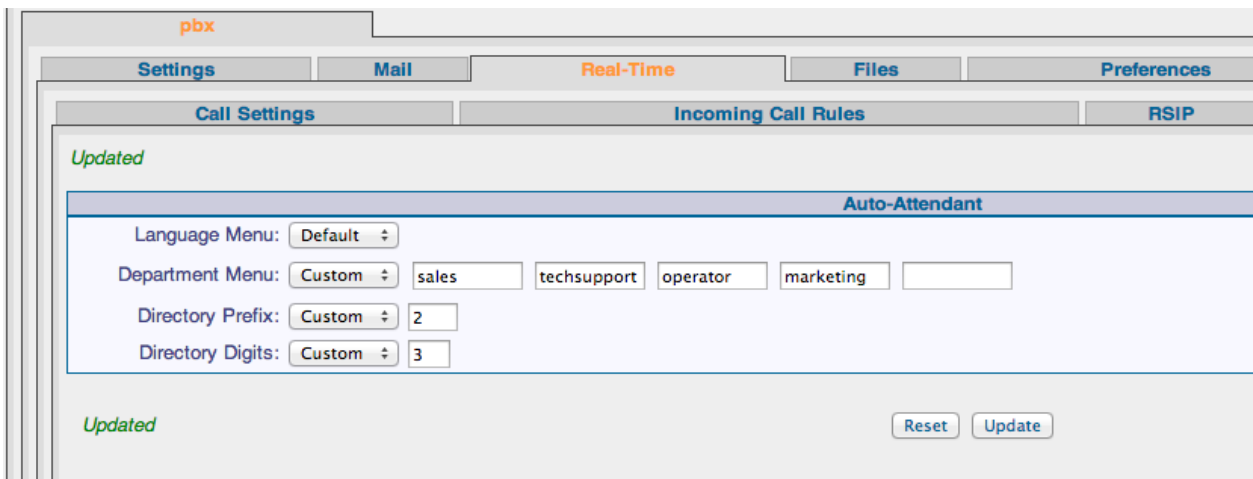


Правило срабатывает на любой сигнал попавший в аккаунт и вызывает программу из файла «pbx.srrg» (PBX центр) от имени аккаунта pbx. Программа pbx.srrg довольно сложная и предоставляет разные функции в зависимости от того как ее вызывать. Поскольку стартовый сигнал передается в программу, мы можем задать разное поведение программы в зависимости от имени/псевдонима пользователя осуществившего вызов, например, при вызове `pbx#pbx@domain.com` запускается авто-секретарь, а при вызове `pbx#conference@domain.com` работает центр конференций.

Кроме этого у пользователя pbx есть алиас 200, который является добавочным номером. Предназначен для звонков с «железных» телефонов.

### Авто-секретарь

Авто-секретарь это программа, реализующая стандартное иерархическое меню с выбором. Поскольку она поставляется в комплекте с сервером, в WebAdmin есть интерфейс для взаимодействия с ней на странице [Account, в данном случае pbx]->Real-Time->Advanced. Посмотрим как эта страница выглядит по-умолчанию:



Алгоритм программы следующий:

1. проигрывает welcome.wav и dialknownextension.wav
2. проигрывает for[name].wav для каждого пункта [name] из department menu. Каждому отделу ставится в соответствии номер по порядку перечисления. При этом цифры из Directory Prefix пропускаются(они зарезервированы под экстеншены) и если в списке есть слово *operator*, то этот пункт всегда проигрывается последним и ему соответствует цифра 0.
3. после выбора отдела звонок отправляется в аккаунт с именем, совпадающим с именем отдела.

Настроим ее таким образом, чтобы при запуске проигрывалось следующее сообщение: *Здравствуйте, это компания «Специалисты», для специалиста Иванова нажмите 1 для специалиста Петрова нажмите 3 или дождитесь ответа секретаря .*

Поскольку сообщения будут на русском, нужно создать отдельную папку для языка в PBX среде домена и переключить язык интерфейсов пользователя pbx, от имени которого запускается программа, на русский. На странице pbx->Preferences в настройке «Language» ставим «Russian». На странице [Домен]->PBX, нажимаем на кнопку «Create Custom Environment», для изменения PBX окружения в домене:



Отобразился список файлов образующих PBX среду в домене. Слово «parent» слева от имени файла означает, что файл берется с общесерверной среды.

Проматываем в нижнюю часть интерфейса и создаем подпапку для русского языка:

parent	<a href="#">voicerecorder.sppr</a>	2515	30 Dec, 12
parent	<a href="#">waitforconference.wav</a>	63K	30 Dec, 12
parent	<a href="#">waitinginqueue.wav</a>	40K	30 Dec, 12
parent	<a href="#">wed.wav</a>	25K	30 Dec, 12
parent	<a href="#">welcome.wav</a>	23K	30 Dec, 12
parent	<a href="#">whenfinished.wav</a>	101K	30 Dec, 12
parent	<a href="#">yen.wav</a>	20K	30 Dec, 12
parent	<a href="#">yesterday.wav</a>	26K	30 Dec, 12
parent	<a href="#">youinvitedtoconference.wav</a>	68K	30 Dec, 12
parent	<a href="#">youinvitedtoconferenceby.wav</a>	72K	30 Dec, 12
parent	<a href="#">youeventstartsnow.wav</a>	55K	30 Dec, 12
parent	<a href="#">youeventwillstartin.wav</a>	48K	30 Dec, 12

Delete Marked Uncache

Created

**Languages**

Create

Записываем 4 файла и загружаем их в только что созданную папку russian.

welcome.wav //Здравствуйте  
 dialknownextension.wav //вы позвонили в компанию Специалисты наберите короткий номер абонента если он вам известен.  
 forivanov.wav // для специалиста Иванова  
 press.wav// нажмите  
 0.wav, 1.wav,3.wav // цифры  
 forpetrov.wav // для специалиста Петрова  
 tospeakoperator.wav // или дождитесь ответа секретаря  
 tryingextension.wav // набираю абонента  
 failure.wav // ошибка

Чтобы файлы for\*\*\*.wav проигрывались нужно изменить настройки [pbx]->Real-Time->Advanced на эти:

pbx

Settings Mail **Real-Time** Files Preferences Status Alerts

Call Settings Incoming Call Rules RSIP PSTN **Advanced**

Updated | Expert | Hel

**Auto-Attendant**

Language Menu: Default +

Department Menu: Custom + ivanov petrov operator

Directory Prefix: Custom + 2

Directory Digits: Custom + 3

Updated Reset Update

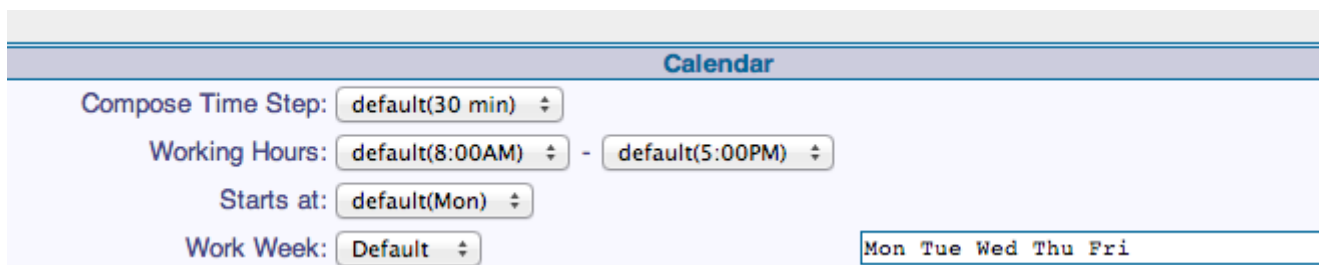
| postmaster | Preference

Также нужно добавить в домен учетные записи с именами ivanov, petrov и operator. Для проверки открываем любую учетную запись в Pronto и набираем в Dialer модуле 200 или pbx.

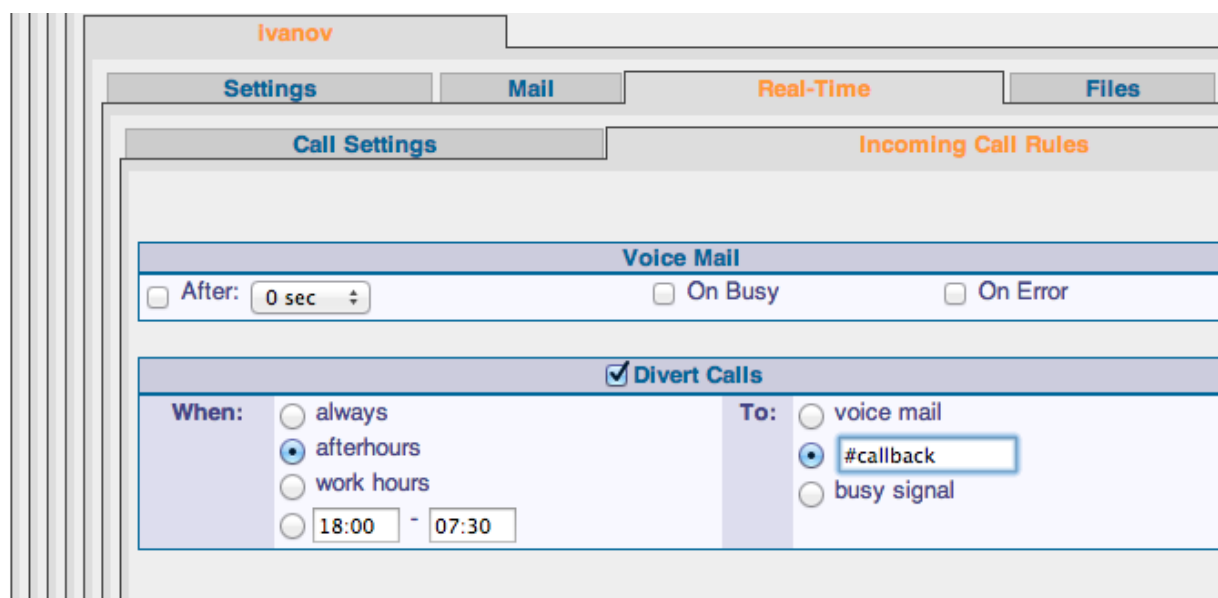
**Устанавливаем приемные часы для специалистов**

Мы хотим, чтобы во вне рабочее для специалиста время звонки отправлялись в PBX программу с определенным именем, например callback (которая спрашивает у пользователя контактный телефон и обещает перезвонить).

Для этого в всех аккаунтах ivanov, petrov и operator заходим на страницу [account]->Preferences и устанавливаем настройку working hours в группе настроек Calendars.



Дальше нужно зайти в раздел [account]->Real-time->Incoming call Rules и включить Divert calls ->when: afterhours, To: #callback:



Поскольку во время настройки нам удобнее, чтобы звонки перенаправлялись всегда, установим временно Divert Call -> When:always, To: #callback.

Если после этих настроек позвонить пользователю, то Pronto выдаст ошибку «Failed to load PBX application code» (при звонке через авто-секретаря будет проигран failure.wav).

Она связана с тем, что программы callback еще не существует в PBX окружении домена, ей мы и займемся в следующем пункте.

### Пишем CG/PL программу

Программа выполняет 3 набора действий:

1. Проигрывает вступление:  
 «На данный момент специалист не может поднять трубку. (callbackbusy.wav)  
 Если вы хотите, мы можем вам перезвонить в ближайшее время.(callbacklater.wav)  
 Наберите свой контактный телефон и нажмите решетку, (callbacknumber.wav)  
 в случае ошибки нажмите звездочку и наберите номер заново»  
 (callbackclearnumber.wav)
2. Считывает телефон.
3. Формируем email и завершаем разговор.  
 «Ваш телефон записан, наши сотрудники свяжутся с вами в ближайшее время»  
 (создаем файл callbacktrailer.wav)  
 Если письмо отправить не получилось предупреждаем об этом и останавливаемся:  
 «Ошибка, сервис не доступен, перезвоните позже» (failure.wav +  
 callbacknotavailable.wav)

Программу можно набирать в любом текстовом редакторе, файл сохраняет в Plain Text формате с именем «callback.sprg».

Текст программы не сложный, порядка 20-ти строчек, дополнительные пояснения в комментариях:

```
// "entry Main" является точкой входа для любой CG/PL программы
entry Main is

// принимаем входящий звонок, если не получилось останавливаемся
if AcceptCall() != null then stop; end if;

// проигрываем .wav файлы
PlayFile("CallbackBusy");
PlayFile("CallbackLater");
PlayFile("CallbackNumber");
PlayFile("CallbackClearNumber");

// иницилируем переменную в которой будет храниться номер
accumulator = "";
loop
// в цикле считываем из буфера по одной DTMF цифре с таймаутом 20 секунд
input = ReadInput(20);

// если решетка или не DTMF символ (функция ReadInput возвращает не только
DTMF) заканчиваем запись номера
exitif not IsString(input) or input == "#";

// если пользователь ошибся в процессе набора, можно начать заново нажав "*"
if input == "*" then accumulator = "";PlayFile("CallbackClearNumber");
end if;
    accumulator = accumulator + input; // добавляем очередной символ
end loop;

//если в итоге номера нет - повесили трубку или просто не стали набирать,
выходим:
if accumulator == "" then stop; end if;
// пытаемся отправить письмо если не получилось просим перезвонить
if SendEmail("CallbackMailer@"+ MyDomain(),"Вам надо перезвонить на номер:"+
accumulator,MyEmail(),null,"") != null
then
    PlayFile("Failure");
    PlayFile("CallbackNotAvailable");
```

```
else
    PlayFile("CallbackTrailer");
end if;

end;
```

По аналогии с аккаунтом pbx, на страницах [ivanov и petrov]->Preferences, «Language» ставим «Russian». Программа callback проигрывается от имени этих аккаунтов и аудио файлы записаны на русском.

Для завершения установки callback.sppr загружаем в [Домен]->PBX, а все .wav файлы — в языковую подпапку «russian» на этой же странице. Большое количество файлов удобно загружать одним .tar архивом (сервер сам его распакует и добавит файлы в окружение)

### **Краткие итоги**

Программы на CG/PL могут быть очень сложными, но в этой статье мы намеренно продемонстрировали только базовую функциональность (прием звонков и DMTF + отправка текстовой почты).

Нашу версию .wav файлов используемых в статье можно найти на [Dropbox](#), так же как и итоговый [callback.sppr](#).

Мы планируем в дальнейшем написать еще несколько статей с более функциональными примерами, так что, если у кого-то есть интересный случай, когда приходилось интегрировать PBX с почтой, календарями или IM, присылайте в личку, постараемся продемонстрировать, как выполнить эту задачу с помощью CGPro.