

Installation von OTRS auf Win32

Table of Contents

<u>Installation von OTRS auf Win32</u>	1
<u>1. Installationen</u>	3
<u>1.1. PERL installieren</u>	3
<u>1.1.1. Zusätzliche PERL Packages installieren</u>	3
<u>1.2. Webserver auswählen und installieren</u>	3
<u>1.2.1. Apache installieren</u>	3
<u>1.2.2. Sambar Server installieren</u>	3
<u>1.3. MySQL installieren</u>	4
<u>1.4. OTRS installieren</u>	4
<u>2. Konfiguration</u>	5
<u>2.1. Perl konfigurieren</u>	5
<u>2.2. Webserver konfigurieren</u>	6
<u>2.2.1. Apache konfigurieren</u>	6
<u>2.2.2. Sambar konfigurieren</u>	7
<u>2.3. MySQL konfigurieren</u>	9
<u>2.4. OTRS konfigurieren</u>	10
<u>2.4.1. CONFIG.PM</u>	10
<u>3. Erster Start</u>	12
<u>3.1. System-eMail-Adressen & POP3</u>	12
<u>3.2. CRON-Job(s) einrichten</u>	12
<u>4. Geschäft!</u>	14

Installation von OTRS auf Win32

Inhaltsverzeichnis

1. [Installationen](#)
 - 1.1. [PERL installieren](#)
 - 1.1.1. [Zusätzliche PERL Packages installieren](#)
 - 1.2. [Webserver auswählen und installieren](#)
 - 1.2.1. [Apache installieren](#)
 - 1.2.2. [Sambar Server installieren](#)
 - 1.3. [MySQL installieren](#)
 - 1.4. [OTRS installieren](#)
2. [Konfiguration](#)
 - 2.1. [Perl konfigurieren](#)
 - 2.2. [Webserver konfigurieren](#)
 - 2.2.1. [Apache konfigurieren](#)
 - 2.2.2. [Sambar konfigurieren](#)
 - 2.2.2.1. [./otrs/bin absichern](#)
 - 2.2.2.2. [PHP and Perl beschleunigen](#)
 - 2.3. [MySQL konfigurieren](#)
 - 2.4. [OTRS konfigurieren](#)
 - 2.4.1. [CONFIG.PM](#)
3. [Erster Start](#)
 - 3.1. [System-eMail-Adressen & POP3](#)
 - 3.2. [CRON-Job\(s\) einrichten](#)
4. [Geschafft!](#)

Diese Anleitung installiert das [Open Ticket Request System OTRS](#) auf *Win32-Systemen*. Getestet ist sie nur auf Windows 2000, funktionieren sollte es aber auch auf Windows 98, 98SE, ME, NT4 und XP.



Kommando Boxen

In diesem Dokument geben wir öfter Kommandos direkt ins System ein. Wir tun dies in einer sogenannten *Kommando Box*. Andere Namen sind *Kommandozeile*, *Eingabeaufforderung*, (*DOS*) *Prompt* oder *Shell*. So ruft man eine Kommando Box auf Windows auf:

Klicke auf *Start*, dann *Ausführen* und gib `cmd` in dem Dialogfeld ein, das aufklappt. Drücke **Return** auf der Tastatur oder klicke *OK*. Eine (gewöhnlich schwarze) Box erscheint und zeigt so etwas wie dies hier:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>
```

Du kannst nun Befehle eingeben:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

C:\>cd /d d:\otrs\bin\cgi-bin

D:\otrs\bin\cgi-bin>dir
Datenträger in Laufwerk D: ist Aurelia
Datenträgernummer: C8F1-F408
```

```
Verzeichnis von D:\otrs\bin\cgi-bin
20.01.03  00:20          <DIR>          .
20.01.03  00:20          <DIR>          ..
08.06.02  23:37                10 .cvsignore
18.01.03  02:14                20 .htaccess
09.01.03  16:06           20.734 customer.pl
09.01.03  16:06           19.014 index.pl
03.01.03  17:17            4.203 installer.pl
03.01.03  17:17            5.450 pic.pl
          6 Datei(en)           49.431 Bytes
          2 Verzeichnis(se),   9.825.726.464 Bytes frei
D:\otrs\bin\cgi-bin>
```

Damit wechselst Du zum OTRS–Installationsverzeichnis und zeigst seinen Inhalt an. Der Schalter */d* weist Windows an, auch direkt das Laufwerk zu wechseln, wenn nötig.

Bitte beachte, dass dies nur eine *beispielhafte Ausgabe* von meinem PC ist – wie man sieht, eine deutsche Windows–2000 Installation. Bei Dir wird es wahrscheinlich anders aussehen.

Du kannst das grundlegende Verhalten und Aussehen der Kommando Box ändern, indem Du in die obere linke Ecke des Fensters klickst und **Eigenschaften** auswählst.

Es hilft sehr, wenn Du Dich mit Kommando Boxen ein wenig beschäftigst und generell mit dem Arbeiten auf der Kommandozeile auskennst. [Suche bei Google zur Hilfe mit Kommandozeilen.](#)

1. Installationen

Wir benötigen drei Hauptkomponenten im System, einen Webserver ([Sambar Server](#) oder/und [Apache](#)), eine Datenbank ([MySQL](#)) und die Programmiersprache [Perl](#).

Ich gehe davon aus, dass die gesamte Installation direkt auf dem Laufwerk *D:* vorgenommen wird, sie funktioniert aber auch in jedem anderen Verzeichnis. Wenn Du also lieber in *E:*, *C:* oder *C:\Programme* installieren möchtest – nur zu, dann musst Du die Pfade unten entsprechend anpassen, d.h. jedes *D:* oder *d:* durch bspw. *C:\Programme* ersetzen. Es ist *keine* gute Idee, Leerzeichen im Installationspfad zu haben, *C:\Program Files\otrs* ist also unglücklich gewählt.

1.1. PERL installieren

Lade *Perl* in der *Version 5.6.1 build 633* von <http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl> herunter. Ich empfehle die MSI-Version.

Installiere Perl mit den Standardeinstellungen nach *D:\Perl*.

1.1.1. Zusätzliche PERL Packages installieren

Lade die OTRS-Win32-Perl-Packages [in einer Datei](#) herunter (kommt per FTP von *ftp.otrs.org*) und entpacke sie nach *D:\Perl*, behalte die Verzeichnisstruktur bei.


1.2. Webserver auswählen und installieren

OTRS benötigt mindestens einen Webserver, um zu laufen ;) Du kannst Dir den Webserver Deines Vertrauens aussuchen, jeder Perl-fähige sollte es tun. Getestet gabe ich OTRS auf [Sambar 5.2 & höher](#) und auf [Apache](#), Versionen [1.3](#) und [2.0](#). `mod_perl` gibt es nur für Apache, es läuft momentan aber nur unter Apache 1.3 für OTRS, also als `mod_perl_1.0`.

1.2.1. Apache installieren

Lade [Apache](#) ([1.3.27](#) und/oder [2.0.43](#)) und installiere ihn. Ich empfehle, Du wählst **D:\Apache** als Basisverzeichnis für beide Versionen. Diese Verzeichnisstruktur ergibt sich:

```
D:\Apache\          --- Heimat von beiden
D:\Apache\Apache\  --- Heimat des Apache 1.3
D:\Apache\Apache2\ --- Heimat des Apache 2.0
```

 Du kannst durchaus beide Versionen, 1.3 und 2.0, auf der selben Maschine installieren und auch parallel laufen lassen, solange Du dafür sorgst, dass beide auf einem unterschiedlichen Port laufen. Z.B. läuft Apache 1.3 auf Port 80, Apache 2.0 auf Port 81, und zusätzlich Sambar auf Port 82. Natürlich reicht ein einziger Webserver auf Port 80, der Rest ist für Geeks ;)

1.2.2. Sambar Server installieren

Lade den Sambar Server von <http://sambar.robertkehl.de> herunter, die Homepage ist <http://www.sambar.com>. Ich empfehle die Version 5.2 Production, es wird auch mit den nachfolgenden Versionen laufen. Benutze keine Version vor der 5.2 Production!

Beachte die Sicherheitshinweise auf <http://www.sambar.com/syshelp/security.htm> und installiere dann Deinen Sambar, wo Du möchtest. Sinnvoll ist *D:\Sambar*.

Auf Windows NT4, 2000 oder XP kannst und solltest Du den Sambar dann als Dienst installieren, wenn die Installationsroutine dies nicht schon angeboten und erledigt hat. Dafür öffnest Du eine Kommando Box und gibst Folgendes ein:

```
C:\>cd /d d:\sambar\bin
D:\sambar\bin>ntserver.exe -i -s Sambar
```

Starte dann den Sambar über die Windows-interne Dienstverwaltung oder auf der Kommandozeile mit

```
D:\sambar\bin>net start Sambar
```

Wenn Du auf Win9x/ME bist, starte ihn über *Start -> Programme -> Sambar Server -> Start Sambar Server*. Auf Win9x/ME gibt es keine Dienste.

Rufe nun den Sambar über <http://localhost> auf. Die Dokumentation findest Du auf <http://localhost/syshelp/index.htm>. Über <http://localhost/session/adminlogin?RCpage=/sysadmin/index.stm> erreichst Du die Administrationsoberfläche.



Sichere *jetzt* unbedingt Deinen Sambar gegen unerlaubte Zugriffe von aussen, indem Du im *User Management* ein gutes Kennwort für den Benutzer *admin* vergibst, per default ist das Passwort für *admin* leer. Gute Kennworte sind länger als acht Zeichen und bestehen aus sinnlosen Kombinationen von groß- und kleingeschriebenen Buchstaben, die mit Zahlen und/oder Sonderzeichen vermischt sind, wie dieses hier: **sie.&Fh9w_iG**

1.3. MySQL installieren

Lade MySQL von <http://www.mysql.com/downloads/mysql-3.23.html> herunter und installiere es in *D:\mysql*, unter Win 2000/XP als Dienst. Starte es.

Für die komfortable Konfiguration der Datenbank empfehle ich, auch phpMyAdmin von <http://www.phpmyadmin.net> zu installieren, z.B: in *D:\mysql\phpMyAdmin*. Dies erfordert natürlich [PHP](#), was aber eh' eine sehr gute Idee ist.

Wir ändern das Passwort für root später.

1.4. OTRS installieren

Last but not least – the beast!

Lade OTRS als tarball (.tar.gz) herunter: <http://otrs.org/download> und entpacke es unter Beibehaltung der Verzeichnisstruktur nach *D:* – ein Unterverzeichnis 'otrs' wird auf *D:* automatisch erstellt.

2. Konfiguration

2.1. Perl konfigurieren

Nun patchen wir unsere Perl-Installation hoch, da es in der Grundkonfiguration noch nicht alle sog. "Packages", also Programmmodule mitbringt.

Öffne eine "Eingabeaufforderung" und gib' folgende Kommandos:

```
C:\>cd /d D:\Perl\packages
D:\Perl\packages>install.bat
[...returned output snipped...]
```

Die Installationsroutine platziert die Datei `mod_perl` in `\Apache\Apache\modules`. Wenn Du Apache woanders installiert hast, musst Du momentan noch den Pfad in `install.bat` anpassen oder `mod_perl` so von Hand in das Verzeichnis `modules` in Deiner Apache-Installation kopieren.

Dein Perl sollte nun so (oder besser) aussehen:

```
PPM> query
Archive-Tar      [0.072      ]
Compress-Zlib    [1.16       ]
DBD-Mysql        [1.2200     ]
DBI              [1.27       ]
Digest           [1          ]
Digest-HMAC      [1.01       ]
Digest-MD2       [2          ]
Digest-MD4       [1.1        ]
Digest-MD5       [2.20       ]
Digest-SHA1      [2.01       ]
File-CounterFile [0.12       ]
Font-AFM         [1.18       ]
GD               [1.27.2     ]
GDGraph          [1.32       ]
GDTextUtil       [0.80       ]
HTML-Parser      [3.26       ]
HTML-Tagset      [3.03       ]
HTML-Tree        [3.11       ]
IO-stringy       [2.108      ]
MD5              [2.02       ]
MIME-Base64      [2.12       ]
MIME-tools       [5.411a     ]
MailTools        [1.58       ]
Net-DNS          [0.33       ]
PPM              [2.1.6      ]
SOAP-Lite        [0.55       ]
Storable         [1.0.12     ]
Test-Simple      [0.47       ]
Tk               [800.023    ]
URI              [1.19       ]
XML-Parser       [2.27       ]
XML-Simple       [1.06       ]
libwin32         [0.19.1     ]
mod_perl         [1.27_01-dev]
```

```
PPM> quit
Quit!
```

Wenn Du möchtest, kannst Du die Dateien in `D:\Perl\packages` löschen, musst Du aber nicht.

Damit ist Perl bedient.

2.2. Webserver konfigurieren

Wieder unterscheiden wir zwischen Apache und Sambar. Denk' dran, Du kannst beide parallel laufen lassen, wenn Du möchtest.

2.2.1. Apache konfigurieren

Im folgenden verwende ich nur 'Apache', der Prozess ist für beide Versionen derselbe.

Öffne Apache's Konfigurationsdatei `httpd.conf`, liegt in `d:\Apache\Apache\conf\`. Füge diese Zeilen am Ende hinzu und pass' sie an Deine Anforderungen an:

```
# Die nächsten beiden für Apache2 auskommentieren!
LoadModule perl_module modules/mod_perl.so
AddModule mod_perl.c

### added for OTRS (http://otrs.org/)

<IfModule mod_alias.c>
    Alias /otrs/ "d:/otrs/bin/cgi-bin/"
    PerlModule Apache::Registry
</IfModule>

<Location /otrs>
    Options ExecCGI
    Order deny,allow
    Deny from all
    allow from 127.0.0.1

    SetHandler cgi-script
    ScriptInterpreterSource registry

    <IfModule mod_perl.c>
        SetHandler perl-script
        PerlHandler Apache::Registry
        PerlSendHeader On
    </IfModule>

</Location>

# load all otrs modules
PerlRequire d:/otrs/scripts/apache-perl-startup.pl

# MaxRequestsPerChild (so no apache child will be to big!)
#MaxRequestsPerChild 400
# depends on your RAM
```

Nimm' Dir ein wenig Zeit und konfiguriere den Rest der `httpd.conf` auch noch. Öffne `d:\otrs\scripts\apache-perl-startup.pl` und ändere diese Zeilen:

```
1:
#! D:/Perl/bin/perl
10,11:
use lib "d:/otrs/";
use lib "d:/otrs/Kernel/cpan-lib";
38,39:
#use Kernel::System::AuthSession::IPC;
```

```
use Kernel::System::AuthSession::DB;
46,47:
#use Kernel::System::Log::SysLog;
use Kernel::System::Log::File;
```

Über mod_perl_2.0 auf Apache2

Momentan läuft OTRS nicht rund unter Win32, wenn mod_perl_2.0 benutzt wird. Uns ist aber sehr wohl bewußt, dass mod_perl_2.0 einen deutlichen Geschwindigkeitszusatz bedeutet und sind dran, es auf Apache2 ans Laufen zu bekommen. Bitte sei ein wenig geduldig oder versuch's selber. [Schreib' uns](#), wenn Du's geschafft hast :)

Das war's für Apache. Starte (oder Restart) Deine Installationen über den Apache Monitor, den Du im Systray unten rechts findest. Manchmal ist es auch eine gute Idee, den gesamten Rechner neu zu starten.

2.2.2. Sambar konfigurieren

Wir stellen nun den Webserver *Sambar* ein. Erstelle in der Datei *D:\Sambar\config\mappings.ini* die folgenden Einträge:

```
[aliases]
/mysql = d:/mysql/Docs
/phpMyAdmin = d:/mysql/phpMyAdmin

[cgi-aliases]
/otrs/ = /otrs/bin/cgi-bin/
/obin/ = /otrs/bin/
```

Dann in *D:\Sambar\config\config.ini* folgende Werte einstellen:

```
[common]
Trace Level = INFO
Trace Performance = true
Network Trace Level = None
Dynamic IP Test = true
License =
Licensee =
DNS Primary = IPadresse.DNS1.beideinem.Provider
DNS Secondary = IPadresse.DNS2.beideinem.Provider

[server]
System Administrator = admin
System Administrator IP = 127.0.0.1
Act As HTTP Server = true
Act As HTTPS Server = false
Act As DNS Server = false
Act As DHCP Server = false
Act As Mail Server = false
Act As TFTP Server = false
Act As FTP Server = false
Act As FTPS Server = false
Act As FTP Proxy = false
Act As NNTP Proxy = false
Act As SMTP Proxy = false
Act As POP3 Proxy = false
Act As IMAP4 Proxy = false
Act As Bridge Proxy = false
Act As SOCKS Proxy = false
Act As Telnet Server = false
Trace FTP = true
```

```
Trace TFTP = true
Trace Bridge = true
Trace DHCP = true
Trace DNS = true
SMTP Server = smtp.deinprovider.de
DOT-File Security = true

[http]
Act As HTTP Proxy = false
Trace Proxy Usage = true
Trace Requests = true
Log Format = performance
Don't Log IPs = 127.0.0.1
Default Page = index.pl index.php index.stm index.shtml index.shtm index.sht index.htm index.html
Perl Executable = D:/Perl/bin/perl.exe
CGI Extensions = *.pl
Enforce .htaccess = true

[events]
Monitor Invalid Logins = true
Monitor Invalid Requests = true
Share User Logins = true
Trace User Logins = true
```

Starte den Sambar neu, um die Änderungen wirksam werden zu lassen.



Einschub:

Der *Mail*- wie der *DNS-Server des Sambar* sind momentan noch nicht notwendig, da OTRS zurzeit (noch) kein SMTP- und DNS-Modul hat. Um sie dennoch bereits jetzt (ausserhalb OTRS) nutzen zu können, benötigst Du eine *Pro-Version* des Sambar, siehe <http://sambar.com/syshelp/pro.htm>. Die http-Funktionen sind in der Freeware-Version, die automatisch aktiv ist, nicht eingeschränkt. Wenn License und Licensee "demo" beinhalten, ist die Demo-Version der Pro-Version aktiviert, die nur zwischen 9:00 und 17:00 Uhr funktioniert:

```
[common]
License = demo
Licensee = demo
```

2.2.2.1. /otrs/bin absichern

Wir *müssen* nun den Alias /obin/, der auf d:\otrs\bin zeigt, grundlegend gegen unbefugten Zugriff absichern, denn in d:\otrs\bin ist genau der Code enthalten, um das System prima zu hacken. Ich persönlich möchte diese "Tools" nicht so gerne öffentlich zugänglich machen.

Erstelle im Windows Explorer die Datei *d:\otrs\bin\htaccess.txt* mit dieser Zeile als Inhalt:

```
require group root
```

Öffne eine Kommandozeile und gib dies ein:

```
C:\>cd /d d:\otrs\bin
D:\otrs\bin>ren htaccess.txt .htaccess
```

Nun haben nur noch User der Gruppe "root" und das System selber (für CRON-Jobs z.B.) die Berechtigung, Skripte in "d:\otrs\bin" auszuführen.

2.2.2.2. PHP and Perl beschleunigen

Wir bringen Sambar nun noch bei, die Dinge (nicht nur für OTRS) ein bisschen zu beschleunigen, indem wir Perl und PHP als ISAPI-DLLs einbinden.

In *D:\Sambar\config\config.ini* leerst Du den Eintrag für *CGI Extensions* wie hier gezeigt:

```
[common]
CGI_Extensions =
```

In *D:\Sambar\config\mappings.ini* setzt Du diese Werte:

```
[isapi]
*.pl = d:\perl\bin\perlis.dll
*.cgi = d:\perl\bin\perlis.dll
*.php = d:\php\php4isapi.dll
*.php4 = d:\php\php4isapi.dll
*.php3 = d:\php\php4isapi.dll
```

Nun kopierst Du noch die Datei *D:\php\sapi\php4isapi.dll* nach *d:\php*. Starte Sambar neu um alle Änderungen wirksam werden zu lassen.

Das war's für Sambar.

2.3. MySQL konfigurieren

Sichere auch Dein MySQL gegen unerlaubte Zugriffe von aussen, indem Du die Benutzer *@%*, *root@%* und *root@localhost* mit guten Kennworten versiehst. Dies kannst Du über *phpMyAdmin* tun.

Machen wir uns daran, die Datenbank zu füllen. Das geht, wie unten gezeigt, per *mysql* auf der Eingabeaufforderung, die Ausgabe der Befehle ist nicht gezeigt.

```
C:\>cd /d d:\mysql
D:\mysql>mysql -u root -p password -e 'create database otrs'
D:\mysql>mysql -u root -p password otrs < d:\otrs\scripts\database\otrs-schema.mysql.sql
D:\mysql>mysql -u root -p password otrs < d:\otrs\scripts\database\initial_insert.sql
D:\mysql>mysql -u root -p -e 'GRANT ALL PRIVILEGES ON otrs.* TO otrs@localhost IDENTIFIED BY "son'
D:\mysql>mysqladmin -u root -p password reload
```

Ich empfehle allerdings, statt der Kommandozeile das *phpMyAdmin* unter <http://localhost/phpMyAdmin> zu nutzen. Die Datenbank *otrs* legst Du direkt im Start-Screen an. Dann klickst Du auf das Register "SQL", und

gibst bei "oder Datei:" *d:\otrs\scripts\database\otrs-schema.mysql.sql* ein. Alternativ kannst Du auch die "Durchsuchen"-Schaltfläche nutzen. Klicke auf OK. Wenn alles glatt gelaufen ist, gebe *d:\otrs\scripts\database\initial_insert.sql* ein und klicke OK.

Führe nun noch den folgenden SQL-Befehl in "phpMyAdmin" aus, wieder auf dem Register "SQL", diesmal aber in der großen Text-Box:

```
GRANT ALL PRIVILEGES ON otrs.* TO otrs@localhost IDENTIFIED BY "some-pass" WITH GRANT OPTION;
```

Damit ist MySQL mit der Datenbank otrs, den notwendigen Tabellen und dem User otrs@localhost versorgt.

2.4. OTRS konfigurieren

2.4.1. CONFIG.PM

Nun bringen wir OTRS noch etwas über uns bei. Erzeuge auf der Kommandozeile eine Config-Datei:

```
C:\>cd /d d:\otrs\Kernel
D:\otrs\Kernel>copy Config.pm.dist Config.pm
```

In dieser Datei (*d:\otrs\Kernel\Config.pm*) stellst Du, neben Deinen Angaben zu FQDN, Datenbank etc. folgende Werte ein:

```
# -----#
# Sendmail
# -----#
$Self->{'SendmailModule'} = 'Kernel::System::Email::SMTP';
$Self->{'SendmailModule::Host'} = 'smtp-server.deines.providers';
$Self->{'SendmailModule::AuthUser'} = 'smtp-username';
$Self->{'SendmailModule::AuthPassword'} = 'smtp-passwort';
# -----#
# directories
# -----#
$Self->{Home} = 'd:/otrs';
$Self->{CounterLog} = '<OTRS_CONFIG_Home>/var/log/TicketCounter.log';
$Self->{ArticleDir} = '<OTRS_CONFIG_Home>/var/article';
$Self->{StatsPicDir} = '<OTRS_CONFIG_Home>/var/pics/stats';
$Self->{TemplateDir} = '<OTRS_CONFIG_Home>/Kernel/Output';
$Self->{TempDir} = '<OTRS_CONFIG_Home>/var/tmp';
# -----#
# LogModule
# -----#
$Self->{LogModule} = 'Kernel::System::Log::File';
# -----#
# own config settings
# config settings taken from Kernel/Config/Defaults.pm
# -----#
$Self->{SessionUseCookie} = 0;
$Self->{SessionModule} = 'Kernel::System::AuthSession::DB';
```



Eine Bemerkung zum FQDN

Das ist der sog. *Full Qualified Domain Name* Deines Rechners, wie die dazugehörige IP-Adresse ebenfalls einzigartig im Internet. Wenn Du eine Wählleitung für den Zugriff auf das Internet benutzt, wird der allerdings bei jedem Einwählvorgang ein anderer sein, ebenso wie die IP-Adresse Deines Rechners auch. Damit kommt OTRS zwar zurecht, aber schön ist es nicht, wenn exim sich "falsch meldet".

Zur Lösung dieses Problemchens gibt es Dienste im Internet wie z.B. <http://www.dyndns.org>, das sind *Dynamic DNS Dienste*. Mit einem Client wie *DirectUpdate* (<http://www.directupdate.net>) kannst Du dann nach jedem Einwählvorgang automatisch Deine momentane IP-Adresse an *dyndns.org* übermitteln, und bist so ständig und auf Dauer unter einem bestimmten Namen erreichbar, z.B. <http://mein-oller-pc.home-ip.net>. Natürlich kannst Du mit solchen Diensten auch Domains wie www.meineolledomain.de auf Deinen Heimatrechner umleiten lassen. Wenn Du Fragen zu diesen Diensten hast, [frag' mich ruhig](#), nun aber zurück zu OTRS :)

3. Erster Start

Nun ist es an der Zeit, Dein OTRS das erste Mal aufzurufen. Ein vorheriger Reboot des Rechners ist angeraten. Rufe in Deinem favorisierten Browser diese Adresse auf: <http://localhost/otrs/index.pl>.

Als Benutzername gibst Du "root@localhost" ein, das Passwort ist "root", jeweils ohne Anführungszeichen.

 Ändere *jetzt* auf der Seite <http://localhost/otrs/index.pl?Action=AdminUser> das Passwort!

3.1. System-eMail-Adressen & POP3

Auf <http://localhost/otrs/index.pl?Action=AdminSystemAddress> stellst Du die System-eMail-Adresse(n) ein. Das sind die Adressen, über die Anfragen an Dein OTRS gestellt werden sollen.

Auf <http://localhost/otrs/index.pl?Action=AdminPOP3> legst Du für diese System-eMail-Adresse(n) den(die) POP3-Account(s) an.


Dein OTRS ist bereit, Du kannst loslegen. Schau Dich ruhig ein wenig in der Oberfläche um, und bearbeite das Test-Ticket.

Schreibe nun eine eMail an eine der System-eMail-Adressen. Dann gibst Du in einem Browser die folgende Adresse ein, damit wird das System per POP3 mit eMails versorgt:

<http://localhost/obin/PostMasterPOP3.pl>

Deine eMail ist in der Queue *Raw* angekommen – hier findest Du sie:

<http://localhost/otrs/index.pl?Action=AgentQueueView>. Beantworte diese erste externe Anfrage.

 Es ist im übrigen *keine* gute Idee, von einer der System-eMail-Adressen aus in das System zu schreiben, das gibt unschöne und unbrauchbare Schleifen.


3.2. CRON-Job(s) einrichten

Zuletzt bringen wir den Sambar-Server noch dazu, regelmäßig (alle 10 Minuten) und vor allem ohne unser Zutun eMails in die OTRS-Datenbank zu pumpen. Dafür trägst Du Folgendes in

D:\Sambar\config\schedule.ini ein:

```
[cron]
# fetch emails every 10 minutes
*/10 * * * * perl d:/otrs/bin/PostMasterPOP3.pl &
```

 Diese Änderung wird 1–2 Minuten nach Speichern der Datei wirksam.

 Wenn Du magst, kannst Du an dieser Stelle bereits die folgenden CRON Jobs einrichten. Was sie tun, ist kurz angedeutet und an anderer Stelle in diesem Manual beschrieben (möglicherweise nur auf Englisch).

```
# start generic agent every 20 minutes
*/20 * * * * perl d:/otrs/bin/GenericAgent.pl &
```

```
# check every 60 min the pending jobs
45 * * * * perl d:/otrs/bin/PendingJobs.pl &

# Rebuild Ticket Index every day
01 01 * * * perl d:/otrs/bin/RebuildTicketIndex.pl &

# delete every six hours old session ids
55 */6 * * * perl d:/otrs/bin/DeleteSessionIDs.pl --expired &

# unlock every hour old locked tickets
35 * * * * perl d:/otrs/bin/UnlockTickets.pl --timeout &
```



Keine CRON Jobs auf Apache

Bitte beachte, dass Apache keine CRON Jobs kennt. Du musst entweder die Jobs von der Kommandozeile aus manuell starten, oder den Windows Taskplaner nutzen. Lass' das folgende Kommando ausführen, um zu verhindern, dass hässliche, schwarze Fenster während der Ausführung eines Skripts aufklappen: **cmd /c start /min perl d:\otrs\bin\NameDesCRONJobs.pl**

4. Geschafft!

Herzlichen Glückwunsch, die grundlegende Win32-Installation ist hoffentlich erfolgreich abgeschlossen!

Für den Fall, dass etwas an einer Stelle unklar geblieben ist und/oder Du statt der OTRS-Oberfläche nur Fehlermeldungen bekommst, zögere nicht, eine eMail an [otrs-win32 at robertkehl.de](mailto:otrs-win32@robertkehl.de) zu senden. Hoffentlich erhältst Du ein Ticket zurück ;-)

Mehr noch als über Trouble Tickets freue ich mich natürlich über Erfolgsmeldungen!

Natürlich wird diese Anleitung analog der Entwicklung des [OTRS](#) angepasst und weiterentwickelt. Ich erhebe keinen Anspruch darauf, mit der hier beschriebenen Architektur den Stein der Weisen gefunden zu haben, d.h. mir ist durchaus bewußt, dass diese Anleitung ganz schön von hinten durch die Brust in's Auge schießt, aber: Der Pfeil steckt – und darauf kommt's an :) Ach ja, natürlich funktioniert OTRS auch auf Apache, demnächst folgt einen Teil hierüber.

Viel Spaß mit Deinem OTRS auf Sambar!