

OTRS CVS - Handbuch

Martin Edenhofer

OTRS Kernteam

martin@otrs.org

Stefan Wintermeyer

OTRS Kernteam

stefan@otrs.org

Sebastian Wormser

OTRS Kernteam

sibbi@sibbi.com

Robert Kehl

((otrs.de)) Team

robert.kehl@otrs.de

OTRS CVS - Handbuch

by Martin Edenhofer, Stefan Wintermeyer, Sebastian Wormser, and Robert Kehl

Violet December Sun Edition

Published 2003

Copyright © 2003 Martin Edenhofer, Stefan Wintermeyer und die doc-de Gruppe auf otrs.org

Die Rechteinhaber können keine Aussage über die Anwendbarkeit dieses Dokuments treffen, für welchen Zeck auch immer. Es wird "wie gesehen" zur Verfügung gestellt, ohne ausdrückliche oder implizite Garantie.

Dedication

Dieses Handbuch ist den netten Menschen vom *Cafe Lucas* (<http://www.cafe-lucas.de/>) und *Enchilada* (<http://www.enchilada.de/>) (zwei Restaurants in Nürnberg). Danke für die Happy Hour! Heutzutage trifft man uns eher in Frankfurt, aber wir erinnern uns immer gerne der guten, alten Zeiten in Nürnberg.

Table of Contents

Vorwort	i
1. Installation von OTRS	1
1.1. Installation des RPM Softwarepackets auf einer SuSE Linux Distribution (schnell&einfach).....	1
1.2. Verwendung der tar.gz Datei zur Installation von ORTS auf einer beliebigen Linux/Unix plattform.....	5
1.2.1. Die Installation.....	5
1.2.2. Die Datenbank	7
1.2.3. Der Webserver.....	9
1.3. Installation von OTRS unter Microsoft Windows	11
1.3.1. otrs4win32 - Der Automatische Installer.....	11
1.3.1.1. Download.....	11
1.3.1.2. Installation.....	11
1.3.1.3. Das OTRS System starten.....	11
1.3.1.4. Die Server stoppen.....	12
1.3.1.5. OTRS deinstallieren (wenn überhaupt).....	13
1.3.2. Aktualisieren (Upgrading)	13
1.3.3. Schritt-für-Schritt: Die manuelle Installation	13
1.3.3.1. Benötigte Software.....	13
1.3.3.1.1. PERL installieren	14
1.3.3.1.1.1. Installation zusätzlicher Perl Pakete	14
1.3.3.1.2. Installation des Webservers.....	14
1.3.3.1.2.1. Apache installieren.....	14
1.3.3.1.3. MySQL installieren.....	14
1.3.3.1.4. OTRS installieren.....	14
1.3.3.2. Konfiguration	15
1.3.3.2.1. Perl konfigurieren.....	15
1.3.3.2.2. Apache konfigurieren.....	16
1.3.3.2.3. MySQL konfigurieren.....	16
1.3.3.2.4. OTRS Konfiguration	17
1.3.3.2.4.1. CONFIG.PM	17
1.3.3.3. Der erste Start	18
1.3.3.3.1. System Email Adresse & POP3.....	18
1.3.3.3.2. No CRON Jobs on Apache	18
1.3.3.4. Fertig!.....	18
2. Trouble Ticket Systeme	21
2.1. Ein einfaches Beispiel für ein kleines Trouble Ticket System.....	21
2.2. Was soll ich mir unter Trouble Ticket vorstellen?	21
2.3. Was muss ich mir unter einer Queue vorstellen?	22
3. Erste Schritte	23
3.1. Als root anmelden und neue Benutzer anlegen.....	23
3.2. Als normaler Benutzer anmelden.....	24
4. Ticket Themen.....	27
4.1. Ticket via eMail beantworten.....	27
4.2. Ticket via Telefon beantworten.....	27
4.3. Ticket 'Inhalt'	28
4.4. Ticket-Historie	28
4.5. Notiz zum Ticket hinzufügen.....	29
4.6. Ticket schliessen	29
4.7. Ticket weiterleiten.....	30
4.8. Ticket-Priorität	31
4.9. Druckansicht	31
5. System-eMail-Benachrichtigungen	33
5.1. Agenten-Benachrichtigungen	33
5.1.1. Neues Ticket	33
5.1.2. Folgenachricht.....	33
5.1.3. Neues Ticket in Favoriten-Queues	33
5.1.4. Ticket-Sperre Timeout	33
5.2. Kunden-Benachrichtigungen (min 1.1)	33
5.2.1. Ticket-Status	33
5.2.2. Ticket-Besitzer	33
5.2.3. Ticket-Queue.....	33

6. Automatische Antworten.....	35
6.1. Antwort	35
6.2. Folgenachricht.....	35
6.3. Abweisung	35
6.4. Geschlossen -> Neues Ticket.....	35
7. eMails empfangen	37
7.1. Via POP3-Konten - der einfache Weg (PostMasterPOP3.pl)	37
7.2. Via Kommandozeilen-Programm und z.B. procmail (PostMaster.pl)	37
7.2.1. eMails via POP3 oder IMAP und fetchmail für PostMaster.pl empfangen	38
7.3. Filterung/Verteilung über OTRS/Webinterface.....	38
7.4. Filterung/Verteilung über OTRS/PostMaster Module (für komplexere Verteilungsszenarien)	38
7.5. Filterung/Verteilung durch Procmail (für komplexere Verteilungsszenarien)	39
7.6. Beispiel für Filterung/Verteilung durch Procmail und ein Webformular.....	41
8. eMails versenden.....	43
8.1. Via Sendmail (Standard)	43
8.2. Via SMTP relay/smarthost (min. OTRS 1.1)	43
9. Benutzer (Agenten)	45
9.1. Wie es funktioniert.....	45
9.1.1. Admin-Interface	45
9.2. Benutzer-Backend	45
9.3. Benutzer-Authentifizierung-Backend	45
9.3.1. Datenbank (Standard)	45
9.3.2. LDAP	45
9.3.3. HTTPBasicAuth.....	46
9.3.4. Radius	47
10. Kunden (min. OTRS 1.1).....	49
10.1. Wie es funktioniert.....	49
10.1.1. Agenten-Interface	49
10.1.2. Admin-Interface	49
10.1.3. Kunden-Interface	49
10.2. Kundenbenutzer-Backend	49
10.2.1. Datenbank (Standard)	49
10.2.2. LDAP	50
10.3. Kunden-Authentifizierungs-Backend.....	51
10.3.1. Datenbank (Standard)	52
10.3.2. LDAP	52
10.3.3. HTTPBasicAuth.....	52
10.3.4. Radius	53
10.4. Kunden-Selbstanmeldung anpassen.....	53
10.4.1. Vorlage	53
10.4.2. Kunden-Mapping	53
10.4.3. Kunden-Tabelle anpassen	54
11. LDAP Integration	55
11.1. Active Directory.....	55
11.1.1. Vorbereitungen.....	55
11.1.2. Konfiguration	55
11.1.2.1. Konfiguration des Active Directory	55
11.1.2.2. OTRS LDAP Optionen	55
11.1.3. Authentifizierung der Agenten.....	55
11.1.4. Authentifizierung der Kundenbenutzer	56
11.1.5. Daten der Kundenbenutzer	56
11.1.6. Tipps & Tricks	57
11.1.6.1. Control who's let in.....	57
11.1.6.2. Konfigurationsbeispiel	58
12. Zugriffsrechte	61
12.1. Gruppen/Benutzer/Queue Zugriffsrechte.....	61
12.2. Module Zugriffsrechte - Agent/Admin Oberfläche	61

13. Konfigurationsdatei	63
13.1. TicketHook (Ticket Kennzeichen)	63
13.2. FQDN.....	63
13.3. Protokoll.....	63
13.4. E-Mail Überprüfungsoption	64
13.5. Max. Postmaster E-Mail	64
13.6. Format der Ticket-Nummer	64
13.7. Datenbankeinstellungen	65
13.8. ASP (Application Service Provider) Optionen	65
13.9. Custom Queue.....	65
13.10. Queue Verzeichnisart	66
13.11. Agent Benachrichtigung	66
13.12. Sitzungsmanagement	67
13.13. URL Login und Logout Einstellungen	68
13.14. Agent-Oberfläche Standardeinstellungen	68
13.15. Rechtschreibüberprüfung	69
13.16. Format der Antwort (Antwort erstellen)	70
14. Cronjobs.....	73
14.1. Wie funktioniert's - Setup	73
14.2. Standard-Cronjobs	73
14.2.1. aaa_base	73
14.2.2. pending_jobs	73
14.2.3. postmaster	73
14.2.4. postmaster_pop3	73
14.2.5. unlock.....	73
14.2.6. session.....	74
14.2.7. rebuild_ticket_index.....	74
14.2.8. generic_agent-database	74
14.2.9. generic_agent	74
15. Generic-Agent	75
15.1. Web-Interface (bevorzugt)	75
15.2. Konfigurationsdatei (optional).....	76
15.2.1. Beispiele:.....	76
15.2.1.1. Spam Tickets schließen.....	77
15.2.1.2. Tickets löschen.....	77
15.2.1.3. Tickets von "tricky" nach "experts" verschieben	78
15.2.1.4. Verschieben von Tickets von abc nach experts und Änderung der Priorität.....	78
15.2.1.5. Verschieben von eskalierten Tickets nach experts und ausführend eines Kommandos	78
15.2.1.6. Löschen aller Tickets mit dem Betreff "I love you" in der Queue "abc".....	79
16. Übersetzung in verschiedene Sprachen	81
16.1. Neue Übersetzungsdateien	81
16.2. Übersetzungen für Aktionen	82
16.3. Eigene Übersetzungen und Zusätze	83
17. Anpassung des Front-End	85
17.1. Die dtl-Syntax	85
17.2. Beispiele für dtl-Dateien	86
17.2.1. Motd.dtl.....	86
17.2.2. Login.dtl.....	86
17.2.3. Header.dtl	88
17.3. Erstellen eines neuen Motivs	88
18. Anpassen des Ticket Status (ab OTRS 1.1)	91
19. Anpassen der Ticket Priorität (ab OTRS 1.1)	93
20. Multi Hosting.....	95
20.1. Multi Gruppen.....	95
20.2. Multi Setup	95

21. Leistungsverbesserung	97
21.1. OTRS	97
21.1.1. TicketIndexModule	97
21.1.2. TicketStorageModule	97
21.2. Datenbank	97
21.2.1. MySQL	98
21.2.2. PostgreSQL	98
21.3. Webserver.....	98
21.3.1. Datenbank Verbindung.....	98
21.3.2. Vorgeladene Module - startup.pl.....	98
21.3.3. Perl Module bei Änderung neu laden	98
21.3.4. Die richtige Strategie wählen.....	98
21.3.5. mod_gzip.....	98
21.3.6. mod_dosevasive	98
22. Das FAQ System.....	101
22.1. Was sind FAQs?	101
22.2. Vorarbeiten	101
22.3. Schnittstellen.....	101
22.4. FAQ Funktionen.....	101
22.4.1. FAQ Funktionen für Agenten.....	101
22.4.2. FAQ Funktionen für Kunden.....	102
22.4.3. FAQ Funktionen für öffentliche FAQs.....	102
23. Fehlersuche.....	103
23.1. Generelle Probleme mit OTRS auf SuSE Linux.....	103
23.1.1. SuSE Linux 8.0	103
23.1.2. SuSE Linux und Postfix	103
23.2. Generelle Probleme mit OTRS auf Mandrake.....	103
23.2.1. Aktualisieren Ihres Systems.....	103
23.3. Generelle Probleme mit OTRS auf anderen Distributionen (z.B. Redhat)	104
23.4. Probleme mit Apache.....	104
23.4.1. Interner Server Fehler (Internal Server Error)	104
23.4.2. Fehler: Kann nicht zur Datenbank verbinden!.....	104
23.4.3. FreeBSD, PostgreSQL und Apache (install_driver(Pg) failed)	104
23.5. Probleme mit MySQL.....	105
23.5.1. Überprüfen Sie die Datenbankverbindung.....	105
23.5.2. Fehlermeldung: Access denied for user: 'otrs@localhost'	105
23.5.3. root Passwort für MySQL unbekannt	105
23.5.4. http://localhost/otrs/installer.pl	105
23.6. Sonstige Probleme	105
23.6.1. Probleme beim Empfang von e-Mails	105
23.6.2. root Passwort für OTRS unbekannt	106
24. Entwicklerhandbuch.....	107
24.1. Kernmodule.....	107
24.2. Programmierstil.....	107
24.3. Fehlerbeseitigung	107
25. Entwicklerhandbuch - OTRS Module schreiben.....	109
25.1. Agenten-Oberfläche Benachrichtigungsmodule	109
25.2. Kunden-Oberfläche Benachrichtigungsmodule	110
25.3. Ticketnummer Generator	110
25.4. Module zur Authentifizierung von Agenten	112
25.5. Module zur Authentifizierung von Kundenbenutzern	114
25.6. Module für Kundenbenutzer	114
25.7. PostMaster Filter Module	115
25.8. Generic Agent Modules	115
25.9. Agenten-Ticketberechtigungen Module	116
25.10. Kunden-Ticketberechtigungen Module	117

26. FAQ - Häufig gestellte Fragen.....	119
Glossar	123
A. Online Quellen.....	127
A.1. Homepage	127
A.2. Mailinglisten	127
A.3. Fehlerverfolgung	127
B. Das OTRS Kernteam	129
B.1. Martin Edenhofer	129
B.2. Stefan Wintermeyer.....	129
B.3. Sebastian Wormser.....	129
B.4. Dipl.-Ing. Robert Kehl	129
C. Danksagungen	131
D. RFC 1297	133
E. GNU Free Documentation License	135
0. PREAMBLE	135
1. APPLICABILITY AND DEFINITIONS	135
2. VERBATIM COPYING	135
3. COPYING IN QUANTITY	136
4. MODIFICATIONS	136
5. COMBINING DOCUMENTS.....	137
6. COLLECTIONS OF DOCUMENTS	137
7. AGGREGATION WITH INDEPENDENT WORKS	137
8. TRANSLATION	137
9. TERMINATION	138
10. FUTURE REVISIONS OF THIS LICENSE.....	138
How to use this License for your documents	138
Literatur.....	139

List of Examples

7-1. .fetchmailrc	38
7-2. .procmailrc	40
11-1. Kernel/Config.pm - AuthModule	55
11-2. Kernel/Config.pm - AuthModule::LDAP Einstellungen	56
11-3. Kernel/Config.pm - AuthModule::LDAP Einstellungen zum SearchUser.....	56
11-4. Kernel/Config.pm - Customer::AuthModule LDAP Einstellungen	56
11-5. Kernel/Config.pm - CustomerUser::LDAP Einstellungen	56
11-6. Kernel/Config.pm - GroupDN für Agenten.....	57
11-7. Kernel/Config.pm - GroupDN for Customers	57
11-8. Kernel/Config.pm - AuthModule LDAP settings	58
13-1. Kernel/Config.pm - TicketHook / Ticket Kennzeichen	63
13-2. Kernel/Config.pm - FQDN	63
13-3. Kernel/Config.pm - LogModule	63
13-4. Kernel/Config.pm - CheckEmailAddresses.....	64
13-5. Kernel/Config.pm - CheckMXRecord.....	64
13-6. Kernel/Config.pm - PostmasterMaxEmails	64
13-7. Kernel/Config.pm - TicketNumberGenerator.....	64
13-8. Kernel/Config.pm - Datenbankeinstellungen	65
13-9. Kernel/Config.pm - ChangeOwnerToEveryone	65
13-10. Kernel/Config.pm - Custom Queue	65
13-11. Kernel/Config.pm - QueueListType	66
13-12. Kernel/Config.pm - Agent notification	66
13-13. Kernel/Config.pm - Session management	67
13-14. Kernel/Config.pm - URL Login und Logout Einstellungen.....	68
13-15. Kernel/Config.pm - Agent-Oberfläche Standardeinstellungen.....	68
13-16. Kernel/Config.pm - Rechtschreibüberprüfung	69
13-17. Kernel/Config.pm - Rechtschreibüberprüfung - Agenten Präferenzen	70
13-18. Kernel/Config.pm - Rechtschreibüberprüfung - Überprüfungspflicht.....	70
13-19. Kernel/Config.pm - Antwortenformat (unix_style)	70
13-20. Kernel/Config.pm - Antwortenformat (ms_style)	70
16-1. Kernel/Language/de.pm - alte Datei.....	81
16-2. Kernel/Language/xy.pm - Neue Datei	82
16-3. Kernel/Language/de_Calendar.pm	83
16-4. Kernel/Language/de_Custom.pm	83

Vorwort

Über dieses Buch

Dieses Handbuch kann durchaus himmelschreiende Inkonsistenzen, fehlende Abschnitte und andere Un-Features enthalten, oft zu finden bei Werken, die in der Erstehung begriffen sind. Bitte zögern Sie nicht, jeden gefundenen Fehler in unserem Fehlerverfolgungssystem zu melden: <http://bugs.otrs.org/>.

Sie finden die aktuelle Online-Version im HTML- und PDF-Format dieses Buches unter <http://doc.otrs.org/> (<http://docu.otrs.org/>).

Wir freuen uns sehr über Ihr Feedback. Bitte zögern Sie nicht, uns eine Nachricht an [<feedback@otrs.org>](mailto:feedback@otrs.org) zu senden.

Ihr OTRS Kernteam

PS. Falls Sie auf der Suche nach kommerziellem Support und Beratung zu OTRS sind, besuchen Sie bitte die ((otrs.de)) Homepage (<http://www.otrs.de/>).

Chapter 1. Installation von OTRS

Der hier beschriebene Weg der Installation von OTRS wurde auf einem "frisch-installierten" System getestet. Sollten Sie mit der Installation auf Ihrer bestehenden Installation Probleme haben, so versuchen Sie es doch bitte mit einer Neuinstallation Ihres Systems. Die meisten Probleme während des Installationsprozesses resultieren aus fehlerhaften Webserver-Konfigurationen oder fehlenden Datenbank Passwörtern. Nochmals: Benutzen Sie bitte eine Neuinstallation Ihres Betriebssystems um Probleme bei der Installation zu umgehen.

Wir versuchen, die Installation so einfach wie möglich zu halten. Allerdings ist OTRS ein umfangreiches und komplexes Programm, welches nicht einfach in ein Verzeichnis entpackt werden kann und das war's.

1.1. Installation des RPM Softwarepakets auf einer SuSE Linux Distribution (schnell&einfach)

Dieser Abschnitt beinhaltet die Anleitung für die Installation von OTRS auf SuSE Linux. Getestet wurden die SuSE Versionen 7.3, 8.0 und 8.1. Bevor Sie mit der Installation beginnen, schauen Sie bitte unter <http://otrs.org/> (<http://www.otrs.org/>) nach, ob eine aktuellere Version von OTRS als .rpm Datei vorliegt. Sollte dies so sein, verwenden Sie bitte diese!

Installieren Sie OTRS mittels YaST (YaST2) oder der Kommandozeile und rpm (je nach Vorliebe). Beachten Sie jedoch, dass OTRS einige Perl-Module benötigt, die nicht standardmäßig in einer typischen SuSE Installation enthalten sind. YaST sollte die bessere Wahl sein, da es alle Abhängigkeiten automatisch beachtet und löst.

Sollten Sie den Kommandozeilen-rpm-Weg bevorzugen, so müssen Sie die Perl-Module manuell vor Beginn der Installation von OTRS installiert haben.

```
ernie:~ # rpm -ivh /tmp/otrs.rpm
otrs                                     #####
Check OTRS user (/etc/passwd)... otrs exists.
```

Next steps:

```
[SuSEconfig]
Execute 'SuSEconfig' to configure the webserver.
```

```
[start Apache and MySQL]
Execute 'rcapache restart' and 'rcmysql start' in case they don't run.
```

```
[install the OTRS database]
Use a webbrowser and open this link:
http://localhost/otrs/installer.pl
```

```
[OTRS services]
Start OTRS 'rcotrs start-force' (rcotrs {start|stop|status|restart|start-force|stop-force}).
```

Have fun!

Your OTRS Team
<http://otrs.org/>

```
ernie:~ #
```

Es ist an der Zeit, SuSEconfig zu starten:

```
ernie:~ # SuSEconfig
Starting SuSEconfig, the SuSE Configuration Tool...
Running in full featured mode.
Reading /etc/sysconfig and updating the system...
Executing /sbin/conf.d/SuSEconfig.aaa_at_first...
Executing /sbin/conf.d/SuSEconfig.apache...
Including /opt/otrs/scripts/apache-httpd.include.conf
Executing /sbin/conf.d/SuSEconfig.bootsplash...
Executing /sbin/conf.d/SuSEconfig.doublecheck...
Executing /sbin/conf.d/SuSEconfig.guile...
Executing /sbin/conf.d/SuSEconfig.hostname...
Executing /sbin/conf.d/SuSEconfig.ispell...
Executing /sbin/conf.d/SuSEconfig.perl...
Executing /sbin/conf.d/SuSEconfig.permissions...
Executing /sbin/conf.d/SuSEconfig.postfix...
Setting up postfix local as MDA...
```

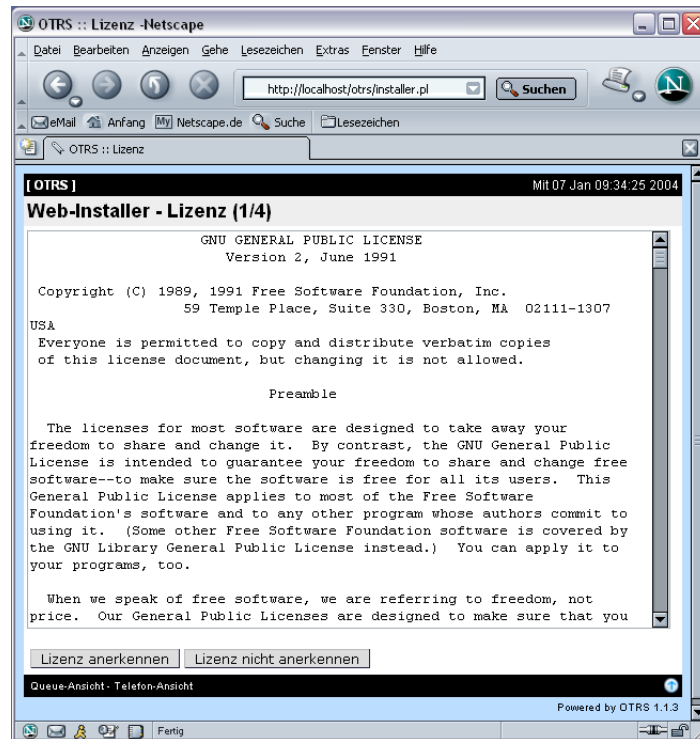
```
Setting SPAM protection to "off"...  
Executing /sbin/conf.d/SuSEconfig.profiles...  
Finished.
```

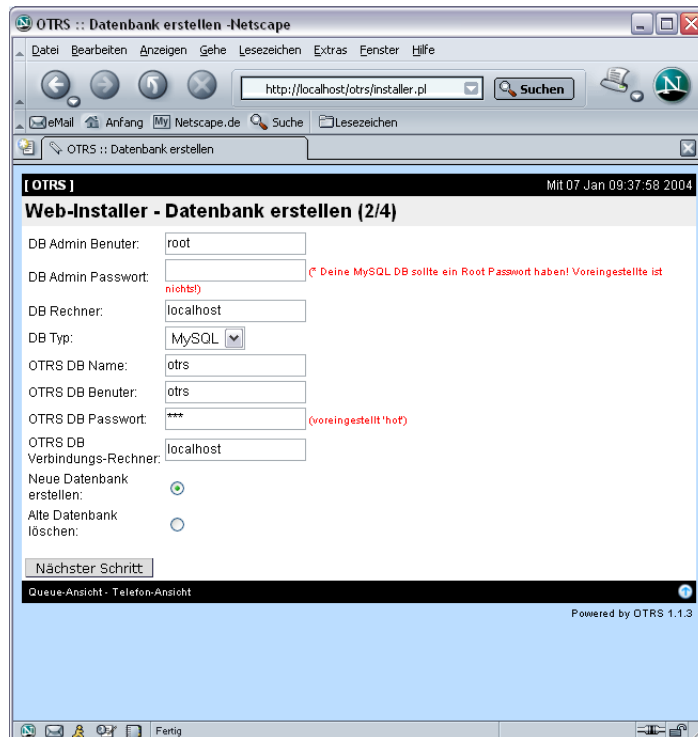
Wenn die Installation des OTRS.rpm abgeschlossen ist, müssen Sie Ihren Webserver (Apache) neustarten (rcapache restart), damit die Änderungen in der Konfiguration übernommen werden.

```
ernie:~ # rcapache restart  
Shutting down httpd  
Starting httpd [ PERL ]  
ernie:~ #
```

done
done

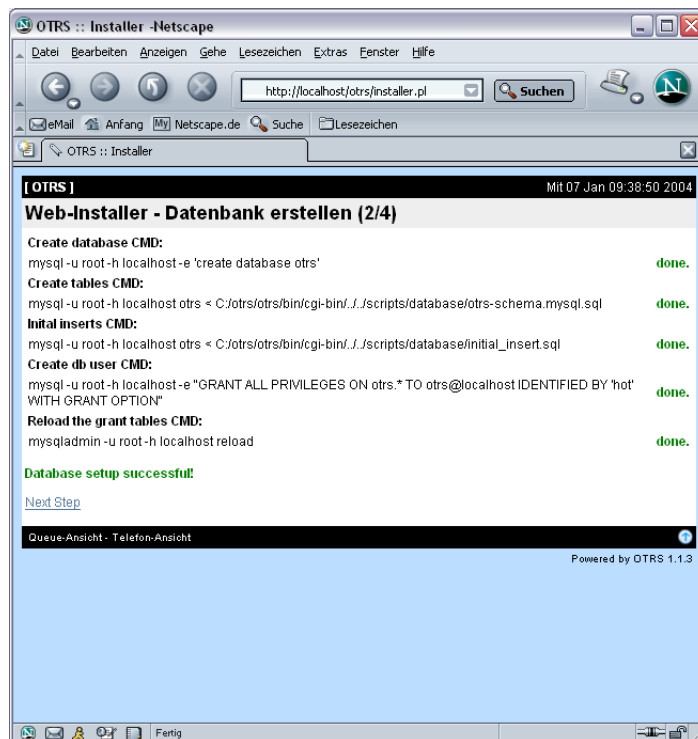
Nun müssen Sie die Datenbank aufsetzen. Öffnen Sie hierzu folgenden Link: <http://localhost/otrs/installer.pl>.

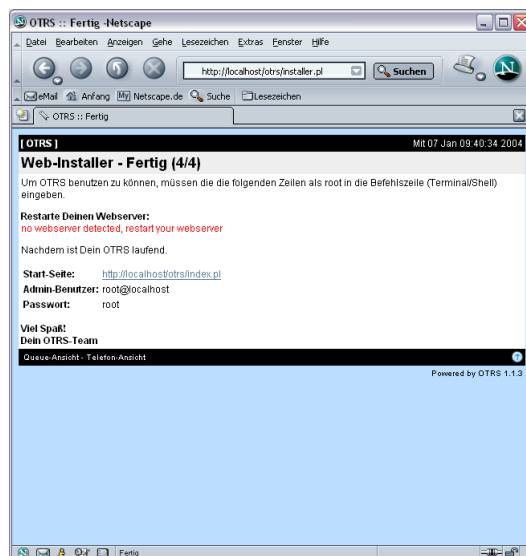
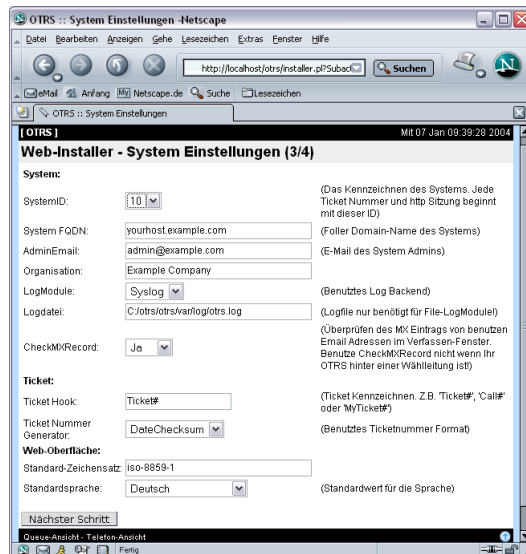




Warning

Es ist niemals eine wirklich gute Idee, Standardpasswörter zu verwenden! Sie installieren hier ein wichtiges Stück Software, bei dem Sie darauf achten sollten, dass niemand Unbefugtes darauf Zugriff hat. Sichere Passwörter sind hierfür unabdingbar.





OK, es ist an der Zeit, OTRS zu starten. (Verwenden Sie den Kommandozeilenbefehl `rcotrs restart-force`)

```
ernie:~ # rcotrs restart-force
Shutting down OTRS
Disable /opt/otrs/bin/PostMaster.pl ... done.
no crontab for otrs
Shutting down cronjobs ... failed!
Shutting down OTRS (completely)
Shutting down Apache ... done.
Shutting down MySQL ... done.

Starting OTRS (completely)
Starting Apache ... done.
Starting MySQL ... done.
Starting OTRS
Checking Apache ... done.
Checking MySQL ... done.
Checking database connect... (It looks Ok!).
Enable /opt/otrs/bin/PostMaster.pl ... done.
Checking otrs spool dir... done.
Creating cronjobs (source /opt/otrs/var/cron/*) ... done.
```

done

```
-->> http://ernie.example.com/otrs/index.pl <--
done
done

ernie:~ #
```

Fertig! War das nicht leicht? ;-)

Sie erreichen OTRS über den folgenden Link: <http://ernie.example.com/otrs/index.pl> oder <http://localhost/otrs/index.pl>

1.2. Verwendung der tar.gz Datei zur Installation von OTRS auf einer beliebigen Linux/Unix plattform

Dieser Abschnitt ist eine Anleitung zur Installation von OTRS auf einer beliebigen Linux Distribution. Bitte verwenden Sie diesen Weg nur, wenn Sie sich mit der manuellen Installation auskennen. Anderfalls steht Ihnen die RPM Version zur Verfügung. Bevor Sie mit der Installation beginnen, schauen Sie bitte unter <http://otrs.org/> (<http://www.otrs.org/>) nach, ob eine aktuellere Version von OTRS als tar.gz Datei vorliegt. Sollte dies so sein, verwenden Sie bitte diese!

1.2.1. Die Installation

```
# --
# INSTALL description of OTRS
# Copyright (C) 2001-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: install-cli.sgml,v 1.4 2004/03/02 08:40:22 robert Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --

Software requirements?
=====

On all Perl-Platforms! You need:
* min. Perl5
* Database (e. g. MySQL, PostgreSQL)
* Webserver (mod_perl isn't required but "very nice" to have)
* LDAPv2 compliant server (OpenLDAP for works fine, however LDAP isn't required)
* some CPAN-Module (DBI, DBD::mysql, Digest::MD5, MIME::Base64 (if Perl < 5.8), Net::DNS)
  for LDAP (Net::LDAP) and for stats (GD, GD::Text, GD::Graph, GD::Graph::lines,
  GD::Text::Align)

Installation:
=====

This few steps describe an OTRS installation incl. webserver and database
settings. The OTRS user is "otrs" and the home (root) directory is /opt/otrs
(of course you can choose an other directory or/and OTRS user).

1. Install CPAN Modules (if needed):
-----
Note: use "bin/otrs.checkModules" to get an overview of all installed and
required cpan modules.

a) Install the RPMs if your distributions provides RPMs for
the required CPAN modules.

b) Install the required modules via CPAN shell (http://www.cpan.org/)
perl -MCPAN -e shell;
...
install Digest::MD5
install Net::DNS
install MIME::Base64 (if Perl < 5.8)
...
```

```

    if you plan to use an LDAP directory service you should install Net::LDAP
    ...
    install Net::LDAP
    ...
    and maybe the GD stuff (stats support, not required!)
    ...
    install GD
    install GD::Text
    install GD::Graph
    install GD::Graph::lines
    install GD::Text::Align
    ...

```

2. Create user:

```

-----
Add user:
-----
$shell: useradd -d /opt/otrs/ -c 'OTRS user' otrs

Add user to webserver group (if the webserver is not running with OTRS user):
-----
$shell: usermod -G nogroup otrs

(SuSE=nogroup, Red Hat=apache)

```

3. Install tar.gz:

```

-----
$shell: cd /opt/
$shell: tar -xvzf otrs-xxxx-xx-xx.tar-gz

```

4. Demo config files:

```

-----
There are several OTRS demo config files in $OTRS_HOME/Kernel/*.dist
and $OTRS_HOME/Kernel/Config/*.dist. Make copies of all demo config files:

cp Kernel/Config.pm.dist Kernel/Config.pm

cd Kernel/Config/
for foo in *.dist; do cp $foo `basename $foo .dist`; done

Or if you are installing OTRS an a Windows system:

copy Kernel/Config.pm.dist Kernel/Config.pm

cd Kernel/Config/
copy *.dist *.

```

5. Check if all needed modules are installed:

```

-----
$shell::~> perl -cw /opt/otrs/bin/cgi-bin/index.pl
/opt/otrs/bin/cgi-bin/index.pl syntax OK
$shell::~> perl -cw /opt/otrs/bin/PostMaster.pl
/opt/otrs/bin/PostMaster.pl syntax OK
$shell::~>

If you get "syntax OK" it seems to be Ok. Go ahead.

```

6. Webserver:

```

-----
[follow README.webserver]

```

7. File Permissions:

```

-----
Set the file permissions with
"$HOME/bin/SetPermissions.sh <OTRS_HOME> <OTRS_USER> <WEBSERVER_USER> [OTRS_GROUP] [WEB_GROUP]"
e. g.
Webserver with OTRS user: "SetPermissions.sh /opt/otrs otrs otrs"
Webserver with wwwrun user (e. g. SuSE): "SetPermissions.sh /opt/otrs otrs wwwrun"
Webserver with apache user (e. g. Redhat): "SetPermissions.sh /opt/otrs otrs apache apache"

```

```
8. Database setup:
-----
If you use MySQL, you can use the Web-Installer (http://yourhost/otrs/installer.pl).
Else follow README.database --> "DB - Setup Example".

9. Config file ($HOME/Kernel/Config.pm):
-----
If you used the Web-Installer, you can skip this point. If not,
set some Kernel::Config ($HOME/Kernel/Config.pm) Options (FQDN, SystemID, TicketHook,
Home, ...)

10. First Login:
-----
http://yourhost/otrs/index.pl
User: root@localhost
PW: root

--> goto AdminArea and set some config settings (UserAdd, Queues, ...).

Finished.

11. First Email:
-----
pipe an email directly into $OTRS_HOME/bin/Postmaster.pl
(e. g. 'cat /opt/otrs/doc/test-email-1.box | /opt/otrs/bin/PostMaster.pl').

12. Cronjobs for the OTRS user:
-----
There are several OTRS default cronjobs in $OTRS_HOME/var/cron/*.dist.
Make copies of all of the default cronjobs:

    cd var/cron
    for foo in *.dist; do cp $foo `basename $foo .dist`; done

Or if you are installing OTRS an a Windows system:

    cd var/cron
    copy *.dist *.

Use $OTRS_HOME/bin/Cron.sh {start|stop|restart} to start or stop this cronjobs
from $OTRS_HOME/var/cron/* (.dist will be ignored).

>> Note: Install this cronjobs as OTRS user. <<

Notes:
=====

Also you should read the OTRS performance tuning chapter on our homepage:
http://doc.otrs.org/cvs/en/html/performance-tuning.html

For installation questions ask otrs@otrs.org (http://lists.otrs.org/).

Have a lot of fun.

Your OTRS Team

(January 2004 Frankfurt/Germany)

EOF
```

1.2.2. Die Datenbank

```
# --
# README.database - database description of OTRS
# Copyright (C) 2001-2002 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: install-cli.sgml,v 1.4 2004/03/02 08:40:22 robert Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --

Where can I find the database description files?
=====

XML:
====
$HOME_OTRS/scripts/database/otrs-schema.xml

The XML description files for torque which generate the SQL for your
target database e. g. MySQL, PostgreSQL, DB2, Oracle, ...)

More Infos: http://jakarta.apache.org/turbine/turbine-2/howto/torque-howto.html

MySQL:
-----
$HOME_OTRS/scripts/database/otrs-schema.mysql.sql

PostgreSQL:
-----
$HOME_OTRS/scripts/database/otrs-schema.postgresql.sql

Initial insert file:
=====
$HOME_OTRS/scripts/database/initial_insert.sql contains all needed standard
values. At first use the otrs-schema.*.sql and the insert this file.

DB - Setup Example (MySQL):
=====
Create OTRS database:
-----
shell> mysql -u root -p -e 'create database otrs'

Create the OTRS tables:
-----
shell> mysql -u root -p otrs < scripts/database/otrs-schema.mysql.sql

Insert initial data:
-----
shell> mysql -u root -p otrs < scripts/database/initial_insert.sql

Create an database user:
-----
shell> mysql -u root -p -e 'GRANT ALL PRIVILEGES ON otrs.* TO otrs@localhost IDENTIFIED BY "some-pass" WITH GR

Reload the grant tables of your mysql-daemon:
-----
shell> mysqladmin -u root -p reload

*****
*
* Change the DB-Settings (host, database, user and password) *
*
* $OTRS_HOME/Kernel/Config.pm
* [...]
* # Database
* # (The database name.)
* $Self->{Database} = 'otrs';
*
```

```

*                                     *
* # DatabaseUser                     *
* # (The database user.)             *
* $Self->{DatabaseUser} = 'otrs';    *
*                                     *
* # DatabasePw                       *
* # (The password of database user.) *
* $Self->{DatabasePw} = 'some-pass'; *
* [...]                             *
*                                     *
*****
EOF

```

1.2.3. Der Webserver

```

# --
# README.webserver - webserver description of OTRS
# Copyright (C) 2001-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: install-cli.sgml,v 1.4 2004/03/02 08:40:22 robert Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --

Which webserver is needed?
=====
I prefer the apache webserver (http://httpd.apache.org).

Configuration:
=====
"After" this steps, you will get the login page at
http://your-host/otrs/index.pl or http://your-host/otrs/installer.pl.

SuSE Linux:
=====
a) Install the RPM-Package (http://otrs.org/ - "rpm -i otrs-xxx.rpm").

b) The manual way:
   Use the "$OTRS_HOME/scripts/suse-httpd.include.conf" include config file.

   Add it to /etc/sysconfig/apache with HTTPD_CONF_INCLUDE_FILES
   [...]
   HTTPD_CONF_INCLUDE_FILES=/opt/otrs/scripts/suse-httpd.include.conf
   [...]

   Start SuSEconfig and restart the webserver (rcapache restart).

Or edit the httpd.conf directly:
=====
a)
*) Change the webserver user (normally wwwrun) to the OTRS user (otrs).

   [...]
   User wwwrun
   [...]
   User otrs
   [...]

*) If you can't change the user and group of your webserver (system-wide),
   because you have other applications running on this server, you can
   also work with group permissions (more tricky).

   Use "$HOME/bin/SetPermissions.sh <OTRS_HOME> <OTRS_USER> <WEBSEVER_USER> [OTRS_GROUP] [WEB_GROUP]"

```

```

e. g.
  Webserver with otrs user: "SetPermissions.sh /opt/otrs otrs otrs"
  Webserver with wwwrun user (e. g. SuSE): "SetPermissions.sh /opt/otrs otrs wwwrun"
  Webserver with apache user (e. g. Redhat): "SetPermissions.sh /opt/otrs otrs apache"

b)
*)
Without mod_perl (just CGI):
=====
Add this to the cgi-bin stuff section in httpd.conf
[...]
Alias /otrs-web/ "/opt/otrs/var/httpd/htdocs/"
ScriptAlias /otrs/ "/opt/otrs/bin/cgi-bin/"
[...]

*)
With mod_perl (speed!):
=====
Add this to the mod_perl stuff section in httpd.conf
[...]
Alias /otrs-web/ "/opt/otrs/var/httpd/htdocs/"
Alias /otrs/ "/opt/otrs/bin/cgi-bin/"

<Location /otrs>
  SetHandler perl-script
  PerlHandler Apache::Registry
  Options ExecCGI
  PerlSendHeader On
  PerlSetupEnv On
</Location>

You may want to use a mod_perl startup script. Compiled modules on startup
(speed!!) Use the mod_perl startup script which comes with otrs
(scripts/apache-perl-startup.pl).

- Change the default startup script location of your httpd to
  $OTRS_HOME/scripts/apache-perl-startup.pl in httpd.conf
  [...]
  # load all otrs modules
  PerlRequire /opt/otrs/scripts/apache-perl-startup.pl
  [...]

Edit the scripts/apache-perl-startup.pl script:

- Establish datababase connections on process startup (httpd).

  [...]
  use Apache ();
  use Apache::DBI ();
  Apache::DBI->connect_on_init('DBI:mysql:otrs', 'otrs', 'some-pass');
  # Apache::DBI->connect_on_init($data_source, $username, $auth, \%attr)
  [...]

- Change the otrs lib dir (if needed)!

  [...]
  # --
  # set otrs lib path!
  # --
  use lib "/path/to/otrs/";
  use lib "/path/to/otrs/Kernel/cpan-lib";
  [...]

Nice! You will love mod_perl! ,-)

PS: If you use mod_perl2 use the scripts/apache2-* scripts!

c)
Restart the webserver
=====

```



```
d)
Web-Installer
=====
http://yourhost/otrs/installer.pl

First login
=====
http://yourhost/otrs/index.pl
User: root@localhost
PW: root

EOF
```

1.3. Installation von OTRS unter Microsoft Windows

In diesem Dokument wird die Installation von OTRS (<http://otrs.org>) auf *Microsoft Windows (Win32)* Systemen beschrieben. Der *Automatic Installer* wurde unter Microsoft Windows 2000 und XP getestet, sollte aber auch unter Windows NT4 und 2003 voll funktionsfähig sein. Achten Sie bitte darauf, dass Ihr Betriebssystem auf dem aktuellsten Stand ist, bevor Sie mit der Installation beginnen. Dies können Sie unter folgender URL überprüfen: <http://windowsupdate.microsoft.com/>

Bitte beachten Sie, dass wir mit diesem Installer keine 16-Bit-Windows Versionen (z.B. Windows 98, 98SE, ME) unterstützen.

1.3.1. otrs4win32 - Der Automatische Installer

Der automatische Installer von OTRS ist die komfortabelste Lösung um alle für OTRS notwendigen Software Komponenten zu installieren. Wir empfehlen ausdrücklich die Verwendung mit einer Neuinstallation Ihres Systems, das zudem alle wichtigen Systemupdates installiert hat!

Important: Folgende Versionen der Software-Komponenten werden installiert: OTRS 1.2.3 (<http://otrs.org/>), CRONw 1.4.1b (<http://cron.sourceforge.net/>), Apache 2.0.47 (<http://httpd.apache.org/docs-2.0/>), MySQL 4.0.14 (<http://www.mysql.com/products/mysql/>), PHP 4.3.3 (<http://www.php.net/>), PEAR (<http://pear.php.net/>), Perl 5.8.0 (<http://www.perl.org/>), mod_php 4.3.3 (<http://www.php.net/manual/en/install.apache2.php>), mod_perl 1.99_10 (<http://perl.apache.org/>), mod_ssl 2.0.47 (<http://www.modssl.org/>), openssl 0.9.7b (<http://www.openssl.org/> (???)), PHPMyAdmin 2.5.1 (<http://www.phpmyadmin.net/>), Webalizer 2.01-10 (<http://www.mrunix.net/webalizer/>), Mercury MailTransport System v3.32 (http://www.pmail.com/overviews/ovw_mercury.htm), JpGraph 1.12.1 (<http://www.aditus.nu/jpgraph/>), FileZilla FTP Server 0.8.5 (<http://filezilla.sourceforge.net/>).

1.3.1.1. Download

Die aktuelle Version des OTRS Installers können sie unter <http://otrs.org/download> downloaden. Der Name der ausführbaren Datei mit der Endung .exe folgt hierbei der folgenden Konvention: *otrs4win32-<OTRS-VERSION>-<STATUS>.exe*.

Wenn Sie die Authentizität der Datei bestätigen möchten, so laden Sie einfach noch die zugehörige md5-Datei herunter. Mit Hilfe des md5-summer (<http://md5summer.org/>) können Sie dann die Echtheit bestätigen!

1.3.1.2. Installation

Beachten Sie bitte, dass Sie Administratorrechte auf dem Computer benötigen, auf dem OTRS installiert werden soll. Hierzu melden Sie sich bitte als Administrator an oder führen den Installer unter Verwendung des "Ausführen als..." Kommandos als Administrator aus. Starten Sie bitte den Installer und folgen Sie den Anweisungen.

Installation auf aktuelles System

Der Installer ist gedacht für frisch zu installierende OTRS Systeme, d.h. Sie sind nicht dazu aufgerufen, ein bestehendes OTRS zu aktualisieren. Tatsächlich bietet der Installer momentan keine Möglichkeit, dies zu tun - bitte schauen Sie in den nachfolgenden Teil der Dokumentation, der sich mit dem Aktualisieren beschäftigt.

1.3.1.3. Das OTRS System starten

Stellen Sie sicher, dass keine anderen Serveranwendungen auf den Ports 25 (smtp), 80 (http), 443 (https), 3306 (MySQL) laufen. Wenn doch, finden Sie heraus, was für Server das sind und ob sie zu dieser Zeit auf der Maschine überhaupt laufen sollen. Beispiele sind Microsoft's IIS Webserver, der unbemerkt im Hintergrund mitläuft oder der dazugehörige SMTP Dienst. Schliessen Sie derartige Anwendungen, bevor Sie fortfahren.

Um die Server des Systems nach der Installation direkt zu starten, nutzen Sie die entsprechende Option auf der letzten Seite des Installers oder Einträge im Start Menü.

Warten Sie nach dem Ende des Installers ein wenig, bis Apache seine Startphase vollendet hat - dies kann einige Minuten dauern. Standardmäßig wird der ApacheMonitor nach der Installation gestartet, wie auch WinMySQLAdmin, ein Monitor für MySQL - beobachten Sie sie unten rechts in der TaskBar. Ihre Symbole wechseln auf 'Grün', wenn sie Anfragen entgegennehmen können:



ApacheMonitor und WinMySQLAdmin zeigen beide 'Rot' - haben Sie Geduld...



Warten Sie, während Apache noch 'Rot' zeigt - MySQL zeigt evtl. für eine kurze Zeit 'Gelb' an.



Apache ist noch nicht bereit - MySQL jedoch schon.



Beide sind nun 'Grün' - fertig!

Bitte beachten Sie, dass beide Hilfsprogramme nur nach der Installation automatisch gestartet werden. Nach einem Reboot können Sie sie über das Start Menü starten.

Besuchen Sie **<http://localhost/otrs/installer.pl>**, um den Web-Installer durchzuführen, der die Datenbank erstellen und mit den Anfangswerten füllen wird. Hernach werden Sie auf die Login-Seite geleitet:

<http://localhost/otrs/index.pl>.

Die Oberfläche für Ihre Kunden ist hier: **<http://localhost/otrs/customer.pl>**

Dieses Softwarepaket baut auf ApacheFriends (<http://www.apachefriends.org/>)' XAMPP auf, daher profitieren Sie auch von XAMPP's Möglichkeiten. Besuchen Sie **<http://localhost/index.html>** (inkl. dem Dateinamen!). Dies ist XAMPP mit seinen Komponenten - Sie finden OTRS im Menü zur Linken - hierdurch gelangen Sie zu einer Info-Seite über Ihr OTRS.

Es gibt auch noch eine Handvoll informativer Links im Start Menü, z.B. zu phpMyAdmin, WinMySQLAdmin und dem ApacheMonitor.

1.3.1.4. Die Server stoppen

Wenn Sie die Server als Systemdienste installiert haben (das ist Standard), können Sie das Start Menü oder die Dienstverwaltung von Windows nutzen, um sie zu stoppen. Dies müssen Sie aber nicht, weil bei einem Herunterfahren des Systems die Server automatisch und korrekt beendet werden. Sie sollten auch gar nicht heruntergefahren werden, um dem OTRS System es zu ermöglichen, Wartungsarbeiten im Hintergrund durchzuführen.

Die Standalone Versionen der Server betreffend, schliessen Sie einfach das Kontrollfenster von Apache, und starten Sie für MySQL `mysql_stop.bat`. Schliessen Sie Mercury über sein **File** Menü oder die tastenkombination **Alt-F4**, wenn das Mercury-fenster den Fokus hat. Stoppen Sie den CRONw Dienst über das Start Menü oder über den Befehl `net stop cron`.

1.3.1.5. OTRS deinstallieren (wenn überhaupt)

Für den Fall, dass Sie `otrs4win32` von Ihrem Computer entfernen wollen, beenden Sie einfach alle Server und führen sie das Wartungsprogramm aus, das sie im Startmenüeintrag unter "Administration" finden. All diese Schritte sind über das Startmenü möglich.

Im Zuge des Deinstallationsprozesses werden sämtliche Einträge, die im Zusammenhang mit OTRS entstanden sind, aus der Registrierung entfernt. Zudem werden die Dateien im Installationsverzeichnis und die Verknüpfungen im Startmenü gelöscht. Ein Neustart sollte nicht erforderlich sein!

Zusätzlich können Sie über das Wartungsprogramm einige Teile de Systems *RE*installieren. Dies sind die Dienste, die Einträge im Start Menü und in Windows' Registrierung.

1.3.2. Aktualisieren (Upgrading)

Folgen Sie diesen Schritten, um eine existierende `otrs4win32` Installation von 1.1.3 oder früher auf 1.2.0 zu aktualisieren:

- Stoppen Sie Apache und MySQL (über die "Administration" im Start Menu)
- Sichern Sie alles (Datenbank, Kernel/Config.pm, Kernel/Config/GenericAgent.pm, var/*)
- Haben Sie alles gesichert? , -)
- Wenn möglich, versuchen Sie zunächst, die Installation auf einer separaten Maschine durchzuführen.
- Installieren Sie die .exe an denselben Ort.
 - Aufgemerkt: Die OTRS Motive (Themes) der 1.2 sind *nicht* kompatibel mit kleineren Versionen - gar nicht erst versuchen!
- Starten Sie Apache und MySQL neu (auch über das Start Menu möglich).
- Aktualisieren Sie die Datenbank via phpMyAdmin: <http://localhost/phpmyadmin>.
 - Wählen Sie die 'otrs' Datenbank über das Dropdown Menü zur Linken.
 - Klicken Sie dann auf den 'SQL' Seitenreiter zur Rechten.
 - Klicken Sie auf 'Durchsuchen' und wählen Sie die Datei `C:\otrs\otrs\scripts\DBUpdate-to-1.2.mysql.sql`
 - Klicken Sie auf OK.
- Als Alternative können Sie auch dies auf einer Kommandozeile ausführen:


```
C:\> cd /d C:\otrs
C:\otrs\> mysql\bin\mysql.exe -f -u root otrs < otrs\scripts\DBUpdate-to-1.2.mysql.sql
```

Achten Sie nicht so sehr auf die Fehlermeldungen, sie können jede Datenbankversion mit diesem Skript aktualisieren, und oft hat die Datenbank schon manche notwendigen Änderungen inne.
- Bitte berichten Sie Fehler auf <http://bugs.otrs.org/>

1.3.3. Schritt-für-Schritt: Die manuelle Installation

Dieser Abschnitt der Installationsanleitung behandelt eine Schritt-für-Schritt Installation von OTRS unter *Microsoft Windows 32-bit-Versionen*. Natürlich empfehlen wir nach wie vor den Gebrauch des `otrs4win32` Installers. Dies ist der einfachere Weg!

1.3.3.1. Benötigte Software

Auf dem System werden 3 Hauptkomponenten benötigt, ein Webserver (Apache (<http://apache.org/>)), eine Datenbank (MySQL (<http://www.mysql.com/>)) und die Programmiersprache Perl (<http://www.perl.org/>).

Nehmen wir an, die gesamte Installation findet auf Laufwerk D: statt. Sie können aber auch ein beliebiges anderes Verzeichniss wählen. Ersetzen Sie einfach *d:* or *D:* mit Ihrer Auswahl. Im Installationspfad sollten sich jedoch keine Leerzeichen befinden (*c:\program Files\otrs* wäre keine gute Wahl.). .

1.3.3.1.1. PERL installieren

Laden Sie *Perl*, Version 5.6.1 build 633 von <http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl>. Wir empfehlen die MSI-Version zu verwenden.

Installieren Sie Perl unter folgendem Pfad: *D:\Perl*.

1.3.3.1.1.1. Installation zusätzlicher Perl Pakete

Laden sie die OTRS-Win32-Perl-Packages von <ftp://ftp.otrs.org/> (<ftp://ftp.otrs.org/pub/otrs/misc/win32/OTRS-Win32-Perl-Packages.zip>) herunter und enpacken Sie es in das Verzeichnis *D:\Perl*, unter Bewahrung der Verzeichisstruktur.

1.3.3.1.2. Installation des Webservers

OTRS benötigt einen Webserver. Natürlich können Sie Ihren Favoriten auswählen. OTRS läuft mit jedem Webserver, der Perl unterstützt. Getestet wurde OTRS auf Sambar (<http://www.sambar.org>) 5.2 & höher (<http://sambar.robertkehl.de>) und auf Apache (<http://httpd.apache.org/>) , sowohl Version 1.3 (<http://httpd.apache.org/docs/>) und Version 2.0 (<http://httpd.apache.org/docs-2.0/>). *mod_perl* ist nur für Apache verfügbar. Derzeit ist nur *mod_perl-1.0* auf Apache 1.3.27 mit dieser Installationsanleitung abgedeckt.

1.3.3.1.2.1. Apache installieren

Laden Sie den Apache (<http://www.apache.org>) Ihrer Wahl (1.3.27 (http://www.apache.org/dist/httpd/binaries/win32/apache_1.3.27-win32-x86-no_src.msi) und/oder 2.0.43 (http://www.apache.org/dist/httpd/binaries/win32/apache_2.0.43-win32-x86-no_ssl.msi)) herunter und installieren Sie ihn. Wir empfehlen das Installationsverzeichnis **D:\Apache** als Pfad für beide Installationen. Folgende Verzeichnisstruktur ergibt sich:

```
D:\Apache\          --- home of both
D:\Apache\Apache\   --- home of Apache 1.3
D:\Apache\Apache2\  --- home of Apache 2.0
```

Tip: Sie können Apache 1.3 und 2.0 auf dem selben Computer installieren und ausführen, wenn sie den verschiedenen Versionen unterschiedliche Ports zuweisen. Konfigurieren Sie Apache 1.3 auf Port 80 und Apache 2.0 auf Port 81. Darüber hinaus können Sie auch noch den Samba-Webserver auf Port 82 installieren. Ein Webserver reicht natürlich vollkommen aus.

1.3.3.1.3. MySQL installieren

MySQL können Sie unter <http://www.mysql.com/downloads/> herunterladen. Installieren Sie MySQL unter *D:\mysql* und darüber hinaus als Service (Win 2000/XP). Starten Sie MySQL. MySQL Version 4.0 ist derzeit noch nicht ausführlich getestet, jedoch sollten Sie diese Version trotzdem an Stelle von Version 3.23 versuchen.

Komfortabler gestaltet sich die Konfiguration von MySQL über phpMyAdmin (<http://www.phpmyadmin.net>). Hierfür wird zusätzlich PHP (<http://www.php.net>) benötigt. PHP ist immer eine gute Idee ;-)

Das Passwort für MySQL werden wir später ändern.

1.3.3.1.4. OTRS installieren

Und zum Schluss - Das Biest!

Laden sie OTRS als tarball (tar.gz) herunter: <http://otrs.org/download> und entpacken Sie OTRS in folgendes Verzeichnis: `d:\` (Achten Sie darauf, dass die Verzeichnisstruktur erhalten bleibt. Es wird automatisch ein Verzeichnis **otrs** unter `D:\` erstellt.

1.3.3.2. Konfiguration

1.3.3.2.1. Perl konfigurieren

Als erstes müssen wir nun Perl erweitern, da es in der Grundkonfiguration nicht unsere Bedürfnisse erfüllt. Ein paar "packages" fehlen uns noch, aber die haben wir soeben heruntergeladen, oder?

Öffnen sie eine MS-DOS Eingabeaufforderung und geben sie folgendes ein:

```
C:\>cd /d D:\Perl\packages

D:\Perl\packages>install.bat
[...]
```

Die soeben ausgeführte Installationsroutine platziert die Datei `mod_perl.so` im Verzeichnis `\Apache\Apache\modules`. Sollte der Pfad bei Ihrer Installation abweichen, muss dies in der Batch-Datei `install.bat` manuell angepasst werden. Sie können die Datei `mod_perl.so` aber auch manuell in das `modules` Verzeichnis von Apache kopieren.

Ihre Installation von Perl sollte jetzt so aussehen:

```
D:\Perl\packages>ppm query
[...]
```

Archive-Tar	[0.072]
Authen-SASL	[2.03]
Compress-Zlib	[1.16]
Convert-ASN1	[0.16]
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-Socket-SSL	[0.92]
IO-stringy	[2.108]
MD5	[2.02]
MIME-Base64	[2.12]
MIME-tools	[5.411a]
MailTools	[1.58]
Net-DNS	[0.33]
Net_SSLeay.pm	[1.22]
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]
libnet	[1.12]
libwin32	[0.19.1]
libwww-perl	[5.64]

```
mod_perl          [1.27_01-dev]
perl-ldap         [0.26      ]
```

Die Dateien unter D:\Perl\packages können (müssen aber nicht) jetzt gelöscht werden.

Das war's für Perl!

1.3.3.2.2. Apache konfigurieren

Im folgenden wird auf Apache im allgemeinen eingegangen. Der Prozess ist jedoch für die Versionen 1.3 und 2.0 ähnlich.

Öffnen Sie die Apache Konfigurationsdatei **httpd.conf**. Diese finden Sie unter: **d:\Apache\Apache\conf**. Fügen Sie die folgenden Zeilen am Ende der Datei ein und passen Sie sie Ihren Bedürfnissen an:

```
# Kommentieren Sie die beiden folgenden Zeilen für Apache 2.0 aus!
LoadModule perl_module modules/mod_perl.so
AddModule mod_perl.c

### Hinzugefügt für OTRS (http://otrs.org/)

<IfModule mod_alias.c>
    Alias /otrs/ "d:/otrs/bin/cgi-bin/"
    Alias /otrs-web/ "d:/otrs/var/httpd/htdocs/"
    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>

# lädt alle Perl Module
PerlRequire d:/otrs/scripts/apache-perl-startup.pl

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

Nehmen Sie sich etwas Zeit um auch den Rest der Konfigurationsdatei anzupassen. Öffnen sie die Datei d:\otrs\scripts\apache-perl-startup.pl und ändern Sie die folgenden Zeilen:

```
1:
#! D:/Perl/bin/perl
Zeile 10 und 11:
use lib "d:/otrs/";
use lib "d:/otrs/Kernel/cpan-lib";
Zeile 38 und 39:
#use Kernel::System::AuthSession::IPC;
use Kernel::System::AuthSession::DB;
Zeile 46 und 47:
#use Kernel::System::Log::SysLog;
use Kernel::System::Log::File;
```

Das war's für Apache. Der Neustart erfolgt später.

1.3.3.2.3. MySQL konfigurieren

Sie sollten MySQL sichern in dem Sie gute Passwörter für die Benutzer `@%`, `root@%` und `root@localhost` verwenden. Das lässt sich einfach über *phpMyAdmin* machen.

Kommen wir zur Datenbank. Diese können Sie über die Eingabeaufforderung direkt oder über *phpMyAdmin* anlegen und füllen.

```
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 "some-pass" WITH
D:\mysql>mysqladmin -u root -p password reload
```

Wir empfehlen allerdings *phpMyAdmin* unter: <http://localhost/phpMyAdmin>. Die Datenbank lässt sich direkt auf der Startseite erstellen. Wechseln Sie zum Reiter "SQL", und geben Sie folgendes im Textfeld mit der Bezeichnung "or File:" ein: `d:\otrs\scripts\database\otrs-schema.mysql.sql`. Klicken Sie auf OK. Wenn alles reibungslos verlaufen ist, geben Sie noch `d:\otrs\scripts\database\initial_insert.sql` ein und klicken Sie nochmals auf OK.

Führen Sie das folgende Kommando aus:

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

Das war's dann auch für MySQL, die Datenbank, OTRS und den Benutzer `otrs@localhost`.

1.3.3.2.4. OTRS Konfiguration

1.3.3.2.4.1. CONFIG.PM

Wie können Sie OTRS an Ihre Bedürfnisse anpassen? - Erstellen Sie zuerst Ihre Konfigurationsdatei:

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

Neben den üblichen Einstellungen für FQDN usw. fügen Sie bitte die folgenden Zeilen in die Konfigurationsdatei `d:\otrs\Kernel\Config.pm` ein:

```
# -----#
# E-Mail Einstellungen - Sendmail                                     #
# -----#
$Self->{'SendmailModule'} = 'Kernel::System::Email::SMTP';
$Self->{'SendmailModule::Host'} = 'smtp-server.of.your.provider';
$Self->{'SendmailModule::AuthUser'} = 'smtp-username';
$Self->{'SendmailModule::AuthPassword'} = 'smtp-password';
# -----#
# Verzeichnisse - 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';
# -----#
# eigene Konfigurationseinstellungen                               #
# Standardwerte aus Kernel/Config/Defaults.pm                     #
# -----#
$Self->{SessionUseCookie} = 0;
$Self->{SessionModule} = 'Kernel::System::AuthSession::DB';
```

Hinweis zum FQDN: Der sogenannte *Full Qualified Domain Name* Ihres PC's ist genau wie Ihre IP-Adresse eindeutig (bezogen auf das Internet). Wenn sie sich über einen ISP (Internet Service Provider) einwählen, wird diese Kennung jedoch jedes mal neu vergeben. Für OTRS ist das durchaus OK, problematisch wird es allerdings, bei der Verwendung von Hilfsprogrammen.

Um dieses Problem zu umschiffen, können Sie einen *Dynamic DNS services* wie <http://www.dyndns.org/> zu rate ziehen. Wenn Sie einen Client verwenden (bspw. *DirectUpdate* - <http://www.directupdate.net/>) wird Ihre IP-Adresse immer dann an [dyndns.org](http://www.dyndns.org) weitergeleitet, wenn sie sich geändert hat. DynDNS liefert dann einen FQDN (z.B. [gogos-pc.home-ip.net](http://www.gogos-pc.home-ip.net)) zurück. Darüber hinaus können Sie Anfragen an Ihre "Homedomain" (<http://www.thisismydomain.com/>) an Ihren PC zu Hause weiterleiten. Wenn Sie weitere Fragen bezüglich Dynamic DNS services haben, dann scheuen Sie sich nicht unter nachzufragen (<mailto:otrs-win32@robertkehl.de?subject=Dynamic%20DNS>) - Aber jetzt zurück zu OTRS :-)

1.3.3.3. Der erste Start

Starten Sie Ihre Apache-Installation(en) neu. (z.B. über den Apache Monitor in der Taskleiste). Ein kompletter Neustart ist auch keine schlechte Idee, sollte aber in den meisten Fällen nicht nötig sein.

Jetzt kann's los gehen! OTRS in Aktion! - Öffnen Sie <http://localhost/otrs/index.pl> in Ihrem bevorzugten Browser.

Der initiale Benutzername lautet: "root@localhost", das zugehörige Passwort: "root" (Beides OHNE Anführungszeichen)

Important: Es ist eine sehr gute Idee, das Passwort genau jetzt über <http://localhost/otrs/index.pl?Action=AdminUser> zu ändern.

1.3.3.3.1. System Email Adresse & POP3

Die System Email Adresse können Sie hier eingeben: <http://localhost/otrs/index.pl?Action=AdminSystemAddress>. Das ist die Adresse, die Ihrer Kunden später verwenden werden um Ihr OTRS zu erreichen.

Die POP3-Konten für jede System Email Adresse können Sie unter <http://localhost/otrs/index.pl?Action=AdminPOP3> erstellen.

Ihr OTRS ist jetzt fertig konfiguriert! Schauen Sie sich ein wenig um und spielen Sie mit den Test-Tickets um ein gefühl für OTRS zu bekommen.

Schreiben Sie eine Email an Ihre OTRS Installation. Über den folgenden Link <http://localhost/obin/PostMasterPOP3.pl> können Sie diese dann abholen. Dies kann auch noch automatisiert werden.

Ihre Email wird in der Queue *Raw* auftauchen - diese finden Sie hier: <http://localhost/otrs/index.pl?Action=AgentQueueView>. Beantworten Sie Ihre Anfrage.

Tip: Sie sollten auf *keinen* Fall eine System Email Adresse verwenden um Ihr System anzusprechen. Hierdurch entstehen Loops.

1.3.3.3.2. No CRON Jobs on Apache

Beachten Sie, dass Apache CRON nicht unterstützt. Hierzu benötigen Sie bspw. CRONW (<http://cronw.sourceforge.net/>), einen Perl basierten Aufgabenplaner für 32-Bit Windows Versionen. CRONW kann als Service installiert werden. Der Webserver Sambar (<http://sambar.robertkehl.de/?page=whatis>) beinhaltet schon einen Aufgabenplaner.

1.3.3.4. Fertig!

Glückwunsch, Sie haben jetzt hoffentlich OTRS auf Windwos erfolgreich installiert!

Wenn etwas noch nicht ganz klar ist, oder einwandfrei läuft, scheuen Sie sich nicht unter [otrs-win32 at robertkehl.de](mailto:otrs-win32@robertkehl.de) (<mailto:otrs-win32@robertkehl.de>) nachzufragen. Hoffentlich bekommen Sie ein Ticket zurück :-)

Natürlich würden wir uns über Erfolgsmeldungen mehr freuen!

Viel Spaß mit OTRS auf Windows!

Starthilfe: MS-DOS Eingabeaufforderung: Im Verlauf des Dokuments wurden von Zeit zu Zeit Kommandozeileneingaben beschrieben. Diese erfolgen in so genannten *MS-DOS Eingabeaufforderung*. Andere Bezeichnungen sind *Eingabeaufforderung*, *DOS-prompt*, *shell* oder *command line*. Hier die Anleitung für die Eingabeaufforderung unter Windows:

Klicken Sie auf *Start*, *Ausführen* und geben Sie **cmd** in das Dialogfenster ein. Klicken Sie auf *OK*. Eine (üblicherweise schwarze) Box erscheint. Hierin steht etwas in der Art:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

```
C:\>
```

Nun können Sie Befehle eingeben.

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 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>
```

Dieser Befehl bringt Sie in das Verzeichnis von OTRS und zeigt dessen Inhalt.

Beachten Sie bitte, dass es sich nur um eine Beispielausgabe handelt. Auf Ihrem System kann (wird) es vielleicht anders aussehen.

Sie können das Erscheinungsbild Ihrer Eingabeaufforderung auch anpassen. Wählen Sie dazu einfach **Eigenschaften** aus dem Menü aus. Dies ist vielleicht für Farbe und Größe des Fensters sinnvoll.

Sie werden mit OTRS schneller und besser zurechtkommen, wenn Sie sich ein wenig mit Eingabeaufforderungen vertraut machen. Für Anleitungen schauen Sie doch einfach mal unter Google (<http://www.google.com/search?num=100&newwindow=1&q=%22command+line%22+%22learning+resources%22+windows>) nach.

Chapter 2. Trouble Ticket Systeme

In diesem Abschnitt soll kurz die Grundlegende Idee, die hinter Trouble Tickets im Allgemeinen und Trouble Ticket Systemen im Speziellen steht, erläutert werden.

2.1. Ein einfaches Beispiel für ein kleines Trouble Ticket System

Wie könnte ein typisches Einsatzszenario für OTRS aussehen?

Beispiel: Max Mustermann ist Fabrikant und produziert Videorekorder. Da die Programmierung der Videorecorder sehr schwierig ist, wenden sich die Kunden von Herrn Mustermann gerne und häufig an ihn und schreiben eine E-Mail. An manchen Tagen kann Herr Mustermann der E-Mail Flut kaum Herr werden und so kommt es, dass seine Kunden sich einige Zeit gedulden müssen, bis die Antwort mit der rettenden Lösung bei ihnen eintrifft. Dem ein oder anderen Kunden dauert dies jedoch zu lange und so schickt er einfach eine weitere E-Mail an Herrn Mustermann. Diese E-Mails laufen alle einer INBOX, wie sie von fast allen Email-Programmen verwendet wird, auf. Wenn die Anfragewelle besonders groß ist und Herr Mustermann sich außerstande sieht, alle E-Mails noch am gleichen Tag zu beantworten, kommandiert er seine Entwickler Meier und Schulze zur Kundenhotline ab. Sie benutzen dann alle das gleiche System und greifen auf die gleiche INBOX zu. Natürlich haben Meier und Schulze kein Ahnung, dass manch ein Kunde in seiner Not gleich zwei E-Mail verfasst hat. So kommt es vor, dass Meier die erste E-Mail mit einem anderen Ratschlag beantwortet als Schulze der sich im selben Moment der zweiten E-Mail annimmt. Das Ergebnis ist, dass der Kunde unterschiedliche Antworten bekommt! Darüber hinaus hat Herr Mustermann keinen Einblick darüber, wer wann was welchem Kunden gesagt hat, welche Probleme besonders häufig auftreten und wie groß sein gesamter Aufwand für Kundensupport ist!

Aber dann erfährt Herr Mustermann, dass es OTRS gibt und installiert dies um zukünftig Unterstützung im Kundensupport zu haben. Von nun an werden die Hilferufe der Kunden nicht mehr länger an seine private INBOX, sondern direkt an den OTRS Account weitergeleitet. Das System hat eine Schnittstelle zum OTRS Account, so dass alle ankommenden E-Mails automatisch ins System eingespeist werden. Unabhängig ob Herr Mustermann nun gerade zu Tisch ist oder nicht, generiert OTRS eine automatische Antwort und teilt dem Kunden mit, dass seine E-Mail angekommen ist und so schnell wie möglich bearbeitet wird. Dabei wird eine eindeutige Trouble-Ticket-Nummer vergeben. Der Kunde ist glücklich, dass sein Flehen erhört wurde und wartet gespannt auf eine Antwort. Sowohl Herr Mustermann als auch die Entwickler Meier und Schulze können sich über einen beliebigen Internetbrowser und die URL von OTRS auf die Anfragen zugreifen und diese einzeln beantworten. Stellen wir uns vor, dass Herr Schmidt eine Anfrage ans System gestellt hat und Herr Meier diese kurz und knapp beantwortet. Herr Schmidt reicht dies Antwort jedoch nicht aus und so antwortet er auf die Lösungsmail am folgenden Tag. Herr Meier ist jedoch gerade mit der Entwicklung des Nachfolgers beschäftigt, so dass sich Herr Mustermann der Sache annimmt. Über die History Funktion kann er jetzt auf alle vergangenen E-Mails zugreifen und eine ausführlichere Antwort versenden. Herr Schmidt erhält nun die Lösung für sein Problem, weiß aber nicht, dass diese von unterschiedlichen Personen stammt.

Natürlich ist dies nur ein sehr kleiner Einblick in die Funktionalitäten von OTRS. Da Herr Mustermann eine kleine Firma führt erhält er vielleicht nur um die 100 E-Mails am Tag, die er vielleicht noch ganz überschaulich mit seiner INBOX handhaben kann und somit kein OTRS braucht. Wenn aber der neue DVD-Rekorder in die Regale kommt, werden es vielleicht schon 500 oder in ein paar Jahren schon 10.000 E-Mails am Tag sein! - Glückliche, wer ein OTRS hat!

2.2. Was soll ich mir unter Trouble Ticket vorstellen?

Ein Trouble Ticket lässt sich im Wesentlichen mit einem Krankenblatt eines Patienten vergleichen. Bei der erstmaligen Einlieferung im Krankenhaus wird dieses im Zuge der Anamnese neu angelegt. Jeder Arzt trägt nun seine Diagnose, sowie die verordnete Therapie und Medikation ein und dokumentiert deren Erfolg. Das Krankenblatt gibt nun einen schnellen Überblick, gewährleistet eine schnelle Einarbeitung und verhindert eine Mehrfachdosierung. Ist die Krankheit besiegt und der Patient entlassen, wird das Krankenblatt archiviert.

Erfassung

Im OTRS werden solche Trouble Tickets als normale E-Mails behandelt und gespeichert. So ist es möglich, dass Trouble Tickets auch Anhänge enthalten, die dann als normale E-Mailanhänge gespeichert werden können. Die Tickets selbst werden auf der Festplatte gespeichert. Eine Sortierung erfolgt über die deren Meta-Informationen welche zusätzlich in der Datenbank gehalten werden. (Nähere Informationen findet man im Quelltext!)

2.3. Was muss ich mir unter einer Queue vorstellen?

Wer verhandlungssicheres Englisch spricht, für den mag es etwas komisch erscheinen, dass OTRS.de sich die Mühe macht, einen englischen Begriff zu erläutern. Queues sind jedoch für den produktiven Einsatz von OTRS sehr wichtig, da sie quasi als Grundkonzept hinter allem stehen!

Normalerweise werden E-Mails (und wie oben schon erwähnt auch Trouble Tickets die üblicherweise als solche ankommen) in einer INBOX verwaltet. Eine INBOX ist im Prinzip nur eine große Datei, in der einfach nur alle E-Mails aneinandergereiht werden und der neue E-Mails einfach am Ende angefügt werden. Das E-Mail Programm strukturiert diese Datei und bereitet den Inhalt für den Anwender auf.

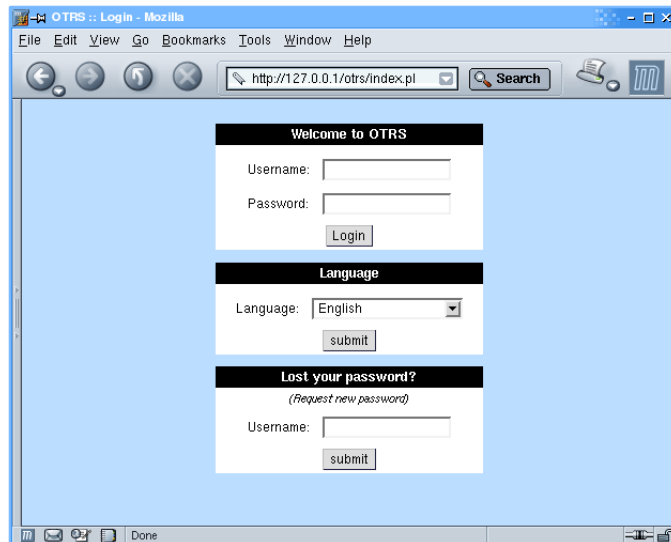
Eine Queue ist ein Mechanismus mit dessen Hilfe viele Tickets gespeichert und verwaltet werden können. Als Anwender ist es völlig unwichtig zu wissen, wo oder wie das Ticket gesichert ist. Wichtig ist nur zu wissen, welcher Queue das Ticket zugeordnet wurde. Anwender sog. Agents können nun Tickets zwischen den Queues verschieben! Warum aber sollte sie das tun? Weil durch das Sortieren der Tickets in unterschiedliche Queues Ordnung geschaffen und somit ein besserer Überblick entsteht. Nehmen wir mal an, das unser Unternehmen täglich 200 E-Mail zu den unterschiedlichsten Themenbereichen erhält. Hier um kümmern sich 3 Teams von Spezialisten. Es wäre nicht besonders sinnvoll oder gar effizient, wenn jeder Spezialist jede E-Mail lesen muss. Zeitverschwendung! Sinnvoller wäre es, einen Agent (den sog. Dispatcher) abzustellen, der die Tickets aus einer Eingangsqueue in die entsprechenden Queues verteilt! Das spart Zeit und somit Geld!

Chapter 3. Erste Schritte

3.1. Als root anmelden und neue Benutzer anlegen

Lassen Sie uns annehmen, dass Sie OTRS bereits installiert haben und natürlich nicht viel Zeit verschwenden und schnell Ergebnisse sehen möchten.

Als erstes müssen Sie Ihren favorisierten Webbrowser starten und auf folgende URL gehen: *http://localhost/otrs/index.pl*

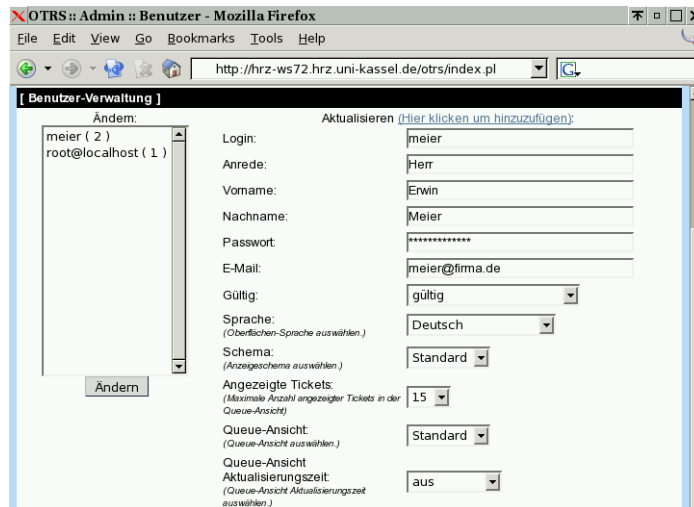


Melden Sie sich als "root@localhost" an, der standardmäßig das Passwort "root" hat. Bitte ändern Sie dieses so bald wie möglich im Admin-Bereich (Natürlich ist es absolut unabhängig von Ihrem normalen Linux root Passwort).

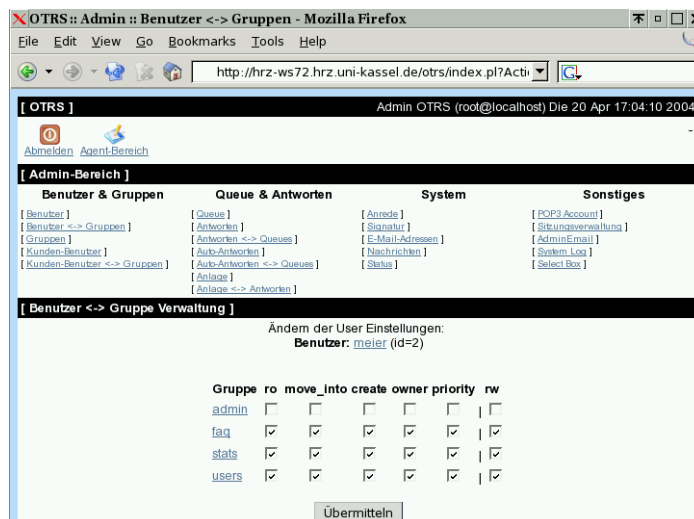
Sie sind nun der root-Benutzer von OTRS. Das bedeutet, dass Sie alles machen können! Sie haben die Macht, Sie sind der Admin des Systems. Normalerweise möchten Sie aber nicht als root arbeiten und natürlich benötigen Sie für jeden Agenten einen eigenen Account. Deshalb gehen Sie als erstes in den Admin-Bereich.



Der Admin-Bereich ist die Zentrale Ihrer Macht. Sie können Benutzer, Gruppen und Queues anlegen und löschen sowie alle möglichen nützlichen Dinge erledigen. Schauen Sie sich ein wenig um und probieren Sie es aus, aber zunächst wollen wir einen neuen Benutzer anlegen.

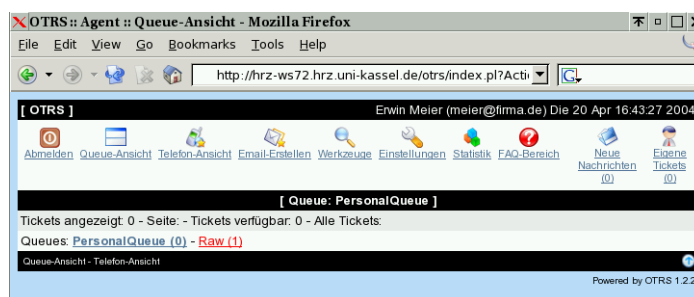


Nun fügen Sie den neuen Benutzer zu den Gruppen "users" und "stats" oder zu einigen anderen Gruppen hinzu.



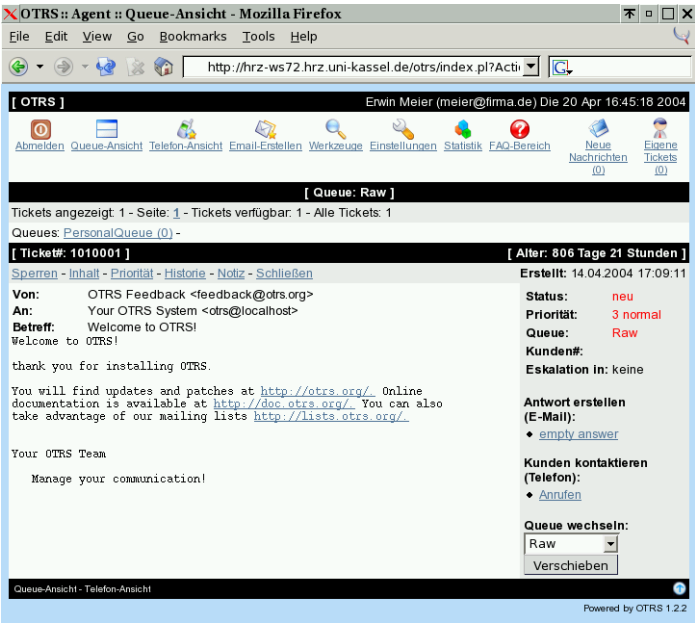
3.2. Als normaler Benutzer anmelden

Nachdem Sie den neuen Benutzer angelegt haben, melden Sie sich ab und mit diesem neuen Benutzer wieder an. Nach dem Login sehen Sie den folgenden Bildschirm:



TODO: folgenden Text je nach Screenshot anpassen

Zwischen dem zweiten und dritten schwarzen Balken (in diesem Beispiel) hat der Benutzer Zugriff auf die verschiedenen Queues. Er/Sie kann in eine Queue wechseln, indem er/sie einfach auf den Link mit dem Queue-Namen klickt.



Chapter 4. Ticket Themen

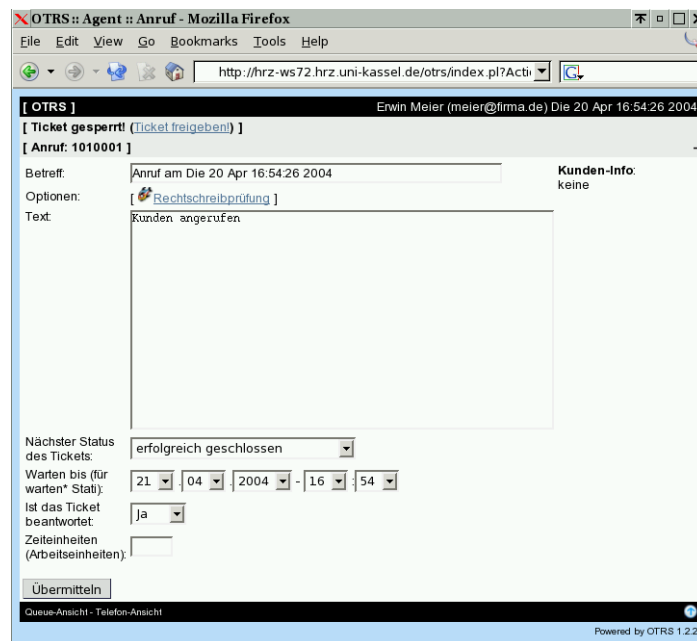
4.1. Ticket via eMail beantworten

Es ist recht einfach, Tickets per eMail zu beantworten. Gehen Sie zur Queue-Übersicht oder zum Ticket-Inhalt und klicken Sie auf einen der aufgeführten Punkte unter "Antwort erstellen (eMail)". Ein neuer Bildschirm wird geöffnet - der Verfassen-Bildschirm. Das Coole daran ist, dass der Verfassen-Bildschirm bereits die entsprechende Standard-Antwort des Punktes enthält! Unter dem Texteingabe-Fenster können Sie weitere Eigenschaften des Tickets verändern, z.B. ob das Ticket mit dem Abschicken der Antwort geschlossen werden oder offen bleiben soll.

4.2. Ticket via Telefon beantworten

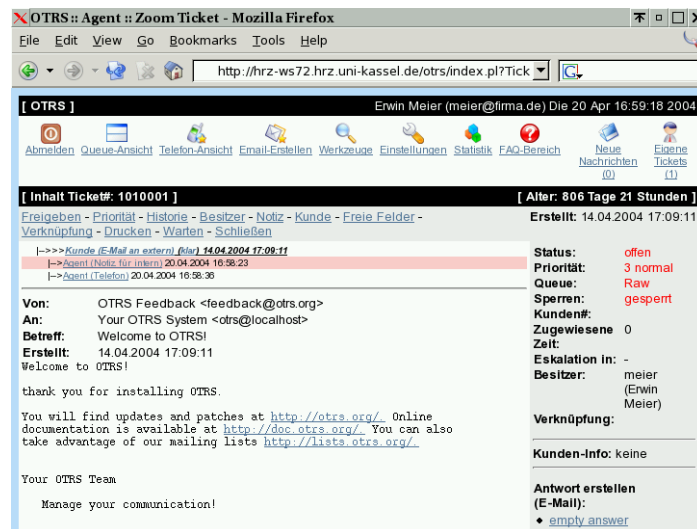
Es ist recht einfach, Tickets per Telefon zu beantworten. Klicken Sie einfach auf "Telefon Anruf". Ein neuer Bildschirm wird geöffnet. Tragen Sie die Telefon-Notizen ein und wählen Sie den neuen Ticket-Status (offen, geschlossen, ...). So können Sie Ihre Antwort im Ticket-System dokumentieren, obwohl Sie dem Kunden keine E-Mail schicken konnten (z.B. wegen

Netzausfall beim Kunden).



4.3. Ticket 'Inhalt'

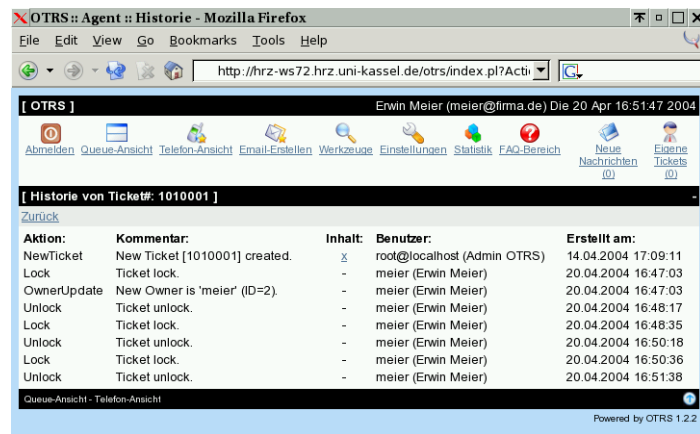
Ticket 'Inhalt' (Ticket Zoom) ist eine detaillierte Ansicht eines Tickets. Das Bildschirmsfoto zeigt Ihnen die Ticket-Daten (Status, Priorität, die Queue des Tickets, den Status der Sperre, Besitzer und die Baumansicht der Artikel). Die Baumansicht zeigt alle Mails und Notizen, die zu diesem Ticket gehören (die gesamte bisherige Kommunikation).



4.4. Ticket-Historie

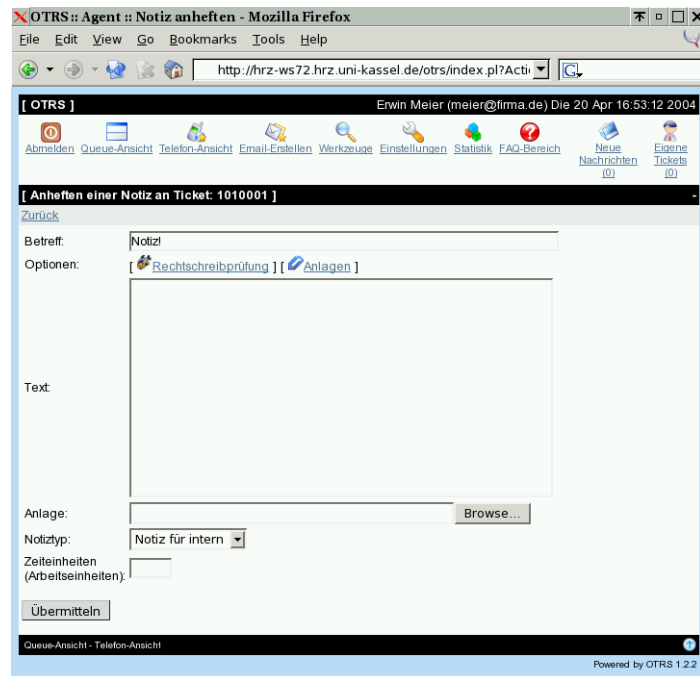
Die Ticket-Historie zeigt Ihnen alle Aktionen dieses Ticket betreffend. Neben allen Mails und Notizen sind auch interne

Aktionen wie z.B. Besitzer-Wechsel und Status-Änderungen aufgeführt.



4.5. Notiz zum Ticket hinzufügen

Jeder Agent hat die Berechtigung, Notizen zu Tickets hinzuzufügen. Vielleicht ist er/sie nicht sicher, um eine qualifizierte Antwort zu geben. Sie können die Art des Tickets (intern, extern, ...) auswählen, der Kunde bekommt keine Info über eine neue Notiz.



4.6. Ticket schliessen

Schliessen Sie Tickets, um einen Fall als erledigt zu markieren.

4.7. Ticket weiterleiten

Leiten Sie Tickets weiter, falls die eMail nicht für Ihr OTRS bestimmt war.

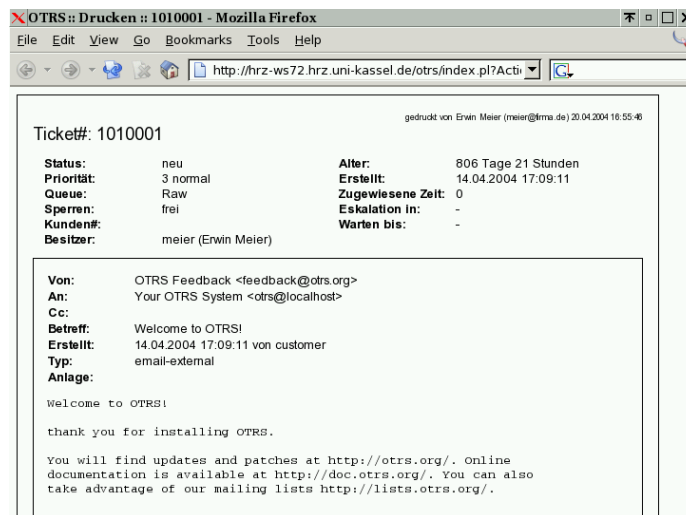
4.8. Ticket-Priorität

Stellen Sie die Priorität eines Tickets ein, falls notwendig.



4.9. Druckansicht

Die Druckansicht eines Tickets. Gedruckt wird die gesamte Kommunikation (alle Mails und Notizen).



Chapter 5. System-eMail-Benachrichtigungen

Dieses Kapitel wird Ihnen mehr Details über die eMail-Benachrichtigungen für Agenten und Kunden zeigen.

5.1. Agenten-Benachrichtigungen

TODO: genaue Bezeichnungen für "Favoriten Queues", "Ticket-Sperre Timeout" und "Voreinstellungen" prüfen

Ein Agent kann Benachrichtigungen über "Neues Ticket", "Folgenachricht", "Ticket in Favoriten-Queue" und "Ticket-Sperre Timeout" erhalten. Um Benachrichtigungen zu erhalten, wählen Sie Ihre gewünschten Queues (genannt "Favoriten Queues") im Agent-Interface unter "Voreinstellungen". Der Benachrichtigungstext kann in "Kernel/Config.pm" definiert werden.

Wichtig: Ein Agent bekommt nur über Änderungen in seinen Favoriten-Queues Benachrichtigungen.

5.1.1. Neues Ticket

Benachrichtigung erfolgt bei neuen Tickets in einer Queue.

5.1.2. Folgenachricht

Benachrichtigung erfolgt bei Folgenachrichten für Tickets, deren Besitzer der Agent ist.

5.1.3. Neues Ticket in Favoriten-Queues

Benachrichtigung erfolgt wenn ein Ticket in eine der Favoriten-Queues verschoben wird.

5.1.4. Ticket-Sperre Timeout

Benachrichtigung erfolgt wenn OTRS die Sperre eines unbeantworteten Tickets aufhebt.

5.2. Kunden-Benachrichtigungen (min 1.1)

Es gibt drei Arten von (automatischen) Kunden-Benachrichtigungen, um die Bearbeitung für den Kunden transparenter zu gestalten. Diese Benachrichtigungen können für Queue einzeln definiert werden (Admin-Bereich). Der Benachrichtigungstext kann in "Kernel/Config.pm" definiert werden.

5.2.1. Ticket-Status

Der Kunde (Absender) bekommt automatisch eine Benachrichtigung wenn sich der Ticket-Status geändert hat.

5.2.2. Ticket-Besitzer

Der Kunde (Absender) bekommt automatisch eine Benachrichtigung wenn der Besitzer eines Tickets geändert hat.

5.2.3. Ticket-Queue

Der Kunde (Absender) bekommt automatisch eine Benachrichtigung wenn sich die Ticket-Queue geändert hat.

Chapter 6. Automatische Antworten

Dieses Kapitel wird Ihnen mehr Details über die automatischen Antworten an Kunden nach Kunden-Aktionen zeigen. Natürlich muss das automatische Antworten via den Admin-Bereich für jede einzelne Queue konfiguriert werden.

6.1. Antwort

Wenn ein neues Ticket erstellt wird, bekommt der Kunde (Absender) die folgende automatische Antwort.

Eine normale Antwort sollte ungefähr so aussehen.

TODO: Standard-Antworten in OTRS bisher nicht übersetzt - hier übersetzen und in OTRS übernehmen?

Thanks for your e-mail. A new ticket has been created.

You wrote:

<OTRS_CUSTOMER_EMAIL[6]>

Your e-mail will be answered by a human asap

Have fun with OTRS! :-)

Your OTRS Team

6.2. Folgenachricht

Wird benutzt wenn ein Kunde eine Folgenachricht via eMail oder Kunden-Interface geschrieben hat.

Eine normale Antwort sollte ungefähr so aussehen.

Thanks for your follow up e-mail

You wrote:

<OTRS_CUSTOMER_EMAIL[6]>

Your e-mail will be answered by a human asap.

Have fun with OTRS!

Your OTRS Team

6.3. Abweisung

Wird benutzt wenn ein Kunde eine Folgenachricht geschrieben hat und die Folgenachricht abgewiesen wird (weil das Ticket geschlossen ist und nicht wieder geöffnet werden kann).

Eine normale Antwort sollte ungefähr so aussehen.

Thanks for your follow up e-mail

You wrote:

<OTRS_CUSTOMER_EMAIL[6]>

Important: Sorry but we can't process your follow up because your ticket is closed. Please create a new ticket.

Have fun with OTRS!

Your OTRS Team

6.4. Geschlossen -> Neues Ticket

Wird benutzt wenn ein Kunde eine Folgenachricht geschrieben hat, die Folgenachricht abgewiesen wird und ein neues Ticket erstellt wird (weil das alte Ticket geschlossen ist und nicht wieder geöffnet werden kann).

Eine normale Antwort sollte ungefähr so aussehen.

Thanks for your follow up e-mail

You wrote:

<OTRS_CUSTOMER_EMAIL[6]>

Important: Sorry but we can't process your follow up because your ticket is closed. A new ticket has been created automatically.

Have fun with OTRS!

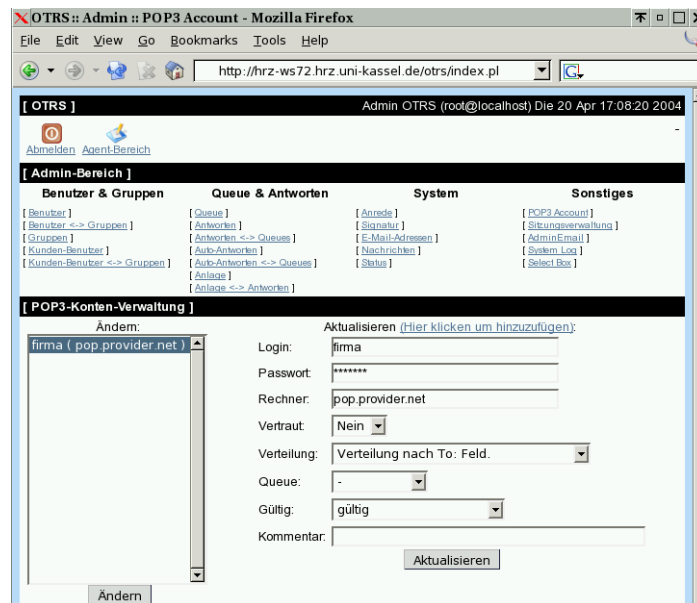
Your OTRS Team

Chapter 7. eMails empfangen

7.1. Via POP3-Konten - der einfache Weg (PostMasterPOP3.pl)

OTRS ist in der Lage, eMails von POP3-Konten zu empfangen.

Konfigurieren Sie Ihre POP3-Konten über das Admin-Interface (POP3 Account).



Starten Sie "bin/PostMasterPOP3.pl" und alle eMails werden in Ihr OTRS geholt.

Ausserdem gibt es einen Beispiel-Cronjob (var/cron/postmaster_pop3.dist), der "bin/PostMasterPOP3.pl" alle 10 Minuten ausführt (s. auch Kapitel Cronjobs).

7.2. Via Kommandozeilen-Programm und z.B. procmail (PostMaster.pl)

OTRS ist in der Lage, eMails über ein Kommandozeilen-Programm (bin/PostMaster.pl) zu empfangen.

Das bedeutet, dass eMails im OTRS angezeigt werden, wenn der MDA (mail delivery agent, z.B. procmail) die eMails nach "bin/PostMaster.pl" durchleitet.

Um "bin/PostMaster.pl" auf der Kommandozeile (ohne MDA) zu testen, benutzen Sie folgendes:

```
shell:~ # cat /opt/otrs/doc/test-email-1.box | /opt/otrs/bin/PostMaster.pl
shell:~ #
```

Wenn die eMail in der Queue-Ansicht angezeigt wird, sind Ihre Einstellungen in Ordnung.

Procmail ist ein sehr bekannter eMail-Filter in der Linux-Umgebung. Es wird wahrscheinlich auf Ihrem System installiert sein. Falls nicht, schauen Sie auf der *procmail Homepage* (<http://www.procmail.org/>).

Um procmail zu konfigurieren (benötigt einen für procmail konfigurierten MDA (z.B. sendmail, postfix, exim oder qmail)), benutzen Sie "~otrs/.procmailrc" und ändern Sie das folgende:

```
SYS_HOME=$HOME
PATH=/bin:/usr/bin:/usr/local/bin
# --
# Pipe all email into the PostMaster process.
# --
:0 :
```

```
| $SYS_HOME/bin/PostMaster.pl
```

Alle an den lokalen OTRS Benutzer gesendeten eMails werden an "bin/PostMaster.pl" durchgeleitet und dann in der Queue-Ansicht angezeigt.

7.2.1. eMails via POP3 oder IMAP und fetchmail für PostMaster.pl empfangen

Um eMails von Ihrem Mailserver via POP3 oder IMAP für den OTRS-Rechner/lokalen OTRS-Benutzer und procmail abzuholen, benutzen Sie fetchmail (<http://www.tuxedo.org/~esr/fetchmail/>). Hinweis: Voraussetzung ist eine funktionierende SMTP-Konfiguration auf dem OTRS-Rechner.

Example 7-1. .fetchmailrc

```
#poll (mailserver) protocol POP3 user (user) password (password) is (localuser)
poll mail.example.com protocol POP3 user joe password mama is otrs
```

Vergessen Sie nicht, .fetchmailrc auf 710 zu setzen ("chmod 710 .fetchmailrc")!

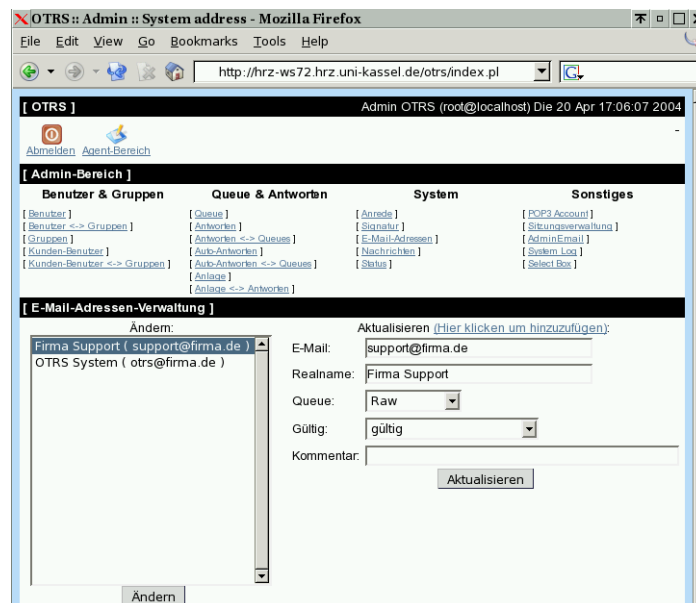
Wenn "fetchmail -a" ausgeführt wird (ggf. via cron), werden alle eMails auf das lokale OTRS-Konto weitergeleitet.

7.3. Filterung/Verteilung über OTRS/Webinterface

OTRS ist in der Lage, ankommende eMails je nach "To" und "Cc" Adresse zu verteilen. Einstellbar über das Admin-Interface.

Beispiel

Fügen Sie eine neue System-eMail-Adresse hinzu (Admin-Bereich --> System-eMail-Adressen).



In diesem Fall werden alle ankommenden eMails (mit To: oder Cc: support@firma.de) in die Queue 'Raw' einsortiert.

7.4. Filterung/Verteilung über OTRS/PostMaster Module (für komplexere Verteilungsszenarien)

Falls Sie die "PostMaster.pl" oder "PostMasterPOP3.pl" Methoden benutzen, können Sie die X-OTRS-Header mit den OTRS/PostMaster-Filtermodulen modifizieren.

Zum Beispiel können Sie die Priorität oder die Queue eines Tickets mit den PostMaster-Filtermodulen einstellen.

Es gibt verschiedene Standard-Filtermodule.

Hinweis: Der Jobname (z.B. \$Self->{'PostMaster::PreFilterModule'}->{'Jobname'}) muss eindeutig sein!

Kernel::System::PostMaster::Filter::Match ist ein Standard-Modul, um einige eMail-Header (z.B. From, To, Subject, ...) zu prüfen und dann den neuen eMail-Header zu setzen (z.B. X-OTRS-Ignore: yes oder X-OTRS-Queue: spam).

```
Kernel/Config.pm
[...]
# Job Name: 1-Match
# (block/ignore all spam email with From: noreply@)
$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} = {
    Module => 'Kernel::System::PostMaster::Filter::Match',
    Match => {
        From => 'noreply@',
    },
    Set => {
        'X-OTRS-Ignore' => 'yes',
    },
};
# Job Name: 2-Match
# (sort emails with From: sales@example.com and Subject: **ORDER**
# into queue 'Order')
$Self->{'PostMaster::PreFilterModule'}->{'2-Match'} = {
    Module => 'Kernel::System::PostMaster::Filter::Match',
    Match => {
        To => 'sales@example.com',
        Subject => '**ORDER**',
    },
    Set => {
        'X-OTRS-Queue' => 'Order',
    },
};
[...]
```

Kernel::System::PostMaster::Filter::CMD ist ein Standard-Modul, um die eMails an ein externes Kommando zu leiten und falls das Ergebnis aus STDOUT true ist, den neuen eMail-Header zu setzen (z.B. X-OTRS-Ignore: yes oder X-OTRS-Queue: spam).

```
Kernel/Config.pm
[...]
# Job Name: 5-SpamAssassin
# (SpamAssassin example setup, ignore spam emails)
$Self->{'PostMaster::PreFilterModule'}->{'5-SpamAssassin'} = {
    Module => 'Kernel::System::PostMaster::Filter::CMD',
    CMD => '/usr/bin/spamassassin | grep -i "X-Spam-Status: yes"',
    Set => {
        'X-OTRS-Ignore' => 'yes',
    },
};
[...]
```

Natürlich ist es auch möglich, eigene PostMaster-Filtermodule zu entwickeln.

Ausserdem gibt es eine Liste aller möglichen X-OTRS-Header in doc/X-OTRS-Headers.txt.

7.5. Filterung/Verteilung durch Procmail (für komplexere Verteilungsszenarien)

Falls Sie die PostMaster.pl Methode benutzen, können Sie X-OTRS-Header zur eMail hinzufügen und PostMaster.pl wird diesen Header dazu benutzen, die eMail an PostMaster.pl zu leiten. TODO: ergibt für mich keinen Sinn (postmaster.pl piped die Mail an sich selbst???)

Der X-OTRS-Queue Mail-Header

Der X-OTRS-Queue Mail-Header wird durch OTRS geprüft und OTRS wird eMails in die entsprechende Queue stellen. Procmail und fromail können dazu genutzt werden, ein mächtiges Verteilungs-Werkzeug zu erstellen.

Beispiele

Die folgenden Beispiele sind von procmailex man-page kopiert. Es ist eine gute Idee, sich diese einmal anzuschauen. Natürlich haben wir den Text ein wenig geändert (um es an die Queue-Idee anzupassen).

Alle eMails von der scuba-dive Maillingliste in die scuba Queue aussortieren.

```
:0 fhw :
* ^TOscuba
| formail -I "X-OTRS-Queue: scuba"
```

Alle eMails von Peter über Compiler an die william Queue weiterleiten

```
:0 fhw :
* ^From.*peter
* ^Subject:.*compilers
| formail -I "X-OTRS-Queue: william"
```

Und hier ein letztes Beispiel, die gesamte .procmailrc.

Example 7-2. .procmailrc

```
# --
# .procmailrc - procmailrc of the OTRS user
# Copyright (C) 2001-2002 Martin Edenhofer (martin+code at otrs.org)
# --
# $Id: receiving_email.sgml,v 1.5 2004/07/29 10:29:11 daniel Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --

SYS_HOME=$HOME

PATH=/bin:/usr/bin:/usr/local/bin
MONTHFOLDER='date +%Y-%m'
YEARFOLDER='date +%Y'
LOGFILE=$SYS_HOME/var/log/procmail-$MONTHFOLDER.log
VERBOSE=on

# --
# Remove all X-OTRS Header (allow this only for trusted email)
# e. g. from *@example.com
# --
:0 fhw :
* !^From.*@example.com
| grep -vi '^X-OTRS-'

# --
# Examples for queue presorting.
# --

:0 fhw :
* ^List-Id:.*OpenAntiVirus
| formail -I "X-OTRS-Queue: OpenAntiVirus"

:0 fhw :
* ^Sender:.*example.com
| formail -I "X-OTRS-Queue: example"
```

```

:0 fhw :
* TO:.*BUGTRAQ
| formail -I "X-OTRS-Queue: BUGTRAQ"

# --
# Backup of all incoming emails.
# It's always better to have a backup of all incoming emails!
# --
:0 c :
$SYS_HOME/var/INBOX.Backup.$MONTHFOLDER

# --
# Pipe all email into the PostMaster process.
# --
:0 :
| $SYS_HOME/bin/PostMaster.pl

# --
# spool all the rest (which the PostMaster.pl can't process!)
# If the database is down or the PostMaster.pl exit was not '0'!
# --
:0 :
$SYS_HOME/var/spool/.

# --
# end of .procmailrc
# --

```

Bitte schauen Sie in die procmailex man-page für mehr Beispiele.

Ausserdem gibt es eine Liste aller möglichen X-OTRS Header in doc/X-OTRS-Headers.txt.

7.6. Beispiel für Filterung/Verteilung durch Procmail und ein Webformular

Dies ist ein Beispiel für ein Webformular, dass eine eMail für OTRS erzeugt. Sie finden dieses Perl-Skript in "\$OTRS_HOME/scripts/webform.pl".

Sie haben ein Thema, Von, eMail, Betreff und Nachrichten Feld.

Ändern Sie die Einstellungen für das Webformular:

```

# --
# web form options
# --
my $Ident = 'ahfiw2Fw32r230ddd12foeo3r';
# sendmail location and options
my $Sendmail = '/usr/sbin/sendmail -t -i -f ';
# email where the emails of the form will send to
my $OTRSEmail = 'otrs-system@example.com';
# topics and dest. queues
my %Topics = (
    # topic => OTRS queue
    'Info' => 'info',
    'Support' => 'support',
    'Bugs' => 'bugs',
    'Sales' => 'sales',
    'Billing' => 'billing',
    'Webmaster' => 'webmaster',
);

```

Achten Sie darauf, dass die benutzte \$OTRSEmail und die OTRS Queue in Ihrem OTRS existieren.

Als nächstes ändern Sie die OTRS .procmailrc von:

```

# --
# Remove all X-OTRS Header (allow this only for trusted email)
# e. g. from *@example.com
# --

```

```
:0 fhw :  
* !^From.*@example.com  
| grep -vi '^X-OTRS-'
```

auf:

```
# --  
# Remove all X-OTRS Header (allow this only for trusted email)  
# just not emails with "X-OTRS-Ident: ahfiw2Fw32r230dddl2foeo3r" header!  
# --  
:0 fhw :  
* !^X-OTRS-Ident: ahfiw2Fw32r230dddl2foeo3r  
| grep -vi '^X-OTRS-'
```

Wenn eine eMail durch webform.pl generiert und an \$OTRSEmail gesendet wird, wird sie an das entsprechende Thema, bzw. Queue verteilt.

Chapter 8. eMails versenden

8.1. Via Sendmail (Standard)

OTRS ist in der Lage, eMails via Sendmail (z.B. Sendmail (<http://www.sendmail.org/>), Postfix (<http://www.postfix.org/>), Qmail (<http://www.qmail.org>) or Exim (<http://www.exim.org>)) zu versenden. Die Standard-Konfiguration sollte gleich ohne Probleme funktionieren.

Es gibt die folgenden Konfigurations-Optionen in "Kernel/Config.pm":

```
# SendmailModule
# (Where is sendmail located and some options.
# See 'man sendmail' for details.)
$Self->{'SendmailModule'} = 'Kernel::System::Email::Sendmail';
$Self->{'SendmailModule::CMD'} = '/usr/sbin/sendmail -t -i -f ';
```

8.2. Via SMTP relay/smarthost (min. OTRS 1.1)

OTRS ist in der Lage, eMails via SMTP (Simple Mail Transfer Protocol / RFC 821 (<http://www.ietf.org/rfc/rfc821.txt>)) zu versenden. Hauptsächlich auf Nicht-Unix-Plattformen (z.B. Win32) benutzt.

Es gibt die folgenden Konfigurations-Optionen in "Kernel/Config.pm":

```
# SendmailModule
$Self->{'SendmailModule'} = 'Kernel::System::Email::SMTP';
$Self->{'SendmailModule::Host'} = 'mail.example.com';
$Self->{'SendmailModule::AuthUser'} = "";
$Self->{'SendmailModule::AuthPassword'} = ";
```


Chapter 9. Benutzer (Agenten)

Dieses Kapitel wird Ihnen mehr Details über die Systembenutzer (Agenten) zeigen.

9.1. Wie es funktioniert

Sie benötigen Agenten, die sich um all die Tickets kümmern.

9.1.1. Admin-Interface

Editieren Sie Ihre Agenten über das Admin-Interface.

9.2. Benutzer-Backend

Es gibt kein existierendes Benutzer-Backend. Im Moment muss sich der OTRS Agent in der otrs-Datenbank befinden.

Benutzer-Backends (für DB und LDAP) sind für OTRS 2.0 geplant.

9.3. Benutzer-Authentifizierung-Backend

9.3.1. Datenbank (Standard)

Das standardmäßige Benutzer-Authentifizierung-Backend ist die otrs-Datenbank.

```
[Kernel/Config.pm]
# This is the auth. module against the otrs db
$Self->{'AuthModule'} = 'Kernel::System::Auth::DB';
[...]
```

9.3.2. LDAP

Falls ein LDAP-Verzeichnis verfügbar ist, können Sie das LDAP-Authentifizierung-Backend benutzen. Dieses Modul ist nur-lesend (d.h. es kann nicht in Ihr LDAP-Verzeichnis schreiben - dies sollte nur für Ihren Verzeichnis-Manager möglich sein), so dass Sie keine Benutzer via Admin-Interface erstellen oder aktualisieren können.

```
[Kernel/Config.pm]
# This is an example configuration for an LDAP auth. backend.
# (take care that Net::LDAP is installed!)
$Self->{'AuthModule'} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'AuthModule::LDAP::UID'} = 'uid';

# Check if the user is allowed to auth in a posixGroup
# (e. g. user needs to be in a group xyz to use otrs)
# $Self->{'AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posixGroups,dc=example,dc=com';
# $Self->{'AuthModule::LDAP::AccessAttr'} = 'memberUid';

# The following is valid but would only be necessary if the
# anonymous user do NOT have permission to read from the LDAP tree
$Self->{'AuthModule::LDAP::SearchUserDN'} = '';
$Self->{'AuthModule::LDAP::SearchUserPw'} = '';

# in case you want to add always one filter to each ldap query, use
# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)'
# $Self->{'AuthModule::LDAP::AlwaysFilter'} = '';

# in case you want to add a suffix to each login name, then
```

```

# you can use this option. e. g. user just want to use user but
# in your ldap directory exists user@domain.
#   $Self->{'AuthModule::LDAP::UserSuffix'} = '@domain.com';

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
#   $Self->{'AuthModule::LDAP::Params'} = {
#       port => 389,
#       version => 3,
#   };
[...]
```

LDAP-Einträge sollen konform zum PosixAccount und inetOrgPerson Schema sein. Ein Beispiel-Eintrag könnte folgendermaßen aussehen:

```

dn: uid=lester,ou=user,dc=example,dc=com
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: officePerson
uid: lester
cn: Lester Adamas
userPassword: {crypt}X5/DBrWPOQQaI
gecos: Lester
loginShell: /bin/csh
uidNumber: 10
gidNumber: 10
homeDirectory: /home/lester
sn: Adams
givenName: Lester
mail: lester@example.com
preferredLanguage: fr
comment: technical support
```

Bitte beachten Sie, dass Sie in Version 1.x einen DB-Eintrag für den LDAP-Benutzer anlegen müssen. Dies wird in den Versionen > 1.1.x automatisch beim ersten Login erledigt.

TODO: soll es wirklich ein Kleiner-Zeichen sein?

Dies sind die Standard-Konfigurations-Optionen, um die LDAP-Attribute auf die Datenbank zu mappen:

```

# UserSyncLDAPMap
# (map if agent should create/synced from LDAP to DB after login)
$Self->{UserSyncLDAPMap} = {
    # DB -> LDAP
    Firstname => 'givenName',
    Lastname => 'sn',
    Email => 'mail',
};
```

9.3.3. HTTPBasicAuth

Falls Sie eine "single sign on"-Lösung benutzen, benutzen Sie http basic authentication (für alle Ihre Systeme) und benutzen Sie das HTTPBasicAuth Modul (kein OTRS-Login benötigt!).

```

[Kernel/Config.pm]
# This is the auth. module against $ENV{REMOTE_USER} (apache
# http-basic-auth)
$Self->{'AuthModule'} = 'Kernel::System::Auth::HTTPBasicAuth';
# Note:
# If you use this module, you should use as fallback the following
# config settings if user isn't login through apache ($ENV{REMOTE_USER})
$Self->{LoginURL} = 'http://host.example.com/not-authorized-for-otrs.html';
$Self->{LogoutURL} = 'http://host.example.com/thanks-for-using-otrs.html';
```

```
[...]
```

9.3.4. Radius

Authentifizierung gegen einen Radius-Server.

```
[Kernel/Config.pm]
# This is example configuration to auth. agents against a radius server
$Self->{'AuthModule'} = 'Kernel::System::Auth::Radius';
$Self->{'AuthModule::Radius::Host'} = 'radiushost';
$Self->{'AuthModule::Radius::Password'} = 'radiussecret';
[...]
```


Chapter 10. Kunden (min. OTRS 1.1)

OTRS ist in der Lage, mit verschiedenen Kundendaten (insbesondere Login, eMail, Telefon) umzugehen. Diese Informationen können im Agenten-Interface angezeigt werden und können für das Kunden-Interface benutzt werden (Benutzer und Authentifizierung).

10.1. Wie es funktioniert

Die benutzten/angezeigten Kundendaten sind frei konfigurierbar (s. Backend), aber es gibt drei benötigte Optionen: BenutzerLogin, BenutzerEmail und BenutzerKundenID.

10.1.1. Agenten-Interface

Wenn Sie die Kundendaten (z.B. Firma, Name, eMail, ...) in Ihrem Agenten-Interface anzeigen möchten, benutzen Sie die folgenden Konfigurations-Optionen.

```
[Kernel/Config.pm]
# ShowCustomerInfo*
# (show customer user info on Compose (Phone and Email), Zoom and Queue view)
$Self->{ShowCustomerInfoCompose} = 1;
$Self->{ShowCustomerInfoZoom} = 1;
$Self->{ShowCustomerInfoQueue} = 0;
[...]
```

10.1.2. Admin-Interface

Über das Admin-Interface können Sie die Kundendaten editieren (<http://yourhost/otrs/index.pl?Action=AdminCustomerUser>).

10.1.3. Kunden-Interface

Die Kundendaten für das Kunden-Interface (<http://yourhost/otrs/customer.pl>) werden für Login und Authentifizierung benutzt.

10.2. Kundenbenutzer-Backend

Es existieren zwei Kundenbenutzer-Backends, DB und LDAP. Falls Sie bereits ein Kunden-Verzeichnis (z.B. SAP, ...) haben, ist es natürlich möglich, dafür ein eigenes Backend zu schreiben.

Notiz: Es ist auch möglich mehr als einen Kunden-Daten Topf zu definieren. Einfach eine weitere Konfig Option wie "CustomerUser" mit einer Nummer zwischen 1 und 10. Zum Beispiel "CustomerUser1".

10.2.1. Datenbank (Standard)

Dies ist ein Beispiel für ein Datenbank-Backend.

```
[Kernel/Config.pm]
# CustomerUser
# (customer user database backend and settings)
$Self->{CustomerUser} = {
    Name => 'Datenbank Quelle',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # if you want to use an external database, add the
        # required settings
        DSN => 'DBI:odbc:yourdsn',
        DSN => 'DBI:mysql:database=customerdb:host=customerdbhost',
        User => "",
        Password => "",
    }
}
```

```

        Table => 'customer_user',
    },
    # customer uniq id
    CustomerKey => 'login',
    # customer #
    CustomerID => 'customer_id',
    CustomerValid => 'valid_id',
    CustomerUserListFields => ['login', 'first_name', 'last_name', 'email'],
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserPostMasterSearchFields => ['email'],
    CustomerUserNameFields => ['salutation', 'first_name', 'last_name'],
    Map => [
        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown, required, storage-type, http-link
        [ 'UserSalutation', 'Salutation', 'salutation', 1, 0, 'var' ],
        [ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var' ],
        [ 'UserLastname', 'Lastname', 'last_name', 1, 1, 'var' ],
        [ 'UserLogin', 'Login', 'login', 1, 1, 'var' ],
        [ 'UserPassword', 'Password', 'pw', 0, 1, 'var' ],
        [ 'UserEmail', 'Email', 'email', 0, 1, 'var' ],
    #      [ 'UserEmail', 'Email', 'email', 1, 1, 'var', '$Env{"CGIHandle"}?Action=\
AgentCompose&ResponseID=1&TicketID=$Data{"TicketID"}&ArticleID=$Data{"ArticleID"}' ],
        [ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var' ],
        [ 'UserComment', 'Comment', 'comment', 1, 0, 'var' ],
        [ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int' ],
    ],
};
[...]
```

Falls Sie Ihre Kundendaten anpassen möchten, ändern Sie die Tabellenspalten oder fügen Sie weitere hinzu:

```
ALTER TABLE customer_user ADD phone VARCHAR (250);
```

Danach fügen Sie Ihre eigenen Spalten dem MAP Array hinzu:

```

    # var, frontend, storage, shown, required, storage-type, http-link
    [ 'UserPhone', 'Phone', 'phone', 1, 0, 'var' ],
```

Natürlich können Sie all diese Kundeninformationen dann auch über das Admin-Interface pflegen.

MultiCustomerIDs: Es ist auch möglich mehr als nur eine Kunden-Nummer einen Kunden zu geben. Z. B. wenn ein Kunde auf Tickets anderer Kunden zugreifen muss.

Einfach eine Tabellenspalten hinzufügen in der die Kunden-Nummern eingetragen werden (wie "CustomerID1, CustomerID2, CustomerID3"):

```
ALTER TABLE customer_user ADD customer_ids VARCHAR (250);
```

Nun einen weiteren Eintrag in der CustomerUser MAP Aarray In Kernel/Config.pm hinzufügen:

```

    # var, frontend, storage, shown, required, storage-type, http-link
    [ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var' ],
```

Natürlich können Sie all die Multi-Kunden-Nummern dann auch über das Admin-Interface pflegen.

10.2.2. LDAP

Falls Sie ein existierendes LDAP-Verzeichnis mit Ihren Kunden-Benutzern haben, sind Sie in der Lage, dieses auch in OTRS zu nutzen.

Dies ist ein Beispiel für ein LDAP-Backend.

```

[Kernel/Config.pm]
# CustomerUser
# (customer user ldap backend and settings)
$Self->{CustomerUser} = {
    Name => 'LDAP Quelle',
    Module => 'Kernel::System::CustomerUser::LDAP',
```



```

Params => {
    # ldap host
    Host => 'bay.csuhayward.edu',
    # ldap base dn
    BaseDN => 'ou=seas,o=csuh',
    # search scope (one|sub)
    SSCOPE => 'sub',
    # The following is valid but would only be necessary if the
    # anonymous user does NOT have permission to read from the LDAP tree
    UserDN => "",
    UserPw => "",
    # in case you want to add always one filter to each ldap query, use
    # this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)'
    AlwaysFilter => "",
    # Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
    Params => {
        port => 389,
        version => 3,
    },
},
# customer uniq id
CustomerKey => 'uid',
# customer #
CustomerID => 'mail',
CustomerUserListFields => ['uid', 'cn', 'mail'],
CustomerUserSearchFields => ['uid', 'cn', 'mail'],
CustomerUserPostMasterSearchFields => ['mail'],
CustomerUserNameFields => ['givenname', 'sn'],
Map => [
    # note: Login, Email and CustomerID needed!
    # var, frontend, storage, shown, required, storage-type
    [ 'UserSalutation', 'Title', 'title', 1, 0, 'var' ],
    [ 'UserFirstname', 'Firstname', 'givenname', 1, 1, 'var' ],
    [ 'UserLastname', 'Lastname', 'sn', 1, 1, 'var' ],
    [ 'UserLogin', 'Login', 'uid', 1, 1, 'var' ],
    [ 'UserEmail', 'Email', 'mail', 1, 1, 'var' ],
    [ 'UserCustomerID', 'CustomerID', 'mail', 0, 1, 'var' ],
    [ 'UserPhone', 'Phone', 'telephonenumber', 1, 0, 'var' ],
    [ 'UserAddress', 'Address', 'postaladdress', 1, 0, 'var' ],
    [ 'UserComment', 'Comment', 'description', 1, 0, 'var' ],
],
};
[...]
```

Falls Sie Ihre Kundendaten anpassen möchten, fügen Sie Ihre neuen Punkte dem MAP Array hinzu (oder entfernen Sie nicht benötigte Punkte):

```

# var, frontend, storage, shown, required, storage-type, http-link
[ 'UserOrganisation', 'Organisation', 'ou', 1, 0, 'var' ],
```

MultiCustomerIDs: Es ist auch möglich mehr als nur eine Kunden-Nummer einen Kunden zu geben. Z. B. wenn ein Kunde auf Tickets anderer Kunden zugreifen muss.

Einfach eine Feld im LDAP Baum für die Multi-Kunden-Nummern auswählen in welche die Kunden-Nummern eingetragen werden (wie "CustomerID1, CustomerID2, CustomerID3"):

Nun einen weiteren Eintrag in der CustomerUser MAP Aarray In Kernel/Config.pm hinzufügen:

```

# var, frontend, storage, shown, required, storage-type, http-link
[ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var' ],
```

Natürlich müssen die Multi-Kunden-Nummern vom LDAP Admin gepflegt werden.

10.3. Kunden-Authentifizierungs-Backend

10.3.1. Datenbank (Standard)

Das Standard-Kunden-Authentifizierungs-Backend ist die otrs Datenbank mit Ihren Kunden-Benutzern, die über das Admin-Interface oder Kunden-Interface (Konto anlegen) angelegt werden.

```
[Kernel/Config.pm]
# This is the auth. module against the otrs db
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';
[...]
```

10.3.2. LDAP

Falls ein LDAP-Verzeichnis mit Ihren Kunden-Benutzern verfügbar ist, können Sie das Kunden-LDAP-Authentifizierungs-Backend benutzen. Dieses Modul ist nur-lesend (d.h. es kann nicht in Ihr LDAP-Verzeichnis schreiben - dies sollte nur dem Verzeichnis-Manager möglich sein), so dass Sie keine Kunden über das Admin- oder Kunden-Interface anlegen oder ändern können.

```
[Kernel/Config.pm]
# This is an example configuration for an LDAP auth. backend.
# (take care that Net::LDAP is installed!)
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::LDAP';
$Self->{'Customer::AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';

# Check if the user is allowed to auth in a posixGroup
# (e. g. user needs to be in a group xyz to use otrs)
# $Self->{'Customer::AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posixGroups,dc=example,dc=com';
# $Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid';

# The following is valid but would only be necessary if the
# anonymous user do NOT have permission to read from the LDAP tree
$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = "";
$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = "";

# in case you want to add always one filter to each ldap query, use
# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)'
# $Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = "";

# in case you want to add a suffix to each customer login name, then
# you can use this option. e. g. user just want to use user but
# in your ldap directory exists user@domain.
# $Self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
# $Self->{'Customer::AuthModule::LDAP::Params'} = {
#     port => 389,
#     version => 3,
# };
[...]
```

10.3.3. HTTPBasicAuth

Falls Sie eine "single sign on"-Lösung für Ihre Kunden benutzen, benutzen Sie http basic authentication (für alle Ihre Systeme) und benutzen Sie das HTTPBasicAuth Modul (kein OTRS-Login benötigt!).

```
[Kernel/Config.pm]
# This is the auth. module against $ENV{REMOTE_USER} (apache
# http-basic-auth)
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::HTTPBasicAuth';
```

```
# Note:
# If you use this module, you should use as fallback the following
# config settings if user isn't login through apache ($ENV{REMOTE_USER})
$Self->{CustomerPanelLoginURL} = 'http://host/not-authorized-for-otrs.html';
$Self->{CustomerPanelLogoutURL} = 'http://host/thanks-for-using-otrs.html';
[...]
```

10.3.4. Radius

Authentifizierung gegen einen Radius-Server.

```
[Kernel/Config.pm]
# This is example configuration to auth. agents against a radius server
$Self->{'Customer::AuthModule'} = 'Kernel::System::Auth::Radius';
$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';
$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';
[...]
```

10.4. Kunden-Selbstanmeldung anpassen

Es ist möglich, die Kunden-Selbstanmeldung für neue Kunden über "customer.pl" anzupassen. Somit können Sie mehr optionale oder benötigte Felder (z.B. Adresse, Ort) hinzufügen.

In unserem Beispiel wollen wir ein benötigtes Adress-Feld hinzufügen.

10.4.1. Vorlage

Editieren Sie "Kernel/Output/HTML/Standard/CustomerCreateAccount.dtl" und fügen Sie Ihre gewünschten Felder hinzu. Wir möchten ein Adress-Feld, also fügen Sie folgendes hinzu:

```
[...]
<tr>
  <td>$Text{"Address"}: </td>
  <td> <input type="text" name="Address" value="$Data{"UserAddress"}" size="35" maxlength="50"></td>
</tr>
[...]
```

10.4.2. Kunden-Mapping

Zusätzlich müssen Sie einen "Adress"-Eintrag zu Ihrem Kunden-Mapping hinzufügen. Fügen Sie "CustomerUser" (Kunden-Mapping) aus "Kernel/Config/Defaults.pm" zu Ihrer "Kernel/Config.pm" hinzu und ergänzen Sie das Adress-Feld. Danach sollte es ungefähr so aussehen:

```
[Kernel/Config.pm]
$Self->{CustomerUser} = {
  Name => 'Interne Kunden',
  Module => 'Kernel::System::CustomerUser::DB',
  Params => {
    Table => 'customer_user',
  },
  Map => [
    # note: Login, Email and CustomerID needed!
    # var, frontend, storage, shown, required, storage-type, link
    [ 'UserSalutation', 'Salutation', 'salutation', 1, 0, 'var' ],
    [ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var' ],
    [ 'UserLastname', 'Lastname', 'last_name', 1, 1, 'var' ],
    # our new option
    [ 'UserAddress', 'Address', 'address', 1, 1, 'var' ],
  ],
}
```

```
[ 'UserLogin', 'Login', 'login', 1, 1, 'var' ],
[ 'UserPassword', 'Password', 'pw', 0, 1, 'var' ],
[ 'UserEmail', 'Email', 'email', 1, 1, 'var' ],
[ 'UserCustomerID', 'CustomerID', 'customer_id', 1, 1, 'var' ],
[ 'UserComment', 'Phone', 'comment', 1, 0, 'var' ],
[ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int' ],
],
Key => 'login',
CustomerID => 'customer_id',
};
[...]
```

10.4.3. Kunden-Tabelle anpassen

Ausserdem müssen wir eine neue Spalte zu unserer "customer_user" Tabelle hinzufügen.

```
mysql> ALTER TABLE customer_user ADD address VARCHAR (200);
Query OK, 0 rows affected (0.21 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

Das ist alles - Starten Sie Ihren Webserver neu (falls Sie mod_perl benutzen) und gehen Sie auf "<http://otrs.example.com/otrs/customer.pl>", um es zu testen.

Chapter 11. LDAP Integration

OTRS kann an mehreren Stellen ein LDAP Verzeichnis als Quelle nutzen. Die folgende Anleitung beschreibt, wie ein Active Directory Server mit OTRS genutzt wird. Das Beispiel dient auch dem grundsätzlichen Verständnis, wie gegen einen LDAP-Server gebunden wird.

11.1. Active Directory

Diese Anleitung beschreibt, wie OTRS mit Microsoft's Active Directory (im folgenden AD genannt) integriert wird. Via LDAP werden sowohl Agenten wie auch Kunden authentifiziert, und die Kunden-informationen beschafft werden.

Wir nehmen an, dass OTRS auf einem Linux Server läuft und dass Sie wissen, wie mit dem AD und dessen administrativen Werkzeugen umzugehen ist.

Die Nennung spezifischer Produkte und Marken erfolgt ohne weitergehende Absicht.

Fühlen Sie sich aufgerufen, vor Änderungen am System ein desselben Backup vorzunehmen, und dies auch regelmäßig zu tun.

11.1.1. Vorbereitungen

Als einzige Voraussetzung für den OTRS Server ergibt sich ein instal.liertes Net::LDAP. Das Modul kann von <http://ldap.perl.org/> heruntergeladen werden.

Zudem ist es eine gute Idee, zunächst einen OTRS-Adminzugänge in der Datenbank anzulegen, oder Sie werden sich nicht einloggen können. Erstellen Sie einen Admin, dessen Name dem im AD entspricht.

Um die LDAP-Fähigkeiten des OTRS mit einem AD nutzen zu können, benötigen Sie mindestens einen AD-Controller. Dieser muss als sog. Global Catalog Server (GCS) fungieren, um Authentifizierungen gegen das AD zu erlauben.

Sie können das AD sowohl von Windows 2000 als auch von Windows 2003 aus nutzen.

Details, die während der Konfiguration von OTRS und LDAP benötigt werden.

- Hostname des AD Servers
- Basis-DN Ihres AD-Baumes
- Username und Passwort, mit denen die LDAP-Abfragen durchgeführt werden (AD erlaubt keine anonymen Abfragen!)

11.1.2. Konfiguration

11.1.2.1. Konfiguration des Active Directory

Erstellen Sie im AD einen neuen Benutzer mit dem werkzeug "AD Benutzer und Computer". Der Benutzer muss lediglich die Rechte eines regulären Domänen-Benutzers besitzen. Existiert eine Richtlinie, die nach einer gewissen Zeit zum Wechsel des Passwortes auffordern wird, konfigurieren Sie den Zugang so, dass das Password nie abläuft.

11.1.2.2. OTRS LDAP Optionen

OTRS kann LDAP an drei Stellen nutzen.

- Authentifizierung der Agenten
- Authentifizierung der Kundenbenutzer
- Daten der Kundenbenutzer

Die beiden Module betreffend Kundenbenutzer werden am besten zusammen verwendet, oder es ist viel Handarbeit notwendig. Diese Anleitung bezieht sich nicht auf die wechselseitige Nutzung der beiden Optionen.

11.1.3. Authentifizierung der Agenten

Bearbeiten Sie die Konfigurationsdatei von OTRS, üblicherweise zu finden in <OTRS_HOME>/Kernel/Config.pm.

Fügen Sie folgendes hinzu:

Example 11-1. Kernel/Config.pm - AuthModule

```
$Self->{'AuthModule'} = 'Kernel::System::Auth::LDAP';
```

Hiermit wird der Modus zur Authentifizierung auf LDAP umgeschaltet.

Nun fügen Sie die folgenden drei Zeilen hinzu:

Example 11-2. Kernel/Config.pm - AuthModule::LDAP Einstellungen

```
$Self->{'AuthModule::LDAP::Host'} = '[AD_server]';
$Self->{'AuthModule::LDAP::BaseDN'} = '[base_dn]';
$Self->{'AuthModule::LDAP::UID'} = 'sAMAccountName';
```

Ersetzen Sie [AD_server] mit dem Hostnamen (w2kad.example.com) oder der IP Adresse Ihres AD Servers (192.168.1.23).

Ersetzen Sie [base_dn] mit der Basis-DN Ihres AD-Baumes. Sie folgt dieser Syntax "dc=example, dc=com".

Ein einfacher Weg, an die Basis-DN zu gelangen, bietet das Werkzeug "AD Benutzer und Computer". Der Name des AD-Baumes zeigt den Domänennamen als "domain.tld" oder "subdomain.domain.tld", je nachdem, wie Ihr AD aufgesetzt wurde. Bei der Konversion in eine DN werden obige Beispiele dann zu "dc=domain, dc=tld" und "dc=subdomain, dc=domain, dc=com".

Finden Sie die Basis-DN Ihres ADs immer noch nicht, können Sie weitere Tools nutzen, wie etwa LDIFDE (finden Sie auf dem AD Server) oder ADSI Edit (von der W2K Support Kit CD/Resource Kit CD).

Da das AD ein spezielles und von einem reinen inetOrg abweichendes Schema nutzt, müssen wir "sAMAccountName" für die UID des Zugangs benutzen. Dieses Merkmal ist bestimmend im AD.

Sie müssen den Zugang angeben, über den OTRS mit dem AD kommuniziert. Dies geschieht mit den folgenden Einstellungen:

Example 11-3. Kernel/Config.pm - AuthModule::LDAP Einstellungen zum SearchUser

```
$Self->{'AuthModule::LDAP::SearchUserDN'} = '[user_dn]';
$Self->{'AuthModule::LDAP::SearchUserPw'} = '[password]';
```

Ersetzen Sie [user_dn] mit der vollständigen DN des gewünschten AD/OTRS Benutzers. Folgen Sie der Standard-LDAP-Syntax. Die vollständige DN richtet sich nach dem Kontainer, in dem Sie den Benutzerzugang mit "AD Benutzer und Gruppen" erstellt haben. Haben Sie den Benutzer im Standard-Kontainer "Benutzer" erstellt, lautet die Syntax "cn=Vorname Nachname, cn=Benutzer, [base_dn]". Wenn Sie unsicher sind, welches die korrekte DN ist, führen Sie LDIFDE.EXE auf Ihrem AD controller aus und durchsuchen sie dessen Ausgabe.

Ersetzen Sie [password] mit dem Passwort dieses Zugangs.

11.1.4. Authentifizierung der Kundenbenutzer

Die Konfiguration der Authentifizierung der Kundenbenutzer wird durch folgende Parameter bestimmt:

Example 11-4. Kernel/Config.pm - Customer::AuthModule LDAP Einstellungen

```
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::LDAP';
$Self->{'Customer::AuthModule::LDAP::Host'} = '[AD_server]';
$Self->{'Customer::AuthModule::LDAP::BaseDN'} = '[base_dn]';
$Self->{'Customer::AuthModule::LDAP::UID'} = 'sAMAccountName';
$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = '[user_dn]';
$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = '[password]';
```

Ersetzen Sie [AD_server] mit dem Hostnamen Ihres AD Servers, z.B.: w2kad.example.com.

Ersetzen Sie [base_dn] mit der DN des Astes, in dem Ihre Kundenbenutzer untergebracht sind. Befinden sich alle Kundenbenutzer in einem "People" Kontainer, spezifizieren Sie diesen als Basis-DN für CustomerAuth: "ou=People, dc=example, dc=com". Haben Sie Kundenbenutzerdaten in mehreren Kontainern, richten Sie mehrere LDAP-Quellen ein. Sie können \$Self->{CustomerUser1} bis \$Self->{CustomerUser10} zusätzlich erstellen.

Ersetzen Sie [user_dn] wie oben mit dem OTRS/AD Benutzer und [password] mit dessen Passwort.

11.1.5. Daten der Kundenbenutzer

Dies sind die notwendigen Einstellungen, um die Daten der Kundenbenutzer aus dem LDAP/AD zu lesen:

Example 11-5. Kernel/Config.pm - CustomerUser::LDAP Einstellungen

```

$Self->{CustomerUser} = {
    Module => 'Kernel::System::CustomerUser::LDAP',
    Params => {
        Host => '[AD_server]',
        BaseDN => '[base_dn]',
        SSCOPE => 'sub',
        UserDN => '[user_dn]',
        UserPw => '[password]',
    },
    CustomerKey => 'sAMAccountName',
    CustomerID => '[customer_id]',
    CustomerUserListFields => ['sAMAccountName', 'cn', 'mail'],
    CustomerUserSearchFields => ['sAMAccountName', 'cn', 'mail'],
    CustomerUserPostMasterSearchFields => ['mail'],
    CustomerUserNameFields => ['givenname', 'sn'],
    Map => [
        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown, required, storage-type
        # [ 'UserSalutation', 'Title', 'title', 1, 0, 'var' ],
        [ 'UserFirstname', 'Firstname', 'givenname', 1, 1, 'var' ],
        [ 'UserLastname', 'Lastname', 'sn', 1, 1, 'var' ],
        [ 'UserLogin', 'Login', 'sAMAccountName', 1, 1, 'var' ],
        [ 'UserEmail', 'Email', 'mail', 1, 1, 'var' ],
        [ 'UserCustomerID', 'CustomerID', 'mail', 0, 1, 'var' ],
        # [ 'UserPhone', 'Phone', 'telephonenumber', 1, 0, 'var' ],
        # [ 'UserAddress', 'Address', 'postaladdress', 1, 0, 'var' ],
        # [ 'UserComment', 'Comment', 'description', 1, 0, 'var' ],
    ],
};

```

Ersetzen Sie [AD_server] mit dem Namen oder der IP Adresse Ihres AD servers.

Ersetzen Sie [base_dn] mit der DN des Astes Ihrer Kundenbenutzer, z.B.: "ou=People, dc=example, dc=com".

Ersetzen Sie [user_dn] und [password] wie gehabt.

Ersetzen Sie [customer_id] mit einem geeigneten Attribut. Üblich ist hier das Attribut 'mail' oder 'o'.

Mit CustomerUserListFields bestimmen Sie, wie Kundenbenutzer angezeigt während der Arbeit werden. Mit den obigen Standardeinstellungen wird der Kundenbenutzer angezeigt als: "Username Nachhname email@adresse"

Möchten Sie lediglich die eMail Adressen anzeigen, ändern Sie den Eintrag so ab, dass er nur noch 'mail' enthält. Sie können hier auch weitere LDAP/AD Attribute hinzufügen.

CustomerUserSearchFields hat dieselben Optionen wie CustomerUserListFields. Hiermit bestimmen Sie, über welche Attribute Kundenbesucher gesucht werden. Fügen Sie bspw. das Attribut 'o' hinzu, genügt die Angabe des Firmennamens, um alle Kundenbenutzer einer Firma aufzulisten, sofern dieser Wert im AD vorhanden ist.

11.1.6. Tipps & Tricks

11.1.6.1. Control who's let in

Verhindern Sie, dass Kundenbenutzer und Agenten sich im OTRS einloggen können. Dies wird umso wichtiger, je größer und unübersichtlicher Ihre AD-Struktur ist/wird. Legen Sie eine Gruppe der Objectklasse posixGroup an und füllen Sie sie mit entweder der UID oder der DN der gewünschten Mitglieder.

Example 11-6. Kernel/Config.pm - GroupDN für Agenten

```

$Self->{'AuthModule::LDAP::GroupDN'} = 'cn=otrsallow_A, ou=posixGroups, dc=example, dc=com';
$Self->{'AuthModule::LDAP::AccessAttr'} = 'memberUid';
# $Self->{'AuthModule::LDAP::UserAttr'} = 'UID';
$Self->{'AuthModule::LDAP::UserAttr'} = 'DN';

```

Example 11-7. Kernel/Config.pm - GroupDN for Customers

```
$Self->{'Customer::AuthModule::LDAP::GroupDN'} = 'cn=otrsallow_C, ou=posixGroups, dc=example, dc=com';
$Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid';
# $Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'UID';
$Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'DN';
```

11.1.6.2. Konfigurationsbeispiel

Dies ist ein komplettes Beispiel, wie OTRS für die Authentifizierung von Kundenbenutzern und Agenten gegen ein AD konfiguriert werden kann. Auch die Kundenbenutzerdaten werden aus dem AD gelesen, die Datenbankeinstellungen wurden hier ebenfalls beibehalten.

Die benutzten Einstellungen:

- Windows AD Server: w2kad.example.com
- Windows OTRS/AD Benutzer: otrs (Vollständiger Name: OTRS Service)
- Windows OTRS/AD Benutzer Passwort: secret
- Windows OTRS/AD Benutzer DN: cn=OTRS Service, ou=People, dc=example, dc=com
- Windows Domäne: example.com
- Windows LDAP Basis-DN: dc=example, dc=com
- Windows Benutzer Container: People
- Windows Benutzer LDAP DN: ou=People, dc=example, dc=com
- Windows Agenten LDAP DN: ou=People, dc=example, dc=com

Example 11-8. Kernel/Config.pm - AuthModule LDAP settings

```
#-----
# Start of Example Config
$Self->{'AuthModule'} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host'} = 'w2kad.example.com';
$Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example, dc=com';
$Self->{'AuthModule::LDAP::UID'} = 'sAMAccountName';

$Self->{'AuthModule::LDAP::SearchUserDN'} = 'cn=OTRS Service, ou=People, dc=example, dc=com';
$Self->{'AuthModule::LDAP::SearchUserPw'} = 'secret';

# This is an example configuration for an LDAP auth. backend.
# (take care that Net::LDAP is installed!)
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::LDAP';
$Self->{'Customer::AuthModule::LDAP::Host'} = 'w2kad.example.com';
$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'ou=People, dc=example, dc=com';
$Self->{'Customer::AuthModule::LDAP::UID'} = 'sAMAccountName';

# The following is valid but would only be necessary if the
# anonymous user do NOT have permission to read from the LDAP tree
$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = 'cn=OTRS Service, ou=People, dc=example, dc=com';
$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = 'secret';

# CustomerUser
# (customer user database backend and settings)
$Self->{'CustomerUser'} = {
    Name => 'Datenbank',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => { Table => 'customer_user',
        # to use an external database
        DSN => 'DBI:odbc:yourdsn',
        DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
        User => "", Password => "",
    },
    # customer uniq id
    CustomerKey => 'login',
    CustomerID => 'customer_id',
    CustomerValid => 'valid_id',
```



```

CustomerUserListFields => ['first_name', 'last_name', 'email'],
# CustomerUserListFields => ['login', 'first_name', 'last_name', 'customer_id', 'email'],
CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
CustomerUserSearchPrefix => "",
CustomerUserSearchSuffix => '*',
CustomerUserSearchListLimit => 250,
CustomerUserPostMasterSearchFields => ['email'],
CustomerUserNameFields => ['salutation', 'first_name', 'last_name'],
# ReadOnly => 1,
Map => [
    # note: Login, Email and CustomerID needed!
    # var, frontend, storage, shown, required, storage-type, http-link
    [ 'UserSalutation', 'Salutation', 'salutation', 1, 0, 'var' ],
    [ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var' ],
    [ 'UserLastname', 'Lastname', 'last_name', 1, 1, 'var' ],
    [ 'UserLogin', 'Login', 'login', 1, 1, 'var' ],
    [ 'UserPassword', 'Password', 'pw', 0, 1, 'var' ],
    [ 'UserEmail', 'Email', 'email', 0, 1, 'var' ],
    [ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var' ],
    [ 'UserComment', 'Comment', 'comments', 1, 0, 'var' ],
    [ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int' ],
],
};

# CustomerUser1
# (customer user ldap backend and settings)
$Self->{CustomerUser1} = {
    Module => 'Kernel::System::CustomerUser::LDAP',
    Params => {
        # ldap host
        Host => 'w2kad.example.com',
        # ldap base dn
        BaseDN => 'ou=People, dc=example, dc=com',
        # search scope (one|sub)
        SSCOPE => 'sub',
        # The following is valid but would only be necessary if the
        # anonymous user does NOT have permission to read from the LDAP tree
        UserDN => 'cn=OTRS Service, ou=People, dc=example, dc=com',
        UserPw => 'secret',
        AlwaysFilter => "",
        SourceCharset => 'utf-8',
        DestCharset => 'iso-8859-1',
    },
    # customer uniq id
    CustomerKey => 'sAMAccountName',
    # customer #
    CustomerID => 'mail',
    CustomerUserListFields => ['sAMAccountName', 'cn', 'mail'],
    CustomerUserSearchFields => ['sAMAccountName', 'cn', 'mail'],
    CustomerUserSearchPrefix => "",
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['mail'],
    CustomerUserNameFields => ['givenname', 'sn'],
    Map => [
        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown, required, storage-type
        [ 'UserSalutation', 'Title', 'title', 1, 0, 'var' ],
        [ 'UserFirstname', 'Firstname', 'givenname', 1, 1, 'var' ],
        [ 'UserLastname', 'Lastname', 'sn', 1, 1, 'var' ],
        [ 'UserLogin', 'Login', 'sAMAccountName', 1, 1, 'var' ],
        [ 'UserEmail', 'Email', 'mail', 1, 1, 'var' ],
        [ 'UserCustomerID', 'CustomerID', 'mail', 0, 1, 'var' ],
        [ 'UserPhone', 'Phone', 'telephonenumber', 1, 0, 'var' ],
        [ 'UserAddress', 'Address', 'postaladdress', 1, 0, 'var' ],
        [ 'UserComment', 'Comment', 'description', 1, 0, 'var' ],
    ],
};
# End example config
#-----

```


Chapter 12. Zugriffsrechte

Dieses Kapitel wird Ihnen einen Einblick in die Rechteverwaltung von OTRS geben.

12.1. Gruppen/Benutzer/Queue Zugriffsrechte

Mit OTRS sind Sie in der Lage, verschiedene Benutzergruppen mit unterschiedlichen Zugriffsrechten (z.B. Lesezugriff auf Tickets einer bestimmten Queue oder eine Gruppe hat in einer Queue keine Schreibrechte (ro/rw)) zu verwalten.

Die Abstufung der Zugriffsrechte für Benutzergruppen beinhaltet: "ro" ("read only" Lesezugriff auf ein Ticket), "move_into" (Erlaubnis, Tickets in eine bestimmte Queue zu verschieben), "create" (Berechtigung, Tickets in dieser Queue zu erstellen), "owner" (Berechtigung, den Besitzer des Tickets zu ändern), "priority" (Berechtigung für die Änderung der Ticket-Priorität) und "rw" (Vollzugriff "read and write").

Sie können die Benutzer den Gruppen zuordnen und diese dann mit der Queue verknüpfen (Benutzer-Gruppe-Queue).

Beispiel: Wenn Sie in einer Queue nur Tickets lesen möchten, oder eine Gruppe keinen Vollzugriff auf eine Queue haben soll, dann müssen Sie:

- a) OTRS - Gruppen erstellen,
- b) die gewünschte Anzahl an Benutzern dieser Gruppe zuweisen
(Hierfür stehen Ihnen die "ro/rw" Optionen zur Verfügung)
- und
- c) Ihre OTRS-Queues den entsprechenden OTRS-Gruppen zuordnen.

Für den Fall, dass "ro"-Queues in der Queue-Ansicht angezeigt werden sollen, bedarf es einer Einstellung in der config.pm.

```
Kernel/Config.pm
[...]  
    # QueueViewAllPossibleTickets  
    # (show all ro and rw queues - not just rw queues)  
    $Self->{QueueViewAllPossibleTickets} = 0;  
[...]
```

12.2. Module Zugriffsrechte - Agent/Admin Oberfläche

Es ist möglich, Rechte für ein Module (welches sich unter Kernel/Modules/*.pm befindet) zu vergeben. Es soll beispielsweise eine Gruppe von Benutzern in der Lage sein, Antworten (Modul default answers) zu bearbeiten. Normalerweise muss der Benutzer hierfür in einer "admin" Gruppe sein.

So geht's:

- a) Erstellen Sie eine neue OTRS-Gruppe mit Namen "responses",
- b) ordnen Sie die Benutzer dieser Gruppe zu ("rw" wird benötigt),
- c) fügen Sie die folgenden Zeile in Kernel/Config.pm ein:
[...]
\$Self->{'Module::Permission'}->{'AdminResponse'} = 'responses';
[...]
- d) fügen Sie einen Link in die Agenten Oberfläche ein (Kernel/Output/HTML/*/AgentNavigationBar.dtl)
[...]
Response
[...]
- und
- e) Starten Sie Ihren Webserver neu!

Jetzt sollten alle Benutzer der Gruppe "responses" in der Lage sein, das AdminResponse Modul zu benutzen und somit Antworten zu bearbeiten.

Darüber hinaus ist es möglich, mehrere Gruppen zugriff auf ein Module zu ermöglichen (durch die Verwendung eines Perl Arrays):

```
c) [Kernel/Config.pm]  
    $Self->{'Module::Permission'}->{'AdminResponse'} = ['admin', 'responses'];  
[...]
```


Chapter 13. Konfigurationsdatei

Innerhalb OTRS gibt es viele Konfigurationsoptionen. Insgesamt gibt es zwei solcher Dateien: 1. Kernel/Config.pm und 2. Kernel/Config/Defaults.pm.

Kernel/Config/Defaults.pm ist die Standard Konfigurationsdatei, die nicht geändert werden sollte. Alle möglichen Konfigurationsoptionen finden Sie in dieser Datei. Das Lesen sollte Ihnen nicht schwer fallen.

Kernel/Config.pm.dist ist die initiale Beispieldatei für Kernel/Config.pm (angepasste Konfiguration Ihres Systems). Config.pm muß noch aus Config.pm.dist erzeugt werden. (cp Kernel/Config.pm.dist Kernel/Config.pm).

So funktioniert's! Die Datei Kernel/Config/Defaults.pm wird zuerst, die Datei Kernel/Config.pm danach geladen. Wenn Sie Änderungen an der Konfiguration vornehmen wollen, kopieren Sie einfach die benötigte Option aus Kernel/Config/Defaults.pm in Kernel/Config.pm und ändern Sie die Werte hier.

Nun werden wir einige, nicht alle Konfigurationsfunktionen erläutern.

13.1. TicketHook (Ticket Kennzeichen)

Sie können den TicketHook / das Ticket Kennzeichen anpassen. Dies ist der erste Teil im Betreff Ihrer E-Mails (z.B. [MyTicket: 007]).

Example 13-1. Kernel/Config.pm - TicketHook / Ticket Kennzeichen

```
[...]
# TicketHook
# (To set the Ticket identifier. Some people want to
# set this to e. g. 'Call#', 'MyTicket#' or 'Ticket#'.)
$self->{TicketHook} = 'Ticket#',
[...]
```

Hinweis: Verwenden Sie keinen TicketHook, der nur aus 2 Zeichen (z.B. TN) besteht. MS Outlook würde nämlich "TN: 54968797" durch "RE: 54968797" ersetzen. OTRS würde dies ziehmlche Probleme bereiten.

13.2. FQDN

Der OTRS "full qualified domain name" wird für E-Mail Nachrichten ID's verwendet.

Example 13-2. Kernel/Config.pm - FQDN

```
[...]
# FQDN
# (Full qualified domain name of your system.)
$self->{FQDN} = 'yourhost.example.com';
[...]
```

13.3. Protokoll

Die OTRS Protokollierungsmethode.

Example 13-3. Kernel/Config.pm - LogModule

```
[...]
# -----#
# log settings                                     #
# -----#

# LogModule
# (log backend module)
$self->{LogModule} = 'Kernel::System::Log::SysLog';
# $self->{LogModule} = 'Kernel::System::Log::File';

# param for LogModule Kernel::System::Log::File (required!)
# $self->{'LogModule::LogFile'} = '/tmp/otrs.log';
# param if the date (yyyy-mm) should be added as suffix to
```

```
# logfile [0|1]
# $Self->{'LogModule::LogFile::Date'} = 0;
[...]
```

13.4. E-Mail Überprüfungsoption

Es gibt zwei Möglichkeiten, die verwendeten E-Mailadressen zu überprüfen:

Example 13-4. Kernel/Config.pm - CheckEmailAddresses

```
[...]
# CheckEmailAddresses
# (Check syntax of used email addresses)
$Self->{CheckEmailAddresses} = 1;
[...]
```

Example 13-5. Kernel/Config.pm - CheckMXRecord

```
[...]
# CheckMXRecord
# (Check mx records of used email addresses)
$Self->{CheckMXRecord} = 1;
[...]
```

Deaktivieren Sie diese Einstellung, wenn Sie sich in einem Netzwerk ohne externe DNS Anbindung befinden. Andernfalls können Sie keine E-Mails von der Agenten Oberfläche verschicken.

13.5. Max. Postmaster E-Mail

OTRS ist in der Lage, automatische E-Mail-Antworten zu verschicken. Hierdurch können Schleifen entstehen.

Example 13-6. Kernel/Config.pm - PostmasterMaxEmails

```
[...]
# PostmasterMaxEmails
# (Max post master daemon email to own email-address a day.
# Loop-Protection!) [default: 40]
$Self->{PostmasterMaxEmails} = 40;
[...]
```

13.6. Format der Ticket-Nummer

Sie können zwischen vier verschiedenen OTRS Formaten wählen. AutoIncrement ist als Standard gesetzt.

Example 13-7. Kernel/Config.pm - TicketNumberGenerator

```
[...]
# TicketNumberGenerator
#
# Kernel::System::Ticket::Number::AutoIncrement (default) --> auto increment
#   ticket numbers "SystemID.Counter" like 1010138 and 1010139.
#
# Kernel::System::Ticket::Number::Date --> ticket numbers with date
#   "Year.Month.Day.SystemID.Counter" like 200206231010138 and 200206231010139.
#
# Kernel::System::Ticket::Number::DateChecksum --> ticket numbers with date and
#   check sum the counter will be rotated daily (my favorite)
#   "Year.Month.Day.SystemID.Counter.CheckSum" like 2002070110101520 and 2002070110101535.
#
# Kernel::System::Ticket::Number::Random -->
#   random ticket numbers "SystemID.Random" like 100057866352 and 103745394596.
```

```
$Self->{TicketNumberGenerator} = 'Kernel::System::Ticket::Number::AutoIncrement';
[...]
```

Sie können auch Ihr eigenes Modul zur Erzeugung von Ticket Nummern schreiben. Das Modul muss lediglich über zwei Funktionen verfügen: 1. "CreateTicketNr()" um eine neue Ticket Nummer zu generieren und 2. "GetTNByString()" um eine Ticket Nummer als Zeichenkette zu erhalten.

Ein nettes Beispiel ist das Kernel::System::Ticket::Number::Random

(<http://otrs.org/cgi-bin/cvsweb.cgi/otrs/Kernel/System/Ticket/Number/Random.pm>) Zufallsmodul.

13.7. Datenbankeinstellungen

Example 13-8. Kernel/Config.pm - Datenbankeinstellungen

```
[...]
# DatabaseHost
# (The database host.)
$Self->{DatabaseHost} = 'localhost';

# Database
# (The database name.)
$Self->{Database} = 'otrs';

# DatabaseUser
# (The database user.)
$Self->{DatabaseUser} = 'otrs';

# DatabasePw
# (The password of database user.)
$Self->{DatabasePw} = 'some-pass';

# DatabaseDSN
# (The database DSN for MySQL ==> more: "man DBD::mysql")
$Self->{DatabaseDSN} = "DBI:mysql:database=$Self->{Database};host=$Self->{DatabaseHost}";

# (The database DSN for PostgreSQL ==> more: "man DBD::Pg")
#$Self->{DatabaseDSN} = "DBI:Pg:dbname=$Self->{Database}";
[...]
```

13.8. ASP (Application Service Provider) Optionen

Soll der Agent in der Lage sein Tickets in Queues zu verschieben, denen er nicht angehört, so sollte die Option verwendet werden. (Nützlich, wenn OTRS im ASP Modus verwendet wird).

Example 13-9. Kernel/Config.pm - ChangeOwnerToEveryone

```
[...]
# ChangeOwnerToEveryone -> useful for ASP
# (Possible to change owner of ticket ot everyone) [0|1]
$Self->{ChangeOwnerToEveryone} = 0;
[...]
```

13.9. Custom Queue

Einige Benutzer von OTRS wollen den "Custom Queue" Namen verändern.

Example 13-10. Kernel/Config.pm - Custom Queue

```
[...]
# CustomQueue
# (The name of custom queue.)
$self->{CustomQueue} = 'PersonalQueue';
[...]
```

13.10. Queue Verzeichnisart

Konfigurieren Sie die Queue-Auswahl in Ihrem System! Die Optionen sind "tree" (Standard) und "list".

Tree:

```
QueueA
  Queue1
  Queue2
  Queue3
QueueB
  Queue1
    Queue-a
    Queue-b
    Queue-c
    Queue-d
  Queue2
    Queue-a
    Queue-b
QueueC
  Queue1
  Queue2
```

List:

```
QueueA
QueueA::Queue1
QueueA::Queue2
QueueA::Queue3
QueueB
QueueB::Queue1
QueueB::Queue2
```

Example 13-11. Kernel/Config.pm - QueueListType

```
[...]
# QueueListType
# (show queues in system as tree or as list) [tree|list]
$self->{QueueListType} = 'tree';
[...]
```

13.11. Agent Benachrichtigung

Sollten neue Tickets eintreffen oder der Verlauf eines Tickets gewünscht sein, so kann OTRS den/die Agenten informieren.

Example 13-12. Kernel/Config.pm - Agent notification

```
[...]
# -----#
# notification stuff                                #
# -----#
# notification sender
$self->{NotificationSenderName} = 'OTRS Notification Master';
$self->{NotificationSenderEmail} = 'otrs@' . $self->{FQDN};
```



```

# new ticket
$Self->{NotificationSubjectNewTicket} = 'New ticket notification! (<OTRS_CUSTOMER_SUBJECT[10]>';
$Self->{NotificationBodyNewTicket} = "
Hi,

there is a new ticket!

<snip>
<OTRS_CUSTOMER_EMAIL[6]>
<snip>

http://$Self->{FQDN}/otrs/index.pl?Action=AgentZoom&TicketID=<OTRS_TICKET_ID>

Your OTRS Notification Master

";

# follow up
$Self->{NotificationSubjectFollowUp} = 'You got follow up! (<OTRS_CUSTOMER_SUBJECT[10]>';
$Self->{NotificationBodyFollowUp} = "
Hi <OTRS_USER_FIRSTNAME>,

you got a follow up!

<snip>
<OTRS_CUSTOMER_EMAIL[6]>
<snip>

http://$Self->{FQDN}/otrs/index.pl?Action=AgentZoom&TicketID=<OTRS_TICKET_ID>

Your OTRS Notification Master

";
[...]
```

13.12. Sitzungsmanagement

Das OTRS Sitzungsmanagement.

Example 13-13. Kernel/Config.pm - Session management

```

[...]
```

```

# -----#
# session settings                                #
# -----#

# SessionModule (replace old SessionDriver!!!)
# (How should be the session-data stored?
# Advantage of DB is that you can split the
# Frontendserver from the DB-Server. fs is faster.)
$Self->{SessionModule} = 'Kernel::System::AuthSession::DB';
# $Self->{SessionModule} = 'Kernel::System::AuthSession::FS';
# $Self->{SessionModule} = 'Kernel::System::AuthSession::IPC';

# SessionCheckRemoteIP
# (If the application is used via a proxy-farm then the
# remote ip address is mostly different. In this case,
# turn of the CheckRemoteID. ) [1|0]
$Self->{SessionCheckRemoteIP} = 1;

# SessionDeleteIfNotRemoteID
# (Delete session if the session id is used with an
# invalid remote IP?) [0|1]
$Self->{SessionDeleteIfNotRemoteID} = 1;

# SessionMaxTime
# (Max valid time of one session id in second (8h = 28800).)
$Self->{SessionMaxTime} = 28800;
```

```

# SessionDeleteIfTimeToOld
# (Delete session's witch are requested and to old?) [0|1]
$Self->{SessionDeleteIfTimeToOld} = 1;

# SessionUseCookie
# (Should the session management use html cookies?
# It's more comfortable to send links ==> if you have a valid
# session, you don't have to login again.) [0|1]
# Note: If the client browser disabled html cookies, the system
# will work as usual, append SessionID to links!
$Self->{SessionUseCookie} = 1;

# SessionUseCookieAfterBrowserClose
# (store cookies in browser after closing a browser) [0|1]
$Self->{SessionUseCookieAfterBrowserClose} = 0;
[...]
```

13.13. URL Login und Logout Einstellungen

Example 13-14. Kernel/Config.pm - URL Login und Logout Einstellungen

```

[...]
```

```

# -----#
# URL login and logout settings          #
# -----#

# LoginURL
# (If this is anything other than "", then it is assumed to be the
# URL of an alternate login screen which will be used in place of
# the default one.)
$Self->{LoginURL} = "";
#   $Self->{LoginURL} = 'http://host.example.com/cgi-bin/login.pl';

# LogoutURL
# (If this is anything other than "", it is assumed to be the URL
# of an alternate logout page which users will be sent to when they
# logout.)
$Self->{LogoutURL} = "";
#   $Self->{LogoutURL} = 'http://host.example.com/cgi-bin/login.pl';
[...]
```

13.14. Agent-Oberfläche Standardeinstellungen

Example 13-15. Kernel/Config.pm - Agent-Oberfläche Standardeinstellungen

```

[...]
```

```

# -----#
# agent area default settings          #
# -----#

# ViewableTickets
# (The default viewable tickets a page.)
$Self->{ViewableTickets} = 25;

# ViewableTicketLines
# (Max viewable ticket lines in the QueueView.)
$Self->{ViewableTicketLines} = 18;

# ViewableTicketLinesZoom
# (Max viewable ticket lines in the QueueZoom.)
$Self->{ViewableTicketLinesZoom} = 6000;

# MaxLimit
# (Max viewable tickets a page.)
```

```

$Self->{MaxLimit} = 150;

# RefreshOptions
# (Refresh option list for preferences)
$Self->{RefreshOptions} = {
    " => 'off',
    2  => ' 2 minutes',
    5  => ' 5 minutes',
    7  => ' 7 minutes',
    10 => '10 minutes',
    15 => '15 minutes',
};

# Highlight*
# (Set the age and the color for highlighting of old queue
# in the QueueView.)
# highlight age1 in min
$Self->{HighlightAge1} = 1440;
$Self->{HighlightColor1} = 'orange';
# highlight age2 in min
$Self->{HighlightAge2} = 2880;
$Self->{HighlightColor2} = 'red';

# -----#
# AgentUtil                                     #
# -----#

# default limit for Tn search
$Self->{SearchLimitTn} = 20;

# default limit for Txt search
$Self->{SearchLimitTxt} = 20;

# viewable ticket lines by search util
$Self->{ViewableTicketLinesBySearch} = 15;

# -----#
# Ticket stuff                                     #
# (Viewable tickets in queue view)                 #
# -----#
# ViewableLocks
# default: ["'unlock'", "'tmp_lock'"]
$Self->{ViewableLocks} = ["'unlock'", "'tmp_lock'"];

# ViewableStateType
# (see http://yourhost/otrs/index.pl?Action=AdminState -> StateType)
$Self->{ViewableStateType} = ['new', 'open', 'pending'];

# ViewableSenderTypes
# default: ["'customer'"]
$Self->{ViewableSenderTypes} = ["'customer'"];
[...]
```

13.15. Rechtschreibüberprüfung

OTRS ist in der Lage, eine Rechtschreibüberprüfungssoftware wie ispell (<http://fmg-www.cs.ucla.edu/fmg-members/geoff/ispell.html>) oder aspell (<http://aspell.sourceforge.net/>) zu verwenden.

Example 13-16. Kernel/Config.pm - Rechtschreibüberprüfung

```

[...]
```

```

# SpellChecker
# (If ispell or aspell is available, then we will provide a spelling
# checker)
#
$Self->{SpellChecker} = "";
$Self->{SpellChecker} = '/usr/bin/ispell';
$Self->{SpellCheckerDictDefault} = 'english';
```

```
# SpellCheckerIgnore
# (A list of ignored words.)
$Self->{SpellCheckerIgnore} = ['www', 'webmail', 'https', 'http', 'html'];
[...]
```

Mögliche Agenten-Wörterbücher.

Example 13-17. Kernel/Config.pm - Rechtschreibüberprüfung - Agenten Präferenzen

```
[...]
$Self->{PreferencesGroups}->{SpellDict} = {
    Colum => 'Other Options',
    Label => 'Spelling Dictionary',
    Desc => 'Select your default spelling dictionary.',
    Type => 'Generic',
    Data => {
        # installed dict catalog (check your insalled catalogues,
        # e. g. deutsch ==> german!)
        # dict => frontend
        'english' => 'English',
        'deutsch' => 'Deutsch',
    },
    PrefKey => 'UserSpellDict',
    Activ => 1,
};
[...]
```

Vielleicht wollen Sie, dass die Antwort standardmäßig überprüft werden muss.

Example 13-18. Kernel/Config.pm - Rechtschreibüberprüfung - Überprüfungspflicht

```
# FrontendNeedSpellCheck
# (compose message must be spell checked)
$Self->{FrontendNeedSpellCheck} = 0;
```

13.16. Format der Antwort (Antwort erstellen)

OTRS ist in der Lage, das Format der Antwort, die in der Agenten-Oberfläche erstellt wurden, zu konfigurieren.

Example 13-19. Kernel/Config.pm - Antwortenformat (unix_style)

```
[...]
# unix_style (default)
$Self->{ResponseFormat} = '$Data{"Salutation"}
$Data{"OrigFrom"} $Text{"wrote"}:
$Data{"Body"}
$Data{"StdResponse"}
$Data{"Signature"}
';
[...]
```

Wenn sie das Format verändern möchten, benutzen sie das Folgende:

Example 13-20. Kernel/Config.pm - Antwortenformat (ms_style)

```
[...]
# ms_style
$self->{ResponseFormat} = '$Data{"Salutation"}
$Data{"StdResponse"}
$Data{"Signature"}
$Data{"OrigFrom"} $Text{"wrote"}:
$Data{"Body"}
';
[...]
```


Chapter 14. Cronjobs

Was ist ein Cronjob? Als Cronjobs werden Aufgaben und Tätigkeiten bezeichnet, die in regelmäßigen Abständen anstehen. Diese werden vom System automatisch ausgeführt und abgearbeitet.

OTRS benötigt einige wenige "Cronjobs" um beispielsweise Tickets zu entsperren, Erinnerungen zu senden oder E-Mail abzuholen.

14.1. Wie funktioniert's - Setup

Einige Cronjobs sind in OTRS standardmäßig im Verzeichnis `$OTRS_HOME/var/cron/*.dist` angelegt. Zu erst sollten sie von den existierenden Cronjobs Kopieen anlegen und anschließend die Endung `.dist` löschen. Unter Linux geht dies wie folgt:

```
cd var/cron
for foo in *.dist; do cp $foo `basename $foo .dist`; done
```

Wen Sie OTRS unter Windows installiert haben, geben Sie das Folgende ein:

```
cd var/cron
copy *.dist *.
```

Mit dem Befehl `$OTRS_HOME/bin/Cron.sh {start|stop|restart}` können Sie diese Cronjobs unter `$OTRS_HOME/var/cron/*` starten oder stoppen. (Die Dateien mit den Endungen `.dist` werden von diesem Script übergangen.)

Hinweis: Installieren Sie diese Cronjobs als OTRS Benutzer.

14.2. Standard-Cronjobs

14.2.1. aaa_base

Dieser Cronjob definiert die E-Mail Adresse des OTRS Administrators, damit alle Fehlermeldungen zugestellt werden können.

```
MAILTO="root@localhost"
```

14.2.2. pending_jobs

Bearbeitet jede Stunde alle eintreffenden "pending auto" Tickets.

```
45 * * * * $HOME/bin/PendingJobs.pl >> /dev/null
```

14.2.3. postmaster

Löscht um 00:10 Uhr alle unbearbeiteten E-Mails, welche via procmail eingegangen sind.

```
10 0 * * * $HOME/bin/otrs.cleanup >> /dev/null
```

14.2.4. postmaster_pop3

Ruft jede 10 Minuten alle E-Mail über die benannten POP3 Accounts ab.

```
* /10 * * * * $HOME/bin/PostMasterPOP3.pl >> /dev/null
```

14.2.5. unlock

Entsperrt einmal pro Stunde alte, gesperrte Tickets.

```
35 * * * * $HOME/bin/UnlockTickets.pl --timeout >> /dev/null
```

14.2.6. session

Löscht alle 6 Stunden alte Sessions-ID's aus der Datenbank, dem Dateiensystem oder dem RAM.

```
55 */6 * * * * $HOME/bin/DeleteSessionIDs.pl --expired >> /dev/null
```

14.2.7. rebuild_ticket_index

Der Ticketindex wird täglich neu aufgebaut. (Notwendig in Verbindung mit: Kernel::System::Ticket::IndexAccelerator::StaticDB - vgl. Kapitel "Performance Tuning").

```
01 01 * * * * $HOME/bin/RebuildTicketIndex.pl >> /dev/null
```

14.2.8. generic_agent-database

Ausführen des GenericAgent.pl mit den Datenbank-Jobs alle 10 Minuten (vgl. Kapitel "Generic-Agent").

```
*/10 * * * * $HOME/bin/GenericAgent.pl -c db >> /dev/null
```

14.2.9. generic_agent

Ausführen des GenericAgent.pl mit den Execute GenericAgent.pl mit der Kernel/Config/GenericAgent.pm Konfigdatei alle 20 Minuten (vgl. Kapitel "Generic-Agent").

```
*/20 * * * * $HOME/bin/GenericAgent.pl >> /dev/null
```


Chapter 15. Generic-Agent

Der Generic-Agent (zu finden unter bin/GenericAgent.pl) ist ein Kommandozeilenprogramm, welches verschiedene Aktionen (bspw. Notizen einfügen, Status setzen, Besitzer ändern...) in Bezug auf die Tickets ausführen kann.

15.1. Web-Interface (bevorzugt)

OTRS 1.3 oder höher kommt mit einem Web-Interface um GenericAgent Jobs zu verwalten.

Admin - GenericAgent Job-Übersicht

The screenshot shows the OTRS Admin GenericAgent Job Overview web interface in a Mozilla Firefox browser. The address bar shows the URL: `http://portal.otrs.de/otrs/index.pl?Action=AdminGenericAgent`. The page title is "OTRS :: Admin :: GenericAgent". The interface is divided into several sections:

- [OTRS]**: Martin Edenhofer (me@otrs.de) Mit 22 Sep 00:37:44 2004. Includes links for Abmelden, Agent-Bereich, and FAQ-Bereich.
- [Admin-Bereich]**: A navigation menu with four categories: Benutzer & Gruppen, Queue & Antworten, System, and Sonstiges. Each category contains a list of links to various administrative functions.
- [GenericAgent]**: The main section for managing GenericAgent jobs. It includes a "Hinzufügen:" form with a "Name:" field and a "Hinzufügen" button. Below this is a "Job-Liste:" table with columns: Name, Letzter lauf, gültig/ungültig, Bearbeiten, Löschen, and Run Now! The table contains one entry: "close spam" with a last run time of "21.09.2004 23:24:35" and a status of "gültig".
- Queue-Ansicht - Telefon-Ansicht**: A link to switch between queue and phone views.
- Powered by OTRS 2.0.x CVS**: A footer note.

Admin - GenericAgent Job-Bearbeiten

The screenshot shows the OTRS Admin GenericAgent Job Edit web interface in a Mozilla Firefox browser. The address bar shows the URL: `http://portal.otrs.de/otrs/index.pl?Action=AdminGenericAgent&Subaction=LoadPrc`. The page title is "OTRS :: Admin :: GenericAgent". The interface is divided into several sections:

- [Admin-Bereich]**: A navigation menu with four categories: Benutzer & Gruppen, Queue & Antworten, System, and Sonstiges. Each category contains a list of links to various administrative functions.
- [GenericAgent]**: The main section for editing a GenericAgent job. It includes a "Job-Liste:" table with columns: Name, Letzter lauf, gültig/ungültig, Bearbeiten, Löschen, and Run Now! The table contains one entry: "close spam" with a last run time of "21.09.2004 23:24:35" and a status of "gültig".
- Speichere Job als?**: A form with a "Name:" field and a "Speichern" button.
- Is Job Valid?**: A form with a "Is Job Valid:" dropdown menu.
- Zeitplan**: A section for setting the job's schedule. It includes a "Minuten:" field (set to 60), a "Stunden:" field (set to 10), and a "Tage:" field (set to Fre).
- Ticket# und Kunden#**: A form with a "Ticket#" field and a "Kunden#" field.
- Volltextsuche in Artikel (z.B. "Mar*in" oder "Baue*")**: A form with a search field.

15.2. Konfigurationsdatei (optional)

Kernel/Config/GenericAgent.pm ist wiederum die zugehörige Konfigurationsdatei. Diese muss, ebenso wie die anderen Konfigurationsdateien von OTRS erst aus der Vorlage Kernel/Config/GenericAgent.pm.dist nach Kernel/Config/GenericAgent.pm kopiert werden.

Es ist genau so gut möglich, verschiedene Konfigurationsdateien zu verwenden. Kopieren Sie die Vorlage einfach nach z.B. Kernel/Config/GenericAgentFileB.pm und ändern Sie die Bezeichnung des Packets in der neuen Datei.

```
[...]
package Kernel::Config::GenericAgentFileB;
[...]
```

Führen Sie den bin/GenericAgent.pl mit der Option "-c Kernel::Config::GenericAgentFileB" aus und schon wird die neue Konfigurationsdatei verwendet.

15.2.1. Beispiele:

Wir wollen nun einige Beispiele vorstellen. Eine Beispielkonfiguration finden sie zudem unter: Kernel/Config/GenericAgent.pm.examples

Hier nun die möglichen Optionen:

```
'name of job' => {
    # get all tickets with these properties
    TicketNumber => '200%01',
    Queue => 'system queue',
    States => ['new', 'open'],
    Priorities => ['1 very low', '2 low', '3 normal'],
    Locks => ['unlock'],
    From => '%spam@example.com%',
    To => '%support@example.com%',
    Cc => '%client@example.com%',
    TicketFreeKey1 => 'Product',
    TicketFreeText1 => 'adasd',
    TicketFreeKey2 => 'Product',
    TicketFreeText2 => 'adasd',
    Subject => '%VIRUS 32%',
    Body => '%VIRUS 32%',
    CustomerID => '123',
    CustomerUserLogin => 'uid123',
    # tickets older the 60 minutes
    TicketCreateTimeOlderMinutes => 60,
    # tickets newer then 60 minutes
    TicketCreateTimeNewerMinutes => 60,
    # tickets with create time older then ...
    TicketCreateTimeOlderDate => '2004-01-19 00:00:01',
    # tickets with create time newer then ...
    TicketCreateTimeNewerDate => '2004-01-09 23:59:59',

    # or escalation tickets
    Escalation => 1,

    # new ticket properties (no option is required, use just the options
    # which should be changed!)
    New => {
        # new queue
        Queue => 'new system queue',
        # possible states (closed successful|closed unsuccessful|open|new|removed)
        State => 'closed successful',
        Priority => '3 normal',
        # new ticket free text
        TicketFreeKey1 => 'Product123',
        TicketFreeText1 => 'adasd123',
        # new ticket owner (if needed)
        Owner => 'root@localhost',
        # set customer id and customer user
```

```

CustomerID => '123',
CustomerUserLogin => 'usd213',
# if you want to add a Note
Note => {
    From => 'GenericAgent',
    Subject => 'Your subject!',
    Body => 'Some comment!',
    ArticleType => 'note-internal', # note-internal|note-external|note-report
},
# your program (/path/to/your/program) will be executed like
# "/path/to/your/program $TicketNumber $TicketID" ARG[0] will
# be the ticket number and ARG[1] the ticket id
CMD => '/path/to/your/program',
# DELETE ticket from database and filesystem
Delete => 1,
},

```

15.2.1.1. Spam Tickets schließen

Wenn Sie auf Ihrer System-E-Mail-Adresse Spam Nachrichten erhalten und Sie nicht viel Zeit damit verschwenden wollen, legen Sie einfach eine Queue (bspw. Spam) an und verschieben Sie diese Spam-Tickets in diese Queue. Wenn der GenericAgent.pl ausgeführt wird, werden alle offenen Tickets in dieser Queue automatisch geschlossen.

Beispiel für die Konfigurationsdatei: Kernel/Config/GenericAgent.pm

```

[...]
# --
# [name of job] -> close all tickets in queue spam
# --
'close spam' => {
    # get all tickets with this properties
    Queue => 'spam',
    States => ['new', 'open'],
    Locks => ['unlock'],
    # new ticket properties (no option is required, use just the options
    # which should be changed!)
    New => {
        # new queue
        Queue => 'spam',
        # possible states (closed successful|closed unsuccessful|open|new|removed)
        State => 'closed successful',
        # new ticket owner (if needed)
        Owner => 'root@localhost',
        # if you want to add a Note
        Note => {
            From => 'GenericAgent',
            Subject => 'spam!',
            Body => 'Closed by GenericAgent.pl because it is spam!',
        },
    },
},
[...]

```

15.2.1.2. Tickets löschen

Wenn Sie Tickets (aus der Datenbank und dem Dateisystem) löschen möchten, verwenden Sie das folgende:

Beispiel für die Konfigurationsdatei: Kernel/Config/GenericAgent.pm

```

[...]
# --
# [name of job] -> close and delete all tickets in queue delete
# --
'delete' => {
    # get all tickets with this properties
    Queue => 'delete',
    States => ['new', 'open'],

```

```

        Locks => ['unlock'],
        # new ticket properties (no option is required, use just the options
        # witch should be changed!)
        New => {
            # DELETE!
            Delete => 1,
        },
    },
    [...]

```

15.2.1.3. Tickets von "tricky" nach "experts" verschieben

Verschieben von Tickets aus der Queue "tricky" in die Queue "experts" und anhängen einer Notiz.

Beispiel für die Konfigurationsdatei: Kernel/Config/GenericAgent.pm

```

[...]
```

```

# --
# [name of job] -> move all tickets from tricky to experts
# --
'move tickets from tricky to experts' => {
    # get all tickets with this properties
    Queue => 'tricky',
    States => ['new', 'open'],
    Locks => ['unlock'],
    # new ticket properties
    New => {
        Queue => 'experts',
        Note => {
            From => 'GenericAgent',
            Subject => 'Moved!',
            Body => 'Moved from "tricky" to "experts" because it was not possible to find a sollution!',
            ArticleType => 'note-internal', # note-internal|note-external|note-report
        },
    },
},
[...]
```

15.2.1.4. Verschieben von Tickets von abc nach experts und Änderung der Priorität

Verschieben von Tickets aus der Queue "abc" in die Queue "experts" und ändern der Priorität von "3 normal" auf "4 hoch".

Beispiel für die Konfigurationsdatei: Kernel/Config/GenericAgent.pm

```

[...]
```

```

# --
# [name of job] -> move all tickets from abc to experts and change priority
# --
'move all abc priority "3 normal" tickets to experts and change priority'=> {
    # get all tickets with this properties
    Queue => ['abc'],
    States => ['new', 'open'],
    Locks => ['unlock'],
    Priorities => ['3 normal'],
    # new ticket properties
    New => {
        Queue => 'experts',
        Priority => '4 high',
    },
},
[...]
```

15.2.1.5. Verschieben von eskalierten Tickets nach experts und ausführend eines Kommandos

Wenn es eskalierte Tickets gibt, werden diese in die Queue experts verschoben und anschließend ein Kommando ausgeführt.

Beispiel für die Konfigurationsdatei: Kernel/Config/GenericAgent.pm

```
[...]
# --
# [name of job] -> move all tickets from xyz to experts
# --
'move escalation ticket to experts and execute CMD' => {
    # get all tickets with this properties
    Queue => 'xyz',
    Escalation => 1,
    # new ticket properties
    New => {
        Queue => 'experts',
        # your program (/path/to/your/program) will be executed like
        # "/path/to/your/program $TicketNumber $TicketID" ARG[0] will
        # be the ticket number and ARG[1] the ticket id
        CMD => '/path/to/your/program',
    },
},
[...]
```

15.2.1.6. Löschen aller Tickets mit dem Betreff "I love you" in der Queue "abc"

Löschen aller Tickets mit dem Betreff "I love you" in der Queue "abc".

Beispiel für die Konfigurationsdatei: Kernel/Config/GenericAgent.pm

```
[...]
'delete all tickets with subject "VIRUS 32" in queue abc' => {
    # get all tickets with this properties
    Queue => 'abc',
    # From => '%spam@example.com%',
    # To => '%support@example.com%',
    # Cc => '%client@example.com%',
    Subject => '%VIRUS 32%',
    # Body => '%testing case 1245%',
    # new ticket properties
    New => {
        # DELETE!
        Delete => 1,
    },
},
[...]
```


Chapter 16. Übersetzung in verschiedene Sprachen

Das OTRS Webfrontend unterstützt verschiedene Sprachen. Die Übersetzungen befinden sich in den Dateien, die unter `Kernel/Language/*.pm` zu finden sind.

Beginnend mit dem Release 1.3 werden die Übersetzungen von mehreren Dateien eingelesen.

1. `Kernel/Language/$Locale.pm` - Standard, Diese Datei wird zuerst gelesen.
2. `Kernel/Language/$Locale_$Action.pm` - Übersetzungen für Zusatzmodule wie den Kalender. Überschreibt die Standardwerte..
3. `Kernel/Language/$Locale_Custom.pm` - Ihre eigenen Übersetzungen und Zusätze. Der Inhalt dieser Datei überschreibt die beiden vorhergehenden.

16.1. Neue Übersetzungsdateien

Wenn sie OTRS in eine Sprache übersetzen möchten, die derzeit noch nicht vorliegt, so müssen Sie nur 5 Dinge tun:

- Laden Sie sich eine aktuelle Deutsch-Übersetzung (`Kernel/Language/de.pm`) aus dem CVS (<http://cvs.otrs.org/>) herunter.
- Ändern Sie den Paketnamen (von `"package Kernel::Language::de;"` zu `"package Kernel::Language::xy;"`) und übersetzen sie jedes Wort und jeden Satz.
- Fügen Sie die neue Sprache ins System ein, indem sie das Folgende in Ihre `Kernel/Config.pm` einfügen:
`"$Self->{DefaultUsedLanguages}->{xy} = 'Sprache XY';"`
- Wenn Sie `mod_perl` verwenden, starten Sie Ihren Webserver neu und Ihre Sprache steht zur Verfügung.
- Senden Sie die neue Übersetzung an [feedback\[at\]otrs.org](mailto:feedback[at]otrs.org) - Vielen Dank!

Jetzt können Sie die neue Sprache im Einstellungsfenster auswählen.

Example 16-1. `Kernel/Language/de.pm` - alte Datei

```
# --
# Kernel/Language/de.pm - provides de language translation
# Copyright (C) 2002-2003 Martin Edenhofer xxxxxxxxxxxxxxxxxxxxxxxx
# --
# $Id: language-translation.sgml,v 1.5 2004/09/21 20:50:28 martin Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --
package Kernel::Language::de;

use strict;

use vars qw($VERSION);
$VERSION = '$Revision: 1.5 $';
$VERSION =~ s/^\.*:\s(\d+\.\d+)\s.*$/$1/;

# --
sub Data {
    my $Self = shift;
    my %Param = @_;
    my %Hash = ();

    # $$START$$
    # Last translation Fri Jan 3 20:39:15 2003 by

    # possible charsets
    $Self->{Charset} = ['iso-8859-1', 'iso-8859-15', ];
    # date formats (%A=WeekDay;%B=LongMonth;%T=Time;%D=Day;%M=Month;%Y=Year;)
    $Self->{DateFormat} = '%D.%M.%Y %T';
    $Self->{DateFormatLong} = '%A %D %B %T %Y';
    $Self->{DateInputFormat} = '%D.%M.%Y - %T';

    %Hash = (
        # Template: AAABase
```

```

' 2 minutes' => ' 2 Minuten',
' 5 minutes' => ' 5 Minuten',
' 7 minutes' => ' 7 Minuten',
'10 minutes' => '10 Minuten',
'15 minutes' => '15 Minuten',
'AddLink' => 'Link hinzufügen',
'AdminArea' => 'AdminBereich',
'all' => 'alle',
'All' => 'Alle',
'Attention' => 'Achtung',
[...]
```

An:

Example 16-2. Kernel/Language/xy.pm - Neue Datei

```

# --
# Kernel/Language/fr.pm - provides fr language translation
# Copyright (C) 2002 Bernard Choppy xxxxxxxxxxxxxxxxxxxxxxxx
# Copyright (C) 2002-2003 Nicolas Goralski xxxxxxxxxxxxxxxxxxxxxxxx
# --
# $Id: language-translation.sgml,v 1.5 2004/09/21 20:50:28 martin Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --
package Kernel::Language::fr;

use strict;

use vars qw($VERSION);
$VERSION = '$Revision: 1.5 $';
$VERSION =~ s/^\.*:\s(\d+\.\d+)\s.*\$/\$/;
# --
sub Data {
    my $Self = shift;
    my %Param = @_;
    my %Hash = ();

    # $$START$$
    # Last translation Fri Jan  3 20:40:04 2003 by

    # possible charsets
    $Self->{Charset} = ['iso-8859-1', 'iso-8859-15', ];
    # date formats (%A=WeekDay;%B=LongMonth;%T=Time;%D=Day;%M=Month;%Y=Year;)
    $Self->{DateFormat} = '%D.%M.%Y %T';
    $Self->{DateFormatLong} = '%A %D %B %T %Y';
    $Self->{DateInputFormat} = '%D.%M.%Y - %T';

    %Hash = (
    # Template: AAABase
    ' 2 minutes' => ' 2 minutes',
    ' 5 minutes' => ' 5 minutes',
    ' 7 minutes' => ' 7 minutes',
    '10 minutes' => '10 minutes',
    '15 minutes' => '15 minutes',
    'AddLink' => 'Ajouter un lien',
    'AdminArea' => 'Zone d\'administration',
    'all' => 'tout',
    'All' => 'Tout',
    'Attention' => 'Attention',
    [...]
```


16.2. Übersetzungen für Aktionen

Die Datei für Aktionen überschreibt die Standardwerte aus Kernel/Language/\$Locale.pm, obwohl sie eher dafür gedacht ist, Übersetzungen für Zusatzmodule bereitzustellen. Sie kommt daher nur in Benutzung, wenn ein bestimmtes Modul geladen ist.

Example 16-3. Kernel/Language/de_Calendar.pm

```
# --
package Kernel::Language::de_Calendar;
# --
use strict;

use vars qw($VERSION);
$VERSION = '$Revision: 1.5 $';
$VERSION =~ s/^\$.*:\W(.*)\W.+?$/\1/;

# --
sub Data {
    my $Self = shift;
    my %Param = @_;

    # $$START$$

    $Self->{Translation}->{'Calendar'} = 'Termine';

    # $$STOP$$
}
# --
1;
```

16.3. Eigene Übersetzungen und Zusätze

Diese Datei wird zuletzt eingelesen, ergo können und sollten Sie hier Einstellungen der vorherigen überschreiben. Sie können auch eigene, neue Übersetzungen hinzufügen.

Example 16-4. Kernel/Language/de_Custom.pm

```
# --
package Kernel::Language::de_Custom;
# --
use strict;

use vars qw($VERSION);
$VERSION = '$Revision: 1.5 $';
$VERSION =~ s/^\$.*:\W(.*)\W.+?$/\1/;

# --
sub Data {
    my $Self = shift;
    my %Param = @_;

    # $$START$$

    # Re-Definitions
    $Self->{Translation}->{'Lock'} = 'Bearbeiten';
    $Self->{Translation}->{'Unlock'} = 'Freigeben';

    # Additions
    $Self->{Translation}->{'Model#'} = 'Modell-Nr.';
    $Self->{Translation}->{'spare part number'} = 'Ersatzteil-Nummer';

    # $$STOP$$
}
# --
1;
```


Chapter 17. Anpassung des Front-End

OTRS macht es möglich, das Erscheinungsbild des Systems (Front-End) an die eigenen Bedürfnisse anzupassen. Das Zauberwort lautet: dtl (Dynamic Template Language). Alle Front-End Masken befinden sich im Verzeichnis: `~otrs/Kernel/Output/HTML/<THEME>/*.dtl`. Standardmäßig ist das "Standard" Motiv ausgewählt.

Sie haben also die Möglichkeit, Ihr OTRS so anzupassen, wie Sie es möchten. Sie können aber auch ein eigenes Motiv entwerfen.

17.1. Die dtl-Syntax

Kommentar

Ein Kommentar wird durch ein einfaches "#"-Zeichen eingeleitet.

```
# --  
# das ist ein Kommentar  
# --
```

Eine Variable setzen

```
<dtl set $Data{"Test1"} = "German">
```

Note: `$Data{"xyz"}` existiert nur in dieser dtl-Datei wobei `$Env{"xyz"}` in allen dtl-Dateien existiert. Neu: `$Config{"xyz"}` ist eine nicht nur "read only" und existiert im gesamten Programm! (2002-05-22 / 0.5 BETA5)

Ausgabe einer Variablen

Um eine Variable auf dem Monitor auszugeben, verwenden Sie einfach:

```
$Data{"xyz"} oderr $Env{"xyz"}
```

Text Übersetzung

```
$Text{"Das wird übersetzt"}
```

Achten Sie darauf, dass die Übersetzung in den entsprechenden Übersetzungsdateien unter `"$HOME_OTRS/Kernel/Language/*.pm"` existiert. Wenn diese nicht existiert, wird der gegebene Text angezeigt.

Bedingungen

```
<dtl if ($Text{"Lock"} eq "Lock") { $Data{"FrontendLanguage"} = "English"; }>
```

Es gibt lediglich die Möglichkeit etwas in `$Data{"xyz"}` und `$Env{"xyz"}` zu speichern.

Auf eine Konfigurationsoption zugreifen - \$Config{}

```
$Config{"Sendmail"}
```

Gemeinsame Umgebungsvariablen - \$Env{}

```
$Env{"SessionID"} --> Die aktuelle Session ID  
$Env{"Time"} --> Die aktuelle Zeit z.B. 'Thu Dec 27 16:00:55 2001'  
$Env{"CGIHandle"} --> Die aktuelle CGI-Datei z.B. 'index.pl'  
$Env{"UserCharset"} --> Der aktuelle Zeichensatz z.B. 'iso-8859-1'  
$Env{"Baselink"} --> Der Basislink --> index.pl?SessionID=...  
$Env{"UserFirstname"} --> z.B. Dirk  
$Env{"UserLastname"} --> z.B. Hohndel  
$Env{"UserLogin"} --> z.B. mgg@x11.org  
$Env{"UserIsGroup[users]"} = Yes --> Benutzergruppen (useful for own links)  
$Env{"UserIsGroup[admin]"} = Yes
```

```
$Env{"Action"} --> Die aktuelle Aktion  
$Env{"Subaction"} --> Die aktuelle Subaktion
```

Inkludieren eines ander dtl Datei - `$Include{}`

Um css.dtl inkludieren:

```
$Include{"css"}
```

System-Aufruf

Um den Output eines Systemkommandos zu bekommen, verwenden Sie:

```
# execute system call  
<dtl system-call $Data{"uptime"} = "uptime">  
  
# print  
$Data{"uptime"}  
  
oder  
  
# execute system call  
<dtl system-call $Data{"procinfo"} = "procinfo | head -nl ">  
  
# print  
$Data{"procinfo"}
```

Beispiele

```
# set variable  
<dtl set $Data{"Test1"} = "English">  
  
# print variable  
Echo: $Data{"Test1"}  
  
# condition  
<dtl if ($Text{"Lock"} ne "Lock") { $Data{"Test2"} = "Not English!"; }>  
  
# print result  
Result: $Data{"Test1"}  
  
oder  
  
# translation test  
Lock: $Text{"Lock"}  
  
# config options  
Sendmail: $Config{"Sendmail"}
```

17.2. Beispiele für dtl-Dateien

17.2.1. Motd.dtl

Wenn Sie diese Datei anpassen möchten, verwenden Sie bitte diese Datei: `~otrs/Kernel/Output/HTML/Standard/Motd.dtl`

```
<p>  
Das ist die Nachricht des Tages. Sie können dies in der Datei Kernel/Output/HTML/Standard/Motd.dtl ändern.  
</p>
```

17.2.2. Login.dtl

Dies ist ein Beispiel für das Loginfenster. Wenn Sie dieses abändern möchten, verwenden Sie bitte folgende Datei:

~otrs/Kernel/Output/HTML/<THEME>/Login.dtl

```
# --
# http headers
# --
Content-Type: text/html; charset=$Env{"UserCharset"};
X-Powered-By: OTRS - Open Ticket Request System (http://otrs.org)

# --
# html comment
# --

<!-- OTRS: Copyright 2002, OTRS Project. This Software is under the GPL. -->
<!--      Web: http://otrs.org/ - Lists: http://lists.otrs.org/      -->
<!--      GNU Public License: http://www.gnu.org/licenses/gpl.txt  -->

# --
# set some html variables
# --
<dtl set $Env{"BGCOLOR"} = "#BDDDF" >
<dtl set $Env{"BGTableColor0"} = "#000000" >
<dtl set $Env{"BGTableColor1"} = "#FFFFFF" >
<dtl set $Env{"BGTableColor2"} = "#EEEEEE" >
<dtl set $Env{"FontColor0"} = "#000000" >
<dtl set $Env{"FontColor1"} = "#FFFFFF" >
<html>
<head>
  <title>OTRS :: $Text{"$Data{"Title"}}</title>
</head>
<!-- end header -->
<body bgcolor="$Env{"BGCOLOR"}" text="$Env{"FontColor0"}">

<center>

<p>
<font color="red">$Data{"Message"}</font>
</p>

<p>
<form action="$Env{"CGIHandle"}" method="post" enctype="application/x-www-form-urlencoded">
<input type="hidden" name="Action" value="Login">

<table border="0" cellspacing="0" cellpadding="3" width="240">
<tr bgcolor="$Env{"BGTableColor0"}">
  <td align="center"><font color="$Env{"FontColor1"}"><b>$Text{"Welcome to OTRS"}</b></font></td>
</tr>
<tr>
  <td align="center" bgcolor="$Env{"BGTableColor1"}">
    <table cellspacing="8" cellpadding="2">
      <tr>
        <td>Username:</td>
        <td><input type="text" name="User" value="$Data{"User"}" size="18"></td>
      </tr>
      <tr>
        <td>Password:</td>
        <td><input type="password" name="Password" size="18"></td>
      </tr>
    </table>
    <input type="submit" value="$Text{"Login"}">
  </td>
</tr>
</table>
</form>
</p>

# --
# Message of the day data!
# --
$Data{"Motd"}
```

```

</center>

</body>
</html>

```

17.2.3. Header.dtl

Dies ist die Standardkopfzeile einer jeden OTRS Seite. Wenn Sie diese anpassen möchten, verwenden Sie bitte die folgende Datei: `~otrs/Kernel/Output/HTML/<THEME>/Login.dtl`

```

# --
# http headers
# --
Content-Type: text/html; charset=$Env{"UserCharset"};

# --
# html comment
# --

<!-- OTRS: Copyright 2002, OTRS Project. This Software is under the GPL. -->
<!--      Web: http://otrs.org/ - Lists: http://lists.otrs.org/      -->
<!--      GNU Public License: http://www.gnu.org/licenses/gpl.txt  -->
# --
# set some html variables
# --
<dtl set $Env{"BGColor"} = "#BDDDFD">
<dtl set $Env{"BGTableColor0"} = "#000000">
<dtl set $Env{"BGTableColor1"} = "#FFFFFF">
<dtl set $Env{"BGTableColor2"} = "#EEEEEE">
<dtl set $Env{"FontColor0"} = "#000000">
<dtl set $Env{"FontColor1"} = "#FFFFFF">
<dtl set $Env{"Box0"} = "[ ">
<dtl set $Env{"Box1"} = " ]">
# --
# check refresh
# --
<dtl if ($Data{"Refresh"} ne "") { $Data{"MetaHttpEquiv"} = "<meta http-equiv='refresh' content='$Data{"Refresh"}'"/>
# --
<html>
<head>
    <title>OTRS :: $Text{"$Data{"Title"}}</title>
</head>
<!-- end header -->

```

17.3. Erstellen eines neuen Motivs

Um ein neues Motiv zu erstellen, erstellen Sie zuerst ein neues Motivverzeichnis (`mkdir ~otrs/Kernel/Output/HTML/NewTheme/`). Kopieren Sie alle existierenden Motive in das neue Verzeichnis (`cp ~otrs/Kernel/Output/HTML/Standart/*.dtl ~otrs/Kernel/Output/HTML/NewTheme/`).

Modifizieren Sie die dtl-Dateien nach belieben.

Legen Sie einen neuen Datenbankeintrag für dieses Motiv an. Dies muss über SQL erfolgen.

```

shell:~> mysql -u root -p some-pass otrs
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.48-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> INSERT INTO theme
->      (theme, valid_id, create_time, create_by, change_time, change_by)
->      VALUES

```

```
->      ('NewTheme', 1, current_timestamp, 1, current_timestamp, 1);  
mysql>
```

Fertig. Öffnen Sie die Einstellungsseite und ändern Sie das Motiv.

Wenn Sie Ihr neues Motiv standardmäßig verwenden möchten (login,...), dann fügen Sie bitte die folgende Konfigurationseinstellung in Ihre Konfigurationsdatei (Kernel/Config.pm) ein.

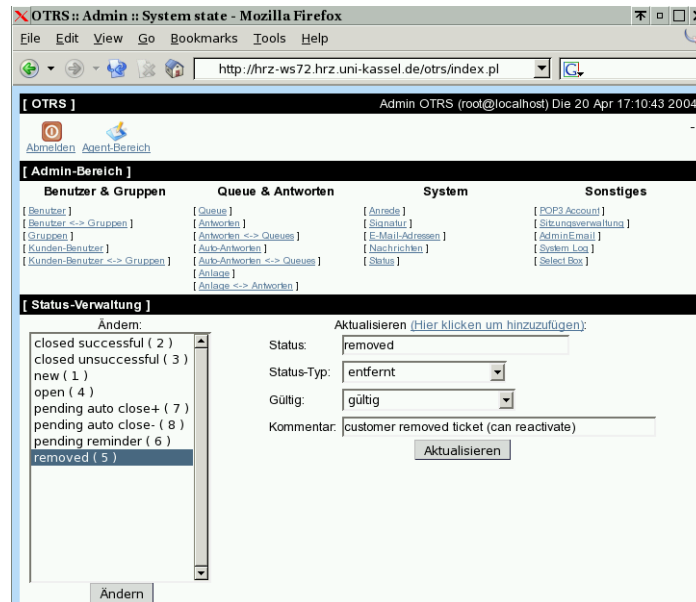
```
$Self->{DefaultTheme} = 'NewTheme' ;
```


Chapter 18. Anpassen des Ticket Status (ab OTRS 1.1)

OTRS erlaubt es Ihnen, die Ticket-Status zu verändern oder neue Status hinzuzufügen. Hierbei gibt es zwei wichtige Optionen. Zum Einen den Namen des Status "state-name" und zum Zweiten den Type des Status "state-type".

- Die standardmäßig voreingestellten Status lauten: 'neu', 'offen', 'erfolgreich geschlossen', 'erfolglos geschlossen', 'warten auf erfolgreich abschließen', 'warten auf erfolglos abschließen' und 'warten zur Erinnerung'.
- Jeder Status besteht aus einem Namen ("state-name") und einem Typen ("state-type"). Der Name ist frei wählbar.

Benutzen Sie einfach das Admin-Interface (Status) wenn Sie einen Status ändern oder hinzufügen möchten.



Beachten Sie, dass Sie bei Änderungen am Status "neu - new" auch die entsprechenden Änderungen in der Konfigurationsdatei "Kernel/Config.pm" vornehmen.

```
[...]
# PostmasterDefaultState
# (The default state of new tickets.) [default: new]
$self->{PostmasterDefaultState} = 'new';

# CustomerDefaultState
# (default state of new customer tickets)
$self->{CustomerDefaultState} = 'new';
[...]
```

Auch bei Änderungen am Status "offen - open" sind Änderungen in der Konfigurationsdatei von Nöten!

```
[...]
# default phone new state
$self->{PhoneDefaultNewNextState} = 'open';

# PostmasterFollowUpState
# (The state if a ticket got a follow up.) [default: open]
$self->{PostmasterFollowUpState} = 'open';
[...]
```


Chapter 19. Anpassen der Ticket Priorität (ab OTRS 1.1)

Wenn Sie die Ticket-Priorität anpassen / ändern wollen, arbeiten Sie einfach die nächsten Schritte ab. Derzeit gibt es hierzu leider kein Maske im Web-Frontend.

- So erhalten Sie die aktuellen Prioritäten:

```
mysql> select id, name from ticket_priority;
```

```
+-----+-----+
| id | name          |
+-----+-----+
| 1  | 1 very low   |
| 2  | 2 low        |
| 3  | 3 normal     |
| 4  | 4 high       |
| 5  | 5 very high  |
+-----+-----+
```

```
5 rows in set (0.00 sec)
```

```
mysql>
```

Wichtig: Das Attribut "id" bestimmt die Reihenfolge der Prioritäten. => 1 entspricht dem Minimum und 5 (oder höher) repräsentiert das Maximum.

Die Nummer, mit welcher der Name der Priorität beginnt, wird verwendet, damit die Reihenfolge korrekt umgesetzt wird.

- Anpassen/Ändern der Prioritäten via SQL. Z.B.:

```
mysql> UPDATE ticket_priority SET name = '3 default' WHERE id = 3;
```

Wenn Sie diesen SQL-Befehl ausführen, wird die Priorität "3 normal" in Zukunft "3 default" lauten.

- Beachten Sie bitte, dass Sie die Änderungen bezüglich der Priorität auch in der Konfigurationsdatei (Kernel/Config.pm) nachpflegen.

```
[...]
# PostmasterDefaultPriority
# (The default priority of new tickets.) [default: '3 normal']
$self->{PostmasterDefaultPriority} = '3 default';
[...]
# default phone priority
$self->{PhoneDefaultPriority} = '3 default';
[...]
# CustomerDefaultPriority
# (default priority of new customer tickets)
$self->{CustomerDefaultPriority} = '3 default';
[...]
```


Chapter 20. Multi Hosting

OTRS ist in der Lage, verschiedene Gruppen und Domains zu verwalten.

20.1. Multi Gruppen

Dies ist ein Beispiel für das Setup eines Unternehmens mit einem Helpdesk, einer Sales- und einer Billing-Abteilung. Jede Abteilung darf keine Tickets der anderen beiden Abteilungen sehen oder bearbeiten. Die Queues der jeweiligen Abteilung sind exklusiv zugeordnet!

Hierfür müssen Sie das folgende in die Konfigurationsdatei (Kernel/Config.pm) einfügen:

```
# ChangeOwnerToEveryone -> useful for ASP
# (Possible to change owner of ticket to everyone) [0|1]
$self->{ChangeOwnerToEveryone} = 0;

# ShowCustomerSelection
# (show customer selection in phone and change customer view
# - disable this for ASP!) [0|1]
$self->{ShowCustomerSelection} = 0;
```

System - E-Mail-Adressen (1/2): Sie müssen zuerst neue System - E-Mail-Adressen für jede Abteilung anlegen (z.B. helpdesk@beispiel.de, sales@beispiel.de und billing@beispiel.de). Hinweis: Beim anlegen der Adressen können Sie die Queues noch nicht zuweisen, da diese noch nicht angelegt wurden.

Gruppen: Fügen Sie für jede Abteilung eine neue Gruppe in Ihrem OTRS ein. In diesem Beispiel also die Gruppen: helpdesk, sales und billing.

Queue: Fügen Sie für jede Abteilung die gewünschte Anzahl von Queues ein (beachten Sie die Namensgebung - z.B. Gruppe-Thema). In diesem Beispiel also die Queues: helpdesk-products, helpdesk-raw, sales-raw, sales-products, billing-raw, billing-claim... Anschließend müssen Sie die Queues den entsprechenden Gruppen zuweisen!

System - E-Mail-Adressen (2/2): Bringen Sie Ihre neuen System - E-Mail-Adressen auf den aktuellen Stand, indem Sie die richtige Queue zuweisen. Hinweis: Natürlich müssen Sie die Zugänge zu den POP3 Konten einpflegen, damit OTRS Ihre E-Mails auch empfangen kann. (Admin-Oberfläche/POP3-Konto)

Benutzer: Pflegen Sie nun noch die Benutzer ins System ein. Ordnen Sie diese nun den entsprechenden Gruppen zu (Also Sales-Benutzer in die Sales-Gruppe usw.).

Hinweis: Wenn Sie einen Benutzer einer Abteilung zudem noch in die "admin"-Gruppe einordnen, dann kann dieser alle Einstellungen des Systems ändern. Also auch Benutzer oder Antworten ändern oder hinzufügen. Sie sollten also Administratorrechte nur und ausschließlich Administratoren einräumen!

Ihr Multi-Gruppen Setup ist nun abgeschlossen.

20.2. Multi Setup

Wenn Sie auf einem Rechner mehrerer OTRS-Systeme parallel im Einsatz haben möchten (Produktion, Test usw.), dann sollten Sie sicherstellen, dass jedes OTRS eine eindeutige SystemID zugewiesen bekommen hat.

Hierzu müssen Sie die folgenden Zeilen in Ihre Konfigurationsdatei einfügen:

```
# SystemID
# (The identify of the system. Each ticket number and
```

```
# each http session id starts with this number)
$self->{SystemID} = n;
```

Ersetzen Sie n durch eine ausreichend große Zahl (integer) die eindeutig im gesamten System ist. Beachten Sie darüber hinaus, dass sie mit mod_perl-1 nur eine Instanz gleichzeitig im Einsatz haben könne. Wenn Sie mehr als eine OTRS-Instanz gleichzeitig betreiben möchten, starten Sie entweder einen weiteren Webserver (auf einer anderen IP-Adresse oder einem anderen Port) oder verwenden Sie einfach mod_perl-2.

Hinweis: Wenn sie das Skript "scripts/apache-perl-startup.pl" verwenden, ändern sie das "use lib" Verzeichnis. (z.B. opt/otrs oder opt/otrs2)

Chapter 21. Leistungsverbesserung

Eine erschöpfende Liste verschiedener Techniken, um das Maximum an Leistung aus Ihrem OTRS System zu bekommen: Konfiguration, Programmierung, Speichernutzung und mehr.

21.1. OTRS

Im folgenden finden Sie die Optionen, die Leistung des Systems via OTRS selbst zu verbessern.

21.1.1. TicketIndexModule

Zur Verfügung stehen zwei Hintergrundmodule für den Ticket Index.

Kernel/Config.pm

```
[...]
$self->{TicketIndexModule} = 'Kernel::System::Ticket::IndexAccelerator::RuntimeDB';
[...]
```

- Kernel::System::Ticket::IndexAccelerator::RuntimeDB (Standard), generiere jede Queue-Ansicht dynamisch aus der Ticket Tabelle. Sie werden keine Probleme mit der Leistung bekommen bis zu etwa 60.000 Tickets (oder 6000 offenen) in Ihrem System.
- Kernel::System::Ticket::IndexAccelerator::StaticDB, das leistungsfähigste Modul. Es sollte ab 80.000 Tickets oder mehr als 6000 offenen eingesetzt werden. Benutzt eine extra ticket_index Tabelle, arbeitet wie eine Ansicht (View). Führen Sie bin/RebuildTicketIndex.pl zum erstmaligen Aufbau des Index' aus.

21.1.2. TicketStorageModule

Es stehen zwei Module für das Speichern der Tickets und Artikel bereit.

Kernel/Config.pm

```
[...]
$self->{TicketStorageModule} = 'Kernel::System::Ticket::ArticleStorageDB';
[...]
```

- Kernel::System::Ticket::ArticleStorageDB (Standard), speichere Anhänge & Co. in der Datenbank. Merke: Benutzen Sie diese Option nicht für größere Systeme.

Pro: Ist der Benutzer, unter dem der Webserver läuft, nicht der OTRS Benutzer, können Sie mit diesem Modul Dateiberechtigungsprobleme vermeiden.

Contra: Es ist nicht wirklich nett, Anhänge in Ihrer Datenbank zu speichern. Achten Sie darauf, dass Ihre Datenbank das kann. Für MySQL setzen Sie in dessen Konfiguration bspw. "set-variable = max_allowed_packet=8M", um 8 MB große Objekte zu speichern (Standard ist 2M).

- Kernel::System::Ticket::ArticleStorageFS, speichere Anhänge & Co. im lokalen Filesystem ab. Merke: Benutzen Sie dies für große Installationen.

Pro: Schneller!

Contra: Der Benutzer, unter dem der Webserver läuft, sollte der OTRS Benutzer sein (Dateisystemberechtigungen!).

Note: Bei OTRS 1.2 oder höher, kann man das TicketStorageModule im Betrieb ändern.

21.2. Datenbank

Einstellungen sind immer spezifisch für die jeweils eingesetzte Datenbank. Bei Problemen lesen Sie die Dokumentation und fragen Sie Ihren Datenbankadministrator.

21.2.1. MySQL

Wenn Sie den Tabellentyp MyISAM (Standard) benutzen, und einen großen Teil einer Tabelle gelöscht haben, oder wenn Sie sehr viele Änderungen an einer Tabelle mit Zeilen variabler Länge vorgenommen haben (Tabellen mit VARCHAR, BLOB oder TEXT Spalten), sollten Sie die Datendateien mit dem "optimize" Kommando behandeln.

Dies bietet sich an, wenn MySQL viel CPU Zeit braucht. Optimieren Sie die Tabellen ticket, ticket_history und article.

```
mysql>optimize table ticket;  
mysql>optimize table ticket_history;  
mysql>optimize table article;
```

21.2.2. PostgreSQL

PostgreSQL konfigurieren Sie am besten in der postgresql.conf Datei in Ihrem PostgreSQL Datenverzeichnis. Hier gibt es Hilfe dazu: <http://www.varlena.com/varlena/GeneralBits/Tidbits/perf.html>

http://www.varlena.com/varlena/GeneralBits/Tidbits/annotated_conf_e.html Ist die Leistung immer noch nicht genügend, empfehlen wir, Fragen auf der "PostgreSQL Performance Mailing Liste" zu stellen. Die Teilnehmer der PostgreSQL Liste sind sehr freundlich und können wahrscheinlich helfen. <http://www.postgresql.org/lists.html>.

21.3. Webserver

Natürlich empfehlen wir mod_perl (<http://perl.apache.org/>). Es ist sehr viel schneller (etwa um den Faktor 100) als pures CGI, braucht aber auch mehr Speicher. Ihr httpd wird mit mod_perl also größer sein.

21.3.1. Datenbank Verbindung

Sie können die Datenbank Verbindung bereits beim Start des httpd-Prozeß herstellen lassen - dies spart ebenso Zeit (siehe auch README.webserver).

21.3.2. Vorgeladene Module - startup.pl

Nutzen Sie das Start Skript scripts/apache-perl-startup.pl, um die Perl Module vorzuladen (siehe README.webserver).

21.3.3. Perl Module bei Änderung neu laden

Standardmäßig wird Apache::StatINC (mod_perl1) oder Apache::Reload (mod_perl2) in scripts/apache-httpd.include.conf eingesetzt. Deaktivieren Sie es und die Geschwindigkeit steigt um etwa 8%. Ab nun müssen Sie den Webserver neu starten, wenn Sie irgendetwas ändern!

21.3.4. Die richtige Strategie wählen

Bei wirklich großen Installationen (über 1000 neue Tickets am Tag, über 40 Agenten) ist es eine sehr gute Idee, den Artikel "Choosing the Right Strategy" (in englisch) zu lesen (<http://perl.apache.org/docs/1.0/guide/strategy.html>).

21.3.5. mod_gzip

Falls Ihre Bandbreite ein wenig schmal sein sollte, benutzen Sie mod_gzip (http://www.schroepl.net/projekte/mod_gzip/). Eine HTML-Seite von 45k wird mod_gzip auf etwa 7k zusammendrücken - nett.

21.3.6. mod_dosevasive

Um http DoS (Denail of Service) Angriffe zu blocken kann mod_dosevasive benutzt werden.
(<http://www.nuclearelephant.com/projects/dosevasive/>).

Chapter 22. Das FAQ System

22.1. Was sind FAQs?

'FAQ' ist die Abkürzung von 'Frequently Asked Question(s)' - Häufig Gestellte Fragen. Je länger Sie Ihr OTRS System betreiben, desto eher werden Sie den Eindruck nicht los, zuviel Zeit mit dem immer wiederkehrenden Beantworten von immer gleichen Fragen zu vertun.

Wir haben daher eine FAQ Datenbank in OTRS (mind. 1.2.x) integriert, die helfen wird, dies Dilemma zu lösen. Nun können Ihre Kunden erst versuchen, dort eine Antwort zu finden - bevor sie Ihr Support Team überladen. Wenn sie Sie doch kontaktieren, können Sie aus Ihrer OTRS Installation heraus ein breites Wissen strukturiert präsentieren. Ihre Agenten tun in der Zeit etwas Produktiveres.

22.2. Vorarbeiten

Die FAQ Datenbank muss nicht noch aktiviert werden, aber Sie müssen Mitglied der 'faq' Gruppe sein, um damit arbeiten zu können. Fügen Sie sich als Mitglied dieser Gruppe hinzu, bevor Sie weiterlesen, oder Sie werden das Folgende nicht nachvollziehen können.

Volle Kontrolle über den FAQ-Bereich inkl. der Möglichkeit neue Einträge zu erstellen, erhalten Sie, indem Sie sich bzw. den entsprechenden Agenten 'rw' Rechte geben. Man benötigt mindestens 'ro' Rechte um die FAQs lesen zu können. Ihre Kundenbenutzer benötigen dieses Recht daher auch.

22.3. Schnittstellen

Agent: Der Agent loggt sich in den FAQ-Bereich ein über das FAQ Symbol in der Menüleiste des OTRS (sofern die Administration eines bereitgestellt hat), oder über diesen Link: <http://localhost/otrs/index.pl?Action=FAQ>. Ist kein Symbol vorhanden, kann es sehr gut sein, dass Sie keine 'ro' Rechte in der FAQ-Gruppe besitzen.

Agenten mit 'rw' Rechten können alle Arten von FAQs sehen, einschliesslich der 'internen'.

Kunden: Ihre Kunden können die FAQs einsehen, indem sie einen ähnlichen Eintrag in ihrer Menüleiste nutzen, der hierauf zeigt: <http://localhost/otrs/customer.pl?Action=CustomerFAQ>.

Kunden können die öffentlichen und Kunden-FAQs sehen. Letztere tragen den Status 'external', um sie von den 'public', also öffentlichen zu unterscheiden.

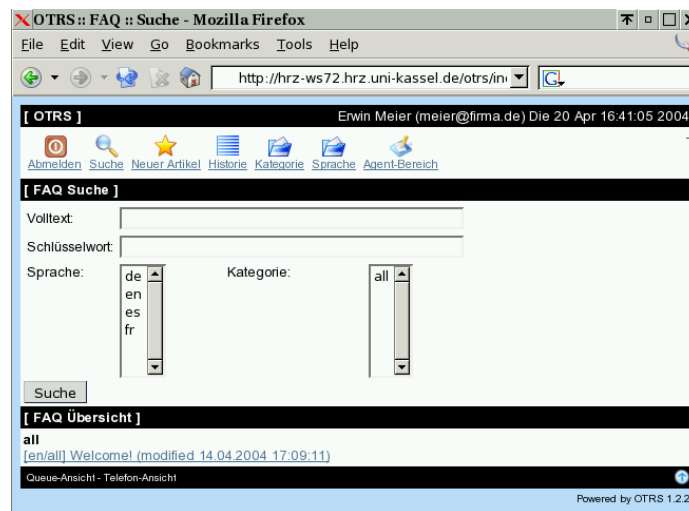
Abhängig von der Konfiguration Ihres Webservers darf jeder unter <http://localhost/otrs/faq.pl> wenigstens die öffentlichen FAQs einsehen.

22.4. FAQ Funktionen

22.4.1. FAQ Funktionen für Agenten

Neben den wohlbekannten Symbolen zum Ausloggen aus dem OTRS sowie für den Agenten- und Administrationsbereich,

beinhaltet der FAQ-Bereich die folgenden Funktionen zum Verwalten der FAQs.



Die verschiedenen Funktionen sollen eine intuitive Handhabung erlauben. Eine Kurzbeschreibung folgt:

- **Suche:** Sie können im Volltext und unter den Schlüsselworten suchen lassen, kombiniert mit Spracheinstellungen und verschiedenen Kategorien.
Sie können auch einen Eintrag aus der Liste aller FAQs wählen, wie unter der Suchfunktion angeboten.
- **Neuer Artikel:** Hier erstellen Sie neue Artikel, d.h. neue FAQ Einträge.
- **Historie:** Die Gesamthistorie lässt Sie sehen, wer welchen FAQ Eintrag erstellt/geändert hat und wann.
- **Kategorie:** Um verschiedene Wissensbereiche voneinander abzugrenzen, verwenden Sie Kategorien. Stellen Sie sich vor, Sie legen eine Kategorienliste an, die zum Beispiel Ihre Produktpalette abbildet.
- **Sprache:** Hier definieren Sie die in den FAQs verwendeten Sprachen.
- **Status:** Hier können Sie noch mehr als die drei anfänglichen Stati 'internal', 'external' and 'public' definieren.

22.4.2. FAQ Funktionen für Kunden

Ihre Kunden können die FAQs durchsuchen, ähnlich der Agentensuche. Auch können Sie aus einer Liste verfügbarer FAQs auswählen.

22.4.3. FAQ Funktionen für öffentliche FAQs

Was für Kunden gilt, gilt für öffentliche FAQs ebenso: Sie können durchsucht werden und es existiert eine Liste von ihnen, aus der ausgewählt werden kann. Sie müssen lediglich nicht eingeloggt sein, um <http://localhost/otrs/faq.pl> zu betrachten. Noch einmal, dies hängt von der Konfiguration ihres Webservers ab.

Öffentliche FAQs können auch von Kunden eingesehen und gesucht werden.

Chapter 23. Fehlersuche

Wir teilen diesen Abschnitt pro Distribution auf. Natürlich können Sie Ihr eigenes System mit Ihrem eigenen Linux aufsetzen, aber verstehen Sie bitte, dass wir keinen Support für derartiges bieten können. OTRS ist zu kompliziert und hat zu viele Verbindungen zu anderen Komponenten des System, dass wir schon froh sind, Zugang zu Mechanismen wie RPM zu haben. Wir versuchen, soviele Plattformen wie möglich zu unterstützen, aber Zeit ist eine wertvolle Resource.:-)

23.1. Generelle Probleme mit OTRS auf SuSE Linux

Der allererste Schritt sollte sein, <http://www.suse.de/de/support/download/updates/> zu besuchen, eine Seite, auf der sich Patches und Aktualisierungen zu Apache, MySQL, Perl für Ihr SuSE Linux finden. Bitte achten Sie besonders auf Aktualisierungen von Apache, MySQL, Perl und natürlich OTRS. Benutzern von YaST2 sollte es möglich sein den YaST Online Update Mechanismus zu nutzen. Anderenfalls laden Sie die -rpm-Dateien herunter und deinstallieren Sie das alte Paket mit "rpm --nodeps -e otrs" und installieren Sie es erneut mit "rpm -i foo.rpm"

Der zweite Schritt sollte ein besuch der *OTRS Homepage* (<http://www.otrs.org/>) sein. Wir stellen dort die letzten Aktualisierungen und Anleitungen bereit.

Als Drittes mögen Sie eine e-Mail an das Entwicklerteam von OTRS schreiben. Wir werden Sie nach besten Kräften unterstützen.

23.1.1. SuSE Linux 8.0

Unglücklicherweise ist das RPM auf der SuSE 8.0 Distribution fehlerhaft (kein SuSE Fehler). Bitte laden Sie die neueste Version von unserer Homepage (<http://www.otrs.org/>) und installieren Sie es mit YaST oder manuell via

```
# altes Paket deinstallieren
shell> rpm --nodeps -e otrs
# neues Paket installieren
shell> rpm -i new-otrs.rpm
```

23.1.2. SuSE Linux und Postfix

Postfix kommt leider nicht mit Procmail vorkonfiguriert, Sie müssen es in der /etc/postfix/main.cf aktivieren:

```
[...]
mailbox_command = /usr/bin/procmail
[...]
```

Starten Sie Postfix neu via 'rcpostfix restart'.

23.2. Generelle Probleme mit OTRS auf Mandrake

OTRS arbeitet wie erwartet auf einem Mandrake System, wenigstens auf einem 9.1 oder 9.2er. Da wir derzeit noch keine RPMs anbieten, müssen Sie den Schritten zur manuellen Installation des Tarballs in der Dokumentation folgen.

23.2.1. Aktualisieren Ihres Systems

Bevor Sie sich daran machen, OTRS auf einem Mandrake System zu installieren, fühlen Sie sich aufgefordert, via Mandrake's Softwarepakete-Management Ihr System auf den neuesten Stand zu bringen. Sie finden eine bequeme Oberfläche in Mandrake's Kontrollzentrum, oder führen Sie diese Kommandos als Superuser über die Kommandozeile aus:

```
shell:~ # urpmi.update --update --auto --auto-select
shell:~ # urpmi --update --auto-select
```

Dies erfordert eine funktionsfähige Aktualisierungsquelle. Wollen Sie den GWDG.de server nutzen, führen Sie etwas wie dies aus (tauschen Sie <rel> mit Ihrer Mandrake Version):

```
shell:~ # urpmi.addmedia update_source ftp://ftp.gwdg.de/pub/linux/mandrake/updates/<rel>/RPMS
```

23.3. Generelle Probleme mit OTRS auf anderen Distributionen (z.B. Redhat)

Ganz ehrlich, wir haben nicht wahrhaft versucht OTRS auf einem RedHat System laufen zu lassen. Es sollte kein Problem darstellen, aber es gibt noch kein fertiges RPM - wir arbeiten dran. Dasselbe gilt für andere, nicht erwähnte Distributionen.

23.4. Probleme mit Apache

Meist, wenn Probleme mit Apache auftreten, kompilieren die Leute sich ihren eigenen, oft sehr speziellen Apache. Bitte schön, tun Sie's, aber falls Sie bei Problemen anfragen, verweisen wir bestimmt auf die mit Ihrer Distribution ausgelieferten Version.

23.4.1. Interner Server Fehler (Internal Server Error)

Überprüfen Sie die Syntax der Datei index.pl, um den Fehler zu finden:

```
shell:~ # cd ~otrs
```

Es ist wichtig, dass Sie sich im \$HOME des 'otrs' Benutzers befinden.

```
shell:/opt/otrs # perl -cw bin/cgi-bin/index.pl
bin/cgi-bin/index.pl syntax OK
shell:/opt/otrs #
```

Falls Fehler kommen, müssen Sie höchswahrscheinlich fehlende Perl Module von CPAN (<http://www.cpan.org/>) nachinstallieren.

So installieren Sie die CPAN Module via der cpan shell:

```
shell:~ # perl -MCPAN -e shell;
-- (evtl. muessen Sie CPAN erst konfigurieren) --

cpan shell -- CPAN exploration and modules installation (v1.59_54)
ReadLine support enabled

cpan> install Digest::MD5
[installing Digest::MD5 module]
cpan>
```

Überprüfen Sie die Syntax nun noch einmal (perl -cw bin/cgi-bin/index.pl) und installieren Sie dann weitere, fehlende Module, falls nötig.

23.4.2. Fehler: Kann nicht zur Datenbank verbinden!

Die englische Übersetzung der fehlermeldung lautet: 'Error: Can't connect to database!'

Wenn Sie diese Meldung in Ihrem Browser erhalten, nachdem Sie die Datenbankeinstellungen verändert haben, starten Sie den Webserver neu. mod_perl liest die Konfigurationsdatei von OTRS nur beim Start ein.

Bleibt die Meldung bestehen, schauen Sie sich das Fehler-Log Ihres Webservers an. (z.B. /var/log/httpd/error_log).

23.4.3. FreeBSD, PostgreSQL und Apache (install_driver(Pg) failed)

Alles funktioniert prima, nur bekomme ich bei jeder Benutzung der Web-Schnittstelle jedes Mal: [...] Software error: install_driver(Pg) failed: [Thu Sep 19 16:52:18 2002] index.pl: [Thu Sep 19 16:52:18 2002] index.pl: Can't load '/usr/local/lib/perl5/site_perl/5.005/i386-freebsd/auto/DBD/Pg/Pg.so' for module DBD::Pg: Shared object "libpq.so.2" not found at /usr/libdata/perl/5.00503/DynaLoader.pm line 169. [Thu Sep 19 16:52:18 2002] index.pl: [Thu Sep 19 16:52:18 2002] index.pl: [Thu Sep 19 16:52:18 2002] index.pl: [Thu Sep 19 16:52:18 2002] index.pl: at (eval 124) line 3 Perhaps a required shared library or dll isn't installed where expected at /usr/local/otrs/bin/cgi-bin/./Kernel/System/DB.pm line 67 [...]

Um dieses Problem zu lösen, setzen Sie in der `httpd.conf` Ihres Apache die Umgebungsvariable `LD_LIBRARY_PATH=/usr/local/pgsql/lib`. Setzen Sie die Zeile `"SetEnv LD_LIBRARY_PATH /usr/local/pgsql/lib"` in Ihrer `httpd.conf` und starten Sie den Webserver neu (siehe <http://lists.otrs.org/pipermail/otrs/2002-September/000248.html>).

23.5. Probleme mit MySQL

Bitte überprüfen Sie Ihre Passwörter sehr genau, was die häufigste Fehlerursache ist. Bei Passwortfehlern hilft meist ein Zurücksetzen des Systems.

23.5.1. Überprüfen Sie die Datenbankverbindung

Um die Datenbankverbindung zu überprüfen, benutzen Sie den Befehl `'~otrs/bin/CheckDB.pl'`. Ist die Ausgabe ein "It looks Ok!", haben Sie keine Probleme mit der Verbindung zu Ihrer Datenbank. Anderenfalls achten Sie auf die Fehlerausgabe.

23.5.2. Fehlermeldung: Access denied for user: 'otrs@localhost'

Überprüfen Sie das Passwort des Datenbankbenutzers in der Datei `Kernel/Config.pm`. Wenn Sie das Passwort nicht (mehr) kennen, setzen Sie es neu:

```
shell> mysql -u root -p
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37 to server version: 3.23.48-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> GRANT ALL PRIVILEGES ON otrs.* TO otrs@localhost IDENTIFIED BY "yourpw" WITH GRANT OPTION;
mysql>
```

Laden Sie die Benutzertabellen neu:

```
$shell> mysqladmin -u root -p reload
```

(oder starten Sie MySQL neu) und ändern Sie das neue Passwort in der `Kernel/Config.pm`.

Notiz: Wenn Sie `mod_perl` nutzen, starten Sie den Webserver neu, da `mod_perl` die Konfigurationsdateien nur beim Start einliest, also auch und vor allem `Kernel/Config.pm`.

23.5.3. root Passwort für MySQL unbekannt

Siehe: http://www.mysql.com/documentation/mysql/bychapter/manual_Problems.html#Resetting_permissions

23.5.4. <http://localhost/otrs/installer.pl>

Wenn Sie die Datenbank erneut aufsetzen wollen, können Sie den Web-Installer aufrufen, indem Sie zu <http://localhost/otrs/installer.pl> navigieren.

Notiz: Dies funktioniert im Moment nur für SuSE RPMs.

23.6. Sonstige Probleme

23.6.1. Probleme beim Empfang von e-Mails

Wenn alles ausser dem Mailempfang funktioniert, überprüfen Sie, ob das `~otrs/bin/PostMaster.pl` Programm korrekt funktioniert. leiten Sie eine der beiden Beispiel-eMails durchm den `PostMaster.pl`-Prozess (`~otrs/doc/test-email-*.box` oder `/usr/share/doc/packages/otrs/test-email-*.box`).

Beispiel:

```
shell:~ # cat /usr/share/doc/packages/otrs/test-email-1.box | /opt/otrs/bin/PostMaster.pl
shell:~ #
```

Falls Fehler kommen, müssen Sie diese beheben. Falls keine Fehler kommen, und Sie die eMail in der Postmaster Warteschlange sehen, ist Ihr MTA nicht für den Betrieb mit Procmail konfiguriert (Logfiles checken!), oder das \$HOME des OTRS Benutzers ist verkehrt gesetzt (überprüfen Sie \$HOME und \$HOME/.procmailrc).

23.6.2. root Passwort für OTRS unbekannt

Erhalten Sie die Meldung "Anmeldung fehlgeschlagen! Benutzername oder Passwort falsch.", so setzen Sie das root-Passwort für OTRS zurück:

```
shell:~ # mysql -u root -p otrs
mysql> UPDATE system_user SET pw='roK20XGbWESM' where login='root@localhost';
mysql>
```

Somit setzen Sie das root-Passwort auf 'root'. Der Eintrag ist per crypt() verschlüsselt, siehe 'man 3 crypt'.

Chapter 24. Entwicklerhandbuch

Dieses Kapitel beinhaltet weitere Informationen zum schreiben von einfachen OTRS Modulen.

24.1. Kernmodule

Die Kernmodule des Systems befinden sich unter: \$OTRS_HOME/Kernel/System/. Dort befinden sich die Module: Log (Kernel::System::Log), Ticket (Kernel::System::Ticket), Auth (Kernel::System::Auth), User (Kernel::System::User), Email (Kernel::System::Email),

24.2. Programmierstil

Um eine einheitliche Entwicklung innerhalb des OTRS Projekts gewährleisten zu können, sind einige Rahmenbedingungen und Vorgaben bezüglich des Stils zu beachten.

1. Formatierung: Ein Tabulator erstreckt sich über 4 Leerzeichen. (----). Bitte beachten Sie auch die Position der Klammern. Hierzu zwei Beispiele:

```
if ($Condition) {
    Foo();
}
else {
    Bar();
}
```

und

```
while ($Condition == 1) {
    Foo();
}
```

2. Namensgebung: Die Namensgebung und die Kommentierung des Programmcodes erfolgen in englischer Sprache. Variablen, Objekte und Methoden werden durch Nomen oder durch aneinandersetzen von Nomen beschrieben. Achten Sie darauf, dass jedes Wort mit einem Großbuchstaben beginnt. (z. B.: '@TicketIDs' oder '\$Output' oder 'BuildQueueView()')

3. Kopfzeile im Code: Fügen Sie die folgende Kopfzeile in jede Datei ein:

```
# --
# (file name) - a short decription what it does.
# Copyright (C) (year) (name of author) (email of author)
# --
# $Id: developer-guide.sgml,v 1.2 2004/01/14 08:18:11 jens Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
```

4. Sonstiges: All das, was sich auf <http://www.perl.com/CPAN-local/doc/manual/html/pod/perlstyle.html> bezieht.

Darüber hinaus können Sie 'pb' den 'Perl Beautifier' zur Verschönerung Ihres Programmcodes verwenden. (<http://www.arachnoid.com/lutusp/ftp/cgi/pb.txt>). Es ist sehr zu empfehlen.

24.3. Fehlerbeseitigung

...

Chapter 25. Entwicklerhandbuch - OTRS Module schreiben

Dieses Kapitel wird Ihnen weitere Informationen über das Schreiben von einfachen OTRS Modulen geben.

25.1. Agenten-Oberfläche Benachrichtigungsmodule

Mit Agent-Benachrichtigungsmodulen können Sie Agenten neue Information zukommen lassen. Die Navigation sieht normalerweise folgendermaßen aus:



Dies ist ein Beispiel für ein simples Benachrichtigungsmodul. Speichern Sie es unter Kernel/Output/HTML/NotificationMotd.pm.

```
# --
# Kernel/Output/HTML/NotificationMotd.pm - message of the day
# Copyright (C) 2003 Hans Mueller mail@example.com
# --
# $Id: developer-guide-custom-modules.sgml,v 1.5 2004/04/23 08:21:17 martin Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --

package Kernel::Output::HTML::NotificationMotd;

use strict;

# --
sub new {
    my $Type = shift;
    my %Param = @_;

    # allocate new hash for object
    my $Self = {};
    bless ($Self, $Type);

    # get needed objects
    foreach (qw(ConfigObject LogObject DBObject LayoutObject UserID)) {
        $Self->{$_} = $Param{$_} || die "Got no $_!";
    }
    return $Self;
}

# --
sub Run {
    my $Self = shift;
    my %Param = @_;
    return $Self->{LayoutObject}->Notify(Info => 'Tägliche Neuigkeiten!');
}

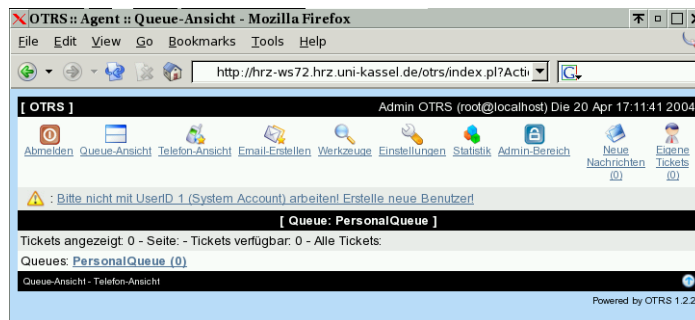
# --
```

1;

Wenn Sie dieses Module nun auch verwenden möchten, so müssen Sie es in der Konfigurationsdatei (Kernel/Config.pm) aktivieren und Ihren Webserver neu starten (letzteres nur, wenn Sie mod_perl verwenden).

```
# Frontend::NotifyModule - module name (50-Motd)
$self->{'Frontend::NotifyModule'}->{'50-Motd'} = {
    Module => 'Kernel::Output::HTML::NotificationMotd',
};
```

Im Ergebnis sehen Sie eine Nachricht in der Agenten Oberfläche:



Normalerweise möchten Sie Nachrichten mit einem gewissen Sinngehalt in Ihre Benachrichtigungsfunktion aufnehmen. Diese Art der Benachrichtigung ist beispielsweise sinnvoll für Eskalationsinformationen.

Die standardmäßigen Benachrichtigungsmodule sind die folgenden (Kernel/Config.pm):

```
# agent interface notification module to check the used charset
$self->{'Frontend::NotifyModule'}->{'1-CharsetCheck'} = {
    Module => 'Kernel::Output::HTML::NotificationCharsetCheck',
};

# agent interface notification module to check the admin user id
# (don't work with user id 1 notification)
$self->{'Frontend::NotifyModule'}->{'2-UID-Check'} = {
    Module => 'Kernel::Output::HTML::NotificationUIDCheck',
};

# show online agents
$self->{'Frontend::NotifyModule'}->{'3-ShowAgentOnline'} = {
    Module => 'Kernel::Output::HTML::NotificationAgentOnline',
};

# show online customers
$self->{'Frontend::NotifyModule'}->{'4-ShowCustomerOnline'} = {
    Module => 'Kernel::Output::HTML::NotificationCustomerOnline',
};
```

25.2. Kunden-Oberfläche Benachrichtigungsmodule

Mit Kunden-Benachrichtigungsmodulen können Sie Agenten neue Information zukommen lassen. Es sind die gleichen Module wie die Agent-Oberfläche Benachrichtigungsmodule.

Die standardmäßigen Benachrichtigungsmodule sind die folgenden (Kernel/Config.pm):

```
# show online agents
$self->{'CustomerFrontend::NotifyModule'}->{'1-ShowAgentOnline'} = {
    Module => 'Kernel::Output::HTML::NotificationAgentOnline',
};
```

25.3. Ticketnummer Generator

Wenn Sie Ihr eigenes Format in der Ticketnummer nutzen wollen, erstellen Sie Ihr eigenes Ticketnummer-Modul.

Die Module befinden sich unter "Kernel/System/Ticket/Number/*.pm". Schauen Sie in der Konfigurationsdatei unter "Ticket number format" nach für die Standard-Module.

Ein Beispiel eines einfachen Ticketnummer-Moduls, speichern Sie es unter Kernel/System/Ticket/Number/Simple.pm. Sie benötigen nur zwei Funktionen, CreateTicketNr() und GetTNByString():

```
# --
# Ticket/Number/Simple.pm - a ticket number auto increment generator
# Copyright (C) 2003 Hans Mueller mail@example.com
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --
# Note:
# available objects are: ConfigObject, LogObject and DBObject
#
# Generates auto increment ticket numbers like ss.... (e. g. 1010138, 1010139, ...)
# --

package Kernel::System::Ticket::Number::Simple;

use strict;

use vars qw($VERSION);
$VERSION = '$Revision: 1.5 $';
$VERSION =~ s/^\$.*:\W(.*)\W.+?$/\1/;

sub CreateTicketNr {
    my $Self = shift;
    my $JumpCounter = shift || 0;
    # get needed config options
    my $CounterLog = $Self->{ConfigObject}->Get('CounterLog');
    my $SystemID = $Self->{ConfigObject}->Get('SystemID');
    my $MinSize = 5;
    my $Count = 0;
    if (-f $CounterLog) {
        open (COUNTER, "< $CounterLog") || die "Can't open $CounterLog: $!";
        my $Line = <COUNTER>;
        ($Count) = split(/:/, $Line);
        close (COUNTER);
        if ($Self->{Debug} > 0) {
            $Self->{LogObject}->Log(
                Priority => 'debug',
                Message => "Read counter from $CounterLog: $Count",
            );
        }
    }
    # count auto increment ($Count++)
    $Count++;
    $Count = $Count + $JumpCounter;
    # write new count
    if (open (COUNTER, "> $CounterLog")) {
        flock (COUNTER, 2) || warn "Can't set file lock ($CounterLog): $!";
        print COUNTER $Count . "\n";
        close (COUNTER);
        if ($Self->{Debug} > 0) {
            $Self->{LogObject}->Log(
                Priority => 'debug',
                Message => "Write counter: $Count",
            );
        }
    }
    else {
        $Self->{LogObject}->Log(
            Priority => 'error',
            Message => "Can't write $CounterLog: $!",
        );
    }
}
```

```

        die "Can't write $CounterLog: $!";
    }
    # pad ticket number with leading '0' to length $MinSize (config option)
    while (length($Count) < $MinSize) {
        $Count = "0".$Count;
    }
    # create new ticket number
    my $Tn = $SystemID . $Count;
    # Check ticket number. If exists generate new one!
    if ($Self->CheckTicketNr(Tn=>$Tn)) {
        $Self->{LoopProtectionCounter}++;
        if ($Self->{LoopProtectionCounter} >= 1000) {
            # loop protection
            $Self->{LogObject}->Log(
                Priority => 'error',
                Message => "CounterLoopProtection is now $Self->{LoopProtectionCounter}!".
                    " Stopped CreateTicketNr()!",
            );
            return;
        }
        # create new ticket number again
        $Self->{LogObject}->Log(
            Priority => 'notice',
            Message => "Tn ($Tn) exists! Creating new one.",
        );
        $Tn = $Self->CreateTicketNr($Self->{LoopProtectionCounter});
    }
    return $Tn;
}
# --
sub GetTNByString {
    my $Self = shift;
    my $String = shift || return;
    # get needed config options
    my $SystemID = $Self->{ConfigObject}->Get('SystemID');
    my $TicketHook = $Self->{ConfigObject}->Get('TicketHook');
    my $TicketDivider = $Self->{ConfigObject}->Get('TicketDivider') || ':';
    my $MinSize = 5;
    my $MaxSize = $MinSize + 5;
    # check ticket number
    if ($String =~ /^$TicketHook:+. {0,1}($SystemID\d{$MinSize,$MaxSize})\~FW/i) {
        return $1;
    }
    else {
        if ($String =~ /^$TicketHook:+. {0,1}($SystemID\d{$MinSize,$MaxSize})/i) {
            return $1;
        }
        else {
            return;
        }
    }
}
# --
1;

```

Um dieses Modul zu nutzen, fügen Sie Ihrer Kernel/Config.pm die folgenden Zeilen hinzu und starten Sie Ihren Webserver neu (falls Sie mod_perl benutzen):

```
$Self->{TicketNumberGenerator} = 'Kernel::System::Ticket::Number::Simple';
```

Nun wird Ihr Ticketnummer-Modul genutzt.

25.4. Module zur Authentifizierung von Agenten

OTRS kommt bereits mit mehreren Module zur Authentifizierung von Agenten (DB, LDAP und HTTPBasicAuth). Darüber hinaus können Sie Ihre eigenen Module entwickeln.

Die Module befinden sich unter "Kernel/System/Auth/*.pm". Schauen Sie in der Konfigurationsdatei unter "User Auth Backend" nach.

Ein Beispiel eines einfachen Ticketnummer-Moduls, speichern Sie es unter Kernel/System/Auth/Simple.pm. Sie benötigen nur drei Funktionen, new(), GetOption() and Auth(). Die Authentifizierung ist erfolgreich, wenn Auth() wahr (also 'true') ist.

```
# --
# Kernel/System/Auth/Simple.pm - provides the db authentication
# Copyright (C) 2001-2004 Martin Edenhofer martin+code at otrs.org
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --
# Note:
# available objects are: ConfigObject, LogObject and DBObject
# --

package Kernel::System::Auth::Simple;

use strict;

# --
sub new {
    my $Type = shift;
    my %Param = @_;

    # allocate new hash for object
    my $Self = {};
    bless ($Self, $Type);

    # check needed objects
    foreach (qw(LogObject ConfigObject DBObject)) {
        $Self->{$_} = $Param{$_} || die "No $_!";
    }

    # Debug 0=off 1=on
    $Self->{Debug} = 0;

    return $Self;
}
# --
sub GetOption {
    my $Self = shift;
    my %Param = @_;
    # check needed stuff
    if (!$Param{What}) {
        $Self->{LogObject}->Log(Priority => 'error', Message => "Need What!");
        return;
    }
    # module options
    my %Option = (
        PreAuth => 0,
    );
    # return option
    return $Option{$Param{What}};
}
# --
sub Auth {
    my $Self = shift;
    my %Param = @_;
    # check needed stuff
    if (!$Param{User}) {
        $Self->{LogObject}->Log(Priority => 'error', Message => "Need User!");
        return;
    }
}
```

```

# get params
my $User = $Param{User} || "";
my $Pw = $Param{Pw} || "";
my $RemoteAddr = $ENV{REMOTE_ADDR} || 'Got no REMOTE_ADDR env!';
my $UserID = "";
my $GetPw = "";
# sql query
my $SQL = "SELECT pw, user ".
    " FROM ".
    " users ".
    " WHERE ".
    " user = '$User'";
$Self->{DBObject}->Prepare(SQL => $SQL);
while (my @RowTmp = $Self->{DBObject}->FetchrowArray()) {
    $GetPw = $RowTmp[0];
    $UserID = $RowTmp[1];
}

my $Salt = $GetPw;
$Salt =~ s/^(...)*$/1/;
my $CryptedPw = crypt($Pw, $Salt);

# just a note
if (!$Pw) {
    $Self->{LogObject}->Log(
        Priority => 'notice',
        Message => "User: $User without Pw!!! (REMOTE_ADDR: $RemoteAddr)",
    );
    return;
}
# login note
elsif (((($GetPw)&&($User)&&($UserID)) && $CryptedPw eq $GetPw) {
    $Self->{LogObject}->Log(
        Priority => 'notice',
        Message => "User: $User logged in (REMOTE_ADDR: $RemoteAddr).",
    );
    return $User;
}
# just a note
elsif (($UserID) && ($GetPw)) {
    $Self->{LogObject}->Log(
        Priority => 'notice',
        Message => "User: $User with wrong Pw!!! (REMOTE_ADDR: $RemoteAddr)"
    );
    return;
}
# just a note
else {
    $Self->{LogObject}->Log(
        Priority => 'notice',
        Message => "User: $User doesn't exist or is invalid!!! (REMOTE_ADDR: $RemoteAddr)"
    );
    return;
}
}
# --

1;

```

25.5. Module zur Authentifizierung von Kundenbenutzern

Verfahren Sie wie bei den Modulen zur Authentifizierung von Agenten, die Module liegen diesmal unter Kernel/System/CustomerAuth/*.pm.

25.6. Module für Kundenbenutzer

...

25.7. PostMaster Filter Module

Siehe auch das Kapitel "PostMaster". Dieses Beispiel sortiert eMails von "email@example.com" in die Queue "Some::System::Queue" und setzt freien Text.

Für eine Übersicht über alle X-Header-Optionen, siehe "doc/X-OTRS-Headers.txt".

Ein Beispiel eines simplen Postmaster Filter Moduls. Speichern Sie unter Kernel/System/PostMaster/Filter/Simple.pm ab. Es werden nur zwei Funktionen benötigt, new() and Run():

```
# --
# Kernel/System/PostMaster/Filter/Simple.pm - sub part of PostMaster.pm
# Copyright (C) 2001-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: developer-guide-custom-modules.sgml,v 1.5 2004/04/23 08:21:17 martin Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --

package Kernel::System::PostMaster::Filter::Simple;

use strict;

use vars qw($VERSION);
$VERSION = '$Revision: 1.5 $';
$VERSION =~ s/^\.*:\s(\d+\.\d+)\s.*$/$1/;

# --
sub new {
    my $Type = shift;
    my %Param = @_;

    # allocate new hash for object
    my $Self = {};
    bless ($Self, $Type);

    $Self->{Debug} = $Param{Debug} || 0;

    # get needed objects
    foreach (qw(ConfigObject LogObject DBObject)) {
        $Self->{$_} = $Param{$_} || die "Got no $_!";
    }

    return $Self;
}

# --
sub Run {
    my $Self = shift;
    my %Param = @_;
    if ($Param{GetParam}->{From} =~ /email@example.com/i) {
        $Param{GetParam}->{'X-OTRS-Queue'} = 'Some::System::Queue';
        $Param{GetParam}->{'X-OTRS-TicketKey1'} = 'Planet';
        $Param{GetParam}->{'X-OTRS-TicketValue1'} = 'Sun';
    }
    return 1;
}

# --
1;
```

25.8. Generic Agent Modules

See also chapter "GenericAgent".

This module will notify the current ticket owner.

A example of a simple generic agent module, save it under Kernel/System/GenericAgent/Simple.pm. You just need 2 functions, new() and Run(). Run() will be get the ticket id:

```
# --
# Kernel/System/GenericAgent/Simple.pm - generic agent notifications
# Copyright (C) 2001-2004 Martin Edenhofer <martin+code otrs.org>
# --
# $Id: developer-guide-custom-modules.sgml,v 1.5 2004/04/23 08:21:17 martin Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --

package Kernel::System::GenericAgent::Simple;

use strict;
use Kernel::System::User;
use Kernel::System::Email;

use vars qw(@ISA $VERSION);
$VERSION = '$Revision: 1.5 $';
$VERSION =~ s/^\$.*:\W(.*)\W.+?$/\1/;

# --
sub new {
    my $Type = shift;
    my %Param = @_;

    # allocate new hash for object
    my $Self = {};
    bless ($Self, $Type);

    # check needed objects
    foreach (qw(DBObject ConfigObject LogObject TicketObject)) {
        $Self->{$_} = $Param{$_} || die "Got no $_!";
    }

    $Self->{UserObject} = Kernel::System::User->new(%Param);
    $Self->{EmailObject} = Kernel::System::Email->new(%Param);

    return $Self;
}

# --
sub Run {
    my $Self = shift;
    my %Param = @_;

    my %Ticket = $Self->{TicketObject}->GetTicket(%Param);

    my %User = $Self->{UserObject}->GetUserData(UserID => $Ticket{UserID});
    if ($User{UserEmail}) {
        $Self->{EmailObject}->Send(
            To => $User{UserEmail},
            Subject => "[${Ticket{TicketNumber}}] Ticket Notification!",
            Body => "Hi $User{Firstname}, some info ... about $Ticket{'TicketNumber'}".
            Loop => 1,
        );
    }

    return 1;
}

# --
1;
```

25.9. Agenten-Ticketberechtigungen Module

...

25.10. Kunden-Ticketberechtigungen Module

...

Chapter 26. FAQ - Häufig gestellte Fragen

1. Ich habe otrs.rpm auf SuSE 8.0 installiert, aber es funktioniert nicht. Warum?

Die standardmäßig in SuSE 8.0 mitgelieferte Installationsdatei ist fehlerhaft. (Dies ist kein Fehler von SuSE). Bitte schauen Sie auf unserer Website <http://otrs.org> (<http://www.otrs.org/>) nach und laden Sie sich die neueste Version der Installationsdatei. Sie können diese dann manuell oder über YaST installieren.

```
rpm -Uvh otrs.rpm
```

2. Was ist eigentlich OTRS?

OTRS ist ein Ticket Request System mit vielen Funktionalitäten um Kundenanrufe oder E-Mails zu managen.

3. Wofür steht die Abkürzung OTRS?

Open Trouble-Ticket Request System.

4. Was kostet OTRS?

OTRS selbst ist kostenlos zu haben (GPL (<http://www.gnu.org/copyleft/gpl.txt>)). Wenn Sie jedoch Beratungs- und/oder Supportbedarf haben, wenden Sie sich bitte an ((otrs.de)) (<http://www.otrs.de>).

5. Dieses Dokument verweist auf Agenten. Was ist mit einem Agenten im Kontext von OTRS gemeint?

Ein Agent ist ein "menschliches Wesen", das mit dem System arbeitet. Benutzer wäre eine alternative Beschreibung.

Natürlich kann ein Skript wie ein Agent agieren. Viel Spaß beim umsetzen!

6. Wie kann ich einen Benutzer, der nicht länger benötigt wird, aus dem System entfernen? (Frage von Andreas Haase - Mittwoch, 21. Aug 2002 11:17:01 +0200 (CEST))

OTRS arbeitet mit Datenbanken und somit auch mit identifizierenden Primärschlüsseln. Wenn Sie einen Benutzer (eine Queue, ...) löschen möchten, würden Sie damit auch wichtige Referenzinformationen löschen. Aus diesem Grund ist eine solche Funktionalität nicht in OTRS selbst implementiert.

Setzen Sie stattdessen einfach den Status des Benutzers auf "ungültig". Dann ist der Eintrag weiterhin zu sehen (z.B. in der Ticket-History), aber nicht aktiv und kann somit auch nicht mehr verwendet werden.

7. Kann OTRS eigentlich E-Mails empfangen?

Ja, sogar mit MIME Unterstützung.

Das \$HOME/bin/PostMaster.pl Skript empfängt die E-Mails und sortiert sie automatisch in die richtige Queue / zum richtigen Ticket.

8. Kann OTRS E-Mails versenden?

Ja. Automatische als auch persönliche Antworten können vom System verschickt werden.

Hinweis: Sie können System-E-Mail-Adressen im Adminbereich von OTRS hinzufügen, löschen und ändern.

9. Gibt es eine Volltextsuche?

Ja.

10. Ist OTRS multi-Benutzer und multi-Gruppen fähig?

Ja, aber natürlich!

11. Ist es möglich, verschiedene Domänen zu verwenden?

Ja, Sie können verschiedene E-Mail Adressen wie z.B. 'support@meinedomain.com', 'sales@meinedomain.com' und verschiedene Domänen 'marketing@kundendomain.de' mit einem System verwenden.

12. Kann ich OTRS nur mit SuSE Linux verwenden?

Nein. Allerdings ist es auf SuSE Linux entwickelt worden und somit haben wir offen gesagt nicht sehr viel auf Redhat, Debian, ... getestet.

Weitere Installationsinformationen finden Sie in diesem Dokument und unter <http://www.otrs.org/>. Wir wären froh, von Ihnen Feedback über eine Installation auf einem anderen System zu erhalten.

13. Welche Software wird benötigt?

Mindestens: Perl5 (<http://www.perl.com/>) (mit einigen CPAN (<http://www.cpan.org/>) Modulen), MySQL (<http://www.mysql.com>) und Apache (<http://www.apache.org>).

Nochmals: Sie sind auf der sicheren Seite, wenn Sie SuSE Linux als Betriebssystem verwenden. Es wird alles nötige bereitstellen.

14. Wie stabil ist OTRS?

OTRS ist ein stabiles System. Sie sollten also keine Probleme bekommen, aber garantieren können wir das nicht.

15. Welche Hardwareanforderungen stellt das System?

Sagen wir mal: IBM s390. *SCNR* ;-)

Einige unserer Testumgebungen sind Pentium II 300 mit 64 MB RAM. Diese verrichten einen wirklich guten Job. Natürlich gilt auch hier: Je mehr RAM und je schneller die CPU, desto besser.

16. Wie skaliert OTRS und wie groß kann es maximal werden?

Dies hängt von der verwendeten Hardware ab. Im Moment ist OTRS ein geschlossenes System. Mit ein wenig Arbeit können Sie aber einen Webserver-Cluster aufsetzen und die Datenbank auf eine eigene Maschine transferieren. Derzeit sind einige Cluster-Mechanismen in Planung. Diese stehen aber nicht ganz oben auf der Liste.

Es gibt OTRS Installationen, die 20.000 Tickets (E-Mails) am Tag bearbeiten und keine Anzeichen von Stress von sich geben.

17. Kann ich auch ein Oracle oder DB2 Datenbanksystem verwenden?

Derzeit unterstützen wir standardmäßig MySQL (<http://www.mysql.com/>) und darüber hinaus PostgreSQL (<http://www.postgresql.org/>). Natürlich vertrauen wir diesen genau so wie z.B. DB2. Jeder, der willens und in der Lage ist, OTRS auf eine andere Datenbank zu portieren, ist im OTRS-Team jederzeit willkommen!

Hinweis: Der Programmcode ist darauf ausgelegt, verschiedene Datenbanken zu unterstützen.

18. Welche Programmiersprache wurde/wird verwendet?

Perl (<http://www.perl.com/>) OO, SQL und dtl.

19. Wie lautet der standardmäßige Administrator-Zugang?

Benutzer: root@localhost, Passwort: root

Es ist natürlich eine sehr gute Idee, dieses Passwort sofort nach der Installation zu ändern!

20. Arbeitet OTRS mit mod_perl (<http://perl.apache.org/>)?

Aber natürlich.

21. Ist es möglich, OTRS für meinen speziellen Anwendungsfall anzupassen?

Sie können OTRS so anpassen wie Sie es für nötig halten. Nutzen Sie z.B. die Vorteile der dtl (dynamic template language) um das OTRS Erscheinungsbild anzupassen.

22. Kann OTRS auch auf Windows Betriebssystemen installiert werden?

Ja, es gibt sowohl eine Schritt-für-Schritt Anleitung als auch einen automatischen Installer für Windows.

23. Wie kann ich Mitglied in der OTRS Entwickler Community werden?

Jeder, der willens und in der Lage ist, uns zu unterstützen, ist mehr als willkommen. Schicken Sie uns einfach eine E-Mail.

24. Eigentlich mag ich OTRS, wohler wäre mir aber bei einem kommerziellen Produkt.

Leider können wir Ihnen hier nicht helfen.

25. Welchen Webbrowser benötige ich?

OTRS funktioniert mit allen verbreiteten Browsern (sicher aber mit: lynx, w3m, Netscape, Mozilla/Firefox, Opera, IE and Konqueror). Sie benötigen nicht einmal JavaScript- oder Java-Unterstützung.

26. Ist es möglich, ein eigenes Ticketnummernformat zu generieren?

Ja, das ist möglich. Sie können zwischen vier verschiedenen OTRS-Formaten wählen (Autowert ("SystemID.Counter" z.B. 1010138 oder 1010139), Datum ("Year.Month.Day.SystemID.Counter" z.B. 200206231010138 oder 200206231010139), Zufallszahl ("SystemID.Random" z.B. 100057866352 oder 103745394596) und Datum-Prüfsumme ("Year.Month.Day.SystemID.Counter.CheckSum" z.B. 2002070110101520 oder 2002070110101535)).

Darüber hinaus ist es möglich, ein eigenes Ticket-Format zu erstellen (vgl. OTRS Dokumentation).

27. Unterstützt OTRS den RFC 1297?

Ja, OTRS unterstützt diesen RFC.

28. Kann ich zusätzliche Ticket-Stati einfügen?

Ja, aber seien Sie vorsichtig mit den bereits existierenden Stati (OTRS benötigt einige Basis-Stati).

29. Kann ich mehr als ein OTRS auf einem Rechner laufen lassen?

Ja, das geht. Installieren Sie einfach ein zweites, drittes ... in einem anderen Verzeichnis (z.B. /opt/otrs01, /opt/otrs02, /opt/otrs03, ...) und konfigurieren Sie den Webserver entsprechend neu.

30. Welche Änderungen werden auf meinem Windows-System vorgenommen, wenn ich den otrs4win32-Installer verwende?

MySQL benötigt eine Konfigurationsdatei im Windows-Verzeichnis. Die Startmenue-Einträge werden in Ihrem Profil angelegt. Ansonsten werden keine weiteren Verzeichnisse oder Dateien außerhalb des Installations-Verzeichnisses geschrieben. Die Systemregistry wird nur dann angetastet, wenn Sie einen oder mehrere Server als System-Service installieren.

31. Warum werden bei der Installation unter Windows (Win32) keine Services installiert?

Wir wollen mit dem OTRS-Installer ein System zur Verfügung stellen, das so wenig wie möglich an der Betriebssystemkonfiguration ändert. Eine Service-Installation greift tief in Ihr System ein. Wenn Sie trotzdem System-Services installieren möchten, können Sie das bequem nach der Installation manuell über die Startmenue-Einträge nachholen.

32. Warum sollte ich dennoch System-Services (System-Dienste) verwenden?

Wir empfehlen die Verwendung von Diensten, wenn Sie OTRS produktiv einsetzen. Zum einen ist dies ein bequemer Weg, wenn die Dienste im Hintergrund laufen und keine lästigen Kontrollfenster die Arbeit stören, zum anderen muss kein Benutzer angemeldet sein, um den Dienst zu nutzen.

33. Ist es sinnvoll, alle Dateien in einem Verzeichnis zu speichern?

Es ist nichts falsch an einer solchen Lösung. Sie mögen vielleicht nicht darüber nachgedacht haben, aber: Es gibt Menschen, die nicht ganz daneben liegen, wenn sie sagen, dass jeder Server seine eigene Partition haben soll, eine Datenbank sollte auf einem RAID-System liegen, ein aktiver Webserver benötigt eine Spiegelpartition und so weiter... *Damit liegen Sie vollkommen richtig!* - also nur zu, verwenden Sie die entsprechenden Windows-Funktionen, um all diese Szenarien umzusetzen (Schlagwörter in diesem Zusammenhang sind *Dynamic Disks* und *DFS*). In anderen Worten, Sie können die Unterverzeichnisse in die verschiedenen Partitionen verschieben. Achten Sie aber darauf, dass Sie die Konfiguration von OTRS entsprechend anpassen. Sollten Fragen oder Probleme während der Partitionierung auftreten, so kontaktieren Sie uns einfach!

Glossar

Dieses Glossar beinhaltet OTRS Begriffe.

Benutzer Gruppen

Agenten

Die Menschen, die mit und an den Tickets arbeiten.

Admin / Administrator

Jeder Agent, der rw-Rechte in der Gruppe Admin besitzt.

Kunde

Der Berichterstatter eines Problems.

Web Oberfläche / Interface

Agenten Oberfläche / Interface

Web Oberfläche / Interface (<http://example.com/otrs/index.pl>) für den Agenten um mit den Tickets zu arbeiten.

Admin Oberfläche / Interface

Web Oberfläche (<http://example.com/otrs/index.pl?Action=Admin>) für den Administrator um Administratortasken wahrzunehmen.

Kunden Oberfläche / Interface

Web Oberfläche (<http://example.com/otrs/customer.pl>) für den Kunden um Ticket zu erstellen, zu senden und zu verfolgen.

Agenten Oberfläche / Interface

Mailbox

Überblick über alle erhaltenen und erfassten Tickets im System.

Telefon Ansicht

Eine Maske zur Erfassung neuer Tickets.

Queue Ansicht

Überblick über alle offenen und nicht gesperrten Tickets sortiert nach Queue. Die Sortierreihenfolge jeder Queue richtet sich nach der Priorität und dem Alter.

Ticket Historie

Ein detaillierter Überblick über jede Aktion (z.B. verschieben, sperren..) eines Tickets.

Ticket Zoom

Eine detaillierte Sicht (jeder Artikel) eines Tickets.

Sie haben 1 neue Nachricht(en) bekommen!

Eine neue Nachricht wartet in der persönlichen Queue des Agenten. Diese Nachricht erscheint dann, wenn der letzte Artikel zu einem Ticket nicht vom Agenten verfasst wurde. Diese Funktionalität hilft dem Agenten, sich in seinen Tickets zurecht zu finden. Die Nachricht verschwindet, wenn das Ticket beantwortet wurde.

Ticket

Ticket Nummer

Eindeutige Ticket Nummer. Referenz für Agenten und Kunden. Es ist möglich, das Ticketnummerformat zu ändern.

Ticket Alter

Alter des Tickets im System.

Ticket Status

Die Ticket Status sind z.B. neu oder offen. Es ist möglich, Ticket Status zu ändern, hinzuzufügen oder zu löschen.

Ticket Priorität

Wie wichtig ist ein Ticket! Ticket Prioritäten sind z.B. 3 normal oder 4 hoch. Es ist möglich, Prioritäten zu ändern, hinzuzufügen oder zu löschen.

Ticket Sperre

Ein Ticket ist entweder gesperrt oder freigegeben. D.h. es ist gesperrt, wenn ein Agent daran arbeitet. Zu dieser Zeit ist es für andere Agenten nicht zugänglich. Wenn ein Agent ein Ticket sperrt, wird er automatisch auch dessen Besitzer.

Ticket Besitzer

Der Bearbeiter des Tickets. Besitzer ist nicht gleich Sperre!

Ticket KundenID

Die KundenID ist die ID des Kunden. Standardmäßig wird die E-Mailadresse verwendet.

Ticket Kunden/Benutzer

Der Kunde/ Benutzer ist der Absender identifiziert durch die E-Mailadresse.

Ticket Eskalationszeit

Wenn eine Eskalationszeit für eine Queue (Admin Oberfläche -> Queue) gesetzt wurde und ein Ticket offen ist und nicht beantwortet wird, so erscheint nach diesem Zeitraum nur noch dieses Ticket in der Queue.

Artikel

Der Abschnitt des Tickets der Antworten, Notizen o.ä. enthält.

Artikel Art

Verschiedene Arten von Artikeln:

email-external: E-Mail zwischen Kunde und Agenten

email-internal: E-Mail zwischen Agenten (z.B. für Feedback, erscheinen nicht in der Kunden Oberfläche.)

email-notification-ext: E-Mail Benachrichtigung an den Kunden

email-notification-int: E-Mail Benachrichtigung an den Agenten (erscheinen nicht in der Kunden Oberfläche)

phone: Telefonnotiz

webrequest: Anfrage über die Kunden Oberfläche

note-internal: Notiz, die nur für Agenten zu sehen ist.

note-external: Notiz, die für Agenten und Kunden zu sehen.

note-report: Notiz zwecks Berichterstattung - sollte von Monitoring-Tools wie Nagios (<http://www.nagios.org/>) verwendet werden.

[Derzeit nicht in Benutzung: fax, sms]

Artikel Absender

Der Ersteller eines Artikels ('agent', 'customer' oder 'system').

Antwort

Modulare standard Antworten, die es Agenten ermöglichen, schnell zu antworten. Konfigurierbar je nach Queue.

Auto-Antwort

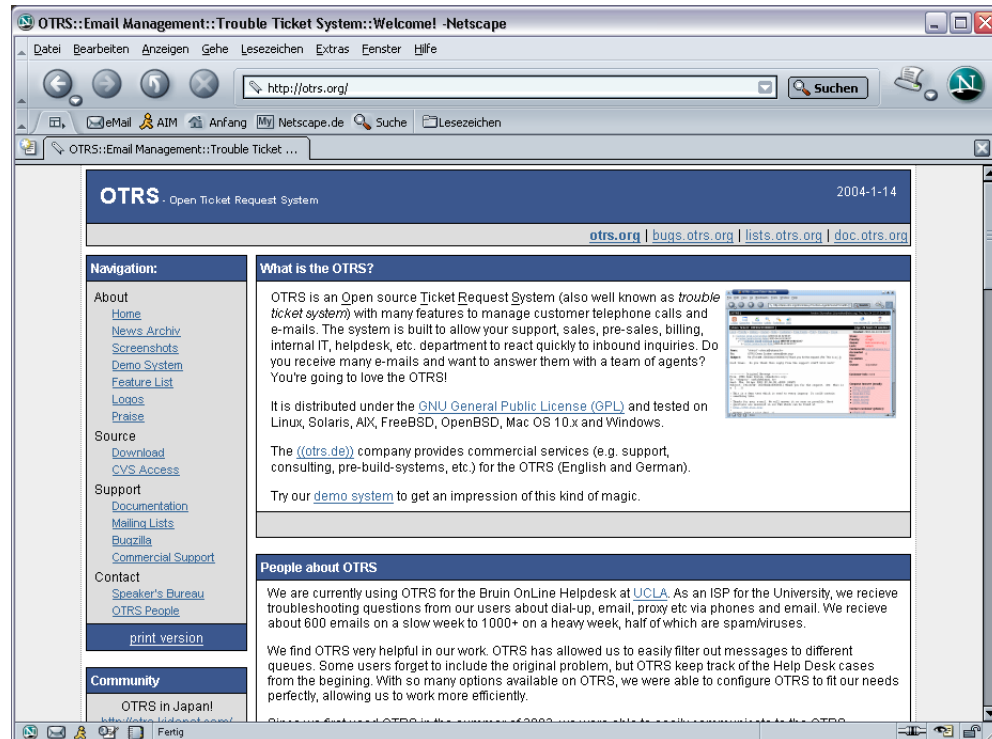
Automatisch generierte Antwort nachdem ein Kunde ein neues Ticket erstellt hat. Konfigurierbar je nach Queue.

Appendix A. Online Quellen

Wir unterstützen Sie mit den aktuellsten Informationen über OTRS und geben Ihnen die Möglichkeit uns mit Feedback zu versorgen.

A.1. Homepage

Unsere Homepage finden Sie unter <http://www.otrs.org/>.



A.2. Mailinglisten

Wir bieten sechs Haupt-Mailinglisten. Sie müssen eine Mailingliste abonnieren, bevor Sie einen Beitrag schreiben könne.

Die erste lautet: <announce at otrs.org>. Dies ist eine Liste mit wenigen Beiträgen, hauptsächlich aber Ankündigungen von neuen OTRS Releases oder Sicherheitsbelange.

Die zweite lautet: <otrs at otrs.org>. Dies ist eine Liste mit hohem Beiträgeaufkommen. Hier werden alle Arten von Fragen zu OTRS gestellt.

Die dritte lautet: <otrs-de at otrs.org>. Dies ist eine Liste mit hohem Beiträgeaufkommen. Hier werden alle Arten von Fragen zu OTRS in deutscher Sprache gestellt.

Die vierte lautet: <dev at otrs.org>. Dies ist eine Liste mit mittelmäßigem bis hohem Beiträgeaufkommen. Hier diskutieren die OTRS Entwickler alle Arten von Problemen und Fragestellungen.

Die fünfte lautet: <i18n at otrs.org>. Dies ist eine Liste mit wenigen Beiträgen. Hier werden alle Fragen bezüglich Sprache und Übersetzung geklärt. Wenn Sie also OTRS übersetzen möchten oder Fragen zur Übersetzung haben, ist hier Ihr Platz.

Die sechste lautet: <cvsv-log at otrs.org>. Die ist eine Liste mit extrem vielen Beiträgen, die automatisch von CVS generiert werden.

Um eine zu abonnieren besuchen Sie folgenden Link: <http://lists.otrs.org/>.

A.3. Fehlerverfolgung

Profis brauchen kein Fehlermanagement Werkzeug! ;-)

War ein Scherz. Um uns Fehler mitzuteilen, besuchen Sie uns unter: <http://bugs.otrs.org/>. Es ist uns bewusst, das bugzilla machmal nicht sehr komfortabel ist. Es ist aber derzeit das beste Fehlerverfolgungssystem, das wir finden konnten.



Indem Sie uns Fehler aufzeigen, helfen Sie uns sehr. Wir bitten Sie um Ihre Mithilfe!

Appendix B. Das OTRS Kernteam

Wie bei jeder guten Sache erinnert sich niemand mehr daran, wann genau OTRS entstanden ist, es muss im dritten oder vierten Quartal 2001 geschehen sein. In diesen glorreichen Tagen bestand das Kernteam aus Martin Edenhofer und Stefan Wintermeyer.

B.1. Martin Edenhofer

E-Mail: [<martin@otrs.org>](mailto:martin@otrs.org)
Homepage: <http://martin.edenhofer.de/>

Ohne Martin würde OTRS nicht existieren. Er ist ein brillianter Coder. Und zu scheu, mehr von sich preiszugeben. ;)

B.2. Stefan Wintermeyer

E-Mail: [<stefan@otrs.org>](mailto:stefan@otrs.org)
Homepage: <http://www.wintermeyer.de/>

So wenig Worte, wo doch so viel zu sagen wäre. ;-)

B.3. Sebastian Wormser

E-Mail: [<sibbi@sibbi.com>](mailto:sibbi@sibbi.com)
Homepage: <http://sibbi.org/>
Stiess zum Team: 01.05.2002

Wir sind sehr froh, dass Sebastian Wormser (Sibbi) zum Team gehört. Wir haben zusammen an einem kommerziellen Trouble Ticket system gearbeitet (stts auf SuSE (<http://www.suse.de/>) - IBM Case Study on SuSE (<http://www.ibm.com/software/success/cssdb.nsf/CS/NAVO-56G3KU?OpenDocument>) / German Linux magazin (<http://www.linux-magazin.de/ausgabe/2001/04/stts/stts.html>)). Ist es auch eine Art Wiedervereinigung, so sind wir doch diesmal OpenSource (und hoffentlich besser).

B.4. Dipl.-Ing. Robert Kehl

E-Mail: [<robert.kehl@otrs.de>](mailto:robert.kehl@otrs.de)
Homepage: <http://www.robertkehl.de/>
Stiess zum Team: 01.09.2003

Robert ist der Vater des Win32-Ports von OTRS. Über viele Wochen suchte und probierte er auf sehr unterschiedlichen Wegen und kam schliesslich zu einem brillanten otrs4win32 Installer.

Appendix C. Danksagungen

Wie die meisten OpenSource-Projekte möchten auch wir vielen Leuten für Ihre Hilfe danken. Dies ist eine offene Liste:

Martin Scherbaum, Carsten Gross, Harald Müller, Stefan Schmidt, Milisav Radmanic, Uli Hecht, Norman Walsh, Heiko Baumann, Atif Ghaffar, Pablo Ruiz Garcia, Dan Rau, Christoph Kaulich, Mark Jackson, Diane Shieh, Bernard Choppy, Carl Bailey, Phil Davis, Edwin D. Vinas, Lars Müller, Bryan Fullerton, Vladimir Gerdjikov, Fred van Dijk, Sebastien Guilbaud, Wiktor Wodecki, Arnold Ligtoet, Antti Kämäräinen, Nicolas Goralski, Robert Kehl, Gilberto Cezar de Almeida, Jorge Becerra, Eddie Urenda, Stella Power, Andreas Haase, Reiner Keller, Covert Jake, Moshe Leibovitch, Bjoern Jacke, Remo Catelotti, Alfons Obermeyer, Michael Rech, Torsten Werner;

Und natürlich gibt es immer Leute, die das Projekt besonders unterstützen. Dies sind:

- Robert Kehl, der den Win32-Installer alleine kreiert hat. Dank Dir, Robert!
- Aus den OTRS Mailinglisten schöpfen wir viele Ideen, und die Benutzer bekommen sehr guten Support darüber. Vielen Dank an alle auf den Listen, dass Ihr uns helft!

Spiegel

Das Projekt erreichte schnell einen Punkt an dem die Bandbreite unseres FTP-Servers nicht mehr ausreichte. Hier ist eine Liste von Leuten, die uns aushelfen, indem sie Spiegel aufsetzten. Vielen herzlichen Dank!

Nils Jeppe (mirror Hamburg), Bryan Fullerton (ftp.samurai.com), Eberhard Moenkeberg (ftp.gwdg.de)

Appendix D. RFC 1297

Viele Leute haben keine Idee, was ein Trouble Ticket System ist und warum sie eines benötigen könnten. Die *RFC 1297* (<http://www.faqs.org/rfcs/rfc1297.html>) ist ein guter Start um einen Überblick zu bekommen. Wir geben Sie hier im englischen Original wieder.

PURPOSES OF A NOC TROUBLE TICKET SYSTEM

A good Network Operations Trouble Ticket System should serve many purposes:

- 1) **SHORT-TERM MEMORY AND COMMUNICATION** ("Hospital Chart"). The primary purpose of the trouble ticket system is to act as short-term memory about specific problems for the NOC as a whole. In a multi-operator or multi-shift NOC, calls and problem updates come in without regard to who worked last on a particular problem. Problems extend over shifts, and problems may be addressed by several different operators on the same shift. The trouble ticket (like a hospital chart) provides a complete history of the problem, so that any operator can come up to speed on a problem and take the next appropriate step without having to consult with other operators who are working on something else, or have gone home, or are on vacation. In single-room NOCs, an operator may ask out loud if someone else knows about or is working on a problem, but the system should allow for more formal communication as well.
- 2) **SCHEDULING and WORK ASSIGNMENT**. NOCs typically work with many simultaneous problems with different priorities. An on-line trouble ticket system can provide real time (or even constantly displayed and updated) lists of open problems, sorted by priority. This would allow operators to sort their work at the beginning of a shift, and to pick their next task during the shift. It also would allow supervisors and operators to keep track of the current NOC workload, and to call in and assign additional staff as appropriate.

It may be useful to allow current priorities of tickets change according to time of day, or in response to timer alerts.

- 3) **REFERRALS AND DISPATCHING**. If the trouble ticket system is thoroughly enough integrated with a mail system, or if the system is used by Network Engineers as well as Network Operators, then some problems can be dispatched simply by placing the appropriate Engineer or Operator name in an "assigned to" field of the trouble ticket.
- 4) **ALARM CLOCK**. Typically, most of the time a trouble ticket is open, it is waiting for something to happen. There should almost always be a timer associated with every wait. If a ticket is referred to a phone company, there will be an escalation time before which the phone company is supposed to call back with an update on the problem. For tickets referred to remote site personnel, there may be other more arbitrary timeouts such as

"Monday morning". Tickets referred to local engineers or programmers should also have timeouts ("Check in a couple of days if you don't hear back from me"). A good trouble ticket system will allow a timeout to be set for each ticket. This alarm will generate an alert for that ticket at the appropriate time. Preferably, the system should allow text to be attached to that timer with a shorthand message about what the alert involves ("Remind Site: TT xxx") (The full story can always be found by checking the trouble ticket). These alerts should feed into the NOC's standard alert system.

The Alarm Clock can also assist (or enforce!) administrative escalation. An escalation timer could automatically be set based on the type of network, severity of the problem, and the time the outage occurred.

- 5) **OVERSIGHT BY ENGINEERS AND CUSTOMER/SITE REPRESENTATIVES**. NOCs frequently operate more than one network, or at least have people (engineers, customer representatives, etc) who are responsible for subsets of the total network. For these individual representatives, summaries of trouble tickets can be filtered by network or by node, and delivered electronically to the various engineers or site representatives. Each of these reports includes a summary of the previous day's trouble tickets for those sites, a listing of older trouble tickets still open, and a section listing recurrent problems. These reports allow the site reps to keep aware the current outages and trends for their particular sites. The trouble ticket system also allows network access to the details of individual trouble tickets, so those receiving the general reports can get more detail on any of their problems by referencing the trouble ticket number.
- 6) **STATISTICAL ANALYSIS**. The fixed-form fields of trouble tickets allow categorizations of tickets, which are useful for analyzing equipment and NOC performance. These include, Mean Time Between Failure and Mean Time to Repair reports for specific equipment. The fields may also be of use for generating statistical quality control reports, which allow deteriorating equipment to be detected and serviced before it fails completely. Ticket breakdowns by network a NOC costs to be apportioned appropriately, and help in developing staffing and funding models. A good trouble ticket system should make this statistical information in a format suitable for spreadsheets and graphics programs.
- 7) **FILTERING CURRENT ALERTS**. It would be possible to use network status information from the trouble ticket system to filter the alerts that are displayed on the alert system. For instance, if node XXX is known to be down because the trouble ticket is currently open on it, the alert display for that node could automatically be acknowledged. Trouble tickets could potentially contain much further information useful for expert system analysis of current network alert information.
- 8) **ACCOUNTABILITY ("CYA"), FACILITATING CUSTOMER FOLLOW-THROUGH, AND NOC IMAGE**. Keeping user-complaint tickets facilitates the kind of follow through with end-users that generates happy clients (and good NOC image) for normal trouble-fixing situations. But also, by their nature, NOCs deal with crises; they occasionally find themselves with major outages, and angry users or administrators. The trouble ticket system documents the NOC's (and the rest of the organization's) efforts to solve problems in case of complaints.

—RFC 1297

Natürlich haben wir dem OTRS viele Features spendiert, die die RFC nicht auflistet. Und wir sind noch nicht fertig.

Wir freuen uns sehr über Ihr Feedback. Bitte zögern Sie nicht, uns eine Nachricht an [<feedback@otrs.org>](mailto:feedback@otrs.org) zu senden.

Ihr OTRS Kernteam

Appendix E. GNU Free Documentation License

Eine deutsche Übersetzung der GNU Free Documentation License finden Sie unter folgender Adresse: <http://www.gnu.de/>

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies,

and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Literatur

Ben Laurie and Peter Laurie, 1999, O'Reilly, *Apache. Das umfassende Referenzwerk.*, 3897211270.

Bryan Costales and Eric Allman, 1997, O'Reilly, *sendmail*, 1565922220.

Richard Blum, 2001, SAMS, *Postfix*, 0672321149.

Eric Rescorla, 2000, Addison Wesley, *SSL and TLS*, 0201615983.

iX, Heise Verlage, "Winnetou und Old SSLeay", Holger Reif, 1998, page 128.

George Reese, Tim King, and Randy Jay Yarger, 2002, O'Reilly, *Managing & Using MySQL (2nd Edition)*, 0596002114.

Illiad, 1999, O'Reilly, *User Friendly*, 1565926730.

