

Migração de e-mail

1. [Zimbra Tech Center](#)
2. [Comunidade Sandbox](#)
3. Migração de e-mail

Conteúdo

- [1 de migração de email](#)
 - [1.1 a partir de IMAP](#)
 - [1.1.1 usando imapsync \(método recomendado\)](#)
 - [1.1.1.1 Kerio](#)
 - [1.1.1.2 Dovecot](#)
 - [1.1.1.3 Cyrus](#)
 - [1.1.2 métodos comerciais](#)
 - [1.1.2.1 YippieMove](#)
 - [1.1.2.2 MigrationWiz](#)
 - [1.1.2.3 MoveMyMail](#)
 - [1.2 POP3](#)
 - [1.2.1 pop2imap](#)
 - [1.2.2 imap_tools](#)
 - [1.2.3 Ferramentas Comerciais](#)
 - [1.3 a partir do Exchange](#)
 - [1.4 a partir do Outlook](#)
 - [1.5 a partir do Lotus Notes / Domino](#)
 - [1.6 a partir iMail](#)
 - [1.7 a partir de arquivos mbox](#)
 - [1.7.1 mbox> Maildir> Zimbra de hash-dirs](#)
 - [1.8 a partir Maildir](#)
 - [1.8.1 Merak](#)
 - [1.8.2 posts no fórum relacionados](#)
 - [1.9 a partir de Thunderbird pastas locais](#)
 - [1.9.1 lotes de pastas locais => mbox => dovecot => imapsync](#)
 - [1.10 A partir de Tobit David Zehn](#)
 - [1.10.1 Passo 1 - TAS → IMAP → Maildir](#)
 - [1.10.2 Passo 2 - Maildir → Zimbra](#)
 - [1.11 a partir DeskNow](#)
 - [1.11.1 DeskNow Versão](#)
 - [1.11.2 DeskNow: Como é armazenar e-mails](#)
 - [1.11.3 Shell Script para automatizar a migração](#)
 - [1.11.3.1 Customisation](#)

- [1.11.3.2 Limitações](#)
 - [1.11.3.3 Para Fazer](#)
 - [1.11.3.4 O código-fonte do script shell](#)
- [1.12 A partir do Exchange para Zimbra](#)
 - [1.12.1 Usando mbsync](#)

Migração de e-mail

- Este artigo é um **trabalho em andamento** , e pode ser seções inacabadas ou ausentes.

De IMAP

Usando imapsync (Método Recomendado)

Atualmente, o método recomendado para migrar os usuários para Zimbra de um servidor IMAP existente é com o [imapsync](#) ferramenta escrito por Gilles Lamiral. Em caso de dúvida, consulte a documentação da sua versão do imapsync para obter as opções e definições actualizadas.

Ubuntu 8.x terá isso já no repositório assim:

```
Sudo aptitude install imapsync
```

GertThiel originalmente publicado este [Guide to imapsync](#) nos fóruns Zimbra.

Links de fórum relacionados:

- [Http://www.zimbra.com/forums/administrators/16-importing-exsisting-folders-messages-address-books.html](http://www.zimbra.com/forums/administrators/16-importing-exsisting-folders-messages-address-books.html)
- [Http://wiki.zimbra.com/index.php?title=Imapsync_from_courier-imap_without_passwords](http://wiki.zimbra.com/index.php?title=Imapsync_from_courier-imap_without_passwords)

Kerio

Somente o correio pode ser migrado no momento. Usando o seguinte comando imapsync para migrar emails de um único usuário para zimbra

```
perl imapsync --buffer-size 8192000 --nosyncacl --subscribe --syncinternaldates \
--host1 kerio.host.com --user1 user@host.com --password1 keriopasswd --sep1 "/" --prefix1 "" \
- Host2 zimbra.host.com --user2 user@host.com --password2 zimbrapasswd
```

Dovecot

Consulte [Migrando do Dovecot com LDAP externo](#)

Cyrus

[Http://www.zimbra.com/forums/migration/6223-migrating-cyrus-zimbra-imapsync.html](http://www.zimbra.com/forums/migration/6223-migrating-cyrus-zimbra-imapsync.html)

Métodos comerciais

YippieMove

Nota: este foi, provavelmente, originalmente adicionado por alguém de YippieMove ([Usuário: Gettyless](#))

[YippieMove](#) é um SaaS para a migração de e-mail. O serviço permite que você se mova entre dois provedores de e-mail que suportam IMAP, incluindo o Zimbra. Nenhum download é necessário e todas as transferências são realizadas em seus servidores. YippieMove cobra \$ 14.95 por conta transferida (desconto de volume disponível). Um screencast está disponível [aqui](#) .

MigrationWiz

[MigrationWiz](#) é uma solução de migração de e-mail baseada em nuvem que pode migrar entre Zimbra e outros sistemas (incluindo POP, IMAP, o Exchange, o Google Apps, etc.). O preço atual da MigrationWiz é de US \$ 9,99 por caixa postal, com descontos de volume disponíveis.

MoveMyMail

[MoveMyMail](#) é um novo serviço que incide sobre as migrações de e-mail IMAP. Atualmente está livre enquanto estiver em versão beta. Eles fazem um bom trabalho de mover para / a partir de Zimbra, Google Apps / Gmail, Yahoo, etc Ele é apenas online, e não requer qualquer download. Eles têm uma opção em massa disponível gratuitamente também.

POP3

Pop2imap

Pop2imap é uma ferramenta usada para obter uma caixa de entrada pop e sincronizá-lo para uma pasta IMAP. Você pode obtê-lo a partir de [\[1\]](#)

É muito semelhante ao imapsync acima.

```
pop2imap \  
--host1 some.pop.com.au --user1 YourAccount --passfile1 yourPasswordFile \  
--host2 zimbra.imap.com.au --user2 yourZimbraAccount --passfile2 yourZimbraPasswordFile \  

```

--folder MyOldPOPMail

Copiará o INBOX em sua conta pop para MyOldPOPMail pasta.

Existem alguns problemas com pop2imap embora.

- Não vai criar a pasta, você deve ter que criado manualmente
- Ele não suporta SSL, no entanto isso é fácil de fazer com "stunnel" - fornecendo o túnel SSL para o servidor
- Ele faz o download de todas as mensagens primeiro para obter cabeçalhos, embora alguns servidores pop fazer isso bem, muitos não e isso é uma operação muito demorada. As modificações no script são simples de saltar este passo e forçar uma cópia do correio (tecnicamente já não é uma sincronização, mas uma cópia em linha reta).

Imap_tools

imap_tools (http://www.athensfbc.com/imap_tools/) tem modo de lote com listas construído em para todas as atividades ... POP3-IMAP, IMAP-IMAP ...

Ferramentas Comerciais

Ambos [MoveMyMail](#) e [MigrationWiz](#) suporte POP3 -> migrações IMAP.

De Exchange

Há pelo menos duas maneiras de migrar o conteúdo de um usuário individual do MS Exchange. O primeiro é usar o Zimbra Migration Wizard. O segundo é exportar o conteúdo das pastas do MS Exchange para um arquivo .pst e importá-los usando o Assistente de Importação Zimbra PST. Uma terceira opção é exportar os dados do MS Exchange para um arquivo .pst e restaurá-lo assim que o perfil do Zimbra for criado para um usuário Zimbra correspondente, através do Zimbra Connector para Outlook ou IMAP.

Veja fóruns: [Migração do Exchange](#)

Informações sobre como usar o Assistente de Migração Zimbra Exchange pode ser encontrada no [Assistente de migração para o Guia de instalação do Exchange](#) .

página wiki separada está aqui - [A migração do Exchange](#)

Do Outlook

Se você estiver usando o Outlook sem o Exchange, você pode importar as pastas de correio locais diretamente no Zimbra com o [Outlook PST ferramenta de importação Microsoft](#) a partir [da página de download](#) .

Do Lotus Notes / Domino

Lotus Notes / Domino para ZCS pode ser encontrada no [Assistente de migração para o Guia de instalação Domino](#)

De iMail

O primeiro passo é exportar as informações dos usuários, incluindo a senha. Eu usei [este](#) utilitário para CommuniGate Pro, que cria um arquivo de comprimento fixo. Eu, então, converti-o para um arquivo delimitado por tabulações com [este utilitário](#) para que eu pudesse usar a [massa Criar](#) processo. Depois de criar seus domínios e provisionar as contas, ative IMAP4 para iMail e use o processo de migração IMAP.

De arquivos mbox

O formato de arquivo mbox é comumente usado por muitos programas, principalmente sendmail. A vantagem da migração de arquivos mbox é que nenhuma senha ou contas especiais são necessárias. Existem vários scripts que as pessoas enviaram para a wiki:

- [Mbox para maildir com Python](#)
- [Mbox para SMTP com Perl](#) - forward de um arquivo mbox para o endereço indicado, preservando anexos
- [Mbox para 'maildir' com Perl](#) - script Perl para migrar uma loja mbox Kmail para IMAP. Alterar check_file para evolução
- [Adicionar mensagens de arquivo mbox](#) - script Python que apela zmmailbox

Mbox> Maildir> Zimbra hash-dirs

Se o arquivo mbox pode ser convertido para o formato maildir, você pode usar o [zmmailbox](#) utilitário para adicionar mensagens à caixa de correio de um usuário. Isso preservará as informações originais de data e hora e permitirá que os administradores classifiquem o e-mail em pastas giorg através de IMAP.

OR zmlmtpinject

Uso tanto para delineado aqui:

<http://www.zimbra.com/forums/installation/12617-recover-data-store-folders.html#post64962>

```
#!/usr/local/bin/perl
#
# mbox2maildir: converte arquivo mbox para o diretório maildir - o inverso da
# maildir2mbox da distribuição qmail.
#
# Uso: mbox2maildir usa as mesmas variáveis de ambiente como maildir2mbox:
# MAILDIR é o nome do seu diretório maildir; MAIL é o nome do seu
# arquivo # mbox; MAILTMP é ignorado. MAIL é excluído após a conversão.
#
# ATENÇÃO: não há nenhum bloqueio; Não corra mais do que um destes! você
# foi avisado.
#
# Com base em converter-e-criar por Russell Nelson <nelson@gmail.org>
# kludged a este por Ivan Kohler <ivan@voicenet.com> 97-Sep-17

exigem 'stat.pl';

Local $ SIG {HUP} = 'IGNORE';
Local $ SIG {INT} = 'IGNORE';
Local $ SIG {QUIT} = 'IGNORE';
Local $ SIG {TERM} = 'IGNORE';
Local $ SIG {TSTP} = 'IGNORE';

($ name, $ passwd, $ uid, gid, $ quota, $ comentário, $ gcos, $ dir, $ shell) =
    getpwuid ($ <);

Die "fatal: home dir $ dir não existe \n" a menos que -e $ dir;
& Stat ($ dir);
Die "fatal: $ name é $ uid, mas $ dir é propriedade de $ st_uid \n" if $ uid! = $ St_uid;

Chdir ($ dir) ou morrer "fatal: incapaz de chdir para $ dir \n";
$ Spoolname = "$ ENV {MAILDIR}";
-d $ spoolname ou mkdir $ spoolname, 0700
    ou morrer ("fatal: $ spoolname não existe e não pode ser criado. \n");

Chdir ($ spoolname) ou die ("fatal: incapaz de chdir para $ spoolname. \n");
-d "tmp" ou mkdir ("tmp", 0700) ou die ("fatal: incapaz de fazer tmp / subdiretório \n");
-d "new" ou mkdir ("new", 0700) ou die ("fatal: incapaz de fazer new / subdiretório \n");
-d "cur" ou mkdir ("cur", 0700) ou morrer ("fatal: incapaz de fazer cur / subdiretório \n");

open (SPOOL, "<$ ENV {MAIL}")
    ou morrer "Não é possível abrir $ ENV {MAIL} \n";
$ I = tempo;
while (<SPOOL>) {
    if (/ ^ From /) {
        $ fn = sprintf ("new /% d $$ mbox..", $ i);
        Aberto (OUT, "> $ fn") ou morrer ("fatal: incapaz de criar nova mensagem");
        $ I ++;
        Próximo;
    }
}
S / ^> From / De /;
```

```

    Imprimir OUT ou morrer ("fatal: incapaz de escrever para a nova mensagem");
}
Close (SPOOL);
fechar);
Unlink (" $ ENV {MAIL}");

```

Aqui está outro método que se expande no exemplo anterior. Este método destina-se especificamente para os arquivos de correio estilo mbox nos diretórios home do usuário e o mbox em / var / spool / mail. Além disso, ele agarra os livros de endereços do OpenWebmail. Ele então pega tudo isso e envia para Zimbra.

Este método já está disponível no github: [git: //github.com/s1037989/owm2zcs.git](https://github.com/s1037989/owm2zcs.git)

```

# Cat / usr / local / bin / owm2zcs

#! / Usr / bin / perl

# Cria Zimbra conta
# Adiciona um alias para First.Last@example.com
# Importa todos em estilo mbox e-mail pastas
# Importações todos os endereços vCard armazenado-OpenWebMail livros

# <USAGE>

# Precisa de um Import.csv exatamente com este formato (delimitado por tabulações):
# uid senha displayName sn givenName

# on owmsrver:
# Certifique-se de que há espaço em disco suficiente no owmsrver: / tmp ou substituir / tmp neste exemplo
# Com outra coisa. Isso pode ser 10, 50, 100 ou mais GB.
# Export DOMAIN = example.com; Mkdir -p / tmp / owm2zcs; cd / tmp / owm2zcs
# 1) Uso: / usr / local / bin / owm2zcs Import.csv
# 2) Uso: prog | / usr / local / biz / owm2zcs

# encontrar ./ -size + 10M -exec cp -al --parents '{ }' ../owm2zcs-large/ \;
# Find ./ -size + 10M -exec rm -f '{ }' \;
# Tar cvzpf - * | ssh zimbra "mkdir -p $ (pwd); cd $ (pwd); tar xvzpf - -C $ (pwd); / opt / Zimbra / bin / zmpov <$ (pwd)
/owm2zcs-11475.prov"
# - ou - descobrir como aumentar max mensagem tamanho de carregamento
# http://wiki.zimbra.com/wiki/Configuring_maxmessagesize

# on zimbraserver:
# Certifique-se de que há espaço em disco suficiente no zimbraserver: / tmp ou substituir / tmp neste exemplo
# com outra coisa . Isso pode ser 10, 50, 100 ou mais GB.
# Transferir a estrutura do arquivo de OWM para ZCS com SSH e TAR (Ou, provavelmente, mais de preferência com Rsync):
# 1) zimbraserver: / # ssh owmsrver "tar cvzpf - / tmp / owm2zcs" | tar xvzpf - -C /
# 2) zimbraserver: / # export RSYNC_PASSWORD = password rsync -vazi rsync: // owmsrver / root / tmp / owm2zcs / tmp
# / opt / Zimbra / bin / zmpov <owm2zcs.prov | tee 2> owm2zcs-err.log owm2zcs-out.log

# </ USO>

# Há alguns itens codificados em aqui ainda, então tome cuidado.
# Eu usei este script para migrar 60GB de arquivos de correio para 50 contas hoje.
# Estará fazendo os 450 restantes amanhã.

```

```
# Script por Stefan Adams <stefan@cogentininnovators.com> 2010-08-13
# baseado no roteiro acima a partir da página Mail_Migration de wiki.zimbra.com e ...
# baseado em converter-e-criar por Russell Nelson <nelson @ gmail .org>
# kludged a este por Ivan Kohler <ivan@voicenet.com> 97-Set-17
```

```
# Converter mbox2Maildir
# find / tmp / Maildir / -type d -nome cur -o -nome novo -o -nome tmp | Perl -pi -e 's / \ / (cur | new | tmp) $ /' | Sort -u | Perl -MFile ::
# Basename -pi -e 'chomp'; $ Maildir = $ _; ($ Mbox) = / Maildir $ /? ('Mbox'): (/Maildir\.(.*)$ /); S / Maildir $ / mail \ / mbox /; S /
# Maildir \ / \ / Mail \ /; $ Root = dirname $ _; $ _ = "Echo $ _"; mkdir -p \ " $ root \ "; env MAILDIR = \ " $ Maildir \ "MAIL = \ " $ _ \
# "MAILTMP = \ root $ tmp \ var / gmail / Bin / maildir2mbox \ n "" | sh
```

```
$ | = 1;
#use strict;
#use advertências;
Use File :: Path;
Use Arquivo :: Basename;
Use Text :: vCard :: Addressbook;
```

```
#local $ SIG {HUP} = 'IGNORE';
#local $ SIG {INT} = 'IGNORE';
#local $ SIG {QUIT} = 'IGNORE';
#local $ SIG {TERM} = 'IGNORE';
#local $ SIG {TSTP} = 'IGNORE';
```

```
Die "variável de ambiente DOMAIN não definida. \ N" a menos que $ ENV {DOMAIN};
$ ENV {CONVMBOX} | = 'OWM';
$ ENV {CONVABOOK} | = 'OWM Contatos';
```

```
My @import = ();
if ($ argv [0]) {
    IMPORT aberta, $ ARGV [0] ou morrer "Não é possível ler $ ARGV [0]: $ \ n";
    @import = <IMPORT>;
    Close IMPORT;
} Else {
    @import = <STDIN>;
}
```

```
Aberta PROV, "> owm2zcs - $$ prov." Ou morrer "Não é possível criar provisionamento script: $! \ N";
```

```
Abra CSV, "> owm2zcs - $$ . Csv" or die "Não é possível criar o relatório-book: $! \ N";
```

```
foreach (@import) {
    my $ selectMailbox = 0;
    My @mailbox = ();
    Meu @abook = ();

    # Obter detalhes de usuário importar
    chomp;
    my ($ uid, $ password, $ displayName, sn $, $ givenName, $ homeDirectory) = mapa {s / "/" g; $ _} divisão / \ t /;
    a menos ($ homeDirectory) {
        ($ uid, undef, undef, undef, undef, undef, undef, $ homeDirectory) = getpwnam ($ uid) ou que {print STDERR
"$ uid não encontrado: $ \ n"; em seguida;};
    }
    não {print STDERR "problema $ homeDirectory \ n "; em seguida;} a menos -d $ homeDirectory;
    $ password = 'password1' se $ password || $ password = ~ /\ ^ {w+} .+$/;!
    $ displayName || = $ uid;
    my ($ f, $ l);
    if ($ displayName = ~ /\ ^ (\ w +) \ s + (\ w +) $ /) {
```



```

        ($ f, $ l) = ($ 1, $ 2);
    } ELSIF ($ displayName = ~ / ^ ( \ w + ), \ s * ( \ w + ) $ / ) {
        ($ f, $ l) = ($ 2, $ 1);
    } ELSIF ($ displayName = ~ / ^ ( \ w + ) \ s + ( \ w + ) \ s + ( \ w + ) $ / ) {
        ($ f, $ l) = ($ 1, $ 3);
    } ELSIF ($ displayName = ~ / ^ ( \ w + ) \ s + ( \ w + ) \ s *, / ) {
        ($ f, $ l) = ($ 1, $ 2);
    } else {
        my ($ f) = ($ displayName = ~ / ^ ( \ w + ) /);
        my ($ l) = ($ displayName = ~ / ( \ w + ) $ /);
    }
    $ givenName || = $ f;
    $ sn || = $ l || $ uid;

    my $ alias = "$ givenName $ sn.";
    $ alias = ~ s / \ s + // g;

    print "Convertendo para $ uid ($ homeDirectory). \ n";

    a menos ($ ENV {NOCREATE}) {
        # Criar Conta e, possivelmente, um alias
        if ($ givenName) {
            PROV print "createAccount $ uid \ @ $ ENV {DOMAIN} \ "$ password \ " displayName \ "$
            displayName \ " sn \ " $ sn \ " givenName \ " $ givenName \ " zimbraPrefFromAddress $ alias \ @ $ ENV {DOMAIN}
            zimbraPrefReplyToAddress $ alias \ DOMAIN} \ n ";
            Print PROV "addAccountAlias $ uid \ @ $ ENV {DOMAIN} $ alias \ @ $ ENV {DOMÍNIO} \ n ";
        } Else {
            PROV print "createAccount $ uid \ @ $ ENV {domain} \ " $ password \ "Selecione \ " $ displayName \
            "sn \ " $ sn \ " \ n";
        }
    }

    A menos ($ ENV {NOMBOX}) {
        # Import mbox (es)
        foreach my $ bobina (grep {-f $ _ && -s _ && -r _ && `arquivo` $ _ "" = ~ / text /} mapa {s / \ "/ \ " / g; $ _} glob (
            "$ homeDirectory / mail / *"), "/ var / spool / mail / $ uid") {
            my $ oldspool = $ carretel;
            $ Spool = basename $ oldspool;
            $ Spool = $ spool eq 'mbox' || $ Oldspool eq "/ var / spool / mail / $ uid"? 'Entrada: "$ spool";
            Imprimir "mbox | $ oldspool -> $ spool ...";
            -e $ uid || Mkdir "$ uid" ou fazer {print STDERR "Ignorado: $!"; último; };
            Do {print STDERR "Omitido."; Próximo; } If -d "$ uid / $ spool";
            -e "$ uid / $ spool" || Mkdir "$ uid / $ spool" ou fazer {print STDERR "Ignorado: $!"; último; };
            Abrir SPOOL, "$ oldspool" ou fazer {print STDERR "Ignorado: $!"; Próximo; };
            $ l = tempo;
            while (<SPOOL>) {
                if (/ ^ From /) {
                    $ fn = sprintf ( "$ uid / $ spool /% d $$ mbox..", $ i);
                    Aberto OUT, "> $ fn" ou fazer {print STDERR "Ignorado: $!"; Próximo; };
                    $ l ++;
                    Próximo;
                }
                S / ^> From / De /;
                imprimir;
            }
            SPOOL perto;
            fechar;
        }
    }

```

```

        Impressão "Done.";
    } Continuar {
        print "\n";
        if (-d "$uid / $carretei") {
            impulso @mailbox, "createFolder \" / $ENV {CONVMBOX} / $spool \" \"n";
            Push @mailbox, "addMessage \" / $ENV {CONVMBOX} / $spool \" \"$uid / $spool \" \"n";
            $SelectMailbox = 1;
        }
    }
}

A menos ($ENV {NOABOOK}) {
    # Import vCard (s)
    foreach my $abook (grep {-f $_ && -s $_ && -r $_ && `arquivo` $_ "" = ~/text | vcard / i} mapa {s / \" / \" / g; $_})
    glob (" $homeDirectory / .openwebmail / webaddr / **") {
        print "livro | $abook ...";
        $Newabook = basename $abook;
        My $address_book = Text :: vCard :: Addressbook-> new ({'fonte_file' => $abook}) ou fazer {print
STDERR "Ignorado: $!"; Próximo; };

        Do {print STDERR "Omitido."; Próximo; } If -e "$uid- $newabook.csv";
        Abrir CSV, "> $uid- $newabook.csv" ou fazer {print STDERR "Ignorado: $!"; Próximo; };
        Imprimir CSV "fullName, email, firstName, lastName \n";
        foreach my $vcard ($address_book-> Vcards ()) {
            próximo a menos que $vcard-> fullName () && $vcard-> e-mail ();
            Meu @fullname = $vcard-> fullName () = ~ /, /? / \ S + /, $vcard-> fullName (), 2: split / \ s +
/, $vcard-> fullName (), 2;

            Print CSV join ',', $vcard-> fullName () || "", $vcard-> email () || "", $fullName [0] || "", $
fullName [1] || "";

            Imprimir CSV "\n";
        }
        CSV perto;
        Impressão "Done.";
    } Continuar {
        print "\n";
        if (-e "$uid- $newabook.csv" && -s _) {
            impulso @abook, "createFolder -V contato \" / $ENV {CONVABOOK} / $newabook \" \"n";
            Push @abook, "postRestURL \" / $ENV {CONVABOOK} / $newabook \" \"$uid- $
newabook.csv \" \"n";

            $SelectMailbox = 1;
        }
    }
}

# User realmente tem algo para importar
if ($selectMailbox) {
    print PROV "selectMailbox $uid \ @ $ENV {domain} \"n";
    Print PROV "createFolder \" / $ENV {CONVMBOX} \" \"n se $ # caixa de correio > = 0;
    Print PROV "createFolder -V contato \" / $ENV {CONVABOOK} \" \"n se $ # abook > = 0;
    Print PROV @mailbox, @abook;
    Imprimir PROV "exit \n";
}

# Assinar o relatório-book.
Print CSV "$uid \ t $ENV {DOMÍNIO} \ t $senha \ t $displayName \ t $sn \ t $givenName \n";
}
CSV perto;

```

Fechar PROV;

```
Print "Inspecione: owm2zcs - $$ {Prov, csv} \n";
```

```
Print "E então importar: \n";
```

```
Print "tar cvzpf - owm2zcs - $$ Prov | ssh zimbra \" mkdir -p \ $ (pwd); Cd \ $ (pwd); Tar xvzpf - C \ $ (pwd); / Opt / zimbra / bin /  
zmprov <\ $ (pwd) / owm2zcs - $$ Prov \" \n";
```

De Maildir

- [Maildir para zmmailbox com o bash](#) - bash script pela nervoso (com base em uma idéia de Jarosław Czarniak)
- [Converter Maildir para IMAP para todos os usuários com perl](#) - por Andreas Kammlott
- [Courier-IMAP Maildir para zmmailbox](#) - por Kevin Stange de Steadfast Networks
 - Suporta importação de bandeiras de mensagens e processamento de hierarquia de subpasta

Merak

Para a migração de e-mails do servidor Merak Mail, use o seguinte script Este script trabalhou com imap e pop3 maildirs

```
#!/usr/bin/python
```

```
# - * - coding: utf-8 - * -  
"""
```

Criado em Qua 04 de maio 13:35:07 2011

```
@author: ucti (Denis Chumachenko)  
"""
```

```
ZMMBOX = "/opt/Zimbra/bin/zmmailbox"  
BFOLDER = "oldbox"
```

sys importação, OS

```
if len(sys.argv) > 1:  
    os.path.realpath  
    se sys.argv[1][-1] == '/':  
        basedir = sys.argv[1]  
    mais:  
        basedir = sys.argv[1] + '/'  
    se os.path.exists(basedir):  
        n = raw_input("AVISO: Programa irá remover todos os arquivos vazios e * .dat arquivos do directório "+" \n você concorda (y / n) ")!?  
        Se n == 'y' ou n == 'Y':  
            findempty = os.popen('find ' + basedir + ' -size 0').readlines()  
            para fi no findempty:  
                print 'Removeing:' + fi.replace('\n', '')  
                os.remove(fi.replace('\n', ''))  
            finddat = os.popen('find ' + basedir + ' -name *.dat').readlines()
```

```

para fi no finddat:
    print 'Removeing:' + fi.replace ( '\ n', '')
    os.remove (fi.replace ( '\ n ',''))
para o caminho, diretórios, arquivos em os.walk (basedir):
    Se o caminho = basedir:
        domain = PATH.replace (basedir,'.') subdivididas ( '/' ) [0]
        USUÁRIO = PATH .replace (basedir + DOMÍNIO + '/', ''). split ( '/') [0]
        DIR = PATH.replace (basedir + DOMÍNIO + '/' + USUÁRIO + '/', '')
        se DIR == PATH: DIR = "";
        Se o usuário = '!':
            k = 0
            print '=====
            print 'PATH = ' + PATH
            impressão "Pesquisar e modificar arquivos imap .."
            para f em arquivos:
                FileExt = os.path.splitext (f) [- 1]
                se FileExt == 'imap.':
                    fhash = f.replace (FileExt, '') [-8:]
                    fr = open (PATH + '/' + f, 'r')
                    tmp = fr.readline ()
                    tmp = tmp.replace ( '\ n', ''). replace ( '\ r', ' ')
                    se tmp [: 1] == ' 0 'e tmp [-8:] == fhash:
                        content = fr.read ()
                        fw = open (PATH + '/' + f, 'w + ')
                        fw.write ( conteúdo)
                        fw.close ()
                        k += 1
                    fr.close ()
            print "Modificar:" + str (k) + "arquivos"
            print '----- '
            print 'Iniciar importação e-mails
            print' domain = '+ DOMÍNIO
            print' USER = '+ USUÁRIO
            print' DIR = '+ DIR
            createcmd = ZMMBOX + ' -z -m '+ USUÁRIO + ' @ ' + DOMÍNIO + 'createFolder "/" + BFOLDER + '/' + DIR + ""
            se DIR = '!':
                addmailcmd = ZMMBOX + '-z -m' + USUÁRIO + '@' + DOMÍNIO + 'addMessage "/" + BFOLDER + '/' + DIR + '/' ""
+ PATH + ' '
            else:
                addmailcmd = ZMMBOX + '-z -m' + USUÁRIO + '@' + DOMÍNIO + 'addMessage" /' + BFOLDER + '/' "" + PATH + '
'''

            os.system (createcmd )
            os.system (addmailcmd)

        else:
            print "Caminho não existe"

else:
    print "Exemplo: path_to_dir_with_domains merak2zimbra_emails.py"

```

Mensagens do fórum relacionadas

[Http://www.zimbra.com/forums/migration/24675-maildir-zimbra.html](http://www.zimbra.com/forums/migration/24675-maildir-zimbra.html)

De pastas locais do Thunderbird

Muitas pastas locais => mbox => dovecot => imapsync

A maneira mais fácil que eu encontrei para importar um monte de pastas locais do Thunderbird é configurar um servidor temporário do dovecot, converter as lojas de correio do thunderbird para mailstores do mbox linux e, em seguida, imapsync-lo do servidor temp para o servidor zimbra:

- Instale o dovecot em um servidor temporário. No Debian etch: apt-get install dovecot-imapd, e depois edite /etc/dovecot.conf e defina os protocolos para incluir o imap. Reinicie o dovecot. Isso é suficiente para os nossos propósitos.
- Baixe o roteiro nstouwimap aqui: <http://www.cs.rice.edu/~dwallach/nstouwimap/> . Ele será usado para converter caixas de correio do Thunderbird em arquivos mbox que podem ser lidos pelo dovecot.
- Copiar Todas as pastas locais de todos os usuários para o servidor.
- Crie um usuário de transferência no servidor dovecot.
- No diretório pessoal do usuário, crie uma pasta de e-mail: / home / transferuser / mail.
- Para cada caixa de correio, faça o seguinte (pode ser scriptado):
 - Limpe a pasta de correio antigo no servidor: rm -rf / home / transferuser / mail / *
 - Execute nstouwimap / caminho / para / usuário / LocalFolders / home / transferuser / mail para converter as Pastas Locais para a caixa de correio dovecot do usuário de transferência.
 - Execute o imapsync para transferir a conta para o zimbra.

De Tobit David Zehn

(Pode se aplicar a outras versões de Tobit David)

Nota: é necessária uma completa experiência de scripting! O autor sugere a aprendizagem Perl :)

Passo 1 - TAS → IMAP → Maildir

- Defina credenciais de acesso remoto na pasta Arquivamento na GUI do Infocenter. Aqui, vamos assumir "exportação" para o nome de usuário, "zimbra" para a senha.
- Copie o *archive.ini* arquivo da pasta de arquivo principal para todas as outras pastas. Depois disso, você pode ter que reiniciar David. Não tenho certeza.

Encontrar / caminho / para / david / archive -type d -exec cp /path/to/david/archive/archive.ini {} \;

- Iniciar maserver se ele não estiver em execução, para fornecer acesso IMAP

/ Caminho / para / david / util / linux / DvISEctl iniciar maserver

- Instale [imapproxy](#) para evitar que falhas maserver de desconexão, e configurar **imapproxy.conf** usar um enorme tempo de espera. Vamos supor que a porta 9143 está disponível.

```
server_hostname localhost
cache_size 3072
listen_port 9143
server_port 143
cache_expiration_time 3000
proc_username ninguém
proc_groupname nogroup
stat_filename / var / run / pimpstats
protocol_log_filename /var/log/imapproxy_protocol.log
syslog_facility LOG_MAIL
send_tcp_keepalives não
enable_select_cache não
foreground_mode há
force_tls há
chroot_directory / var / lib / imapproxy / chroot
enable_admin_commands não
```

- Localizar ou criar algo que mapeia o TAS (Tobias Archive System) através da leitura *archive.dir* arquivos que listam os subdiretórios (aka subarchives, subpastas) por pasta. Ao fazer o script, leia 430 bytes por sub-arquivo, ignore os primeiros 113 bytes e leia três elementos de informação por sub-arquivo. (Perl *descompactar* modelo: "x113 aZ * AZ * As AZ * As" .) Cada elemento consiste de um byte de identificação (1 = nome interno, 3 = caminho, 4 = nome de exibição, 38 = usuário) e uma string terminada em null, e Você precisará do nome exibido para usar como nome da pasta IMAP e do caminho para saber onde encontrar o arquivo *archive.dir* seguinte para percorrer.

- Encontre uma maneira de lidar com barras em nomes de arquivos do Tobit; Zimbra tem uma distinção pura entre separadores de caminho e caracteres permitidos, enquanto Tobit permite "/" em um nome de arquivo, mas também usa-o como o separador de caminho. Por exemplo, escolhemos exportar "/ Users / example / friends / family" como "/ Users / example / friends ~ family".
- Além disso, renomear arquivos que têm um & ou " neles. Você pode usá-los em Zimbra, mas você não pode exportar essas pastas via IMAP. Tanto quanto eu sei, o caminho mais seguro para renomear essas entradas é usar a GUI Infocenter E faça-o manualmente.
- Exporte o e-mail via IMAP. Eu achei [imapsync](#) ser suficientemente capaz de lidar com a implementação IMAP quebrado de Tobit, enquanto [mbsync](#) fez muito bem. Nós utilizado com sucesso o seguinte **.mbsyncrc por pasta** onde \$ REMOTO é o nome da pasta IMAP, e \$ LOCAL é o nome Maildir local.

```
Expurgar nenhum
Criar Ambos
Maildirstore locais
Path ~ / Mail /
IMAPStore tobit
host 127.0.0.1
```

Porta 9143
a exportação do usuário
Passe zimbra
RequireSSL sem
Canal foo
Master ": Tobit: \$ REMOTO"
Slave ": Local: \$ LOCAL"

- Você notará que esta configuração mbsync exporta apenas uma pasta. Isso ocorre porque mbsync não tenta novamente se algo der errado, enquanto o Tobit *maserver* deu erros muito. Sempre que isso acontecer (mbsync não sair 0), reinicie maserver e tente a pasta novamente. Eu sugiro limitar o número de tentativas de 15 por pasta, registrando pastas que não funcionou em tudo.
- A exportação do correio é SLOW. Foram necessários cinco dias completos para exportar 33 GB de correio (1764 pastas, 199020 mensagens) em um P4 2.4 GHz a partir de discos SATA. Você vai notar que, como caixas de correio são maiores, o tempo necessário para exportá-los cresce *exponencialmente* ; como o número de mensagens de uma pasta aumenta, exportador também leva mais tempo *por mensagem* . (Parece (strace, hurrah!) Que ele lê todas as mensagens da pasta apenas para expor um único.)) Sugiro manter pastas menores do que 2000 mensagens antes de migrar. top (1) mostra que maserver é muito ativo enquanto mbsync e imapproxy permanecer imperceptível.

Passo 2 - Maildir → Zimbra

Isso é surpreendentemente fácil, em comparação com a tarefa complexa de puxar o correio de Tobit David. Para cada usuário: criar um arquivo de texto com três comandos zmmailbox por Maildir e executar zmmailbox -z -m username@example.com <\$ TextFile

```
createFolder "/" $ FOLDERNAME"  
addMessage "/" $ FOLDERNAME" "/" path / to / $ FOLDERNAME / cur"  
addMessage "/" $ FOLDERNAME" "/" path / to / $ FOLDERNAME / new"
```

Infelizmente, o status de leitura / não lido é perdido, e cada mensagem aparece lida no Zimbra. Seria bom se addMessage tivesse um switch para marcar as mensagens como não lidas. Acho que isso não é grande coisa. Você ou seus usuários podem discordar.

Demorou duas horas para importar isso em um Core2 Quad com discos SATA em RAID1.

Boa sorte! - [Juere](#) 18:15, 26 de maio de 2009 (UTC)

De Desknow

Existem as seguintes opções que podem ser migradas para o Zimbra:

1. Desknow contas de correio, pastas de correio e os próprios mails
2. Arquivos Desknow
3. Fóruns do Desknow
4. Calendário Desknow
5. Tarefas Desknow
6. Desknow Contatos

O foco aqui é no ponto 1.

Versão do Desknow

O estudo baseia-se em:

- Desknow v3.2.10 Modo Lite. (Nota: É por isso que o IMAPsync não pode ser utilizado .. Desknow desactiva IMAP no modo Lite.)
- A versão do Zimbra é 5.0.16 Community Edition

Desknow: Como armazena e-mails

O seguinte é um layout de diretório típico do Desknow para armazenar e-mails para cada domínio, conta de e-mail e pasta de correio:

```
desknowdata \  
  usermail \  
    {domain} \  
      {user} \  
        Inbox \  
          11a8ccfd307_N93M_34-0.eml  
          11a8ccfd307_N93M_50-0.eml  
          11a9e179bdc_49LX_8-0.eml  
          :  
          ... mais mails ...  
          :  
        Rascunhos \  
        Itens Enviados \  
        Itens Excluídos \  
        Spam \  
        { folder1} \  
          {sub-folder1} \  
          {sub-folder2} \  
          :  
          ... mais sub-pastas ...  
          :  
        {folder2} \  
        :  
        ... mais pastas ...  
        :
```

Desknow armazena emails sob o diretório desknowdata \ usermail.

No próximo nível, existem subdiretórios para cada domínio no Desknow. Esse é o {domínio} no acima.

Em cada {domínio}, existem subdiretórios para cada conta de e-mail. Esse é o {usuário} no acima.

Em cada {usuário}, existem subdiretórios para cada pasta de e-mail. Existem várias pastas predefinidas da seguinte forma:

- Caixa de entrada (o mesmo que a caixa de entrada no Zimbra)
- Rascunhos (o mesmo que Rascunhos em Zimbra)
- Itens Enviados (mesmo que Enviados no Zimbra)
- Itens Excluídos (Igual ao Lixo no Zimbra)
- Spam (Igual a Junk em Zimbra)

Shell Script para automatizar a migração

O Unix shell script **dknw2zmb** (código fonte na seção mais tarde que segue) irá gerar outro script shell (nome e local são configuráveis). Quando o script de shell gerado é executado, ele realizará as contas de e-mail, pastas de e-mail e migração de e-mail do Desknow para o Zimbra.

customização

O script shell tem um número de variáveis que permitirão a personalização da instalação do Desknow e da instalação do Zimbra. Verifique o início do código fonte do shell script abaixo.

Limitações

- A senha das novas contas de e-mail no Zimbra é sempre {user} 1a2b3c. Por exemplo, a senha para 'chantaiman' é 'chantaiman1a2b3c'.
- A migração está agora em um único domínio. Se houver vários domínios, a migração tem que ser executada número múltiplo de tempo, cada um com personalização diferente.
- Alguns e-mails podem não ter a data recebida como "Data:" ou a data não está no formato reconhecível. Nesse caso, a data de recebimento no Zimbra será definida para a data / hora de migração.

Façam

- Gerencie erros melhor. Por exemplo. Se uma pasta de correio não puder ser criada, pule a migração dos e-mails na pasta etc.
- Manter a senha do usuário. Parece que 'zmpov cm' pode ser usado. No entanto, parece que as senhas criptografadas dos usuários podem precisar ser obtidas de Desknow de antemão. Precisa de mais estudos sobre como obter as senhas criptografadas Desknow.

Código-fonte do script de shell

```
#!/ bin / bash
# #####
#
# dknw2zmbr - migrar mails DeskNow para Zimbra
#
# uso:> sudo dknw2zmbr
#
# ##### FORMAÇÕES #####
# --- + --- + --- + --- + --- + --- + --- + --- + - + --- + --- + --- + --- +
#
Configurações # saída
# -----
# BATCH_FILE: o shell script gerado que realmente executar o
# migração
# ERROR_CMMD_FILE: Para uso futuro
# FINISH_CMMD_FILE: Para uso futuro
# Z_SERVER: nome do servidor do Zimbra instalação
# Z_DOMAIN: domínio de destino no Zimbra
# Z_ADMIN_PASSWORD: Senha para 'admin' em Zimbra
#
BATCH_FILE = / tmp / desknow2zimbra
ERROR_CMMD_FILE = / tmp / desknow2zimbra_error_commands
FINISH_CMMD_FILE = / tmp / desknow2zimbra_finish_commands
#
Z_SERVER = mail.xxxx.com
Z_DOMAIN = xxxx.com
Z_ADMIN_PASSWORD = PpAaSsWwRrDd123 # mudar isso para o seu um
#
Configurações # entrada
# -----
# DESKNOWDATA: Localização do 'desknowdata \ usermail \ {domain}' diretório
#
DESKNOWDATA = ~ / desknowmail / xxxx .com
# #####
#
exportação BATCH_FILE
exportação ERROR_CMMD_FILE
exportação FINISH_CMMD_FILE
exportação DESKNOWDATA
exportação Z_SERVER
exportação Z_DOMAIN
exportação Z_ADMIN_PASSWORD

função exec_n_log ()
#
# Descrição
# -----
# Execute o comando passado como parâmetro $ 1. Registrar erros para US $ ERROR_CMMD_FILE,
# e log comandos de sucesso para $ FINISH_CMMD_FILE
#
# Parâmetros
# -----
# $ 1: o comando a ser executado
#
{
```

```

    echo "$ 1" >> $ BATCH_FILE
##
# $ 1 # Eu não consigo fazer este trabalho. Eu pretendia fazer isso para executar o
comando # # transmitido como $ 1. No entanto, o erro ocorre com o comando 'sed'.
## Outro comando simples como 'ls -F' funciona. Qualquer ideia?
##
    If [$? -gt 0]
    , em seguida,
        echo "$ 1" >> $ ERROR_CMMD_FILE
        echo "ERRO:" "$ 1"
    else
        echo "$ 1" >> $ FINISH_CMMD_FILE
    fi
}

```

exportação -f exec_n_log # exportar isso para fazer "encontrar ... -exec ... 'trabalho

```

process_folders function () {
#
# Descrição
# -----
# Chamado por "encontrar ... -exec ...". "encontrar ... -exec ... 'passar os diretórios /
arquivos # a esta fundtion um por um. Em seguida, ele determina o que fazer com base em
# se é um diretório ou um arquivo.
#
# Parâmetros
# -----
# $ 0: o caminho completo passado do "encontrar ... -exec ... '
# $ 1: a conta de correio
#
    if [-d" $ 0 "]
    # Se é um diretório, use zmmailbox para criar pastas Zimbra
    seguida
        echo "DIR:" $ 1 ":" "$ {0}"
        if [ "$ 0" "="! ". -a "$ 0"! = ". /Inbox" -a "$ 0"! = ". /Sent Items" -a "$ 0"! = ". /Drafts" -a "$ 0"! = ". /Deleted Items" -a "$ 0"! = ". /Spam"]
        # não criar essas pastas DeskNow predefinidas que tem homólogos Zimbra ...
        então
            exec_n_log "/ opt / Zimbra / bin / zmmailbox -a admin @ $ {Z_DOMAIN} -p $ {Z_ADMIN_PASSWORD} -m $ {1} @ $ {}
Z_DOMAIN cf '$ {0 # \.}' "
        fi
    elif [-f" $ 0 "]
    # Se for uma interface de arquivo, onda uso e REST para importar o correio ...
    então
        # obter pasta para hierarquia do caminho completo
        fpath = $ {0% / *}
        ufolder = $ {fpath # \.}

        # traduzir pastas DeskNow predefinidos para Zimbra homólogos
        ufolder = $ {ufolder / # \ / Itens enviados / \ / Enviados}
        ufolder = $ {ufolder / # \ / Itens excluídos / \ Lixo}
        ufolder = $ {ufolder / # \ / Spam / \ Junk}

        # codificar caracteres especiais em nome da pasta como o nome da pasta irá
        # se tornar URL no RESTO interface. *** A ordem importa
        ufolder = $ {ufolder // \% / \% 25}
        ufolder = $ {ufolder // \% 20}
        ufolder = $ {ufolder // \% 21}
    fi
}

```

```
#
# Configure os arquivos de saída
#
'#! / bin / bash "echo> $ BATCH_FILE
echo" # `date`" >> $ BATCH_FILE
cp $ BATCH_FILE $ ERROR_CMMD_FILE
cp $ BATCH_FILE $ FINISH_CMMD_FILE

#
# mude para o diretório desknowdata
#
pushd $ dESKNOWDATA> / dev / null
```

```

#
# MAIN
# ----
# percorre 'desknowdata / usermail / {domain}'. Cada entrada deve ser
# um subdiretório com o nome sendo o usuário conta de correio.
#
for i in `
fazer
    if [ "$i" = "admin" ]
    # não criar a conta 'admin', é already lá ...

então

    # criar a conta de utilizador de correio
    exec_n_log "/ opt / Zimbra / bin / zmpov ca $ {i} @ $ {Z_DOMAIN} $ {i} 1a2b3c displayName $ {i} "
fi

diretório # mudança na conta do usuário mail ...
pushd $ i > / dev / null

# criar pasta de correio e correio de importação um-por-um
achado . -exec / bin / bash -c 'process_folders "$ 1" '{i}' $ i \;

# Saltar para fora a partir da conta do usuário mial
popd > / dev / null
feito
popd > / dev / null

# fazer o shell script gerado executável
chmod 755 $ BATCH_FILE

```

De Troca para Zimbra

Usando mbsync

Eu [migrou uma conta de correio](#) a partir de um servidor de troca para Zimbra, utilizando [mbsync](#). Isso permite sincronização contínua e mapear pastas para subpastas.

```

# O servidor IMAP do qual você deseja copiar e-mails.
IMAPAccount imap-src-conta
anfitrião imap.host.ac.uk
student0192 usuário
# Passe "xxxxx" # se você não se importa de armazená-lo no arquivo; caso contrário, irá pedir
UseIMAPS sim
CertificateFile /etc/ssl/certs/ca-certificates.crt

# O servidor IMAP que você deseja copiar e-mails para.
IMAPAccount-dest-conta IMAP
Anfitrião zimbra.corporate.com
usuário worker@corporate.com
# Passe "xxxxx" # se você não se importa de armazená-lo no arquivo; caso contrário, irá pedir

```

```

UseIMAPS sim
CertificateFile /etc/ssl/certs/ca-certificates.crt

servidor # Fazer a ligação IMAP para remoto utilizado abaixo
IMAPStore imap-src
conta IMAP-src-conta

IMAPStore imap-dest
Conta-dest-conta IMAP

# garante que as datas de mensagens será definido corretamente
CopyArrivalDate sim

transferência Canal
master: imap-src:
Slave: imap-dest:
# transferência de todas as pastas
Patterns *
Criar Slave
sincronização Puxe
# importante, caso contrário você vai ter 'Erro: loja ... não suporta in- Caixa de estado de sincronização '
SyncState ~ / .mail / imap-transfer

```

Você pode então executar `mbsync -I transferpara` para listar as pastas de correio serão sincronizados.

Para executar realmente a transferência, execute `mbsync transfer`. A coisa agradável sobre isso é que você pode executar isso periodicamente e ele vai fazer uma sincronização eficiente.

Notas adicionais:

- **senhas** : Você também pode especificar um `PassCmd` usar um método diferente de obter a senha (por exemplo, [a partir de um token ring](#)).
- **backups** : A maioria das instruções on-line para `mbsync` estão usando isso para backup para um sistema de arquivos local. Você pode até mesmo backup para o sistema de arquivos local e, em seguida, empurre-o para o novo servidor, [conforme descrito aqui](#) , mas isso é mais complicado do que o necessário se você quiser apenas para sincronizar os servidores
- **mapeamento pasta** : se você quiser transferir pastas para uma subpasta no sistema de destino (como eu fiz), você pode especificar isso dizendo `Slave ":imap-dest:parent-folder/"` na Channel configuração.

Referencia

Traduzido por Google translate e

Retirado de " https://wiki.zimbra.com/index.php?title=Mail_Migration&oldid=60609 "