

Технически университет - София

Дипломна работа

Задание: Проектиране на оптимизирана програма за възстановяване на имейли

Студент:

Ръководител:

София 2011

Съдържание:

1. Въведение.....	
1.1. Електронна поща. Основни понятия.....	3
1.2. Историята на електронна поща.....	4

1.3. Начин на работа.....	5
1.4. Потребител и e-mail.....	7
2. Основни понятия за Windows Server 2003, Exchange Server 2007 и Power Shell.....	
2.1. Windows Server 2003.....	9
2.2. Exchange Server 2007.....	13
2.3. PowerShell.....	22
3. Описание на топологията, хардуерната и софтуерната част....	

3.1. Exchange топология.....	23
3.2. Хардуерни изисквания за Exchange организация.....	24
3.3. Софтуерни изисквания за Exchange организацията.....	25
4. Дефиниране на проблема.....	
4.1. Пример с организацията „Exchange”.....	26
5. Създаване на програма за разрешаването на проблемаю.....	
5.1. Сопс код на PowerShell script.....	29

6.	Наръчник за администратора.....	
6.1	Инструкции за използване на програмата.....	52
7.	Приложимост.....	56
8.	Изводи.....	57
	Речник.....	58.
	Литература.....	59

1. 1. Въведение

1.1. Електронна поща. Основни понятия

Електронната поща, която често се нарича просто имейл (от англ. e-mail или email), е метод за обмен на цифрови съобщения през Интернет или други компютърни мрежи. Първоначално, електронна поща се изпраща директно от един потребител към друг компютър. Това налага и двата компютъра да са онлайн по едно и също време, а-ла-месинджър. Днешните системи за електронна поща се базират на модела съхрани-и-препрати. Имейл сървърите приемат, препращат, доставят и съхраняват съобщения. Потребителите вече не е необходимо да бъдат на линия по едно и също време, а трябва само да свържат за кратко, обикновено към имейл сървър (сървър представлява компютър, който работи съвместно с други компютри в компютърна мрежа, като се различава от тях по това, че за работата му не е директно необходимо човешко участие. При комуникацията сървърът предоставя услуги към останалите компютри, наричани клиенти), за толкова дълго, колкото е необходимо, да се изпратят и получат съобщенията.

Дадено имейл съобщение се състои от два компонента, заглавие на съобщението и тялото на съобщението, което е всъщност съдържанието на имейла. В заглавието на съобщението съдържа контролна информация, в това число поне имейл адреса на изпращача и един или повече адреси на получателя. Обикновен има и допълнителна информация като тема на писмото.

Първоначално мейлите се ползват само за текстова комуникация, по-късно могат да предават мултимедийно съдържание посредством прикрепвания - процес, който е стандартизиран от RFC 2045 до RFC 2049. Заедно, тези RFC-а започват да се наричат Многоцелеви добавки към Интернет поща, (на английски: Multipurpose Internet Mail Extensions) или съкратено MIME. Първоначално текстови комуникационна среда,

електронна поща е удължен за изпълнение мултимедийно съдържание прикачени файлове, процес, стандартизирани в RFC 2045 през 2049 г. Заедно, тези RFC-та са дошли да се нарича комплексно Internet Mail Extensions (MIME).

Историята на съвременните, глобални Интернет услуги за електронна поща започва още в началото на ARPANET. Стандарти за кодиране на електронна поща съобщения са предложени още през 1973 г. (RFC 561). Преминването от ARPANET към Интернет в началото на 1980, създава основата на сегашните услуги. Едно писмо, изпратено в началото на 1970 г. изглежда доста подобно на едно такова, изпратено по Интернет днес.

Мрежово-базираната електронна поща първоначално се обменя по ARPANET посредством разширения към Протокола за трансфер на файлове (на английски: File Transfer Protocol) (FTP), а сега това става чрез Simple Mail Transfer Protocol (SMTP), публикувано за първи път като Интернет стандарт 10 (RFC 821) през 1982 година. В процеса на транспортиране на електронни съобщения между системи, SMTP обменя параметрите за доставяне посредством пакет към съобщението, който е отделен от самото съобщение (заглавна част и тяло).

1.2. Историята на електронна поща

Електронната поща е предшественик на Интернет и дори играе важна роля при неговото възникване. В началото на 60-те години се появяват първите системи, позволяващи едновременен достъп на множество потребители до един и същ мейнфрейм компютър чрез отдалечени терминали. Електронната поща възниква през 1965 година като средство, използващо тези възможности за обмен на информация между потребителите. Година по-късно вече е възможно съобщенията да се пренасят на различни компютри чрез създадена между тях мрежа.

Възникналият през 1969 година предшественик на съвременния Интернет, ARPANET, изиграва огромна роля за развитието на електронната поща. През 1971 година е създадено първото приложение за електронна поща, предназначено за работа чрез

ARPANET, и е утвърдено използването на знака @, за да се отдели името на потребителя от името на компютъра в адреса на електронната поща. През 1972 година протоколът FTP е адаптиран, така че да улесни предаването на имейл съобщения. Появяват се първите приложения, включващи основните функции, използвани и днес - команди за отговор и препредаване на съобщенията. Те значително допринасят за нарастването на популярността на електронната поща и скоро тя се превръща в една от основните услуги в ARPANET.

През 1977 година е публикуван RFC 733, първият стандарт, поставящ си за цел унифицирането на услугите за електронна поща в ARPANET. През следващите години надеждното предаване на електронна поща става един от водещите мотиви за разработването на протоколите TCP/IP, основата на съвременния Интернет. През 1982 година RFC 733 е заменен от новия стандарт за електронна поща RFC 822, в който за пръв път е описан и синтаксисът на имената на домейни. По същото време е издаден и RFC 821, дефинирайки протокола SMTP, насочен специално към електронната поща заместител на FTP, който и днес е основният стандарт за пренос на електронна поща. В края на 80-те години се появяват първите комерсиални услуги, предоставящи достъп до електронна поща, които стават масови след 1993 година.

1.3. Начин на работа

Схемата вдясно показва обичайната поредица от събития,[3] когато потребител (Alice) изпраща съобщение по електронната поща със своя клиент за електронна поща (MUA). Тя попълва имейл адреса на получателя и натиска бутона за изпращане. Как работи електронната поща

1. Клиентът преобразува съобщението във формата на електронната поща и използва протокола SMTP, за да го изпрати на локалния пощенски сървър (MTA), в този пример - smtp.a.org, поддържан от интернет доставчикът (ISP) на изпращача.

2. Сървърът намира адреса на получателя, дефиниран от протокола SMTP protocol, в този пример - bob@b.org. Интернет имейл адресите са низове от вида

localpart@example.com. Частта преди знака @ е локалната част на адреса, много често потребителското име на получателя, а частта след знака @ е името на неговия домейн. Пощенският сървър на изпращача отделя името на домейна, за да определи абсолютното име на домейна на пощенския сървър на получателя в рамките на системата от имена на домейни (DNS).

3. DNS сървърът за домейна b.org, ns.b.org, отговаря с MX запис, съдържащ пощенските сървъри в домейна, в примера - mx.b.org, сървър, поддържан от интернет доставчика на получателя (Bob).

4. smtp.a.org изпраща съобщението на mx.b.org, използвайки протокола SMTP, а той от своя страна го поставя в пощенската кутия на получателя.

5. Когато получателят провери входящата си поща със своя клиент за електронна поща, той прочита съобщението от пощенския сървър, използвайки протокола POP3.

Описаната поредица от събития е валидна за повечето потребители на електронна поща. В същото време съществуват много други възможности и усложнения при предаването на съобщенията:

* Изпращачът и получателят могат да използват клиенти, свързани с корпоративна система за електронна поща, като Lotus Notes или Microsoft Exchange Server. Тези системи често имат свой вътрешен формат на съобщенията и клиентите им контактуват със сървъра със специфичен за приложението затворен протокол. Сървърът, от своя страна, изпраща и получава съобщения през Интернет чрез специален шлюз, който ги трансформира в стандартните за Интернет формати.

* Възможно е изпращачът или получателят да нямат клиент за електронна поща на своя компютър, а да използват уеб поща (Yahoo, Gmail, Hotmail, ABV...).

* Компютърът на изпращача може да използва собствен пощенски сървър, при което се избягва трансферът от точка 1.

* Получателят може да прочете съобщението по различни начини, например с помощта на протокола IMAP, направо в пощенския сървър mx.b.org или чрез уеб поща.

* Домейните често имат по няколко пощенски сървъра, за да могат да продължат да получават съобщения, дори когато основният сървър не функционира.

В миналото много пощенски сървъри приемат съобщения за всеки получател в Интернет, опитвайки се да ги доставят до него. Те играят важна роля в първите години на Интернет, когато мрежовите връзки се относително ненадеждни. Когато даден пощенски сървър не успее да се свърже със сървъра на получателя, той може да се опита да достави съобщението на такъв отворен сървър, по-близък до крайния получател, който от своя страна има по-добри шансове да се свърже с него в по-късен момент. Този механизъм на предаване се оказва силно уязвим за разпространение на спам, поради което днес приложението му е ограничено.

Снимка 1. Начин на работа на електронна поща

1.4. Потребител и e-mail

Всеки човек ежедневно ползвайки своя имейл клиент (Outlook, Lotus Notes, Thunderbird, IncrediMail, Eudora) ежедневно изпраща имейли – относно работа и служебни контакти, за контакт с роднини и близки, изпращане на реклами и новини. Въпреки, че някои изследвания показват че класическите писма написани на лист хартия в плик, няма да изчезнат, всекидневния сърф в интернет и проток на данни, показва точно обратното.

Нека да се спрем на един от най-известните софтуерни клиенти за електронна поща – Microsoft Outlook. Клиента е част от Microsoft Office пакета, който се ползва за изпращане и получаване на имейли, създаване на календари, слагане на флагове за напомняне (Calendar, Appointments, Tasks), създаване на виртуални срещи, във виртуални стаи (Meeting Rooms), споделяне на ресурси (Shared Resources), както и още много други функции свързани с бизнеса или за лично ползване.

Софтуерния клиент Outlook се инсталира върху работните станции (workstations) на потребителите (users), които са включени в интернет (мрежа) и по този начин ползва ресурсите, които се намират на сървърите, конфигурирани за предлагане на услугите свързани с имейл. Сървърите ползват продукт на Microsoft, инсталиран върху сървърна операционна система за имейл обмен, която се нарича Exchange Server. Сървърите се конфигурират и подържат от администратори. Клиентите (корпоративни или малки клиенти), в зависимост от финансовите възможности, поставя изисквания как да изглежда тяхната „имейл среда“. Повечето малки предприятия, спестяват от екстрите които предлагат администраторите, но за сметка на това искат качествени услуги. Спестяването обикновено се получава при закупуване на хардуер. Нека да разгледаме един прост пример :

Предприятието X решава да създаде Exchange Environment. Архитектите проектират средата в няколко варианта, след което го предлагат на клиента. Клиента решава че няма достатъчно финансови средства и започва да обмисля по икономични решения. Най-често срещаните варианти са: UX система за ролята на Edge (защото няма да се плаща лиценз), съответно UX системата може също така да се конфигурира като firewall (по този начин се спестява от хардуерните firewall устройства), спестяване от SAN (всеки сървър да си има собствени хард дискове, защото по този начин се спестява и от SAN-а и от оптиката чрез, която ще е свързан) и т.н.

В една подобна среда, Exchange администратора трябва да има готовност за решаване на проблеми въпреки липсата на ресурси. Един от интересните случаи е: как ще възстановим изтрит имейл на потребител в един така „орязана“ среда, която няма достатъчно ресурси.

MS Exchange изисква колко е възможно повече дисково пространство но, от друга страна е клиента който няма възможности да си го позволи, но иска всичко да му функционира качествено.

Какво ще се случи ако изтрием много важен за нас имейл? Дали може да ни помогне администратора при липса на дисково пространство? Как трябва да е подготвен един Exchange администратор ?

Отговор на следните въпроси, както и на още много други ще видим в разработка на дипломната работа.

1. 2. Основни понятия за Windows Server 2003, Exchange Server 2007 и Power Shell

2.1. Windows Server 2003

Windows Server 2003 R2 е актуализация на Майкрософт за Windows Server 2003 разработена върху стабилната основа на Windows Server 2003 със Server Pack 1.

Windows Server 2003 R2 улеснява свързването между офисите, контрола върху самоличностите и управлението на данните и приложенията вътре в корпорацията и извън нея. Той е разширена версия на сървърната мрежова операционна система Windows Server 2003. Разработен е, така че да се внедрява в работеща Windows Server 2003 инфраструктура, без да се налага повторно сертифициране и тестване на приложенията и закупуването на нови клиентски лицензи за достъп. Windows Server 2003 R2 интегрира средства, позволяващи да се изгради сигурна и надеждна система и по този начин за пореден път доказва намеренията на Microsoft да продължи да подобрява и усъвършенства сървърната си платформа.

Windows Server 2003 R2 предлага технологии, които помагат за намаляване на общите разходи за експлоатация за свързани бизнес приложения:

- Услугата Active Directory – разширяема и мащабируема услуга по управление на каталозите на базата на области на имената, базирани на Интернет-стандарта Domain Name System (DNS).
- IntelliMirror - Промяната и конфигурирането на компютрите може да става на едно място и след това настройките да се копират на други компютри. Това дава възможност администрацията да се централизира.
- Security Architecture - Повишената сигурност в Windows 2003 се дължи на подобренията в новите идентификационни карти с чип (smart cards), в ключовете за шифриране и в подобряване на сигурността на цялата система. Добавени са и няколко нови инструмента за анализ на сигурността на системата.
- Terminal Services - Тези услуги позволяват дистанционно да се влезе и да се управляват Windows 2003 или да се изпълняват приложения, все едно че се работи локално.

- Windows Script Host - Това е среда за изпълнение на скриптове. Скриптовите могат много да улеснят администратора за автоматизирането на някои обикновени задачи. Например, може да създава потребителски акаунти или да генерира отчети от събитийния регистър (event log).

Domain Controllers и Member Servers

При инсталиране на Windows Server 2003 на нова система, може да се настрои сървър да бъде member server, domain controller или самостоятелен (stand-alone server). Разликите между тези типове сървъри е много важна. Сървърите тип Member са част от домейн, но не съхраняват директно информация за домейна. Домейн контролерите (domain controller) се различават от Member servers, защото те съхраняват данни за домейна и предоставят услуга за идентификация на потребителите. Самостоятелните сървъри не са част от домейн и съхраняват информацията си в собствена база данни. Поради този факт, самостоятелните сървъри сами идентифицират потребителите си.

Във Windows Server 2003 не е проектиран да има първичен и резервен (Backup) контролер. Вместо това се използва модел за разпространение на данните (replication). Чрез този модел някой от домейн контролерите може да направи промяна в директорията и след това да я разпространи към други домейн контролери. Това може да става автоматично. Това доста различава от стария Windows NT модел, в който един основен контролер копира данните в резервен (Backup).

Домейни, които използват Active Directory, се наричат Active Directory домейни. Освен това Active Directory домейните могат да работят както само с един домейн контролер, но така и с много домейн контролери. По този начин, ако някой от контролерите е извън строя, то друг работещ може да го замести.

Всеки сървър може да поддържа следните роли:

- Application server - Програмен сървър. Тази роля предлага/предоставя XML Web услуги, Web приложения и разпределени приложения. Когато се конфигурира, следните услуги и програми се инсталират автоматично: IIS, COM+ и Microsoft .NET Framework. Освен това като допълнителна опция може да се добави Microsoft FrontPage Server Extensions и да се разреши или забрани ASP.NET.
- DHCP server - Сървърът ще стартира Dynamic Host Configuration Protocol (DHCP) и може автоматично да раздава интернет адреси (IP) на клиентите в мрежата. Тази опция стартира New Scope Wizard.
- DNS server - DNS сървър. Чрез тази роля сървърът ще може да превръща IP адресите на компютрите в имена и обратното. При конфигурирането на тази роля се стартира DNS Server Wizard.
- Domain controller - Домейн контролер. Сървърът предоставя услуга по управление на каталозите за домейн и за съхранение на данните на домейна. Домейн контролерите освен това контролират процеса на влизане в компютрите, част от домейна и търсенето в домейна. Тази опция ще инсталира и DNS и Active Directory.
- File server - Файлов Сървър. Сървърът ще може да управлява достъпа до файлове. Тази опция позволява бързо да се конфигурират дисковите квоти и индексването на файловете. Може освен това да се инсталира Web-базиран инструмент за администриране, което инсталира IIS и разрешава Active Server Pages (ASP).
- Mail server (POP3, SMTP) - Пощенски сървър. Ролята предоставя услугата Post Office Protocol 3 (POP3) и Simple Mail Transfer Protocol (SMTP). Потребители, които използват POP3 протокола, ще могат чрез програмите си за поща да изпращат и получават писма в домейна. След като се инсталира услугата, се задава име по подразбиране за пощата (например microsoft.com) и се създават пощенските кутии. Това

е услуга, която е най-добра за малки офиси или за отдалечен достъп до пощата, когато няма много голяма кореспонденция. При голяма кореспонденция се препоръчва продукта Microsoft Exchange Server.

- **Print server** - Сървър за печат. Чрез конфигурирането на тази роля може да се управлява достъпа до мрежовите принтери, драйверите им и опашката с документите за печат. След инсталирането лесно може да се настройват принтерите и техните драйвери.
- **Remote access/VPN server** - Отдалечен достъп/ Virtual private Network сървър. Позволява създаването на виртуални мрежи в организацията. Тези мрежи в някои случаи са много ценни, когато се ползват извън офиса/компанията и се искат да имат достъп до офис-мрежата. Дава се възможност за конфигуриране на връзките за отдалечено свързване, за задаване на права на потребителите да могат да се свързват към офисната мрежа и дали може да се осъществява свързване от офис-мрежата навън. Осигурява се възможност да се управлява мрежовия трафик и dial-up networking (комуникацията през телефон) и virtual private networking (VPN). При конфигурирането на тази опция се стартира помощника „Routing and Remote Access Setup Wizard”.
- **Server cluster node** - Клъстерен сървър. Когато група сървъри работят съвместно и изглеждат като един за външния свят, те формират т.нар. „клъстер” (cluster). Сървър, който работи в такава група сървъри, се нарича „клъстерен сървър”. Тази опция стартира помощника „New Server Cluster Wizard, който позволява да създаде нова клъстерна група или да добави нов сървър към съществуващ вече клъстер чрез „Add Nodes Wizard”. (Тази роля е валидна само за версиите Enterprise и Datacenter на сървър 2003)
- **Streaming media server** - Медияен сървър. Тази роля дава възможност да разпространяват потокова мултимедийна (музика/видео/звук) информация по мрежата. При конфигурирането на тази роля се инсталира Windows Media Services. (Тази роля е валидна само за Standard и Enterprise версии на сървър 2003.)
- **Terminal Server** - Терминален сървър. Тази роля дава възможност да се обслужват множество клиенти, свързани към сървъра посредством терминална сесия. Не е необходима инсталация на терминалния сървър, за да може да се осъществява отдалечено свързване към компютъра. Услугата, наречена „Remote Desktop”, чрез която

се осъществява отдалечено свързване, се инсталира по време на инсталацията на операционната система.

- WINS server - Чрез тази роля сървъра ще може да преобразува NetBIOS имена до IP адреси, и обратното.

2.2. Exchange Server 2007

Exchange Server 2007 предлага функционалности, които да позволят на потребители да имат достъп до електронна поща, календари и др. от всяко място. Exchange Server се инсталира върху платформен сървър който е с операционна система Windows Server 2003 или Windows Server 2008.

Основните характеристики за Exchange Server 2007 :

Calendar Attendant

Calendar Attendant намалява риска от дублиране на ангажименти като следи за последните промени в календарното планиране (покани за срещи, потвърждения, откази). Освен това, Calendar Attendant маркира поканите за срещи като „несигурни“ в календарите на адресатите докато потребителите не потвърдят/отхвърлят поканата. Calendar Attendant разчита на уеб услугата „свободен/зает“ на Exchange Server 2007 за актуална информация относно планираните ангажименти.

Resource Booking Attendant

Resource Booking Attendant позволява автоматично управление на ресурси, включително заседателни зали и техническо оборудване. Могат да се получават автоматични потвърждения за налични ресурси или автоматични откази със съответната обосновка за тях. Администраторите могат да задават детайлни политики за ползване на наличните ресурси – свободни часове за ползване на техниката или планиране на разрешения за нейното ползване.

Scheduling Assistant

Scheduling Assistant помага на потребителите да организират срещите си по-ефективно чрез предоставяне на информация за удобни дати и часове съгласувани с времето на участниците в срещата, както и информация за нужните технически ресурси.

Програмируеми съобщения „Out of Office”

Съобщенията „Out of Office” (OOF) могат бъдат настроени така, че да бъдат изпращани на определени дати или часове. Друга опция, която администраторите могат да активират или дезактивират е изпращането на различни от първоначално зададените „Out of Office” съобщения до външни адресати.

Достъп до електронна поща през мобилни устройства

Възможност за търсене на информация

С възможностите за търсене на Exchange ActiveSync, нужната информация може да бъде бързо открита от всяко мобилно устройство. При команда търсене от мобилно устройство, заявката се изпълнява както в мобилното устройство, така и в пощенската кутия на потребителя. Резултатите от заявката за търсене лесно могат да бъдат достъпни от мобилното устройство. Тази възможност позволява достъп до информация изпратена или получена преди дни, седмици и дори месеци, без значение от паметовите ресурси на мобилното устройство.

Direct Push

Мобилните устройства, разполагащи с Exchange ActiveSync поддържат сигурна връзка с Exchange Server 2007 и получават нова или актуализирана електронна поща, календари, контакти и задачи веднага щом бъдат доставени. Методът „Direct Push” оптимизира

пропускателната способност и доставя актуална информация на потребителите.

Богато потребителско изживяване с богата гама от устройства

Потребителите могат да се радват на познато изживяване с богата гама от мобилни устройства, без да се нуждаят организацията да внедрява скъп софтуер и услуги от трета страна. Протоколът Exchange Server 2007 ActiveSync е лицензиран за ползване от Windows Mobile, Nokia, Symbian, Motorola, Sony Ericsson, Palm и DataViz. Като се има предвид богата гама от партньори, изборът на устройства продължава да се разширява.

Защита и управление на мобилни устройства

Администраторите имат възможност да налагат политики върху ползването на мобилни устройства в тяхната организация като поставяне на изисквания за дължина и степен на защита на ПИН кодовете и могат да изпълняват отдалечено изтриване на данните и приложенията в устройствата, в случай че те бъдат откраднати или изгубени. Тези контроли стават по-прецизни с Exchange Server 2007 и позволяват налагане на политики за всеки отделен потребител. Ползването на мобилните устройства може да се проследи и управлява от средата на Exchange Server. Service Pack 1 (SP1) включва 28 нови политики за мобилни устройства, за мрежата, приложенията и контролите за сигурност.

LinkAccess

Когато потребителите получат линк за Windows SharePoint Services сайт или за споделен файл на мобилно устройство, Exchange Server 2007 използва LinkAccess, за да покаже документа. Не е необходим VPN или тунел.

Календари и „Out of Office” съобщения

С Exchange Server 2007 потребителите разполагат с много повече опции, когато достъпват своите календари през мобилни устройства чрез Exchange ActiveSync. Те могат да потвърждават ангажименти за срещи със съобщения, да препращат покани за срещи на други лица и да проследяват всички потвърждения от страна на предвидените участници в дадена среща. Съобщения от типа „Out of Office” също могат да бъдат настроени за изпращане от мобилни устройства.

Уеб-базирана електронна поща

Outlook 2007 изживяване

Outlook Web Access, AJAX приложение още с първото си издание с Exchange Server 5.5, предлага богато, Outlook изживяване в браузър. Новите функционалности в Outlook Web Access 2007 позволяват на потребителите да:

- Създават „Out of Office” съобщения и да ги изпращат на вътрешни и/или външни адресати.
- Използват Scheduling Assistant, за да организират ефективно срещи.
- Да имат достъп до SharePoint документи без VPN или тунел чрез LinkAccess
- Да използват WebReady Document Viewing, за да преглеждат прикачени файлове в HTML формат, дори и без инсталиране на приложението, създадо документа.
- Да имат достъп до RSS абонаменти
- Да преглеждат съдържание в управляеми E-mail папки (Managed E-mail Folders)
- Да получават и манипулират с гласова поща и факс съобщения чрез Unified Messaging

- Да търсят в Global Address List
- Да редактират и управляват „server-side” правила за пощенската кутия
- Да изпращат и получават S/MIME съобщения
- Да имат достъп до Public Folder информация

Защита в Access

Възможностите за защита в Outlook Web Access 2007 са подобрени. Поддържа се двойна автентификация, а администраторите могат да налагат преглед на документи само в HTML формат, с цел предпазване от загуба на данни при ползване на обществени компютърни центрове.

Self-Service Support

Менюто „Опции” в Outlook Web Access 2007 позволява на потребителите бързо и лесно да разрешават типични „helpdesk” проблеми самостоятелно. Потребителите на OWA могат да изискват промяна на ПИН кода на гласова поща чрез Unified Messaging, да извършват отдалечено изтриване на цялата информация в мобилни устройства в случай на загубване или кражба, да прибавят адресанти в списъците

„сигурни адресанти” и „блокирай адресант” от средата на Outlook Web Access. Service Pack 1 (SP1) предлага опция за потвърждаване от Exchange при изпълнение на операция за отдалечено изтриване на данни от мобилни устройства (remote wipe request).

Outlook Web Access Light

Outlook Web Access Light предлага богато, Outlook Web Access изживяване при бавни връзки и включва много от новите функционалности на Outlook Web Access 2007 като възможности за изпращане на „Out of Office” съобщения (вътрешни и външни), Really Simple Syndication (RSS) абонаменти и достъп до управляеми E-mail папки (Managed E-Mail Folders).

Възможности за търсене

Пощенските кутии в Exchange Server 2007 са напълно индексирани по подразбиране и позволяват на потребителите да търсят информация в тях чрез Outlook Web Access. Реиндексването е значително по-бързо в сравнение с Exchange Server 2003, а заявките за търсене обхващат както съдържанието на самите имейли, така и съдържанието на прикачените файлове.

Отдалечен достъп до документи: LinkAccess

Когато потребителите получат линк за Windows SharePoint Services сайт или за споделен файл докато работят с Outlook Web Access, Exchange Server 2007 използва LinkAccess, за да визуализира този документ без използване на VPN или тунел.

Отдалечен достъп до документи: Преглед на документи в HTML формат

Outlook Web Access 2007 може да транскодира множество типове документи – Microsoft Word, Microsoft Excel, Microsoft PowerPoint и PDF файлове – в HTML формат, за да могат да бъдат видими за потребителите в клиентския браузър, дори и в случаи, когато приложението, създало документа не е инсталирано на

клиента. Това позволява на потребителите да бъдат продуктивни, без да бъдат повлияни от машината, която използват. При ползване на обществени компютри, разглеждането на документи е сигурно, тъй като HTML документите са защитени от Outlook Web Access при излизане от акаунт или при приключване на сесия. Service Pack 1 (SP1) включва поддръжка на Microsoft Office 2007 файлови формати.

Унифицирана поща

Система за гласова поща

Гласовата поща може да бъде съхранявана и достъпна от унифицираната пощенска кутия през Outlook, Outlook Web Access, мобилни устройства или телефон. Тази унифицирана възможност за достъп подобрява продуктивността на служителите благодарение на улеснената процедура за манипулиране с най-използваните форми на комуникация. Освен това, унифицираната поща значително намалява разходите на организацията, елиминирайки нуждата от самостоятелна система за гласова поща и възползвайки се от вече направените инвестиции в Active Directory. Exchange Server 2007 Unified Messaging може да се свърже с „legacy private branch exchange” (PBX) инфраструктура чрез IP gateway или да бъде директно свързан с определени IP PBX инсталации.

Факс система

Факс съобщенията могат да бъдат съхранявани и достъпни от унифицираната пощенска кутия през Outlook, Outlook Web Access или мобилни устройства. Unified Messaging централизира управлението на услугите за входящи факс съобщения в инфраструктурата на Exchange.

Speech-Enabled Automated Attendant

Приложението Speech-Enabled Automated Attendant приема обаждания чрез автоматичен оператор с програмируеми менюта (например „натиснете 1 за отдел „Продажби“) и проверки в директорията „Global address list“ (например „с кого желаете да се свържете?“) Потребителите могат да взаимодействат с Automated Attendant чрез „touch tone“ менюта или чрез техния глас и функцията за разпознаване на глас.

Self-Service Voice Mail Support

С помощта на „Outlook Web Access“, потребителите могат да променят ПИН кода на гласовата си поща, да записват поздрав за гласовата си поща, да записват „out-of-office“ гласови съобщения, да избират папки за достъп до електронна поща през телефон и да прослушват входящите имейли от тях с функцията „text-to-speech“.

Outlook Voice Access

Гласовият достъп до Outlook, позволява на потребителите да разполагат с достъп до Exchange пощенската си кутия през стандартен телефон, от всяко място. Чрез „touch tone“ или гласови менюта, те могат да прослушат и реагират на ангажиментите в календарите си, да прослушват имейл съобщения (функция „text to speech“), да прослушват гласовата си поща, да се обаждат на контактите си, или да се обаждат на други потребители от директорията.

Play on Phone

Exchange Unified Messaging позволява на потребителите да прослушват гласовите си съобщения от определен, избран от тях телефон. Тази услуга е подходяща, когато се намирате на обществено място и не искате да прослушвате гласовата си поща, така че всички да чуят съобщенията ви. Play on Phone рутира гласовата поща до мобилни телефони, стационарни телефони или до други номера, зададени от потребителя.

Нови известия за гласова поща

В комбинация с Office Communication Server 2007 (OCS), потребителите получават известия на техния Office Communicator клиент или на ОС-телефон за входяща гласова поща.

Директно избиране на Outlook Voice Access

С Office Communicator, потребителите могат да изберат Outlook Voice Access с едно единствено кликване на мишката, без да въвеждат вътрешен номер или ПИН.

Сървърни роли

Exchange Server 2007 е модулна система с 5 сървърни роли - Edge Transport, Hub Transport, Mailbox, Client Access и Unified Messaging, които намаляват времето за инсталиране, редуцират ръчното, слединсталационно конфигуриране от администраторите, намаляват повърхността за атаки и повишават защитата. Администраторите притежават гъвкавата възможност да внедряват само определени функционалности и услуги в сървъра и да управляват също толкова гъвкаво. Всички сървърни роли с изключение на Edge Transport могат да бъдат внедрени на един единствен сървър и само Hub Transport и Mailbox се изискват за инсталиране на Exchange Server 2007.

Edge Exchange Server има разпределена роля на spam filter или rule, или както му е самото име „острие“ – реже „вредните“ имейли. Този вид сървъри никога не се намират в домейна. Намират се на границата, като пропускателен пункт, между интернет и Exchange организацията. Edge сървърите се грижат за сигурността на имейлите. За по-голяма сигурност съществува възможност, самия Edge сървър да се намира между два firewall-a (т.н. DMZ зона). Върху Edge сървърите може да се инсталира допълнителен софтуер като SMEX, ForeFront, AntiGen. Филтрирането на имейли може да стане по разширение, контекст, размер, домейн ...

Hub Transport сървър има роля на транспортиране на имейлите от exchange организацията към интернет, и обратното (след като са минали Edge сървъра). Съответния Hub Transport сървър има ролята да намери Mailbox сървъра и да достави писмото в mailboxa му. Механизъма, на който работи разпределянето на имейли, е откриване и разпознаване на домейна (DNS) и MX record-a. На този начин Hub Transport сървъра разбира дали имейла е за самата вътрешна организация, дали става въпрос за Trust domain, или за изпращането на имейл извън вътрешната организация, т.е. в интернет. Върху Hub сървърите, също така може да се инсталират анти-спам и анти-вирусен софтуер (като например SMEX), който да се грижи за „вътрешната“ сигурност.

Mailbox сървъра има роля на контейнер в който се намират exchange базите с mailbox-ите. Също така, в Mailbox сървъра се записват всички действия на изпратените, както и получените имейли, в така наречените транзакционни логове.

Базите може се намират в Storage група . Всяка от тях може да има определен лимит, който ще наследи всяка база която се намира в съответната Storage група, както и всеки mailbox, който се намира в тази база. Един mailbox, може да съдържа няколко SMTP адреса, от които само един е primary. Също така, може да се правят настройки, разрешения и забрани за съответния потребител.

Client Access сървърите, имат роля за достъп на потребителите, до техните имейл пощенски кутии. Върху тях може да е инсталиран имейл клиент (на пример MS Outlook), chat assistant (като например Office Communicator) , ActiveSync (дава възможност за достъп до пощата на потребителя чрез телефон), както и Outlook Web Access (уеб версия на Outlook-а), който дава възможност на потребителите достъп на mailbox-а през интернет, чрез авторизация (user name & password) или срещу сертификат.

Unified Messaging Server в Exchange Server 2007 позволява на потребителите да получават електронна поща, гласова поща и факс съобщения в една унифицирана пощенска кутия, достъпна от всяко място. Потребителите могат да управляват всичките си съобщения от едно единствено място, точно както управлението на електронна поща. Гласовата поща, например, може да се препраща, може да се изпълнява търсене на всякакъв тип обекти – писма, задачи, контакти с помощта на вградените възможности за търсене.

Снимка 2. Топология на Exchange сървърните роли

2.3. PowerShell

Exchange Management Shell е базиран на Microsoft Windows PowerShell и представлява разширяема и гъвкава среда за управление, допълваща графичния интерфейс на Exchange Management Console. Командният шел на Exchange дава възможност за бързо управление с командни скриптове – автоматизиране, „batching”, докладване - и се интегрира с Active Directory. За да помогне на администраторите бързо да научат синтаксиса на Exchange Management Shell и вградените шаблонни скриптове, различните помощници от графичната конзола за управление показват синтаксиса на всяко действие от командния ред, което е зададено в помощника. Текстът може да бъде „изрязан” и „поставен” директно в Exchange Management Shell или в скриптов файл.

1. 3. Описание на топологията, хардуерната и софтуерната част

3.1. Exchange топология

Планирането на Exchange топология се гради на 3 нива:

- Мрежово

- Active Directory

- Exchange

Мрежовото ниво представлява основа за комуникация, както между сървърите, така и между клиентските персонални компютри и сървърите. Физическото ниво на мрежата дефинира пътя, който ползват сървърите и работните станции, за да получават и изпращат данни. То съдържа IP адресите, IP subnet адресите, LAN или WAN, router-и, firewall-и, кабели...

Логическото мрежово ниво следи конвенционната схема на превръщането на името в IP адрес, базиран на Domain Name System (DNS). Чрез DNS-а настройваме зоните чрез, които ще се осъществява превръщането на IP адресите в имена и обратното. Основната част за Exchange 2007 DNS конфигурирането са MX записите. Тези записи представляват ресурсни записи в DNS-а, които специфицират Mail сървъра, който е отговорен за получаване на имейли, от името на получателите и техния домейн. В зависимост от това как е настроен MX записа в домейна, по този начин съобщенията ще бъдат препратени чрез SMTP протокола.

Active Directory нивото се грижи за инфраструктурата, която може да ползват членовете на домейна, за връзка с на най-близкия сървър, както и за контрол на трафика на репликациите между сървърите. Active Directory нивото има две поднива: физическо и логическо. Физическото ниво представляват – домейн контролери, глобални каталози и активната директория (Domain Controllers, global catalog servers, Active Directory). Логическото ниво представляват - потребители, групи, дървета, сайтове, домейни, организационни единици.

Exchange нивото дефинира рутирането на имейлите между exchange сървърите. Това ниво, както и предишните две, също така има две поднива: физическо – exchange сървърите и mail конекторите; логическо: mailbox, public folder, database, distribution list, organization.

3.2. Хардуерни изисквания за Exchange организация

В зависимост от организацията за която ще се дизайнира Exchange топология, следва проектирането на хардуера. Сървърите трябва да имат следните хардуерни характеристики:

Процесор: Intel или AMD с x64 архитектура.

RAM памет: Минимум 2 GB RAM на сървър. В случай че един сървър има инсталирани две роли (примерно Mailbox и Hub Transport), препоръчителни са 4GB

Видео карта: Видео карта поддържаща DirectX9

Дисково пространство: Минимум от 1.2 GB свободно пространство на хард диска за да се инсталира Exchange Server 2007 . Дисковете задължително трябва да са RAID5 или RAID10, защото при неизправност на „обикновения“ диск ще има отрицателен ефект върху exchange базите, съответно потребителите няма да имат достъп до пощенските кутии. При конфигуриран клъстер е препоръчително да се използва SAN. Следователно, всяка база да положи на отделен дял. Големината на дяла зависи от

предвидената големина на базата. Exchange Transaction Logs са логове в които се записва всяко действие случило се върху exchange сървърите. Те заемат голяма част от дисковото пространство.

Firewall: Juniper, Cisco или сървър с конфигурирана и програмирана за тази функция UX-система.

Routers and switches : Cisco

Кабели: UTP LAN, оптични

Backup ленти: Seagate, Sony, Tandberg Data

3.3. Софтуерни изисквания за Exchange организацията

Операционна система: За инсталирането на Exchange Server 2007 , сървъра трябва да има инсталирана Windows операционна система (Windows Server 2003 или Windows Server 2008) . Зависимост от нуждите и конфигурацията се инсталира Edition-а, който най-добре отговаря на изискванията. Примерно ако искаме да конфигурираме Mailbox сървъра като single copy cluster (SCC) или cluster continuous replication (CCR), трябва да имаме инсталиран indows Server 2003 Enterprise Edition or Windows Server 2008 Enterprise.

Относно Edge сървърната роля, тя може да бъде заместена с UX операцияна система, като например RedHat, HP-UX и т.н.

Windows server features: IIS – Internet Information Service : Exchange 2007 изисква инсталирането на IIS за Mailbox сървърна роля, който е един от характеристиките (feature) на Windows Server;

World Wide Web Publishing Service (W3SVC): Exchange 2007 изисква инсталирането на W3SVC за Client Access сървърна роля, както и Remote procedure call (RPC), Hypertext Transfer Protocol (HTTP), Proxy Windows networking component, ASP.NET version 2.0;

Unified Messaging сървърната роля има нужда от: Microsoft Exchange Speech Engine service; Microsoft Windows Media Encoder; Microsoft Windows Media Audio Voice Codec; Microsoft Core XML Services (MSXML) 6.0;

Анти-вирус и анти-спам: ForeFront, AntiGen, Trend Micro SMEX, Message Labs, Symantec, McAfee

Клиентски софтуер: MS Office Outlook 2003/2007; Web browser (за Outlook Web Access); Windows Mobile 5.0-6.1 (за мобилни телефони – нужно е да имат Messaging and Security Feature Pack); Exchange ActiveSync (за мобилни телефони които ползват операционна система различна от Windows Mobile)

Backup софтуер: Volume Shadow Copy Service (VSS); NTBackup; Data Protector; BackupExec; Legato; Snap Mirror for Exchange (SME)

Archive софтуер: IAP – Integrated Archive Platform, Enterprise Vault

1. 4. Дефиниране на проблема

4.1. Пример с организацията „Exchange”

Имаме компания, която има нужда от Exchange среда. Архитекта на топологията прави схема, в границите на изискванията на компанията. Exchange средата е следната:

снимка 3. Архитектура на организацията наречена Exchange

MBX1, MBX2 – Mailbox server

Edge – Edge Server

HUB – Hub Transport Server

DC – Domain Controller

CAS – Client Access Server

ARC – Archive Server

Започва се калкулацията на разходите, и компанията решава да замести някои елементи от Exchange средата, за да спести от разходите. След обмислянето за по-икономично решение и cost saving (спестяване на финансови средства), архитекта на топологията предлага Edge сървъра да е с UNIX операционна система, както и съкращението на дисковото пространство от Mailbox сървърите (предназначени за процеса на възстановяване на имейли), за да има повече дисково пространство на архивния сървър, върху който ще се архивират имейлите.

Едно от важните неща, в цялата Exchange архитектура е предвиждане на дисково пространство специално предопределено за процеса на възстановяване на имейли.

Възстановяването на един потребителски имейл включва няколко стъпки.

Първоначално ни е нужна следната информация:

Трябва да знаем

1. Кога е изтрил имейла?
2. В кой Exchange сървър и в коя Exchange база е бил потребителя когато е имал изтрилият имейл ?

Запитването се отправя към потребителя. Потребителя няма достатъчно технически знания, съответно отговора на първия въпрос е много по-лесен от отговора на втория въпрос – къде се (е) намира(ла) пощенската кутия на потребителя когато искания имейл е бил в пощенската му кутия.

Exchange базите, са контейнери които се намират в така наречените Storage Group. Една Storage Group може да съдържа няколко бази. Всяка база се намира на отделен дисков дял. По някога се налага местенето на потребителите от една база в друга с цел баланс на дисковото пространство. Това е един от процесите който се записва в Transaction Log-овете .

Какво се случва когато нямаме достатъчно дисково пространство? – потребителите се местят от една база в друга, за да се запази баланс и по този начин не може да бъдем сигурни къде е било местоположението на потребителя когато си е изтрил имейла. Може да го видим в Transaction Log-овете но, понеже нямаме достатъчно дисково пространство, Transaction Log-овете се или презаписват или Backup-ват. В случай че се Backup-ват, удължава се процеса на търсенето на отговора на втория въпрос. Трябва да правим възстановяване на Transaction Log-овете, които не са на бройка малко, след което трябва да отваряме log след log за да търсим потребителя, но разбира се, в тези логове не се записва само местенето на пощенските кутии, а много повече неща (путирането на имейлите, препращането, промяна на настройки и т.н.)

Възстановяването на имейла е процес при който се прави възстановяване на Exchange базата на файлово ниво, както и логовете ѝ, за съответната дата. Exchange базите са с разширение .edb. От там се намира потребителя който има нужда за възстановяване на изтрития имейл.

Процеса се изпълнява през Exchange Management Console, чрез който се създава Recovery Storage Group, след което се добавя Exchange базата, прави се възстановяване на файлово ниво на базата, от която се „измъква“ потребителската пощенска кутия, в онзи вид, в каквато е била на съответната дата, преди да се изтрие имейла.

Целия процес се случва върху отделно чисто дисково пространство, предназначено само за този процес. Не се препоръчва да се прави възстановителен процес, върху дял на който вече съществува база която се ползва. Риска е че може да свърши дисковото пространство при възстановяване на другата база, която вече се ползва, след което ще се откачи от Exchange сървъра, след което вече всички потребители намиращи се в тази база, няма да имат достъп до пощенските си кутии. Проблемата е че, по този начин и ще спре възстановяването на базата на файлово ниво, както и на логовете.

Препоръчително е, всеки Mailbox сървър да има дял предназначен за този процес. Но в случая не е така – от двата Mailbox сървъра, само единия има предназначен дял за възстановяване на имейли. Веднага следва въпроса – какво ще стане ако искаме да възстановим изтрити имейли на потребите, намиращ се на другия сървър и възможно ли е ? Отговора е да, можем да го направим. В Exchange 2007 и Management Console не съществува тази възможност. Единствения начин е през Exchange PowerShell-а. В Exchange 2010 не съществува графичен вариант за който и да е от двата случая, при възстановяване на имейл. Microsoft са заложили на мощния команден шел, с цел че exchange администраторите да разбират от писане на скриптове. В зависимост от конфигурацията на Exchange сървърите, редовете писани в PowerShell-а в някои случаи стават много дълги, по някога са и доста объркващи, след което при неправилно написан синтаксис, процеса не се изпълнява, след което започва търсене в MS Exchange PowerShell Help-а (кое, къде, как трябва да се напише). По този начин се удължава времето за разрешаване на потребителския проблем. Потребителя, следователно остава недоволен.

1. 5. Създаване на програма за разрешаването на проблемаю

5.1. Сорс код на PowerShell script

Създаване на функция за създаване на формата

```
function GenerateForm {
```

```
[reflection.assembly]::loadwithpartialname("System.Windows.Forms") | Out-Null
```

```
[reflection.assembly]::loadwithpartialname("System.Drawing") | Out-Null
```

Създаване на формата и обекти

```
$form1 = New-Object System.Windows.Forms.Form
```

```
$tabs = New-Object System.Windows.Forms.TabControl
```

```
$tab1 = New-Object System.Windows.Forms.TabPage
```

```
$tab1_comboBox3 = New-Object System.Windows.Forms.ComboBox
```

```
$tab1_comboBox2 = New-Object System.Windows.Forms.ComboBox
```

```
$tab1_comboBox1 = New-Object System.Windows.Forms.ComboBox
```

```
$tab1_result_label = New-Object System.Windows.Forms.Label
```

```
$tab1_check_btn = New-Object System.Windows.Forms.Button
```

```
$tab1_db_label = New-Object System.Windows.Forms.Label
```

```
$tab1_sg_label = New-Object System.Windows.Forms.Label
```

```
$tab1_mailbox_label = New-Object System.Windows.Forms.Label
```

```
$tab1_displayname_textbox = New-Object System.Windows.Forms.TextBox
```

```
$tab1_displayname_label = New-Object System.Windows.Forms.Label
```

```
$tab2 = New-Object System.Windows.Forms.TabPage
```

```
$tab2_mountDBRsg = New-Object System.Windows.Forms.Button
```

```
$tab2_DeleteRSG_btn = New-Object System.Windows.Forms.Button
```

```
$tab2_OpenLogPath_btn = New-Object System.Windows.Forms.Button
```

```
$tab2_OpenDBPath_btn = New-Object System.Windows.Forms.Button
```

```
$tab2_RsgSRV_Cbox = New-Object System.Windows.Forms.ComboBox
```

```
$tab2_RSGSRV_lbl = New-Object System.Windows.Forms.Label
```

```
$tab2_output_txt = New-Object System.Windows.Forms.TextBox
```

```
$tab2_checkRSG_btn = New-Object System.Windows.Forms.Button
```

```
$tab2_CreateRsg_btn = New-Object System.Windows.Forms.Button
```

```
$logpath = New-Object System.Windows.Forms.Label
```

```
$dbpath = New-Object System.Windows.Forms.Label
```

```
$tab3 = New-Object System.Windows.Forms.TabPage
```

```
$tab3_output_txt = New-Object System.Windows.Forms.TextBox
```

```
$tab3_restoreMbx_btn = New-Object System.Windows.Forms.Button
```

```
$tab3_restoreFolder_txt = New-Object System.Windows.Forms.TextBox
```

```
$tab3_restoreFolderTrue_chbx = New-Object System.Windows.Forms.CheckBox
```

```
$tab3_RestFolder_lbl = New-Object System.Windows.Forms.Label
```

```
$tab3_mbxWA_txt = New-Object System.Windows.Forms.TextBox
```

```
$tab3_winMA_lbl = New-Object System.Windows.Forms.Label
```

```
$tab3_mbxFromRSG = New-Object System.Windows.Forms.ComboBox
```

```
$tab3_mbxRSG_lbl = New-Object System.Windows.Forms.Label
```

```
$openFileDialog1 = New-Object System.Windows.Forms.OpenFileDialog
```

```
$openFileDialog2 = New-Object System.Windows.Forms.OpenFileDialog
```

```
$InitialFormWindowState = New-Object System.Windows.Forms.FormWindowState
```

```
### Дефиниране на променливите за нов ред, сканиране на mailbox
```

```
### сървърите
```

```
$nl = [Environment]::NewLine
```

```
$mbxSrvScan=$false
```

```
function OnApplicationExit {
```

```
$script:ExitCode = 0 #Set the exit code for the Packager
```

```
}
```

```
### Дефиниране на бутона за проверяване на display name, извеждане на ### детайли  
за mailbo  
x-a ,  
извеждане на съобщение
```

```
$check_btn_click={
```

```
$predifined_DN = $tab1_displayname_textbox.text
```

```
if($predifined_DN -ne "" -and $predifined_DN -ne $null){
```

```
$pre_search_mailbox = Get-MailboxServer | Get-MailboxStatistics | ?{$_.DisplayName -like  
"$predifined_DN"};
```

```
if($pre_search_mailbox -eq $null){
```

```
$tab1_result_label.text="A mailbox with this display name cannot be found"
```

```
}else{
```

```
foreach($search_mailbox in $pre_search_mailbox){
```

```
$foundMbxDB = $search_mailbox.Database;
```

```
$foundMbxDD = $search_mailbox.DisconnectDate;
```

```
$foundMbxDN = $search_mailbox.DisplayName;
```

```
$foundMbxSRV = $search_mailbox.ServerName;
```

```
$foundMbxSG = $search_mailbox.StorageGroupName;
```

```
$foundMbxDBO = $search_mailbox.DatabaseName;
```

```
$tab1_result_label.text+="$nl"
```

```
$tab1_result_label.text+="$foundMbxDN - $foundMbxDD - $foundMbxDB"
```

```
$tab1_comboBox1.Items.add($foundMbxSRV)
```

```
$tab1_comboBox2.Items.add($foundMbxSG)
```

```
$tab1_comboBox3.Items.add($foundMbxDBO)
```

```
$tab1_comboBox1.SelectedItem = $foundMbxSRV
```

```
$tab1_comboBox2.SelectedItem = $foundMbxSG
```

```
$tab1_comboBox3.SelectedItem = $foundMbxDBO
```

```
}
```

```
}
```



```
}else{ $tab1_result_label.text = "Please Enter Display Name for which the check to be  
performed"; }
```

```
}
```

Дефиниране на променлива и функция за изтриване на RSG

```
$deleteRSG={
```

```
$getRsgDB = Get-StorageGroup -server $tab2_RsgSRV_Cbox.SelectedItem | ?{$_.Recovery  
-eq $true} | Get-MailboxDatabase
```

```
$Error.Clear();
```

```
Dismount-Database $getRsgDB -Confirm:$false
```

```
if(!$Error) {
```

```
$tab2_output_txt.text += "$nl"
```

```
$tab2_output_txt.text += "The RSG Database was dismantled successfully"
```

```
$RSG = Get-StorageGroup -server $tab2_RsgSRV_Cbox.SelectedItem | ?{$_Recovery -eq  
$true}
```

```
Remove-StorageGroup $RSG
```

```
}else {
```

```
$tab2_output_txt.text += "$nl"
```

```
$tab2_output_txt.text += "The RSG Database was not dismantled successfully $error"
```

```
}
```

```
}
```

```
$open_log_btn={
```

```
$openFileDialog2.ShowDialog();
```

```
}
```

```
$open_db_btn={
```

```
$openFileDialog1.ShowDialog();
```

```
}
```

```
$RSG_Tab_Click={
```

```
if($mbxSrvScan -eq $false) {
```

```
$mbxSrvList = Get-MailboxServer
```

```
foreach($obj in $mbxSrvList){
```

```
$tab2_RsgSRV_Cbox.Items.add($obj)
```

```
}
```

```
$mbxSrvScan=$true
```

```
}
```

```
}
```

```
### Дефиниране на променлива и функция за проверяване за вече съществуваща RSG
```

```
$CheckForRSG={
```

```
$selectedServer = $tab2_RsgSRV_Cbox.SelectedItem
```

```
$getRsgDatabase = Get-StorageGroup -server $selectedServer | ?{$_Recovery -eq $true}
```

```
$RSGName = $getRsgDatabase.Name;
```

```
if($getRsgDatabase -eq $null -or $getRsgDatabase -eq ""){
```

```
$tab2_output_txt.text += "$nl"
```

```
$tab2_output_txt.text += "There isn't any existing RSG on the server"
```

```
}else {
```

```
$tab2_output_txt.text += "$nl"
```

```
$tab2_output_txt.text += "There is existing RSG on the server"
```

```
}
```

```
}
```

```
### Дефиниране на променлива и функция за създаване на RSG
```

```
$CreateRSG={
```

```
$error.clear();
```

```
if(new-storagegroup -server $selectedServer -logfolderpath $logPathConv -name "Recovery  
Storage Group" -SystemFolderPath $dbPathConv -recovery){
```

```
$tab2_output_txt.text += "$nl"
```

```
$tab2_output_txt.text += "The RSG was created successfully"
```

```
}else{
```

```
$tab2_output_txt.text += "$nl"
```

```
$tab2_output_txt.text += "The RSG creation FAILED - $error"
```

```
}

}

$handler_openFileDialog2_FileOk=[System.ComponentModel.CancelEventHandler]{

$tab2_output_txt.text += "$nl";

$log_strPath = openFileDialog2.FileName

$logPathConv= $log_strPath.Substring(0,$log_strPath.LastIndexOf("\"));

$tab2_output_txt.text += "Current Location For LOGs - $logPathConv"

}
```

```
$handler_openFileDialog1_FileOk=[System.ComponentModel.CancelEventHandler]{
```

```
$tab2_output_txt.text += "$nl";
```

```
$db_strPath = $openFileDialog1.FileName
```

```
$dbPathConv= $db_strPath.Substring(0,$db_strPath.LastIndexOf("));
```

```
$tab2_output_txt.text += "Current Location For DB - $dbPathConv";
```

```
}
```

Дефиниране на променливи и функции на процеса за възстановяване

```
$LoadRestoreTab={
```



```
$tab3_mbxFromRSG.Items.Clear();
```

```
$SRV = $tab2_RsgSRV_Cbox.SelectedItem;
```

```
$getRsgSG = Get-StorageGroup -server $SRV | ?{$_Recovery -eq $true}
```

```
$getRsgDB = $getRsgSG | Get-MailboxDatabase
```

```
$RSGName = $getRsgSG.Name;
```

```
$availableMbxRsg = Get-ExchangeServer | ?{$_Name -like $SRV} | Get-MailboxStatistics |  
?{$_StorageGroupName -like $RSGName -and $_DisplayName -notlike "*System*"};
```

```
$tab3_output_txt.Text += "$nl"
```

```
$tab3_output_txt.Text += ""
```

```
foreach($avMbxRsg in $availableMbxRsg) {
```

```
$mbxName = $avMbxRsg.DisplayName
```

```
$tab3_mbxFromRSG.Items.Add($mbxName);
```

```
}
```

```
$tab3_mbxFromRSG.Sorted = $true;
```

```
$selectMbx = $tab1_displayname_textbox.Text
```

```
$tab3_mbxFromRSG.SelectedItem = $selectMbx
```

```
$tab3_mbxWA_txt.Text = $tab1_displayname_textbox.Text
```

```
}
```

```
$handler_tab3_restoreFolderTrue_chbx_CheckedChanged={
```

```
if($tab3_restoreFolderTrue_chbx.Checked -eq $true){
```

```
$tab3_restoreFolder_txt.Text = "Restored From "+$tab3_mbxWA_txt.Text;
```

```
}else { $tab3_restoreFolder_txt.Text = "" }
```

```
}
```

```
$handler_tab3_restoreMbx_btn_Click={
```

```
$Error.Clear();
```

```
if($tab3_mbxFromRSG.SelectedItem -ne $null -and $tab3_mbxWA_txt.Text -ne ""){
```

```
if(get-user $tab3_mbxFromRSG.SelectedItem){
```

```
if($tab3_restoreFolderTrue_chbx.Checked -eq $true){
```

```
Restore-Mailbox -RSGMailbox $tab3_mbxFromRSG.SelectedItem -RSGDatabase $getRsgDB  
-id $tab1_displayname_textbox.Text -TargetFolder $tab3_restoreFolder_txt.Text  
-Confirm:$false
```

```
}else {
```

```
Restore-Mailbox -Identity $tab3_mbxFromRSG.SelectedItem -RSGDatabase $getRsgDB  
-Confirm:$false
```

```
}
```

```
if(!$Error){
```

```
$tab3_output_txt.text += "$nl"
```

```
$tab3_output_txt.text += "The Restore of the mailbox was successfull"
```

```
}else {
```

```
$tab3_output_txt.text += "$nl"
```

```
$tab3_output_txt.text += "The Restore of the mailbox FAILED - $error"
```

```
}
```

```
}else {
```

```
$tab3_output_txt.Text += "$nl"
```

```
$tab3_output_txt.Text += "The provided User's SamAccountName is invalid Please check."
```

```
}
```

```
}
```

```
}
```

```
$handler_tab2_mountDBRsg_Click={
```

```
$getRsgDB = Get-StorageGroup -server $tab2_RsgSRV_Cbox.SelectedItem | ?{$_Recovery  
-eq $true} | Get-MailboxDatabase
```

```
$RSGName = $getRsgSG.Name;
```

```
$Error.Clear();
```

```
set-mailboxdatabase -identity $getRsgDB -AllowFileRestore:$True
```

```
mount-database -identity "$getRsgDB"
```

```
if(!$Error) {
```

```
$tab2_output_txt.text += "$nl"
```

```
$tab2_output_txt.text += "The RSG Database was mounted successfully"
```

```
}else {
```

```
$tab2_output_txt.text += "$nl"
```

```
$tab2_output_txt.text += "The RSG Database could not be mounted - $error"
```

```
}
```

```
}
```

```
$LoadRSGTab={
```

```
}
```

```
$OnLoadForm_StateCorrection=
```

```
{#Correct the initial state of the form to prevent the .Net maximized form issue
```

```
$form1.WindowState = $InitialFormWindowState
```

```
}
```

```
#-----
```

```
#region Generated Form Code
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 465
```

```
$System_Drawing_Size.Width = 706
```

```
$form1.ClientSize = $System_Drawing_Size
```

```
$form1.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$form1.Name = "form1"
```

```
$form1.Text = "Mailbox Restore Utility"
```

```
$tabs.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 12
```

```
$System_Drawing_Point.Y = 21
```

```
$tabs.Location = $System_Drawing_Point
```

```
$tabs.Name = "tabs"
```



```
$tabs.SelectedIndex = 0
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 432
```

```
$System_Drawing_Size.Width = 682
```

```
$tabs.Size = $System_Drawing_Size
```

```
$tabs.TabIndex = 0
```

```
$form1.Controls.Add($tabs)
```

```
$tab1.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 4
```

```
$System_Drawing_Point.Y = 22
```

```
$tab1.Location = $System_Drawing_Point
```

```
$tab1.Name = "tab1"
```

```
$System_Windows_Forms_Padding = New-Object System.Windows.Forms.Padding
```

```
$System_Windows_Forms_Padding.All = 3
```

```
$System_Windows_Forms_Padding.Bottom = 3
```

```
$System_Windows_Forms_Padding.Left = 3
```

```
$System_Windows_Forms_Padding.Right = 3
```

```
$System_Windows_Forms_Padding.Top = 3
```

```
$tab1.Padding = $System_Windows_Forms_Padding
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 406
```

```
$System_Drawing_Size.Width = 674
```

```
$tab1.Size = $System_Drawing_Size
```

```
$tab1.TabIndex = 0
```

```
$tab1.Text = "User info"
```

```
$tab1.UseVisualStyleBackColor = $True
```

```
$tabs.Controls.Add($tab1)
```

```
$tab1_comboBox3.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$tab1_comboBox3.FormattingEnabled = $True
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 137
```

```
$System_Drawing_Point.Y = 212
```

```
$tab1_comboBox3.Location = $System_Drawing_Point
```

```
$tab1_comboBox3.Name = "tab1_comboBox3"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 21
```

```
$System_Drawing_Size.Width = 247
```

```
$tab1_comboBox3.Size = $System_Drawing_Size
```

```
$tab1_comboBox3.TabIndex = 12
```

```
$tab1.Controls.Add($tab1_comboBox3)
```

```
$tab1_comboBox2.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$tab1_comboBox2.FormattingEnabled = $True
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 137
```

```
$System_Drawing_Point.Y = 146
```

```
$tab1_comboBox2.Location = $System_Drawing_Point
```

```
$tab1_comboBox2.Name = "tab1_comboBox2"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 21
```

```
$System_Drawing_Size.Width = 247
```

```
$tab1_comboBox2.Size = $System_Drawing_Size
```

```
$tab1_comboBox2.TabIndex = 11
```

```
$tab1.Controls.Add($tab1_comboBox2)
```

```
$tab1_comboBox1.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$tab1_comboBox1.FormattingEnabled = $True
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

`$System_Drawing_Point.X = 137`

`$System_Drawing_Point.Y = 83`

`$tab1_comboBox1.Location = $System_Drawing_Point`

`$tab1_comboBox1.Name = "tab1_comboBox1"`

`$System_Drawing_Size = New-Object System.Drawing.Size`

`$System_Drawing_Size.Height = 21`

`$System_Drawing_Size.Width = 247`

`$tab1_comboBox1.Size = $System_Drawing_Size`

`$tab1_comboBox1.TabIndex = 10`

```
$tab1.Controls.Add($tab1_comboBox1)
```

```
$tab1_result_label.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 18
```

```
$System_Drawing_Point.Y = 313
```

```
$tab1_result_label.Location = $System_Drawing_Point
```

```
$tab1_result_label.Name = "tab1_result_label"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```


\$System_Drawing_Size.Height = 78

\$System_Drawing_Size.Width = 639

\$tab1_result_label.Size = \$System_Drawing_Size

\$tab1_result_label.TabIndex = 9

\$tab1.Controls.Add(\$tab1_result_label)

\$tab1_check_btn.DataBindings.DefaultDataSourceUpdateMode = 0

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 178
```

```
$System_Drawing_Point.Y = 262
```

```
$tab1_check_btn.Location = $System_Drawing_Point
```

```
$tab1_check_btn.Name = "tab1_check_btn"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 148
```

```
$tab1_check_btn.Size = $System_Drawing_Size
```

```
$tab1_check_btn.TabIndex = 8
```

```
$tab1_check_btn.Text = "Check"
```

```
$tab1_check_btn.UseVisualStyleBackColor = $True
```

```
$tab1_check_btn.add_Click($check_btn_click)
```

```
$tab1.Controls.Add($tab1_check_btn)
```

```
$tab1_db_label.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 18
```

`$System_Drawing_Point.Y = 212`

`$tab1_db_label.Location = $System_Drawing_Point`

`$tab1_db_label.Name = "tab1_db_label"`

`$System_Drawing_Size = New-Object System.Drawing.Size`

`$System_Drawing_Size.Height = 23`

`$System_Drawing_Size.Width = 85`

`$tab1_db_label.Size = $System_Drawing_Size`

`$tab1_db_label.TabIndex = 6`

`$tab1_db_label.Text = "Database"`

```
$tab1.Controls.Add($tab1_db_label)
```

```
$tab1_sg_label.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 18
```

```
$System_Drawing_Point.Y = 146
```

```
$tab1_sg_label.Location = $System_Drawing_Point
```

```
$tab1_sg_label.Name = "tab1_sg_label"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 85
```

```
$tab1_sg_label.Size = $System_Drawing_Size
```

```
$tab1_sg_label.TabIndex = 3
```

```
$tab1_sg_label.Text = "Storage Group"
```

```
$tab1.Controls.Add($tab1_sg_label)
```

```
$tab1_mailbox_label.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 18
```

```
$System_Drawing_Point.Y = 86
```

```
$tab1_mailbox_label.Location = $System_Drawing_Point
```

```
$tab1_mailbox_label.Name = "tab1_mailbox_label"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 85
```

```
$tab1_mailbox_label.Size = $System_Drawing_Size
```

```
$tab1_mailbox_label.TabIndex = 2
```

```
$tab1_mailbox_label.Text = "Mailbox Server"
```

```
$tab1.Controls.Add($tab1_mailbox_label)
```

```
$tab1_displayname_textbox.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 137
```

```
$System_Drawing_Point.Y = 24
```

```
$tab1_displayname_textbox.Location = $System_Drawing_Point
```

```
$tab1_displayname_textbox.Name = "tab1_displayname_textbox"
```



```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 20
```

```
$System_Drawing_Size.Width = 247
```

```
$tab1_displayname_textbox.Size = $System_Drawing_Size
```

```
$tab1_displayname_textbox.TabIndex = 1
```

```
$tab1.Controls.Add($tab1_displayname_textbox)
```

```
$tab1_displayname_label.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 18
```

```
$System_Drawing_Point.Y = 24
```

```
$tab1_displayname_label.Location = $System_Drawing_Point
```

```
$tab1_displayname_label.Name = "tab1_displayname_label"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 20
```

```
$System_Drawing_Size.Width = 85
```

```
$tab1_displayname_label.Size = $System_Drawing_Size
```

```
$tab1_displayname_label.TabIndex = 0
```

```
$tab1_displayname_label.Text = "Display name"
```

```
$tab1.Controls.Add($tab1_displayname_label)
```

```
$tab2.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 4
```

```
$System_Drawing_Point.Y = 22
```

```
$tab2.Location = $System_Drawing_Point
```

```
$tab2.Name = "tab2"
```

```
$System_Windows_Forms_Padding = New-Object System.Windows.Forms.Padding
```

```
$System_Windows_Forms_Padding.All = 3
```

```
$System_Windows_Forms_Padding.Bottom = 3
```

```
$System_Windows_Forms_Padding.Left = 3
```

```
$System_Windows_Forms_Padding.Right = 3
```

```
$System_Windows_Forms_Padding.Top = 3
```

```
$tab2.Padding = $System_Windows_Forms_Padding
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 406
```

```
$System_Drawing_Size.Width = 674
```

```
$tab2.Size = $System_Drawing_Size
```

```
$tab2.TabIndex = 1
```

```
$tab2.Text = "RSG"
```

```
$tab2.UseVisualStyleBackColor = $True
```

```
$tab2.add_Enter($LoadRSGTab)
```

```
$tabs.Controls.Add($tab2)
```

```
$tab2_mountDBRsg.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 252
```

```
$System_Drawing_Point.Y = 156
```

```
$tab2_mountDBRsg.Location = $System_Drawing_Point
```

```
$tab2_mountDBRsg.Name = "tab2_mountDBRsg"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 120
```

```
$tab2_mountDBRsg.Size = $System_Drawing_Size
```

```
$tab2_mountDBRsg.TabIndex = 10
```

```
$tab2_mountDBRsg.Text = "Mount DB in RSG"
```

```
$tab2_mountDBRsg.UseVisualStyleBackColor = $True
```

```
$tab2_mountDBRsg.add_Click($handler_tab2_mountDBRsg_Click)
```

```
$tab2.Controls.Add($tab2_mountDBRsg)
```

```
$tab2_DeleteRSG_btn.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 378
```

`$System_Drawing_Point.Y = 156`

`$tab2_DeleteRSG_btn.Location = $System_Drawing_Point`

`$tab2_DeleteRSG_btn.Name = "tab2_DeleteRSG_btn"`

`$System_Drawing_Size = New-Object System.Drawing.Size`

`$System_Drawing_Size.Height = 23`

`$System_Drawing_Size.Width = 120`

`$tab2_DeleteRSG_btn.Size = $System_Drawing_Size`

`$tab2_DeleteRSG_btn.TabIndex = 9`

`$tab2_DeleteRSG_btn.Text = "Delete Existing RSG"`

`$tab2_DeleteRSG_btn.UseVisualStyleBackColor = $True`

`$tab2_DeleteRSG_btn.add_Click($deleteRSG)`


```
$tab2.Controls.Add($tab2_DeleteRSG_btn)
```

```
$tab2_OpenLogPath_btn.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 95
```

```
$System_Drawing_Point.Y = 112
```

```
$tab2_OpenLogPath_btn.Location = $System_Drawing_Point
```

```
$tab2_OpenLogPath_btn.Name = "tab2_OpenLogPath_btn"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 122
```

```
$tab2_OpenLogPath_btn.Size = $System_Drawing_Size
```

```
$tab2_OpenLogPath_btn.TabIndex = 8
```

```
$tab2_OpenLogPath_btn.Text = "Select Location"
```

```
$tab2_OpenLogPath_btn.UseVisualStyleBackColor = $True
```

```
$tab2_OpenLogPath_btn.add_Click($open_log_btn)
```

```
$tab2.Controls.Add($tab2_OpenLogPath_btn)
```

```
$tab2_OpenDBPath_btn.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 95
```

```
$System_Drawing_Point.Y = 57
```

```
$tab2_OpenDBPath_btn.Location = $System_Drawing_Point
```

```
$tab2_OpenDBPath_btn.Name = "tab2_OpenDBPath_btn"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 122
```

```
$tab2_OpenDBPath_btn.Size = $System_Drawing_Size
```

```
$tab2_OpenDBPath_btn.TabIndex = 7
```

```
$tab2_OpenDBPath_btn.Text = "Select Location"
```

```
$tab2_OpenDBPath_btn.UseVisualStyleBackColor = $True
```

```
$tab2_OpenDBPath_btn.add_Click($open_db_btn)
```

```
$tab2.Controls.Add($tab2_OpenDBPath_btn)
```

```
$tab2_RsgSRV_Cbox.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$tab2_RsgSRV_Cbox.FormattingEnabled = $True
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 95
```

```
$System_Drawing_Point.Y = 13
```

```
$tab2_RsgSRV_Cbox.Location = $System_Drawing_Point
```

```
$tab2_RsgSRV_Cbox.Name = "tab2_RsgSRV_Cbox"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 21
```

```
$System_Drawing_Size.Width = 253
```

```
$tab2_RsgSRV_Cbox.Size = $System_Drawing_Size
```

```
$tab2_RsgSRV_Cbox.Sorted = $True
```

```
$tab2_RsgSRV_Cbox.TabIndex = 6
```

```
$tab2_RsgSRV_Cbox.add_Click($RSG_Tab_Click)
```

```
$tab2.Controls.Add($tab2_RsgSRV_Cbox)
```

```
$tab2_RSGSRV_Ibl.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 26
```

```
$System_Drawing_Point.Y = 16
```

```
$tab2_RSGSRV_lbl.Location = $System_Drawing_Point
```

```
$tab2_RSGSRV_lbl.Name = "tab2_RSGSRV_lbl"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 100
```

```
$tab2_RSGSRV_lbl.Size = $System_Drawing_Size
```

```
$tab2_RSGSRV_lbl.TabIndex = 5
```

```
$tab2_RSGSRV_lbl.Text = "RSG Server"
```

```
$tab2.Controls.Add($tab2_RSGSRV_lbl)
```

```
$tab2_output_txt.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 26
```

```
$System_Drawing_Point.Y = 196
```

```
$tab2_output_txt.Location = $System_Drawing_Point
```

```
$tab2_output_txt.Multiline = $True
```

```
$tab2_output_txt.Name = "tab2_output_txt"
```

```
$tab2_output_txt.ScrollBars = 2
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 188
```



```
$System_Drawing_Size.Width = 619
```

```
$tab2_output_txt.Size = $System_Drawing_Size
```

```
$tab2_output_txt.TabIndex = 4
```

```
$tab2.Controls.Add($tab2_output_txt)
```

```
$tab2_checkRSG_btn.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 26
```

```
$System_Drawing_Point.Y = 156
```

```
$tab2_checkRSG_btn.Location = $System_Drawing_Point
```

```
$tab2_checkRSG_btn.Name = "tab2_checkRSG_btn"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 139
```

```
$tab2_checkRSG_btn.Size = $System_Drawing_Size
```

```
$tab2_checkRSG_btn.TabIndex = 3
```

```
$tab2_checkRSG_btn.Text = "Check for Existing RSG"
```

```
$tab2_checkRSG_btn.UseVisualStyleBackColor = $True
```

```
$tab2_checkRSG_btn.add_Click($CheckForRSG)
```

```
$tab2.Controls.Add($tab2_checkRSG_btn)
```

```
$tab2_CreateRsg_btn.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 171
```

```
$System_Drawing_Point.Y = 156
```

```
$tab2_CreateRsg_btn.Location = $System_Drawing_Point
```

```
$tab2_CreateRsg_btn.Name = "tab2_CreateRsg_btn"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 75
```

```
$tab2_CreateRsg_btn.Size = $System_Drawing_Size
```

```
$tab2_CreateRsg_btn.TabIndex = 2
```

```
$tab2_CreateRsg_btn.Text = "Create RSG"
```

```
$tab2_CreateRsg_btn.UseVisualStyleBackColor = $True
```

```
$tab2_CreateRsg_btn.add_Click($CreateRSG)
```

```
$tab2.Controls.Add($tab2_CreateRsg_btn)
```

```
$logpath.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 26
```

```
$System_Drawing_Point.Y = 112
```

```
$logpath.Location = $System_Drawing_Point
```

```
$logpath.Name = "logpath"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 100
```

```
$logpath.Size = $System_Drawing_Size
```

```
$logpath.TabIndex = 1
```

```
$logpath.Text = "Log Path"
```

```
$tab2.Controls.Add($logpath)
```

```
$dbpath.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 26
```

```
$System_Drawing_Point.Y = 62
```

```
$dbpath.Location = $System_Drawing_Point
```

```
$dbpath.Name = "dbpath"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

`$System_Drawing_Size.Width = 76`

`$dbpath.Size = $System_Drawing_Size`

`$dbpath.TabIndex = 0`

`$dbpath.Text = "DB path"`

`$tab2.Controls.Add($dbpath)`

`$tab3.DataBindings.DefaultDataSourceUpdateMode = 0`

`$System_Drawing_Point = New-Object System.Drawing.Point`

`$System_Drawing_Point.X = 4`

`$System_Drawing_Point.Y = 22`

```
$tab3.Location = $System_Drawing_Point
```

```
$tab3.Name = "tab3"
```

```
$System_Windows_Forms_Padding = New-Object System.Windows.Forms.Padding
```

```
$System_Windows_Forms_Padding.All = 3
```

```
$System_Windows_Forms_Padding.Bottom = 3
```

```
$System_Windows_Forms_Padding.Left = 3
```

```
$System_Windows_Forms_Padding.Right = 3
```

```
$System_Windows_Forms_Padding.Top = 3
```

```
$tab3.Padding = $System_Windows_Forms_Padding
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 406
```



```
$System_Drawing_Size.Width = 674
```

```
$tab3.Size = $System_Drawing_Size
```

```
$tab3.TabIndex = 2
```

```
$tab3.Text = "Restore"
```

```
$tab3.UseVisualStyleBackColor = $True
```

```
$tab3.add_Enter($LoadRestoreTab)
```

```
$tabs.Controls.Add($tab3)
```

```
$tab3_output_txt.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 12
```

```
$System_Drawing_Point.Y = 170
```

```
$tab3_output_txt.Location = $System_Drawing_Point
```

```
$tab3_output_txt.Multiline = $True
```

```
$tab3_output_txt.Name = "tab3_output_txt"
```

```
$tab3_output_txt.ScrollBars = 2
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 230
```

```
$System_Drawing_Size.Width = 656
```

```
$tab3_output_txt.Size = $System_Drawing_Size
```

```
$tab3_output_txt.TabIndex = 8
```

```
$tab3.Controls.Add($tab3_output_txt)
```

```
$tab3_restoreMbx_btn.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 12
```

```
$System_Drawing_Point.Y = 128
```

```
$tab3_restoreMbx_btn.Location = $System_Drawing_Point
```

```
$tab3_restoreMbx_btn.Name = "tab3_restoreMbx_btn"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 208
```

```
$tab3_restoreMbx_btn.Size = $System_Drawing_Size
```

```
$tab3_restoreMbx_btn.TabIndex = 7
```

```
$tab3_restoreMbx_btn.Text = "Restore The Mailbox"
```

```
$tab3_restoreMbx_btn.UseVisualStyleBackColor = $True
```

```
$tab3_restoreMbx_btn.add_Click($handler_tab3_restoreMbx_btn_Click)
```

```
$tab3.Controls.Add($tab3_restoreMbx_btn)
```

```
$tab3_restoreFolder_txt.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 222
```

```
$System_Drawing_Point.Y = 88
```

```
$tab3_restoreFolder_txt.Location = $System_Drawing_Point
```

```
$tab3_restoreFolder_txt.Name = "tab3_restoreFolder_txt"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 20
```

```
$System_Drawing_Size.Width = 446
```

```
$tab3_restoreFolder_txt.Size = $System_Drawing_Size
```

```
$tab3_restoreFolder_txt.TabIndex = 6
```

```
$tab3.Controls.Add($tab3_restoreFolder_txt)
```

```
$tab3_restoreFolderTrue_chbx.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 116
```

```
$System_Drawing_Point.Y = 88
```

```
$tab3_restoreFolderTrue_chbx.Location = $System_Drawing_Point
```

```
$tab3_restoreFolderTrue_chbx.Name = "tab3_restoreFolderTrue_chbx"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 24
```

```
$System_Drawing_Size.Width = 104
```

```
$tab3_restoreFolderTrue_chbx.Size = $System_Drawing_Size
```

```
$tab3_restoreFolderTrue_chbx.TabIndex = 5
```

```
$tab3_restoreFolderTrue_chbx.Text = "Enabled"
```

```
$tab3_restoreFolderTrue_chbx.UseVisualStyleBackColor = $True
```

```
$tab3_restoreFolderTrue_chbx.add_CheckedChanged($handler_tab3_restoreFolderTrue_chbx_CheckedChanged)
```

```
$tab3.Controls.Add($tab3_restoreFolderTrue_chbx)
```

```
$tab3_RestFolder_lbl.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 12
```

```
$System_Drawing_Point.Y = 93
```

```
$tab3_RestFolder_lbl.Location = $System_Drawing_Point
```

```
$tab3_RestFolder_lbl.Name = "tab3_RestFolder_lbl"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 23
```

```
$System_Drawing_Size.Width = 111
```

```
$tab3_RestFolder_lbl.Size = $System_Drawing_Size
```

```
$tab3_RestFolder_lbl.TabIndex = 4
```

```
$tab3_RestFolder_lbl.Text = "Restore to a Folder"
```

```
$tab3.Controls.Add($tab3_RestFolder_lbl)
```

```
$tab3_mbxWA_txt.DataBindings.DefaultDataSourceUpdateMode = 0
```



```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 222
```

```
$System_Drawing_Point.Y = 51
```

```
$tab3_mbxWA_txt.Location = $System_Drawing_Point
```

```
$tab3_mbxWA_txt.Name = "tab3_mbxWA_txt"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 20
```

```
$System_Drawing_Size.Width = 249
```

```
$tab3_mbxWA_txt.Size = $System_Drawing_Size
```

```
$tab3_mbxWA_txt.TabIndex = 3
```

```
$tab3.Controls.Add($tab3_mbxWA_txt)
```

```
$tab3_winMA_lbl.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 12
```

```
$System_Drawing_Point.Y = 51
```

```
$tab3_winMA_lbl.Location = $System_Drawing_Point
```

```
$tab3_winMA_lbl.Name = "tab3_winMA_lbl"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 29
```

```
$System_Drawing_Size.Width = 194
```

```
$tab3_winMA_lbl.Size = $System_Drawing_Size
```

```
$tab3_winMA_lbl.TabIndex = 2
```

```
$tab3_winMA_lbl.Text = "Display Name of the mailbox which will be merged with the Restored one"
```

```
$tab3.Controls.Add($tab3_winMA_lbl)
```

```
$tab3_mbxFromRSG.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$tab3_mbxFromRSG.FormattingEnabled = $True
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 222
```

```
$System_Drawing_Point.Y = 15
```

```
$tab3_mbxFromRSG.Location = $System_Drawing_Point
```

```
$tab3_mbxFromRSG.Name = "tab3_mbxFromRSG"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 21
```

```
$System_Drawing_Size.Width = 249
```

```
$tab3_mbxFromRSG.Size = $System_Drawing_Size
```

```
$tab3_mbxFromRSG.TabIndex = 1
```

```
$tab3.Controls.Add($tab3_mbxFromRSG)
```

```
$tab3_mbxRSG_lbl.DataBindings.DefaultDataSourceUpdateMode = 0
```

```
$System_Drawing_Point = New-Object System.Drawing.Point
```

```
$System_Drawing_Point.X = 12
```

```
$System_Drawing_Point.Y = 15
```

```
$tab3_mbxRSG_lbl.Location = $System_Drawing_Point
```

```
$tab3_mbxRSG_lbl.Name = "tab3_mbxRSG_lbl"
```

```
$System_Drawing_Size = New-Object System.Drawing.Size
```

```
$System_Drawing_Size.Height = 19
```

```
$System_Drawing_Size.Width = 124
```

```
$tab3_mbxRSG_lbl.Size = $System_Drawing_Size
```

```
$tab3_mbxRSG_lbl.TabIndex = 0
```

```
$tab3_mbxRSG_lbl.Text = "Mailbox From RSG"
```

```
$tab3.Controls.Add($tab3_mbxRSG_lbl)
```

```
$openFileDialog1.FileName = "openFileDB"
```

```
$openFileDialog1.ShowHelp = $True
```

```
$openFileDialog1.Title = "dbPathOD"
```

```
$openFileDialog1.add_FileOk($handler_openFileDialog1_FileOk)
```

```
$openFileDialog2.FileName = "openFileLOG"
```

```
$openFileDialog2.ShowHelp = $True
```

```
$openFileDialog2.Title = "logPathOD"
```

```
$openFileDialog2.add_FileOk($handler_openFileDialog2_FileOk)
```

```
$InitialFormWindowState = $form1.WindowState
```

```
$form1.add_Load($OnLoadForm_StateCorrection)
```

```
$form1.ShowDialog()| Out-Null
```

```
} GenerateForm
```

1. 6. Наръчник за администратора

6.1 Инструкции за използване на програмата

Наименование на програмата е Mailbox Restore Utility. Тя може да бъде стартирана от Exchange Power Shell , както може да се направи и shortcut на скрипта който да се стартира от desktop-а. Разширението на powershell скриптовите файловете е .ps1

При създаване на shortcut-а е нужно да се въведе в настройките му следващия ред:

```
Powershell -command "& {C:Scriptsmailboxrestoreutility.ps1 }"
```

Също така, скрипта може да се стартира от CMD/BAT файл и да се наглася за изпълнение по определено време.

Снимка 4. Tab User info

При стартирането на скрипта се зарежда графичния интерфейс на програмата. Първоначално забелязваме че се състои от 3 таба. (*снимка 3.*)

В първия таб „User Info” , имае поле за въвеждане на display name, който може да видим от АД.

При натискането на бутона Check, изпълнява се модула от скрипта при който се проверява дали е въведено нещо във полето. При невъвеждане на някаква string стойност, може да видим съобщение за грешка под бутона Check , където се намира скрит инфо-борд. При въвеждане на неправилен display name ще получим съобщение, че такъв не съществува, което съответно означава че или display name е написан грешно или съответния потребител няма създаден mailbox.

При въвеждане на правлен display name, автоматично започва зареждането на останалите полета – mailbox server, storage group, database. Функцията взема тези детайли по идентификация на display name от Exchange атрибутите. Следователно получаваме съобщение че съществува такъв потребител и информацията каде е пътя на пощенската му кутия. След това може да продължим на следващия таб който може да видим на *снимка 4*.

Снимка 4. Таб RSG

В таба RSG имаме опции за настройване и избиране свързани с Recovery Storage Group. Понеже политката на Exchange Server е, че може да създадем само едно RSG за един mailbox server, имаме бутона Check for Existing RSG. По този начин може да проверим дали вече има създадено RSG на избрания сървър. В падащото меню се зареждат всички сървъри с Mailbox роля в съответната организация. Ако няма създадено RSG, имаме възможност да създадем такова чрез бутона Create RSG. При създаването, трябва да посочим папката в която ще се намира базата и транзакцион логовете, което може да направим с бутоните Select Location за DB path и Logs Path. След като сме направили restore на базата в която се намира потребителската пощенска кутия, трябва да „закачим“ базата за RSG. Тази възможност ни дава бутона Mount DB.

В случай че има вече съществуващо RSG , може да го изтрием чрез бутона Delete Existing RSG.

Всички грешки, които ще получим, както и информацията, кой бутон каква операция изпълнява, може да видим в TextBox-а.

Грешките се появяват съвместимо Exchange Management-а, така че може да ги проверяваме в сайта на Microsoft за да разберем, какъв е проблема. След успешно направени горе посочените стъпки, може да преминем на таба Restore

В таба RSG имаме възможност да изберем пощенската кутия, която се намира в RSG, както и възможността да изберем в кой потребителска пощенска кутия ще „налеем“ възстановените имейли. Отментката Restore to a Folder, се използва за да напишем името на фолдера в пощенската кутия, в който ще възстановим имейлите. Във отсрещното текстово поле може да зададем име на папката.

С натискането на бутона Restore The Mailbox започва процеса възстановяване на изтритите от потребителя имейли. Както и в предишния таб, така и в този може да видим всички грешки, които ще получим, както и информацията, кой бутон каква операция изпълнява и всички грешки, които се появяват съвместимо Exchange Management-а, така че може да ги проверяваме в сайта на Microsoft за да разберем какъв е проблема.

1. 7. Приложимост

Приложимостта на програмата е разнообразна – както може да се ползва за малки фирми, в които има към 1000 потребителя, които ползват Exchange услугите, така и във корпоративни фирми, които имат към няколко сайта с по 50 000 потребителя. Скрипта на програмат може да се адаптира и редактира с изсискванията на клиента. Може да се

използва във всеки Environment, който използва Exchange 2007. С едно доста елементарно редактиране (премахване на SG от меюто, както и функциите и cmdlet-ите свързани със SG), програмата става използвана и в Exchange 2010 организация, защото в Exchange 2010 не съществуват Storage Group-и.

Във Exchange 2007 съществува графичен интерфейс за възстановяване на имейли само ако се намират на един и същ сървър потребителската пощенска кутия и RSG-то, но не и ако са на различни сървъри. В Exchange 2010, изобщо не съществува тази графичен интерфейс за възстановяване на имейли.

1. 8. Изводи

Графичния интерфейс дава на администратора по-голяма ефективност и бързина на решаване на проблема, но по този начин ограничава възможностите.

Скриптовете дават много по гъвкави решения и възможности, но за сметка на това администраторите губят доста време за създаването му.

При обединение на PowerShell команди с графичен интерфейс, администратора може да прави много неща едновременно, гъвкаво, за кратко време и с много възможности.

Microsoft имплементира PowerShell-a , за да даде на администраторите пълно разнообразие от възможности, но не се съобразява, че все пак Windows-a работи с „прозорци“, и те са били създадени за удобно администриране.

Именно по тази причина администраторите вече може сами да си прават комбинации от скриптове с графични интерфейси , и да си ги ползват спрямо нуждите. Трябват им само PowerShell и .NET.

Microsoft Exchange е един от основните компоненти на решението за обединени комуникации на Microsoft. Той представлява гъвкава и надеждна платформа, която може да Ви помогне да:

- намалите своите разходи за съобщения с 50 до 80%
- увеличите продуктивността чрез достъп до бизнес комуникациите от всяка точка
- предпазите бизнеса си чрез механизми за защита и съвместимост с регулаторните изисквания, които могат да Ви бъдат от полза при управление на риска.

Речник:

Exchange – размяна, обмен

Mailbox – пощенска кутия

Role – роля

Edge – острие

Hub – център

User – потребител

Recovery Storage Group (RSG) – Група за складиране на възстановяване (на имейли)

Storage Group (SG) - Група за складиране

Database – база данни

Active Directory – Активна директория

Server – Съвър

Restore – Възстановяване

Utility – удобство, полза

Client Access – достъп на клиента

Литература

MCTS 70-236 Exchange 2007, Configuring (*Microsoft official certification training book*)

Windows PowerShell Step-by-step (*Microsoft, Ed Wilson*)

Exchange 2007 Help file

<http://technet.microsoft.com>

<http://msdn.microsoft.com>