Exchange Plugins

Bacula Systems Documentation

Contents

Contents

Microsoft Exchange is a messaging and collaborative platform that enables businesses to communicate, share information and collaborate securely and efficiently. It provides email, calendaring, task management, and contacts functionalities, as well as allows users to access their data from anywhere with Internet connection. Exchange is used by organizations of all sizes, from small businesses to large enterprises, and is a critical part of many companies' communication infrastructure. It is available both as an on-premises solution and as a cloud-based service through Microsoft 365 Software as a Service platform.

Bacula Enterprise Exchange EWS Plugin is a tool to provide backup and restore operations at item level of elements managed by a Microsoft Exchange platform, meaning getting, downloading, cataloging and restoring individual emails, attachments, calendar events, tasks or contacts. Note that the functionality described here is all about traditional on-premise Exchange instances, *not* about Microsoft 365 Software-as-a-Service email functionality. For information about backing up and restoring emails or any other service of Microsoft 365, refer to:

1 Exchange EWS Plugin

The following article aims at presenting the reader with information about the **Bacula Enterprise Exchange EWS Plugin** (Exchange Plugin based on EWS - Exchange Web Services). The document briefly describes the target technology of the plugin, defines the scope of its operations, and presents its main features.

Through subchapters, more in-depth information can be found about the following topics:

1.1 Scope

Bacula Enterprise Exchange Plugin currently supports the following platforms:

- Exchange Server 2019
- Exchange Server 2016
- Exchange Server 2013 with Service Pack 1 or later.

The underlying version of the Windows operative system can be any of the supported ones associated to each Exchange Server product version according to the official Microsoft documentation. As an example, at the time of writing this document, this information can be found here: https://learn.microsoft.com/en-us/exchange/plan-and-deploy/system-requirements?view=exchserver-2019 .

This plugin is available since **Bacula Enterprise 16.2**, and needs to be deployed in a Linux host.

See also:

- Go to *Exchange EWS Features*
- Go to *Exchange EWS Installation*
- Go to *Exchange EWS Configuration*

- Go to *Exchange EWS Operations*
- Go to *Exchange EWS Best Practices*
- Go to *Exchange EWS Limitations*
- Go to *Exchange EWS Troubleshooting*

Go back to the *Exchange EWS plugin main page*.

1.2 Features

The main feature of **Bacula Enterprise Exchange EWS Plugin** is to offer backup and restore of Exchange Server environments at item level, which is the major possible granularity for Exchange services. This includes: emails, attachments, calendar appointments, tasks, contacts and folder structures.

In addition to the main goal, this plugin permits the user to adjust the overall functions to fit his environment offering large flexibility to select the target information to protect, to filter it because of privacy reasons or to do it efficiently through different strategies involving the parallelization of the implied operations.

General Features

Below, there is a list of general features this plugin offers:

- Backup and restore Exchange Server items
- Microsoft EWS API based backups
- Multi-service backup in the same job (email, calendar, contact and/or task)
- Multi-service parallelization capabilities
- Multi-thread single service processes
- Automatic parallelization of fetching processes
- Generation of user-friendly report for restore operations
- Network resiliency mechanisms
- Mailbox discovery capabilities
- List/query and auto-generation capabilities if combined with ScanPlugin
- Restore objects to Exchange Server
  - To original mailbox
  - To any other mailbox
  - To a different Exchange Server (cross-server restore)
- Restore any object to filesystem
- Full, Incremental & Differential backups
  - Advanced delta function for improved performance
- Mail folder, messages, appointments, contacts, tasks and attachments granularity for backup and restore
- Email addresses and folders selection capabilities for backup
- Backup and restore MIME objects for migration purposes
- Emails indexed at item level into Bacula Catalog

- Advanced search capabilities for restore operations
- Generation of user-friendly restore report into the destination mailbox
- Privacy filters for emails:
  - Ability to exclude email message fields from the catalog
  - Exclude private or spam messages through powerful filtering capability by rules.

Protected Items

Below, there is a list of all the items that can be backed up and restored using this plugin:

- User mailboxes
- Shared mailboxes
- Mailbox folder structure
- In-place archiving
- Emails and associated attachments
- Calendar appointments and associated attachments
- Contacts and associated attachments
- User tasks and associated attachments.

See also:

- Go back to *Exchange EWS Scope*
- Go to *Exchange EWS Architecture*
- Go to *Exchange EWS Installation*
- Go to *Exchange EWS Configuration*
- Go to *Exchange EWS Operations*
- Go to *Exchange EWS Best Practices*
- Go to *Exchange EWS Limitations*
- Go to *Exchange EWS Troubleshooting*

Go back to the *Exchange EWS plugin main page*.

1.3 Architecture

Bacula Enterprise Exchange EWS Plugin is a Bacula File Daemon plugin built over the **Exchange EWS (Exchange Web Services) API** to perform all of its operations to retrieve from and feed to the target Exchange service. The plugin runs a Java Daemon which uses a custom extension of the EWS Managed API SDK originally built by Microsoft.

All the information is obtained using secure and encrypted HTTPS queries to Exchange Server from the File Daemon (and through the mentioned Java Daemon), where the plugin is installed. All the requests are performed over the following endpoint: https://exchange.hostname/EWS/EWS/Exchange.asmx

To get more information about EWS, visit: https://learn.microsoft.com/en-us/exchange/client-developer/exchange-web-services/start-using-web-services-in-exchange

The metadata of every backed up item is stored in Bacula using JSON format. If MIME option is enabled, the information is also stored with that format (RFC 2077 for emails). Any attachment associated to a given item is downloaded and stored as it is. The download process is done locally to the host, and then sent to the Bacula Storage Daemon.

Backup and restore processes use different parallelization techniques in order to maximize performance, and overcome latency times when doing each needed request to EWS. Parallelization of several backup jobs is also supported.

Below, there is a simplified vision of the architecture of this plugin within a generic **Bacula Enterprise** deployment:

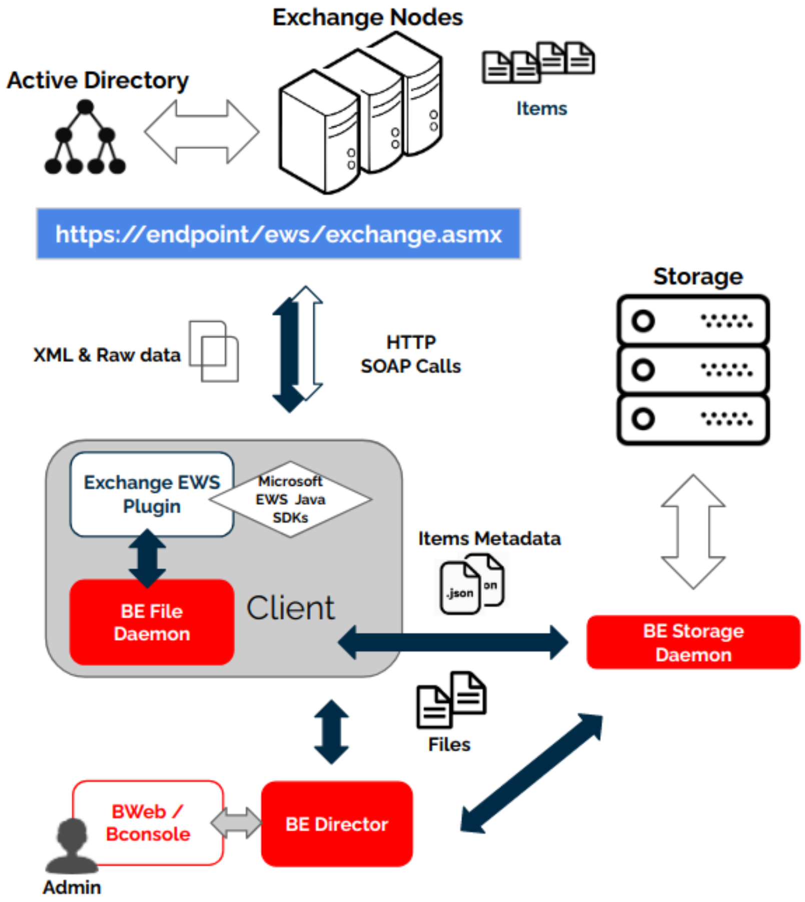


Fig. 1: Exchange EWS Plugin Architecture

See also:

- Go back to *Exchange EWS Scope*
- Go back to *Exchange EWS Features*
- Go to *EWS Installation*
- Go to *Exchange EWS Configuration*
- Go to *Exchange EWS Operations*
- Go to *Exchange EWS Best Practices*
- Go to *Exchange EWS Limitations*
- Go to *Exchange EWS Troubleshooting*

Go back to the *Exchange EWS plugin main page*.

1.4 Installation

This article describes how to install Bacula Enterprise Exchange EWS Plugin.

Prerequisites

- The Bacula File Daemon and the Exchange EWS Plugin need to be installed on the host that is going to connect to the Exchange Server.
- The plugin is implemented over a Java layer, and even if it backs up a Windows product, it must be deployed in a host running Linux. It is possible to use any of the supported **Linux** distributions of Bacula Enterprise, including Red Hat Linux, Debian, Ubuntu or Suse Linux Enterprise Server as some examples.
- The plugin works through a Java daemon, therefore Java needs to be installed into the host through a JRE or JDK package (openjdk-8-jre for example). Installed Java environment needs to be in version 8 or above and the Java binary must be available in the system PATH.
- Memory and computation requirements completely depend on the plugin configuration and usege (parallelization, size of data to backup, etc.). However, it is expected to have a minimum of **4GB RAM** in the server where the File Daemon is running. By default, every job could end using up to 512Mb of RAM in demanding scenarios (usually it will be much less). In some situations this could be higher. Memory limits can be adjusted (see Out of memory).
- Exchange EWS Service is used to perform all plugin operations. Therefore, it must be up and accessible through HTTPS from the host where Bacula FD and the plugin are going to be deployed. To do this, Outlook service needs to be installed on the host where the Bacula Enterprise Exchange Plugin is going to connect to fetch and protect the data.
- In order to fetch the data, the connection to EWS is done using Basic Authentication with username and password. An administrator user (a user belonging to the 'Organization Management' group) properly configured to access the mailboxes of any target user is needed. Details about how to configure such user are given in the next sections.
- EWS endpoint is usually served through HTTPS protocol. The certificate needs to be valid and the included CN (example: myhost.com) needs to match the endpoint configured in the plugin parameters.

Installation Methods

- EWSInstallationWithBIM (recommended)
- EWSInstallationPackageManagers
- EWSInstallationManual

Result

The package installs the following elements:

- Jar libraries in /opt/bacula/lib (such as bacula-exchange-ews-plugin-x.x.x.jar and bacula-exchange-ews-plugin-libs-x.x.x.jar). Note that the version of those jar archives is not aligned with the version of the package. However, that version will be shown in the joblog in a message like 'Jar version:X.X.X'.
- Plugin connection file (e2ws-fd.so) in the plugins directory (usually /opt/bacula/plugins). Note that e2ws acronym means Exchange EWS.

- Backend file (e2ws_backend) that invokes the jar files in /opt/bacula/bin. This backend file searches for the most recent bacula-exchange-ews-plugin-x.x.x.jar file in order to launch it, even thought usually we should have only one file.

Once the plugin is installed, it should be possible to see it loaded through a status client command in bconsole ('Plugin:' line must contain 'e2ws'):

Listing 1: **Status client**

```
*st client
Automatically selected Client: 127.0.0.1-fd
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102

127.0.0.1-fd Version: 16.0.5 (05 April 2023)  x86_64-pc-linux-gnu ubuntu 22.04
Daemon started 14-abr-23 10:14. Jobs: run=2 running=0 max=100.
Ulimits: nofile=1024 memlock=2026356736 status=ok
Heap: heap=827,392 smbytes=436,939 max_bytes=5,100,087 bufs=153 max_bufs=248
Sizes: boffset_t=8 size_t=8 debug=600 trace=1 mode=1,2010 bwlimit=0kB/s
Crypto: fips=no crypto=OpenSSL 3.0.2 15 Mar 2022
APIs: !GPFS
Plugin: bpipe(2) e2ws(1.0.0)
```

See also:

- Go back to *Exchange EWS Scope*
- Go back to *Exchange EWS Features*
- Go back to *Exchange EWS Architecture*
- Go to *Exchange EWS Configuration*
- Go to *Exchange EWS Operations*
- Go to *Exchange EWS Best Practices*
- Go to *Exchange EWS Limitations*

Go back to the *Exchange EWS plugin main page*.

1.5 Configuration

The following chapter present the information on how to configure admin user, and fileset.

Admin User Configuration

Bacula Enterprise Exchange EWS Plugin needs an administrator user to access to the server and to retrieve the information to back up.

This admin user needs to be able to: - Impersonate other users with Full rights (and therefore access their mailboxes) - Have mailbox discovery abilities.

To configure impersonation, it is necessary to run the following command in Powershell:

Listing 2: **Impersonation**

```
New-ManagementRoleAssignment -name:impersonationAssignmentName -
↪Role:ApplicationImpersonation -User:<AdminUserName>
```

For more information about impersonation, visit: https://learn.microsoft.com/en-us/exchange/client-developer/exchange-web-services/how-to-configure-impersonation

In order to provide full permissions to the admin user for the impersonated mailboxes, this command needs to be run in Powershell for every user:

Listing 3: **Mailbox permissions**

```
Add-MailboxPermission -Identity <user-to-impersonate@mydomain.com> -User <admin-
↪user@mydomain.com>   -AccessRights fullaccess
```

Mailbox discover capabilities are enabled with the following command in Powershell:

Listing 4: **Discovery Management**

```
Add-RoleGroupMember -Identity "Discovery Management" -Member <AdminUserName>
```

Note: Be aware that the effect of all these commands can take time. Especially the Mailbox discovery capabilities may need more than 30 minutes to be activated.

Important: In addition to running this command, the Admin user needs to have his mailbox activated. Otherwise, discovery capabilities won't work.

See also:

- Go to *Fileset Configuration*

Go back to the *main configuration page*.

Fileset Configuration

Once the plugin is successfully authorized, it is possible to define regular filesets for backup jobs in Bacula, where we need to include a line similar to the one below, in order to invoke the Exchange EWS Plugin:

Listing 5: **Fileset E2WS**

```
FileSet {
   Name = FS_E2WS
   Include {
      Options {
        signature = MD5
        ...
      }
      Plugin = "e2ws: <e2ws-parameter-1>=<e2ws-value-1> <e2ws-parameter-2>=<e2ws-value-2>
↪ ..."
   }
}
```

It is **strongly recommended** to use only one ‘Plugin’ line in every fileset. The plugin offers the needed flexibility to combine different modules backup inside the same plugin line. Different exchange servers, in case of existing, should be using different filesets and different jobs.

In this plugin, any parameter allowing a list of values can be assigned with a list of values separated by ‘,’.

Below, in the subsections, there are lists that present all the parameters you can use to control Exchange EWS Plugin behavior.

Fileset Connection Parameters

The following parameters control the connection of the Exchange EWS Plugin to the Exchange Server.

Option	Required	Default	Values	Example	Description
endpoint	**Yes**		A hostname or IP address	win19-cl1-exch	Hostname or IP address that matches the DN of the SSL Certificate of the Exchange service
admin_domain	No		A domain name	MYEXCHANGEDOMAIN	The users domain name. If admin_user is including already the domain, this parameter must not be set
admin_user	**Yes**		Email address or username (with or without the domain prefix)	myadmin@mydomain.com	An email address, or the username of the admin user that has permissions to impersonate all the other users. The format can be an email address, a single username (then admin_domain needs to be filled in) or domainusername. For simplicity, it is recommended to use the email address
admin_password	**Yes**		A password string	G3934kdk!/0348	The password associated to the admin user

Note: The admin user must have his mailbox enabled and working. Otherwise, discovery operations will fail.

See also:

- Go to *Fileset Backup Parameters*
- Go to *Fileset Common Parameters*
- Go to *Fileset Tuning Parameters*
- Go to *Fileset Advanced Parameters*
- Go to *Fileset Examples*

Go back to the *Fileset Configuration* article.

Fileset Backup Parameters

The following list of parameters control what is going to be included into the associated backup:

Option	Required	Default	Values	Example	Description
service	No		email, contact, calendar, task, (list parameter: it can contain 0, 1 or more elements separated by ‘,’)	email	Establish the service or services that will be backed up. If this is not set, the plugin will try to backup all supported services.
user	No		Valid email addresses of existing users on the selected exchange service separated by ‘,’	AlexW@yourdomain.com, LeeY@yourdomain.com	Backup mailboxes associated to this list of users. If no user is provided the plugin will use discovery mechanism and include any user with an active mailbox
user_exclude	No		Valid email addresses of existing users on the selected exchange service separated by ‘,’	LauraG@yourdomain.com, AmandaT@yourdomain.com	Exclude selected mailboxes. If this is the only parameter found for selection, all elements will be included and this list will be excluded
user_regex_include	No		Valid regex	.*@management\.mydomain.com	Backup matching user mailboxes. Please, only provide list parameters (user + user_exclude) or regex ones. But do not try to combine them
user_regex_exclude	No		Valid regex	.*@guests\.mydomain.com	Exclude matching user mailboxes. Please, only provide list parameters (user + user_exclude) or regex ones. But do not try to combine them. If this is the only parameter found for user selection, all users will be included and matching users will be excluded
folder	No		Folder names separated by ‘,’	inbox,company	Backup only the list of folders of this parameter from the mailboxes of the selected users. If no folder is provided, all folders will be included
folder_exclude	No		Folder names separated by ‘,’	travel,personal	Exclude selected folders from the mailboxes of the selected users. If this is the only parameter found for folder selection, all folders will be included and this list will be excluded
folder_regex_include	No		Valid regex	.*mycompany	Backup matching folders by name. Please, only provide list parameters or regex ones. But do not try to combine them
folder_regex_exclude	No		Valid regex	.*private	Exclude matching user folders by name. Please, only provide list parameters or regex ones. But do not try to combine them. If this is the only parameter found for folder selection, all folders will be included and matching folders will be excluded
exclude_attachments	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Exclude any attachment from backup
mime	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Backup raw MIME file of items, in addition to the item objects themselves (regular objects are backed up as json formatted objects)
archive	No	No	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	Yes	Include In-place archiving tree folders and items of the selected mailboxes
email_exclude_expr	No		String representing a valid Boolean Javascript expression regarding email message fields	emailSubject.includes(‘private’) && !emailIsRead	Exclude from backup all messages that match the provided expression
email_exclude_index_expr	No		String representing	/.*pri-	Exclude only from indexing (catalog email tables)

See also:

- Go back to *Fileset Connection Parameters*
- Go to *Fileset Common Parameters*
- Go to *Fileset Tuning Parameters*
- Go to *Fileset Advanced Parameters*
- Go to *Fileset Examples*

Go back to the *Fileset Configuration* article.

Fileset Common Parameters

These parameters are controlling generic aspects of the behavior of the Exchange EWS Plugin, it is possible to find also these parameters in other Bacula Enterprise Plugins with similar effects, so you may be familiar with them.

Option	Required	Default	Values	Example	Description
abort_on_error	No	No	No, Yes	Yes	If set to **Yes**: Abort job as soon as any error is found with any element. If set to **No**: Jobs can continue even if it they found a problem with some elements. They will try to backup or restore the other and only show a warning
config_file	No		The path pointing to a file containing any combination of plugin parameters	/opt/bacula/etc/e2ws.settings	Allows to define a config file where configure any parameter of the plugin. Therefore you don't need to put them directly in the Plugin line of the fileset
log	No	/opt/bacula/working/e2ws/e2ws-debug.log	An existing path with enough permissions for File Daemon to create a file with the provided name	/tmp/e2ws.log	Generates additional log in addition to what is shown in job log. This parameter is included in the backend file, so, in general, by default the log is going to be stored in the working directory.
debug	No	0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	Debug level. Greater values generate more debug information	Generates the working/e2ws/e2ws-debug.log* files containing debut information which is more complete with a greater debug number
path	No	/opt/bacula/working	An existing path with enough permissions for File Daemon to create any internal (and usually temporary) plugin file	/mnt/my-vol/	Uses this path to store metadata and temporary files

See also:

- Go back to *Fileset Connection Parameters*
- Go back to *Fileset Backup Parameters*
- Go to *Fileset Tuning Parameters*
- Go to *Fileset Advanced Parameters*
- Go to *Fileset Examples*

Go back to the *Fileset Configuration* article.

Fileset Tuning Parameters

These set of parameters can be used to fine-tune the behavior of the plugin to be more flexible in cases of bad network environments, or when significant job concurrency is happening, etc. It is not necessary to modify them for the great majority of the cases:

Option	Required	Default	Values	Example	Description
backup_queue_size	No	10	0-500	1	Number of maximum enqueued internal operations between service static internal threads (there are 3 communicating through queues with the set size: service fetcher, service opener and general publisher to bacula core). This could potentially affect api concurrent requests and consequently, throttling and cpu/memory consumption for both, the FileDaemon and the Exchange server. It is only needed to modify this parameter, in general, if you are need a ver precise control of your concurrency levels
concurrent_threads	No	10	0-100	1	Number of maximum concurrent backup threads running in parallel in order to open data for running download actions. If you want to have a precise control of your parallelization through different jobs, please set up this value to 1. Please be careful also with the memory requirements, multi-threaded increases very significantly memory consumption per job
concurrent_listing_threads	No	2	0-20	1	Number of maximum concurrent backup page listing threads running in parallel in order to fetch sets of data. This parameter will also affect api concurrent requests
api_list_page_size	No	100	10-500	1	Number of items got in each page for multi-page requests to EWS API
general_network_retries	No	5	Positive integer (number of retries)	10	Number of retries for failed requests to the EWS API
general_network_delay	No	50	Positive integer (seconds)	100	Delay between retries to the EWS API

See also:

- Go back to *Fileset Connection Parameters*
- Go back to *Fileset Backup Parameters*
- Go back to *Fileset Common Parameters*
- Go to *Fileset Advanced Parameters*
- Go to *Fileset Examples*

Go back to the *Fileset Configuration* article.

Fileset Advanced Parameters

The following parameters are advanced ones, and they should not be modified in the great majority of cases:

Option	Required	Default	Values	Example	Description
stream_sleep	No	1	Positive integer (1/10 secconds)	5	Time to sleep when reading header packets from FD and not having a full header available
stream_max_wait	No	120	Positive integer (seconds)	360	Max wait time for FD to answer packet requests
time_max_last_modify_log	No	86400	Positive integer (seconds)	43200	Maximum time to wait to ovewrite a debug log that was marked as being used by other process
logging_max_file_size	No	50MB	String size	300MB	Maximum size of a single debug log fileGenerates the working/e2ws/e2ws-debug.log* files containing debut information which is more complete with a greater debug number
logging_max_backup_index	No	25	Positive integer (number of files)	50	Maximum number of log files to keep
log_rolling_file_pattern	No	“e2ws.log.%d{dd-MMM}.log.gz”	No, Yes	Yes	Log patter for rotated log files
split_config_file	No	=	Character	:	Character to be used in config_file parameter as separator for keys and values
opener_queue_timeout_secs	No	1200	Positive integer (seconds)	3600	Timeout when internal object opener queue is full
publisher_queue_timeout_secs	No	1200	Positive integer (seconds)	3600	Timeout when internal object publisher queue is full

The internal plugin logging framework presents some relevant features that we are going to describe:

- The “.log” files are rotated automatically. Currently, each file can be 50Mb at maximum and the plugin will keep 25 files.
    - This behavior can be changed using the internal advanced parameters: logging_max_file_size and logging_max_backup_index
- The “.err” file can show contents even if no real error happened in the jobs. It can show contents too even if debug is disabled. This file is not rotated, but it is expected to be a small file in general. If you still need to rotate it, you can include it in a general rotating tool like ‘logrotate’.
- Backups in parallel and also failed backups will generate several log files. For example: e2ws-debug-0.log, e2ws-debug-1.log…

See also:

- Go back to *Fileset Connection Parameters*
- Go back to *Fileset Backup Parameters*
- Go back to *Fileset Common Parameters*
- Go back to *Fileset Tuning Parameters*
- Go to *Fileset Examples*

Go back to the *Fileset Configuration* article.

Fileset Examples

In this section, some fileset examples are presented:

Listing 6: **Fileset: for all data belonging to a user**

```
FileSet {
   Name = fs-e2ws-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: endpoint=myexchange.myorg.com admin_user=ex-admin@myorg.com admin_
↪password=xxxxxxx user=adelev@myorg.com"
   }
}
```

Listing 7: **Fileset: using a config file**

```
FileSet {
   Name = fs-e2ws-adelev
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings user=adelev@myorg.com"
   }
}

Config file contents in stored in the same File Daemon host in /opt/bacula/etc/e2ws.
↪settings:
endpoint=myexchange.myorg.com
admin_user=ex-admin@myorg.com
admin_password=xxxxxxx
```

Listing 8: **Fileset: Backup only emails**

```
FileSet {
   Name = fs-e2ws-adelev-email
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email␣
↪user=adelev@myorg.com"
   }
}
```

Listing 9: **Fileset: Backup emails and appointments of all users**

```
FileSet {
   Name = fs-e2ws-email-calendar
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email,calendar"
   }
}
```

Listing 10: **Fileset: Backup only email folders: inbox and important custom folder**

```
FileSet {
   Name = fs-e2ws-all-inbox-important
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email␣
↪folder=inbox,important"
   }
}
```

Listing 11: **Fileset: Backup emails in mime format for two users**

```
FileSet {
   Name = fs-e2ws-mime-u1-u2
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email mime=true␣
↪user=user1@myorg.com,user2@myorg.com"
   }
}
```

Listing 12: **Fileset: Backup emails and contact in mime format for two users, but exclude attachments**

```
FileSet {
   Name = fs-e2ws-user1-user2-no-attach
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings service=email,contact␣
↪mime=true exclude_attachments=true user=user1@myorg.com,user2@myorg.com"
   }
}
```

Listing 13: **Fileset: Backup all services from all users starting with 'org'**

```
FileSet {
   Name = fs-e2ws-org-users
   Include {
      Options { signature = MD5 }
```

(continues on next page)

(continued from previous page)

```
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings user_regex_include=\"org.
↪*\""
   }
}
```

Listing 14: **Fileset: Backup one user reducing the concurrency configuration**

```
FileSet {
   Name = fs-e2ws-user1-min
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings user=user1@myorg.com␣
↪concurrent_threads=1 concurrent_listing_threads=1"
   }
}
```

Listing 15: **Fileset: Backup one user maximizing the concurrency configuration**

```
# Warning: This configuration could provoke throttling issues
FileSet {
   Name = fs-e2ws-user1-max
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings user=user1@myorg.com␣
↪concurrent_threads=50 concurrent_listing_threads=10 backup_queue_size=500 api_list_
↪page_size=500"
   }
}
```

Listing 16: **Fileset: Backup all services, all users, but exclude emails where the subject contains 'private'**

```
FileSet {
   Name = fs-e2ws-exclude-private
   Include {
      Options { signature = MD5 }
      Plugin = "e2ws: config_file=/opt/bacula/etc/e2ws.settings email_exclude_index_
↪expr=\"emailSubject.includes('private')\""
   }
}
```

See also:

- Go back to *Fileset Connection Parameters*
- Go back to *Fileset Backup Parameters*
- Go back to *Fileset Common Parameters*
- Go back to *Fileset Tuning Parameters*
- Go back to *Fileset Advanced Parameters*

Go back to the *Fileset Configuration* article.

See also:

- Go back to *Admin User Configuration*.

Go back to the *main configuration page*.

See also:

- Go back to *Exchange EWS Scope*
- Go back to *Exchange EWS Features*
- Go back to *Exchange EWS Architecture*
- Go back to *Exchange EWS Installation*
- Go to *Exchange EWS Operations*
- Go to *Exchange EWS Best Practices*
- Go to *Exchange EWS Limitations*
- Go to *Exchange EWS Troubleshooting*

Go back to the *Exchange EWS plugin main page*.

1.6 Operations

The following article describes details regarding backup, restore or list operations with **Bacula Enterprise Exchange EWS Plugin**.

Backup in Exchange EWS Plugin

Backup jobs in Exchange EWS plugin behave as any other backup job in Bacula Enterprise once the fileset has been created, as described in the configuration section. Below, some special features of the plugin that happen at backup time are described, as well as the file structure that a backup creates.

Backup File Structure

Items are formatted in the backup catalog in order to not include sensitive information. They are included in a path in the following format:

```
/@e2ws/domain.name/users/user@domain.name/foldername/shortId_itemDate.itemExtension
```

Depending on the type of the item, `itemExtension` wil be:

- Message: .msg
- Appointment: .pp
- Task: .task
- Contact: .con
- Contact group: .con.gr

Mime files will have the extra word ‘mime’ in their extension. For example:

```
/@e2ws/testlab.local/users/ex-admin@testlab.local/regress_20230417125041/
AAPfLdTrAAA=_r20230417-125325.mime.msg
```

Attachments will be stored together with item objects:

- They include their original name (file name)
- They have a special extension ".att"
- They include the attachment type (file or item)
- The first part of the attachment name is the name of the parent message.

Here is an example of an attachment :

```
/@e2ws/testlab.local/users/ex-admin@testlab.local/inbox/AAPfLdTuAAA=_r20230417-125329.
msg.Prompta.gen.file.att
/@e2ws/testlab.local/users/ex-admin@testlab.local/inbox/AAPfLdTuAAA=_r20230417-125329.msg
```

→ *Parent message*

MIME Objects Backup

Based on the fileset parameter **mime**, it is possible to get mime formatted items as well as regular objects (which are in json format). This kind of objects can be useful to get if there is any plan of using the information outside the Exchange service (e.g. for migration purposes).

Note: Activating this option has a performance penalty, and the backup time will be significantly higher, as for every email the information will be requested twice (one for the regular format, so .json message file plus attachments; another one for mime format, so to get a unified .mime.msg file containing the message and attachments).

At restore time, if the restore operation is done via EWS services and not to any local filesystem, selected mime objects are automatically ignored. It means if only those .mime.msg files were manually selected during a restore session over Exchange, the restore won't restore any file. While doing the same over a Local filesystem, destination will end up with those .mime.msg files restored. Usually, the selection would include both kind of files (a folder, a whole backup). In that situation, the restore will be simply successful, while those .mime.msg won't be used.

Email Privacy Filters

Bacula Systems is aware of one of many privacy concerns that may arise when tools like the Exchange EWS Plugin enables the possibility to backup and restore data coming from different users, so the backup administrator can restore potentially private data at his will. Moreover, emails are usually one of the most critical items in terms of privacy.

One of many strategies the plugin offers in order to deal with that problem is the possibility to exclude messages. This is a very powerful feature where it is possible to use quite flexible expressions that allow to select a subset of messages and simply exclude them from the backup:

- `email_exclude_expr` fileset parameter will exclude completely the selected messages
- `email_exclude_index_expr` fileset parameter will exclude the selected messages from the index (MetaEmail catalog table).

Not only messages can be excluded, but also select only a subset of email fields to be included in the indexed information using `email_fields_exclude_index` fileset parameter.

All three discussed expressions are based on an internal structure of fields to work with. Below, you can see the entire list of fields that you can use:

- emailTags

- emailSubject
- emailFolderName
- emailFrom
- emailTo
- emailCc
- emailBodyPreview
- emailImportance
- emailTime
- emailIsRead
- emailIsDraft

Note: It is very important to write the fields exactly as written above.

These fields can be used in a comma separated list in the `email_fields_exclude_index` parameter.

Then, for `email_exclude_index_expr` and `email_exclude_expr`, use them in a valid boolean expression in **Javascript** language syntax. Some examples are provided below:

Listing 17: **Expression to exclude messages where subject includes the word 'private'**

```
emailSubject.includes('private')
```

Listing 18: **Complex expression to exclude messages that are not read and are Draft or their folder name is named Private**

```
!emailIsRead && (emailIsDraft || emailFolderName == 'Private')
```

Listing 19: **Expression to exclude messages based on the received or sent date**

```
!emailTime < Date.parse('2012-11-01')
```

Listing 20: **Expression to exclude messages using a regex based on emailFrom**

```
/.*private.com/.test(emailFrom)
```

Note: This feature is available since Bacula Enterprise version 14.0

Expression Tester

This expression mechanism can sometimes be uncertain for end users as they can have doubts about the correct behavior of their prepared expressions. In order to help with that, Exchange EWS Plugin presents a query method that allows to test those expressions against a static preloaded set of data.

There are two commands available:

- Show command
- Test command

The show command will show the static data in json format, so it is possible to see the contents to adapt the expressions to test command - it will apply the expression parameters to the preloaded static data.

The test command has the following format:

Listing 21: **Expression tester Show command**

```
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_user=
↪<username> admin_password=<password>" parameter=email-expr-show
```

The show command has the following format:

Listing 22: **Expression tester Test command**

```
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_user=
↪<username> admin_password=<password> email_exclude_expr = \"<your-js-expression>\""␣
↪parameter=json|email-expr-test
// Or
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_user=
↪<username> admin_password=<password> email_exclude_index_expr = \"<your-js-expression>\
↪"" parameter=json|email-expr-test
```

Note: You need to provide a valid endpoint and user credentials, even if it's not really used to process any data.

The test command produces JSON output with objects with the exact format that is received from Microsoft and, consequently, the same format that is stored in backup. Note that 'total' value at the end, where the value of 12 total preloaded messages is shown.

Listing 23: **Expression tester Show command output**

```
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_user=
↪<username> admin_password=<password>" parameter=json|email-expr-show
....
```

(continues on next page)

(continued from previous page)

```
    "email-12": {
      "body": {
        "content": "These are the contents in text format of the 12 email of test data.␣
↪It has the following categories:orange, black, white, purple. You can try to filter␣
↪this body using any JS method like /.*12.*/.test(emailBody) or emailBody.includes(12)",
        "contentType": "TEXT"
      },
      "ccRecipients": [
        {
          "emailAddress": {
            "address": "danny@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "lucas@other.com"
          }
        },
        {
          "emailAddress": {
            "address": "terese@other.com"
          }
        }
      ],
      "from": {
        "emailAddress": {
          "address": "elon@other.com"
        }
      },
      "hasAttachments": false,
      "isDraft": false,
      "isRead": false,
      "replyTo": [
        {
          "emailAddress": {
            "address": "elon@other.com"
          }
        }
      ],
      "sentDateTime": {
        "dateTime": {
          "date": {
            "year": 2021,
            "month": 12,
            "day": 5
          },
          "time": {
            "hour": 11,
            "minute": 30,
            "second": 0,
            "nano": 0
          }
```

(continues on next page)

(continued from previous page)

```
    },
    "offset": {
     "totalSeconds": 0
    }
   },
   "subject": "This is private subject 12",
   "toRecipients": [
    {
     "emailAddress": {
      "address": "laura@other.com"
     }
    },
    {
     "emailAddress": {
      "address": "jack@other.com"
     }
    },
    {
     "emailAddress": {
      "address": "john@other.com"
     }
    }
   ],
   "categories": [
    "orange",
    "black",
    "white",
    "purple"
   ]
  }
 },
 {
  "total": "12"
 }
```

The test command on its side will produce two different outputs. The first part presents the same format as the show format, and those are the messages that would be included in the backup. The second part presents a different format, so an output like:

Listing 24: **Expression tester Test command, index part output**

```
.query client=<your-fd-client> plugin="e2ws: endpoint=<ews-endpoint> admin_user=
↪<username> admin_password=<password>" parameter=json|email-expr-show
....
   {
  "meta-email-12": {
   "EmailId": "",
   "EmailOwner": "test@test.com",
   "EmailTenant": "ews.test",
   "EmailTags": "orange,black,white,purple",
   "EmailSubject": "This is private subject 12",
   "EmailFolderName": "/",
```

(continues on next page)

(continued from previous page)

```
      "EmailFrom": "elon@other.com",
      "EmailTo": "laura@other.com,jack@other.com,john@other.com",
      "EmailCc": "danny@other.com,lucas@other.com,terese@other.com",
      "EmailInternetMessageId": "",
      "EmailBodyPreview": "",
      "EmailImportance": "",
      "EmailConversationId": "",
      "EmailSize": 235,
      "EmailIsRead": 0,
      "EmailIsDraft": 0,
      "EmailHasAttachment": 0,
      "Type": "EMAIL",
      "Version": 1,
      "Plugin": "e2ws"
    }
  },
  {
    "total-backup": "12"
  },
  {
    "total-index": "12"
  }
```

That part represents the information that would be indexed in the backup (included into the Catalog). You can also see the total entries at the end, which are very useful to quickly compare with the original 12 value and so, knowing if our expression is filtering the expected data or not. Below, we provide an example where some filtering to the backup is applied, but also it is applied to the index:

Listing 25: **Expression tester Test command, index part output**

```
.query client=127.0.0.1-fd plugin="e2ws: endpoint=<ews-endpoint> admin_user=<username>␣
↪admin_password=<password> email_exclude_expr=\"emailFrom == 'elon@other.com'\" email_
↪exclude_index_expr=\"emailSubject.includes('private')\"" parameter=json|email-expr-test
...
     {
    "meta-email-4": {
      "EmailId": "",
      "EmailOwner": "test@test.com",
      "EmailTenant": "ews.test",
      "EmailTags": "orange,black,white,purple",
      "EmailSubject": "This is orange subject 8",
      "EmailFolderName": "/",
      "EmailFrom": "bob@company.com",
      "EmailTo": "laura@company.com,jack@company.com,john@company.com",
      "EmailCc": "danny@company.com,lucas@company.com,terese@company.com",
      "EmailInternetMessageId": "",
      "EmailBodyPreview": "",
      "EmailImportance": "",
      "EmailConversationId": "",
      "EmailSize": 232,
      "EmailIsRead": 0,
      "EmailIsDraft": 0,
```

(continues on next page)

(continued from previous page)

```
      "EmailHasAttachment": 0,
      "Type": "EMAIL",
      "Version": 1,
      "Plugin": "e2ws"
    }
  },
  {
    "total-backup": "6"
  },
  {
    "total-index": "4"
  }
```

In case your expression is not valid, the plugin will also inform about that with the following message:

```
``error=Error listing elements. Cause:  Predicate test error!! Review your query .....
```

Delta Backup

The Microsoft EWS API provides a Delta function to track changes of some objects. Bacula Enterprise Exchange EWS Plugin uses this function in order to speed up Incremental/Differential processes.

Delta function has the following important characteristics:

- Delta tokens can expire at some point, or even become invalid due to internal Microsoft issues. If that happens, the plugin will try to start a new Delta cycle.
- There are two delta types of tokens implied: one for the folder structure, another for every folder that has changes inside.
- Any situation where the Delta function cannot be used will trigger a regular Full/Inc/Diff, where every element is listed and selected or discarded according to the item dates.

The Delta backup cycle is described below:

- Full backup: All entity elements are backed up. A token (token_1) is generated and the token is stored locally by the FD.
- Incremental 1 backup: token_1 is used to retrieve changes since token_1's generation, so every change is backed up. A new token is generated and stored locally by the FD.
- Incremental 2 backup: token_2 is used to retrieve changes since token_2's generation, so every change is backed up. A new token is generated and stored locally by the FD.
- And so on…

Tokens are stored in a file placed in a path defined by the **path** parameter of the plugin. The name is: `jobname.deltaLink`.

The file stores tokens required for every execution, and it is renewed (emptied) during every Full backup execution.

This file is also backed up in the backup itself, so it can be restored manually, before an Incremental/Differential execution in case it was lost and in case you don't want to run a Full backup again.

Here, we can see an example of the contents of the file, with one execution and one user entity involved. The structure is tree-based, so it is easy to understand what would be generated in case of backing up other folders or users:

Listing 26: **deltaLink**

```
{
   "deltaServices" : {
      "e2ws" : {
         "entities" : {
            "ex-admin@testlab.local" : {
               "containers" : {

↪"AAMkADkwMWYyMWQwLWZjZmMtNDU3NS1iMmM3LWVmMTRkNTQ0MjVjYQAuAAAAAAD2ghxkrOnXTJN4mEgvv12nAQAtUQmYk+IgRb3K
↪AAPfLblCAAA=" : {
                     "deltaEntries" : [
                        {
                           "date" : "Apr 14, 2023, 10:14:42 AM",
                           "delta" :
↪"H4sIAAAAAAAAEAGNgYGcAAotqE0tHE2NTA0ddZ3NHC10TR2djXSdnJ2ddNyMnC2cnczdLU1OD2vBgveDKvOTgksSSVOfEvMSiSgYr
↪j8GaaK3+QMuKS4JSk1Mzy1JTQjJzU0nwrU9icYlnXnFJYl5yqncqKb71zS9K9SxJzS32zwtOLSpLLSLByXDfhgNxUW5iUTYklrgYG
↪sckYG3UDOGZMfKbjuPaV2Yc5p/
↪f0MzPd1dzoxMDAy8DEwg7RwM9jViBcesLntwSAEFOUFYqB1rIwMDL6OAZ6+jn4gRQxupm5hYOVooB2I5ZD4S9H4MHAYiM2Q+OjOYm
↪Xj0sKYLgXwr5brZ7Dd7L+v6q5Sj/2soE1YfomCCoO0paMRRbo2b1aYNIYr94knDYyMTA0AAAccVKZJwMAAA==",
                           "job" : "pluginTest.2023-04-14_10.14.42_03"
                        }
                     ],
                     "description" : "regress_20230414101303",
                     "id" :
↪"AAMkADkwMWYyMWQwLWZjZmMtNDU3NS1iMmM3LWVmMTRkNTQ0MjVjYQAuAAAAAAD2ghxkrOnXTJN4mEgvv12nAQAtUQmYk+IgRb3K
↪AAPfLblCAAA="
                  },
                  "MailboxFolders" : {
                     "deltaEntries" : [
                        {
                           "date" : "Apr 14, 2023, 10:14:42 AM",
                           "delta" :
↪"H4sIAAAAAAAAEAK2be1ST5x3HX6HeQJCgwqzIXaVqJEhCiEBpLsSkBpRytG6oECBcLAaXAkqFeQFXT6TUiU5AgWCptlQp6Oqsh4qX
↪h83t/veZ/
↪3uSMIowX7Fb1BqlBKI2USpVgtV0aLpUp1pFilVqnF2vmqaLVKrlXIZJKyV1PmpZSYM1MKjYUmtdFstJQIC/
↪hJbUF+lsmizxIU/
↪Owyk+X1vAKzIFHn55nMhU9+lqaWSzQRURKpWKJIkIul0YpssSoiWiXWqCIj5SqFMlIr0QpR6iJLislSbLIkGs152abXC/
↪8/7kn5uAmCjz2MHyLX5ZksRktmbokjKkFk/7XE/pnn+Lt/
↪bfHLOnSzz1C9vkYX3rny4ChBnDy+pvpqYMInF2Z277sY3un4q1Hj7B/B0/6xX/bvQqXjC5fFtU/
↪fM0UQXPgpH4jyhSg/
↪QXDlp6ZDlD9EBUBUIEQFQVQwRIVAVChEzYComRA1C6LCIOoFiJoNUXO43y9HO+IBUQboXZ4I3cuTuzQcFH+NclD8NUoIbFP6QM/
↪rR1CE/DXKEeEUiJoMRYi1AJMgij9CBzUVop6HqGlQGYY4enduyo2bcu0aMgVDFBbhRIji780dVCx0rwkQ5SUIY/
↪gpoOTtNWoVRKVDlBGisiEqByp5+9h5ND/
↪lDtX5IIgKhagZEDUTomZBVBhEvQBRsyFqDkTN5ad6XWzfQ9QwQjUJwPulHVoEUQaISoRaAC+IEkGUN0QBdcNOiSFqHkSFQ5QEoiIga
↪gKidEFUNUU0QtR+iDkMl38p/
↪ryvij1VQhBkQlQlRWRCVC1F5ELUaoswQVQBRayGqEKKKIKoYouohqgGiGqGVivEQ5Q2Nvjz5KfvMF5in2GejoyDKBSoN/
↪tGXg+IfETko/jGbg3oyZuPfjBWebun62L/9j0e6KFpUWj18nr8mTGFKfaR91wr4pT5M6TRRW/sW4vQN6UP/
↪HGjij9SPKTW+W77nDrU0Q9T1HWeZPjP9VVnLVxdkUEtNB1ed30YtzTY/fNTMX6ZBTGnOnbo/fUYtzT3R/
↪fa31OnnfSwXfsIfaShTurrs7cV66khfC72oXE8szbCpA3qXEEvXhLSM+VUctTT04dW014ilFkum3Pcjaun6B2GH91NKHdde/
↪0e33uOXerCkXjebZz88RCm1pz+r8drPjzbQlunLj2/8UXf/
↪U36pJyt991c81j43SC1dGXNkFNCbMqSBbbrTva31LZRS+4OKdb+aqwDeKHb61pzSTi11+rve9MrfTS2t6GxP+j11+tqBQB3nsOeZDU
↪STqWsl9uf28P23ycWDp5IOXS1+uo36iQ0bb3DNRl2hDTOW4mdeX3LdnWkkabvuHH1/
↪9Q7tiAJJUaH4tNtcP8DUo6S5oh3Drp0UgcqXlX3e7fcg7QRg70OJXWJsR4Ax0fu0o1+ja8JaOWfIpT2/
↪xJy7Tr3o6yfxQZf0cstf307spbocTS5rr3C9bc5W+kZzCljeaCSs41lGdG2rJp++V1JmJp6/
↪tuwYGf86cfxpS2Hq8+Qjvh7bp3rPxbzUvU9fTEId+BA5yD3pFDVE6l7eM8vJeRRtrreqMs/Xq+B7F08EBs/
↪tI3udepfzih5VT6wVd7z5A2fa59OlncwqkXgFk0UxqXIDMhj3Oxa+T4l1NpyqDKnTjSRXOaRB/
↪UWYml4Rdqd1sX83d8Iqa069wS8zjiSEs11tZdHxJLN61ZXnX5KH/
↪6YqbU0lJxu5840h1bj+0fWEos3dXfIJsPrPYwpbWRYS6/OUMsrU/LmnvsKrG0adtR246z/E8/
↪kjl9643Hnb7FkXZ4nQqqCqGWin79TSD1flSHT9JSaSqx9JR+xW1JPLH0nQbmh7HbjaWffuL97rIQYuml8tXNw7QJpV8c+uJGeCV/
```

(continues on next page)

(continued from previous page)

```
                           "job" : "pluginTest.2023-04-14_10.14.42_03"
                        }
                     ],
                     "description" : "mailboxfolders",
                     "id" : "MailboxFolders"
                  }
               },
               "id" : "ex-admin@testlab.local",
               "name" : "ex-admin@testlab.local"
            }
         }
      }
   },
   "jobName" : "pluginTest"
}
```

See also:

- Go to *Restore in Exchange EWS Plugin*
- Go to *List & Query*

Go back to the *Operations* article.

Restore in Exchange EWS Plugin

Exchange EWS plugin is able to restore to any local filesystem mounted over the host where the File Daemon is running, or to the Exchange environment. The restore method is selected based on the value of the `where` parameter at restore time:

- Empty or '/' (example: `where=/`) → Exchange EWS restore will be triggered
- Any other path for where (example: `where=/tmp`) → Local file system restore will be triggered.

When using the Exchange EWS restore method, the following parameters are available to control the restore behavior under 'Plugin Options' menu during a bconsole restore session:

Option	Required	Default	Values	Example	Description
destination_user	No		Existing email address on the target Exchange service	AlexW@yourdomain.com	Destination User where restore data will be uploaded. If no user is set, every selected file will be restored in the original account
destination_archive	No		0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	yes	Restore using the in-place archving tree instead the regular mailbox tree
send_report	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	0	Send an email to the destination user with a report containing the result of all restored (or failed) items
foreign_container_generation	No	1	0, no, No, false, FALSE, false, off ; 1, yes, Yes, TRUE, true, on	0	Generate a general folder to put inside restored items coming from different mailboxes. For example, if we restore emails from user a@domain.com into Mailbox of user b@domain.com, with this option enabled the plugin will generate an automatic folder a@domain.com inside the destination restore folder used over destination user b@domain.com
endpoint	No	Original backup value	A hostname or IP address	win19-cl1-exch	Cross-server restore: Hostname or IP address that matches the DN of the SSL Certificate of the Destination Exchange service
admin_domain	No	Original backup value	A domain name	MYEXCHANGEDOMAIN	Cross-server restore: The users domain name. If admin_user is including already the domain, this parameter must not be set
admin_user	No	Original backup value	Email address or username (with or without the domain prefix)	myadmin@mydomain.com	Cross-server restore: An email address, or the username of the admin user that has permissions to impersonate all the other users. The format can be an email address, a single username (then admin_domain needs to be filled in) or domainusername. For simplicity, it is recommended to use the email address
admin_password	No	Original backup value	A password string	G3934kdkV0348	Cross-server restore: The password associated to the admin user
debug	No	0	0, 1, 2 ,3, 4, 5, 6, 7, 8, 9	3	Change debug level

Restore Use Cases

The following restore scenarios are supported:

- Restore whole directories, specific items (email, task, contact, appointments or attachments) to original user or to a different user:
  - Restore parameters implied: `destination_user`.
- Restore directories, emails, or attachments to original path or to a different path:
  - Restore parameters implied: `destination_path`.
- Restore to the local filesystem (general restore `where` parameter must be set to a path).
- It is possible to control replacement behavior (items are compared using exchange id) with the generic replace option of Bacula.
- Restore to a different exchange server:
  - Restore parameters implied: `endpoint`, `admin_domain`, `admin_user`, `admin_password`.

Some particularities to remark:

- If no `destination_user` is set, every message will be restored into its original mailbox.
- If no `destination_path` is set, every message will be restored into its original path.
- If the selection contains messages from several users:
  - Original user messages will be restored in their original location
  - For other users, a special folder will be created with the email address of each of them, containing the full path and messages of the restored objects, unless the parameter `foreign_container_generation` is disabled
  - Example: Restore of emails from 2 different users over a third mailbox without destination_path result in auto-generated Restore_date folder containing those 2 foreign users with the restored folder inside of them.

Restore Example Session

In the following restore example session, we restore into the original mailbox all the emails of the backup, inside the ‘restored’ folder.

Note: It is also possible to run backup or restore operations from any of the Bacula Graphical User Interfaces.

Listing 27: **Restore bconsole session**

```
Connecting to Director 127.0.0.1:8101
1000 OK: 10002 127.0.0.1-dir Version: 16.0.5 (05 April 2023)
Enter a period to cancel a command.
*restore
Automatically selected Catalog: MyCatalog
Using Catalog "MyCatalog"

First you select one or more JobIds that contain files
to be restored. You will be presented several methods
of specifying the JobIds. Then you will be allowed to
```

(continues on next page)

(continued from previous page)

```
select which files from those JobIds are to be restored.

To select the JobIds, you have the following choices:
   1: List last 20 Jobs run
   2: List Jobs where a given File is saved
   3: Enter list of comma separated JobIds to select
   4: Enter SQL list command
   5: Select the most recent backup for a client
   6: Select backup for a client before a specified time
   7: Enter a list of files to restore
   8: Enter a list of files to restore before a specified time
   9: Find the JobIds of the most recent backup for a client
   10: Find the JobIds for a backup for a client before a specified time
   11: Enter a list of directories to restore for found JobIds
   12: Select full restore to a specified Job date
   13: Select object to restore
   14: Cancel
Select item:  (1-14): 5
Automatically selected Client: 127.0.0.1-fd
Automatically selected FileSet: FS_E2WS
+-------+-------+----------+------------+---------------------+-------------------+
| jobid | level | jobfiles | jobbytes   | starttime           | volumename        |
+-------+-------+----------+------------+---------------------+-------------------+
|     1 | F     |       32 | 10,632,924 | 2023-04-17 12:52:41 | TEST-2023-04-17:0 |
+-------+-------+----------+------------+---------------------+-------------------+
You have selected the following JobId: 1

Building directory tree for JobId(s) 1 ...
31 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ mark *
31 files marked.
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.2.bsr

The Job will require the following (*=>InChanger):
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

   TEST-2023-04-17:0         File                      FileStorage

Volumes marked with "*" are in the Autochanger.


31 files selected to be restored.

```

(continues on next page)

(continued from previous page)

```
Using Catalog "MyCatalog"
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:           /tmp/regress/tmp/bacula-restores
Replace:         Always
FileSet:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2023-04-17 13:16:33
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): mod
Parameters to modify:
   1: Level
   2: Storage
   3: Job
   4: FileSet
   5: Restore Client
   6: When
   7: Priority
   8: Bootstrap
   9: Where
   10: File Relocation
   11: Replace
   12: JobId
   13: Plugin Options
Select parameter to modify (1-13): 9
Please enter the full path prefix for restore (/ for none): /
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
Where:
Replace:         Always
FileSet:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2023-04-17 13:16:33
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (Yes/mod/no): mod
Parameters to modify:
   1: Level
   2: Storage
   3: Job
   4: FileSet
   5: Restore Client
   6: When
```

(continues on next page)

(continued from previous page)

```
   7: Priority
   8: Bootstrap
   9: Where
   10: File Relocation
   11: Replace
   12: JobId
   13: Plugin Options
Select parameter to modify (1-13): 13
Automatically selected : e2ws: service=email endpoint=w16-cl02-exch admin_user=ex-
↪admin@testlab.local admin_password=Bacula18 debug=4 user="ex-admin@testlab.local"␣
↪folder="REGRESS_20230417125041"
Plugin Restore Options
Option                          Current Value       Default Value
destination_user:               *None*              (*None*)
destination_path:               *None*              (*None*)
send_report:                    *None*              (1)
foreign_container_generation:   *None*              (1)
send_invitations_mode:          *None*              (AllCopy)
endpoint:                       *None*              (*None*)
admin_domain:                   *None*              (*None*)
admin_user:                     *None*              (*None*)
admin_password:                 *None*              (*None*)
debug:                          *None*              (*None*)
Use above plugin configuration? (Yes/mod/no): mod
You have the following choices:
   1: destination_user (Destination User)
   2: destination_path (Destination Path in Exchange)
   3: send_report (Send report of the restore operation to the affected user)
   4: foreign_container_generation (Generate a general container (usually a folder) to␣
↪put inside restored objects coming from different entities)
   5: endpoint (Destination Exchange endpoint)
   6: admin_domain (Destination Exchange endpoint admin user domain)
   7: admin_user (Destination Exchange endpoint admin user)
   8: admin_password (Destination Exchange endpoint admin password)
   9: debug (Change debug level)
Select parameter to modify (1-10): 2
Please enter a value for destination_path: restore
Plugin Restore Options
Option                          Current Value       Default Value
destination_user:               *None*              (*None*)
destination_path:               restore             (*None*)
send_report:                    *None*              (1)
foreign_container_generation:   *None*              (1)
endpoint:                       *None*              (*None*)
admin_domain:                   *None*              (*None*)
admin_user:                     *None*              (*None*)
admin_password:                 *None*              (*None*)
debug:                          *None*              (*None*)
Use above plugin configuration? (Yes/mod/no): yes
Run Restore job
JobName:         RestoreFiles
Bootstrap:       /tmp/regress/working/127.0.0.1-dir.restore.2.bsr
```

(continues on next page)

(continued from previous page)

```
Where:
Replace:         Always
FileSet:         Full Set
Backup Client:   127.0.0.1-fd
Restore Client:  127.0.0.1-fd
Storage:         File
When:            2023-04-17 13:16:33
Catalog:         MyCatalog
Priority:        10
Plugin Options:  User specified
OK to run? (Yes/mod/no): yes
Job queued. JobId=3
```

Listing 28: **Restore job result**

```
*llist joblog jobid=3
time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Start Restore Job RestoreFiles.2023-04-17_13.16.54_11

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Restoring files from JobId(s) 1

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Connected to Storage "File" at 127.0.0.1:8103 with TLS

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Using Device "FileStorage" to read.

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-dir JobId 3: Connected to Client "127.0.0.1-fd" at 127.0.0.1:8102␣
↪with TLS

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: Connected to Storage at 127.0.0.1:8103 with TLS

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-sd JobId 3: Ready to read from volume "TEST-2023-04-17:0" on File␣
↪device "FileStorage" (/tmp/regress/tmp).

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-sd JobId 3: Forward spacing Volume "TEST-2023-04-17:0" to addr=260

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Plugin log of this job available in: /tmp/regress/
↪working/e2ws/e2ws-debug-0.log

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Backend connection to testlab.local stablished

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Jar Version: 1.0.0 | Java version: 11.0.18
```

(continues on next page)

(continued from previous page)

```
time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Starting backend restore process

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Restore to Microsoft Exchange Service

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: No destination entity provided. Trying to restore␣
↪each item into its original owner entity

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Destination Path: restore

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Generate report: enabled

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Foreign container generation: enabled

time: 2023-04-17 13:16:57
logtext: 127.0.0.1-fd JobId 3: e2ws: Send invitations mode for appointments: AllCopy

time: 2023-04-17 13:17:03
logtext: 127.0.0.1-sd JobId 3: End of Volume "TEST-2023-04-17:0" at addr=10694381 on␣
↪device "FileStorage" (/tmp/regress/tmp).

time: 2023-04-17 13:17:03
logtext: 127.0.0.1-sd JobId 3: Elapsed time=00:00:06, Transfer rate=1.780 M Bytes/second

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-fd JobId 3: e2ws: Report sent to:ex-admin@testlab.local

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-fd JobId 3: e2ws: No more items to restore. Restore ended

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: Bacula 127.0.0.1-dir 16.0.5 (05Apr23):
   Build OS:               x86_64-pc-linux-gnu ubuntu 22.04
   JobId:                  3
   Job:                    RestoreFiles.2023-04-17_13.16.54_11
   Restore Client:         "127.0.0.1-fd" 16.0.5 (05Apr23) x86_64-pc-linux-gnu,ubuntu,22.
↪04
   Where:
   Replace:                Always
   Start time:             17-abr-2023 13:16:57
   End time:               17-abr-2023 13:17:19
   Elapsed time:           22 secs
   Files Expected:         31
   Files Restored:         31
   Bytes Restored:         10,669,371 (10.66 MB)
   Rate:                   485.0 KB/s
   FD Errors:              0
```

(continues on next page)

(continued from previous page)

```
   FD termination status:  OK
   SD termination status:  OK
   Termination:            Restore OK

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: Begin pruning Jobs older than 6 months .

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: No Jobs found to prune.

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: Begin pruning Files.

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: No Files found to prune.

time: 2023-04-17 13:17:19
logtext: 127.0.0.1-dir JobId 3: End auto prune.

            jobid: 3
              job: RestoreFiles.2023-04-17_13.16.54_11
             name: RestoreFiles
      purgedfiles: 0
             type: R
            level: F
         clientid: 1
       clientname: 127.0.0.1-fd
        jobstatus: T
    jobstatuslong: Completed successfully
        schedtime: 2023-04-17 13:16:33
        starttime: 2023-04-17 13:16:57
          endtime: 2023-04-17 13:17:19
      realendtime: 2023-04-17 13:17:19
    realstarttime: 2023-04-17 13:16:57
         jobtdate: 1,681,730,239
     volsessionid: 3
   volsessiontime: 1,681,728,758
         jobfiles: 31
         jobbytes: 10,669,371
        readbytes: 10,629,087
        joberrors: 0
  jobmissingfiles: 0
           poolid: 0
         poolname:
       priorjobid: 0
         priorjob:
        filesetid: 0
          fileset:
         hascache: 0
          comment:
         reviewed: 0
    isvirtualfull: 0
```

(continues on next page)

(continued from previous page)

```
            rate: 485
    compressratio: 0
       statusinfo:
     writestorage:
      writedevice:
  lastreadstorage: File
   lastreaddevice: FileStorage
```

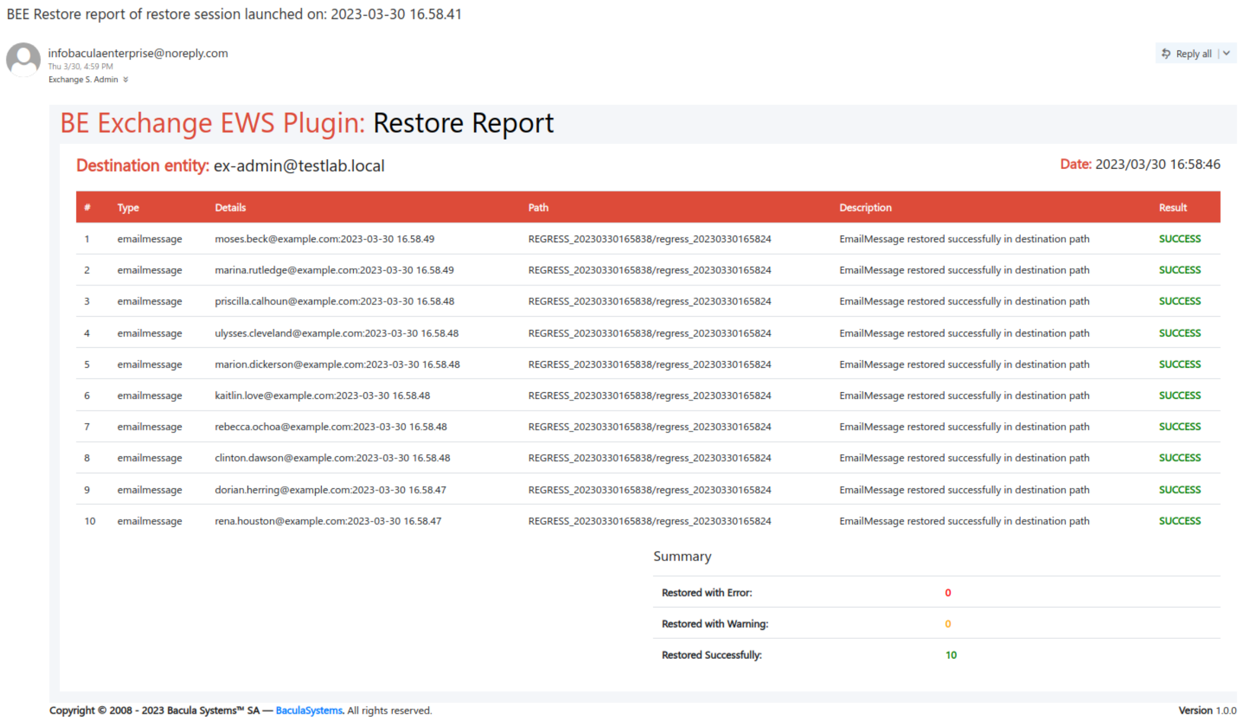
User Restore Report

Files and emails can represent very sensitive information for end-users. For that reason, information included in backup/restore logs is not exhaustive by default. For example, email restores do not include information such as the subject or sender when they are displayed in the backup log. However, for reporting and controlling purposes, the information of what has been exactly restored, what permissions have been applied, and other information can be useful and necessary for the affected user.

Bacula Enterprise Exchange EWS Plugin includes an option to generate a restore report in the user mailbox destination. The restore report contains detailed information about the items that have been restored successfully, if any of them had any trouble during the restore, and it also reports the date when the action was performed.

The generation of the report can be enabled/disabled in the bconsole restore session. If enabled, depending on the service, the report can generate an HTML file or an email in the Inbox of the affected user.

The image below shows an example report from an Email restore session:



BEE Restore report of restore session launched on: 2023-03-30 16.58.41

infobaculaenterprise@noreply.com
Thu 3/30, 4:59 PM
Exchange S. Admin

BE Exchange EWS Plugin: Restore Report

Destination entity: ex-admin@testlab.local

Date: 2023/03/30 16:58:46

#	Type	Details	Path	Description	Result
1	emailmessage	moses.beck@example.com:2023-03-30 16.58.49	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
2	emailmessage	marina.rutledge@example.com:2023-03-30 16.58.49	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
3	emailmessage	priscilla.calhoun@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
4	emailmessage	ulysses.cleveland@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
5	emailmessage	marion.dickerson@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
6	emailmessage	kaitlin.love@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
7	emailmessage	rebecca.ochoa@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
8	emailmessage	clinton.dawson@example.com:2023-03-30 16.58.48	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
9	emailmessage	dorian.herring@example.com:2023-03-30 16.58.47	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS
10	emailmessage	rena.houston@example.com:2023-03-30 16.58.47	REGRESS_20230330165838/regress_20230330165824	EmailMessage restored successfully in destination path	SUCCESS

Summary

Restored with Error:	0
Restored with Warning:	0
Restored Successfully:	10

Copyright © 2008 - 2023 Bacula Systems™ SA — BaculaSystems. All rights reserved.

Version 1.0.0

Fig. 2: Restore Email Example Report

See also:

- Go back to *Backup in Exchange EWS Plugin*

- Go to *List & Query*

Go back to the *Operations* article.

List & Query

It is possible to list information using the bconsole `.ls` command and providing a path. In general, we need to provide a path representing a folder inside the user mailbox. In addition, it is also possible to list the users from a given exchange endpoint through a `.query` command.

Below, there are some examples:

List users:

Listing 29: **Query example: Users**

```
*.query client=127.0.0.1-fd plugin="e2ws: endpoint=xxxx admin_user=xxxxx@my.domain admin_
↪password=xxxxx" parameter=user
user=ex-admin@testlab.local
displayName=Exchange S. Admin
guid=82190796-3221-4f26-8efb-c52ffdc2c4d2
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=df20c005a2b64a3a82d431e97f12ce78-Exchange S
user=support@testlab.local
displayName=First S. User
guid=48b2b400-44ab-40e5-995c-fede3b4453c3
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=8efecafa2b724e85990b1047fca7cba3-First S. U
user=beuser@testlab.local
displayName=Backup Exec
guid=ca497f2d-5b03-4959-8dc8-78f475905a10
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=3f1b5ef6f9ef4a00bd5ea92206d7675c-Backup Exe
user=support2@testlab.local
displayName=Second D. User
guid=527e07d5-a961-4d4b-bc7a-ec00ebc09c32
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=ccc701ae568643329c87dbcaa8f0acfd-Second D.
user=CompanyMeeting@testlab.local
displayName=Bsys room
guid=8b357da2-1249-4fbb-ad27-070272ef7524
reference=/o=BaculaSystems/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/
↪cn=Recipients/cn=93b1df2aff8a4fadbf15d58b73c69999-Bsys room
```

List inbox emails:

Listing 30: **List example: Inbox emails**

```
*.ls client=127.0.0.1-fd plugin="e2ws: endpoint=xxxxxxxx admin_user=xxxx@mydomain.com␣
↪admin_password=xxxx user=ex-admin@testlab.local service=email" path=Inbox
Connecting to Client 127.0.0.1-fd at 127.0.0.1:8102
drwxr-xr-x   1 nobody   nogroup                  -1 1970-01-01 00:59:59   /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/
-rw-r-----   1 nobody   nogroup              386774 2023-04-14 10:16:15   /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPfLckyAAA=_r20230414-101615.msg
```

(continues on next page)

(continued from previous page)

```
-rw-r-----   1 nobody   nogroup              381535 2023-03-30 17:04:26  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgqFAAA=_r20230330-170426.msg
-rw-r-----   1 nobody   nogroup              381535 2023-03-30 17:04:09  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgqEAAA=_r20230330-170409.msg
-rw-r-----   1 nobody   nogroup              385825 2023-03-30 17:03:22  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgqDAAA=_r20230330-170322.msg
-rw-r-----   1 nobody   nogroup              385598 2023-03-30 17:02:10  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgqCAAA=_r20230330-170210.msg
-rw-r-----   1 nobody   nogroup              386081 2023-03-30 17:00:59  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgqBAAA=_r20230330-170059.msg
-rw-r-----   1 nobody   nogroup              382465 2023-03-30 16:59:48  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgqAAAA=_r20230330-165948.msg
-rw-r-----   1 nobody   nogroup              381644 2023-03-30 16:59:05  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgp%2FAAA=_r20230330-165905.msg
-rw-r-----   1 nobody   nogroup              384487 2023-03-30 16:58:09  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgp+AAA=_r20230330-165809.msg
-rw-r-----   1 nobody   nogroup              381568 2023-03-30 16:57:44  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgp9AAA=_r20230330-165744.msg
-rw-r-----   1 nobody   nogroup              381978 2023-03-30 16:57:18  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgp8AAA=_r20230330-165718.msg
-rw-r-----   1 nobody   nogroup              381402 2023-03-30 16:55:51  /@e2ws/testlab.
↪local/users/ex-admin@testlab.local/Inbox/AAPXSgp7AAA=_r20230330-165551.msg
...
```

See also:

- Go back to *Backup in Exchange EWS Plugin*
- Go back to *Restore in Exchange EWS Plugin*

Go back to the *Operations* article.

See also:

- Go back to *Exchange EWS Scope*
- Go back to *Exchange EWS Features*
- Go back to *Exchange EWS Architecture*
- Go back to *Exchange EWS Installation*
- Go back to *Exchange EWS Configuration*
- Go to *Exchange EWS Best Practices*
- Go to *Exchange EWS Limitations*
- Go to *Exchange EWS Troubleshooting*

Go back to the *Exchange EWS plugin main page*.

1.7 Best Practices

The following article presents best practices regarding jobs distribution, concurrency and performance.

Jobs Distribution

It is recommended to split the target backup between different users, or even having one job per user. This way errors in one job will not invalidate a whole backup cycle, where some users have been successful, and others experienced errors. This also makes it easier to identify the cause of the error.

See also:

- Go to *Concurrency*
- Go to *Performance*

Go back to the *Best Practices* article.

Concurrency

When using Exchange EWS APIs, it is possible to find a variety of boundaries that need to be considered. We highlight some of them below:

- Exchange EWS Throttling: https://learn.microsoft.com/en-us/exchange/client-developer/exchange-web-services/ews-throttling-in-exchange
- Capabilities of the host serving the Exchange Service
- Usage of the service during the backup window
- Internet Information Server (IIS) limits: https://learn.microsoft.com/en-us/exchange/architecture/client-access/client-message-size-limits?view=exchserver-2019

If a boundary is crossed, the corresponding request will usually fail. Bacula Exchange EWS Plugin is prepared to wait some amount of time and retry it, so it has a certain level of resiliency. However, it is crucial to plan an adequate strategy to backup all the elements without needing to reach any boundary on a regular basis. This means to control how many concurrent requests are done during the backup window.

A single job implements some parallelism which can be reduced until a point, if necessary, using the following parameters:

- `backup_queue_size` - this variable controls the size of internal queues communicating internal threads, that are designed to fetch, open and send every item to Bacula core. Reducing its size will produce, ultimately (with a value of 1 for example), an execution very similar to a single threaded process.
- `concurrent_threads` which controls the number of simultaneous processes fetching and downloading data. This can be reduced or increased to directly affect the concurrency level of a single job.
- `concurrent_listing_threads` controls a different pool of threads intended only to fetch information from the API. It can be reduced to 1, but increasing it over the default values won't change significantly the behavior of the plugin.

The recommended strategy to backup a new environment is to plan a step-by-step testing scenario before putting it into production, where the number of users and the concurrency of the jobs are increased progressively. Other important point is the timing schedule as some boundaries are related to time-frames (number of request per amount of time). If you detect you reach boundaries when running all your backups during a single day of the week, try to increase the time window, and spread the load through it in order to achieve better performance results.

See also:

- Go back to *Jobs Distribution*
- Go to *Performance*

Go back to the *Best Practices* article.

Performance

The performance of this plugin is highly dependent on many external factors:

- Exchange latency and bandwidth
- Network infrastructure
- FD Host hardware
- FD Load
- Ratio number of elements/size
- And many more.

In summary, it is not possible to establish an exact reference about how much time a backup will need to complete.

As a reference and regarding the number of elements and their size:

- Many little objects to protect: More objects per second, but smaller speed (MB/s).
- Big files to protect: Fewer objects per second, but greater speed (MB/s).

It is recommended to benchmark your own environment in base to your requirements and needs.

The automatic parallelization mechanism (using `concurrent_threads=x`) should work well for most scenarios, however, fine-tune is possible if we define one job per user, and we control how many of them run in parallel, together to decrease the `concurrent_threads` value in order to avoid throttling or Exchange server capacity problems.

There are many possible strategies to use this plugin, so it is recommended to study what suits your needs best before deploying the jobs in your entire environment, so you can get the best possible results:

- You can have a job per user and all services.
- You can have multiple entities and only some services inside a job.
- You can split your workload through a schedule, or try to run all your jobs together.
- You can run jobs in parallel or take advantage of `concurrent_threads` and so, run less jobs in parallel.
- You can select what services to backup or backup them all.
- You can backup whole data to backup or select precisely what elements you really need inside each service (folders).
- And more.

See also:

- Go back to *Jobs Distribution*
- Go back to *Concurrency*

Go back to the *Best Practices* article.

See also:

- Go back to *Exchange EWS Scope*
- Go back to *Exchange EWS Features*

- Go back to *Exchange EWS Architecture*
- Go back to *Exchange EWS Installation*
- Go back to *Exchange EWS Configuration*
- Go back to *Exchange EWS Operations*
- Go to *Exchange EWS Limitations*
- Go to *Exchange EWS Troubleshooting*

Go back to the *Exchange EWS plugin main page*.

1.8 Limitations

The following article presents limitations of Exchange EWS Plugin.

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

More detailed limitations:

Protection Scope

Only the items that are listed in the *features' section* of this document are backed up. This means that backup with this plugin **does not include** elements such as the mailbox configuration, mailbox rules or any other database element outside the items the users can directly work with.

If you are interested in including also those elements into your backup strategy, consider the combination of this plugin with the Bacula Enterprise Exchange VSS Plugin, that works at database level.

Legacy public folders, when not connected to a shared and accessible mailbox are not supported. They will be supported in future versions of this plugin.

See also:

- Go to *Backup of Attachments and Files*
- Go to *Empty Files*

Go back to the main *Limitations* article.

Backup of Attachments and Files

In general, this plugin backups two types of information:

- Objects
- Files.

Objects are elements representing some item in Exchange such as a calendar event, a contact, an email, etc., while files are attachments of those items.

While objects are directly streamed from memory to the backup engine, files need to be downloaded to the FD host before being sent. This is done in order to perform metadata checks and to improve overall performance, as this the way operations can be parallelized. Every file is removed just after being completely downloaded and sent to the backup engine.

The path used for this purpose is established by the `path` plugin variable, that usually is set up in the backend script (e2ws_backend) with the value: `/opt/bacula/working`.

Inside the `path` variable, a `spool` directory will be created and used for those temporary download processes.

Therefore, it is necessary to have at least enough disk space available for the size of the largest file in the backup session. If you are using concurrency between jobs, or through the same job (by default this is the case through the `concurrent_threads=5` parameter), you would need at least that size for the largest file multiplied by the number of operations you run in parallel.

See also:

- Go back to *Protection Scope*
- Go to *Empty Files*

Go back to the main *Limitations* article.

Empty Files

In general, empty files (files with 0 byte contents) are simply not backed up by the Exchange EWS plugin. In particular, item attachments will show a message in the joblog to inform about empty files detected and so, not processed.

See also:

- Go back to *Protection Scope*
- Go back to *Backup of Attachments and Files*

Go back to the main *Limitations* article.

See also:

- Go back to *Exchange EWS Scope*
- Go back to *Exchange EWS Features*
- Go back to *Exchange EWS Architecture*
- Go back to *Exchange EWS Installation*
- Go back to *Exchange EWS Configuration*
- Go back to *Exchange EWS Operations*
- Go back to *Exchange EWS Best Practices*
- Go to *Exchange EWS Troubleshooting*

Go back to the *Exchange EWS plugin main page*.

1.9 Troubleshooting

In this article, there are suggested solutions to common situations that can cause trouble during the usage of the Exchange EWS plugin.

Certificate Problem

The certificate associated to the configured endpoint should be a valid one, or the plugin will reject to connect to it for security reasons.

The Common Name (CN) of the certificate should match the hostname used in the endpoint variable. Otherwise, the plugin will raise errors like: “The request failed. The request failed. Host name ‘10.10.10.99’ does not match the certificate subject provided by the peer”.

If the certificate CN uses a hostname, we need to use that hostname from the plugin configuration, instead of the IP address. If needed, because the DNS Server used by the File Daemon host cannot resolve that hostname, the IP-hostname association should be added to the local /etc/hostname file (or equivalent local hostname configuration, depending on the Operative System).

On the other hand, if the certificate is not valid but it is needed to tell the plugin to trust it, the procedure is to add it to the local keystore of the Java Virtual Machine running on the File Daemon. An example of how to do it is given below:

Listing 31: **Throttling unlimited**

```
keytool -cacerts -storepass changeit -importcert -alias ewscert -file certificate.cer
```

Note that:

- keytool should be available in the PATH of the system. If it’s not, you should look for it inside the Java JRE installation path (bin directory).
- changeit is the default password of the Java keystore. You should change it.
- certificate.cer is the file containing the Exchange certificate to import. You should download it, for instance, using a browser, and connect to Outlook.

See also:

- Go to *Out of Memory*
- Go to *Throttling*

Go back to the main *Troubleshooting* article.

Out of Memory

If you ever face *OutOfMemory* errors of the Java daemon (you will find them in the e2ws-debug.err file), you are very likely using a high level of concurrency through internal `concurrent_threads` parameter and/or parallel jobs.

To overcome this situation you can:

a) Reduce `concurrent_threads` parameter.

b) Reduce the number of jobs running in parallel.

c) If you cannot do that, you should increase JVM memory.

To increase JVM memory, you will need to:

Create this file: ‘/opt/bacula/etc/e2ws_backend.conf’.

Below, an example of the contents: E2WS_JVM_MIN=2G E2WS_JVM_MAX=8G

Those values will define the MIN (E2WS_JVM_MIN) and MAX (E2WS_JVM_MAX) memory values assigned to the JVM Heap size. In this example, we are setting 2Gb for the minimum, and 8Gb for the maximum. In general, those values should be more than enough. Be careful if you are running jobs in parallel, as very big values and several jobs at a time could quickly eat all the memory of your host.

The '/opt/bacula/etc/e2ws_backend.conf' won't be modified through package upgrades, so your memory settings will be persistent.

See also:

- Go back to *Certificate Problem*
- Go to *Throttling*

Go back to the main *Troubleshooting* article.

Throttling

It is possible to manage Exchange throttling policies, and increase them if it detected a high number of requests rejected while doing backup jobs. Below, there is an example of how to configure an unlimited throttling policy for a given account:

Listing 32: **Throttling unlimited**

```
New-ThrottlingPolicy BaculaNoThrottling
Set-ThrottlingPolicy BaculaNoThrottling -RCAMaxConcurrency unlimited -RcaMaxBurst␣
↪unlimited -RcaRechargeRate unlimited -RcaCutoffBalance unlimited
Set-Mailbox <user or service account> -ThrottlingPolicy BaculaNoThrottling
```

See also:

- Go back to *Certificate Problem*
- Go back to *Out of Memory*

Go back to the main *Troubleshooting* article.

See also:

- Go back to *Exchange EWS Scope*
- Go back to *Exchange EWS Features*
- Go back to *Exchange EWS Architecture*
- Go back to *Exchange EWS Installation*
- Go back to *Exchange EWS Configuration*
- Go back to *Exchange EWS Operations*
- Go back to *Exchange EWS Best Practices*
- Go back to *Exchange EWS Limitations*

Go back to the *Exchange EWS plugin main page*.

Bacula Enterprise also provides a traditional plugin to backup and restore Exchange instances through a different plugin, based on Windows VSS technology that is capable of working at a database level. That option is recommended for disaster recovery strategies, while the plugin discussed in the current document is intended to be used for item level capabilities. For information about Exchange Plugin based on VSS, refer to:

2 Exchange VSS Plugin

See also:

2.1 Exchange Single Item Restore

Overview

This user's guide presents how to use the Exchange Single Item Restore feature with **Bacula Enterprise** and the VSS Exchange Plugin allowing to interactively restore selected Exchange Mailboxes.

Note: Since 16.0.7, Bacula Enterprise offers the Exchange EWS Plugin which uses the advanced technology from Microsoft to restore single item in a very easy way. Review *this documentation* for further information about it.

Features Summary

The **Bacula Enterprise** Exchange Single Item Restore provides the following main features:

- Console interface
- BWeb Management Suite interface
- Exchange interface

Scope

This document will present solutions for **Bacula Enterprise** 8.4 and later, which are not applicable to prior versions. The Exchange Single Item Restore has been tested and is supported on Red Hat Enterprise Linux (RHEL) 6 and 7, Ubuntu 14.04, Debian 7 and 8, working with Microsoft Exchange Server 2010 and 2013.

Installation

Packages of the Exchange Single Item Restore plugin are available for supported platforms in the “single-item-restore” download area. The package delivered is usually a previous **Bacula Enterprise** version, therefore please search for previous versions in your download area or keep the version installed when upgrading your Storage Daemon. Please contact Bacula Systems to get access to them or if you would have any question.

Download the plugin package to your **Storage Daemon** server and then install using the package manager:

```
# rpm -ivh bacula-enterprise-single-item-restore*.rpm
```

The package manager will ensure that your **Bacula Enterprise** version is compatible with the Exchange Single Item Restore plugin and will install dependencies. On Redhat, installing the `perl-JSON` package from **rpmforge** is required.

Note: On Redhat 8.X and 9.x, you must have the the AppStream repository enabled to install the perl-File-Copy. The perl-File-Copy module is a dependency required by the bacula-enterprise-single-item-restore package.

Since Bacula Enterprise 16.0.13.

```
# cat /etc/yum.repos.d/dag.repo
[dag]
name = Red Hat Enterprise  - RPMFORGE
baseurl = https://www.baculasystems.com/dl/DAG/rhel6-64
enabled = 1
protect = 0
gpgcheck = 0

# cat /etc/yum.repos.d/baculasystems.repo
[singleitemrestore]
name = Bacula Enterprise
baseurl = https://www.baculasystems.com/dl/<id>/rpms/single-item-restore/<version>/rhel6-
↪64
enabled = 1
protect = 0
gpgcheck = 0
```

(continues on next page)

(continued from previous page)

```
[bacula]
name = Bacula Enterprise
baseurl = https://www.baculasystems.com/dl/<id>/rpms/bin/<version>/rhel6-64
enabled = 1
protect = 0
gpgcheck = 0
```

Where <id> is your customer download area identifier and <version> is the current Bacula Enterprise version.

```
# yum install bacula-enterprise-single-item-restore
```

The Exchange Single Item Restore also contains a specific Bacula Enterprise service called “bee-exchange” that must be installed and running on each Exchange server that is used to run the restore process. This installation is done by executing the dedicated installer program `bacula-enterprise-win64ExchangeSingleItemRestore-VERSION.exe`, where `VERSION` represents the product version number like `8.4.0-1` or `8.6.5-1`.

Note: Upgrading the “bee-exchange” service may not work in all cases. **Bacula Systems** recommends to explicitly deinstall an older version before installing a newer one.

The service will start a REST / HTTP daemon that will listen on `localhost:8081` by default. It is possible to configure the service to listen on an external IP address and use HTTPS and / or http authentication. If you decide to keep the default values for the bind address (localhost), you will need to run the internet browser used for the restore operation on the Exchange Server itself. Assuming the Exchange Server to be properly protected against unauthorized access, this provides a reasonably safe default protection against misuse.

Notes about the “bacula” Account on Redhat

All shell commands in this document use the “bacula” Unix account to run.

On Red Hat, the Unix “bacula” account is locked by default. This implies that it is not possible to execute a command such as `su - bacula` successfully.

It is possible to unlock the “bacula” account, or to use “sudo -u bacula” to execute commands. For example:

```
bacula@storage# /opt/bacula/bin/bconsole
```

This could be run from the root account using the following command:

```
root@storage# sudo -u bacula /opt/bacula/bin/bconsole
```

It is also possible to start a shell session using

```
root@storage# sudo -u bacula /bin/bash
```

Alternatively, unlock the “bacula” unix account and use `su` with a command such as:

```
root@storage# chsh -s /bin/bash bacula
root@storage# su - bacula
bacula@storage# whoami
bacula
```

Samba SMB Shares

The **Bacula Enterprise** Exchange Single Item Restore plugin will use Samba SMB shares automatically to provide the needed data to the Exchange system. It will set up those shares automatically.

To use Samba SMB network shares, installing and configuring the "samba" package is mandatory. To configure the /etc/samba/smb.conf file for use with Bacula the script install-single-item-restore.sh needs to be run.

```
root@storage# /opt/bacula/scripts/install-single-item-restore.sh install
Do you want to initialise Samba smb.conf [yes/No]: yes
Choose a Workgroup [BACULA]:

Need to set a password to "bacula" user
New SMB password: *******
Retype new SMB password: ******

Configuration done.

root@storage# cat /etc/samba/smb.conf
[global]
workgroup = BACULA
wide links = yes
follow symlinks = yes
unix extensions = no
include = /etc/samba/bacula.d/main.conf
```

At this point, it is possible to modify /etc/samba/smb.conf to add your own configuration directives. See below for some hints regarding security-related settings.

The network share configurations used by the Exchange Single Item Restore will be stored in the directory /etc/samba/bacula.d. Customizing the template used by Bacula to generate configuration files is possible by using a template file.

Depending on the version of the Samba server and its actual configuration, modifications to the template share will be required. In particular, the Exchange Single Item Restore will create symbolic links inside the created share which point to locations outside the shared directory tree. As this can pose a security risk, Samba by default may prevent clients access to such links. To allow these client accesses, which are reasonable as the target location is under control, and the "bacula" user account is assumed to be safe against misuse, a template such as the following should be put in place:

```
[root@storage ~]# cat /etc/samba/bacula.d/custom.tpl
[__share__]
    path = __path__
    browseable = No
    level2 oplocks = No
    oplocks = No
    posix locking = No
    wide links = Yes
    read only = No
    valid users = bacula
    follow symlinks = Yes
```

Note the explicit setting of **follow symlinks = Yes** and **wide links = Yes**. These, along with the global Samba configuration setting of **allow insecure wide links = Yes**, will allow a client to access the symbolically linked data locations that are required by the Exchange Single Item Restore. We suggest to review the impact of those settings in the manual describing the smb.conf configuration file.

The Samba server can also be configured to join your AD domain, and it is possible to allow a group of users to mount and use the Bacula network share. However, configuring the Samba server to use an existing Active Directory server is not covered by this document.

The Samba share will be used by the Exchange server to access the database and log files. We advise testing the creation of a Samba share and accessing it from the Exchange server to make sure that everything is configured properly at the network level.

```
bacula@storage# /opt/bacula/bin/smbadd --add --share test1 --path /opt/bacula/bin
bacula@storage# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[test1]"
Loaded services file OK.

# Global parameters
[global]
        workgroup = BACULA
        unix extensions = No
        idmap config * : backend = tdb
        include = /etc/samba/bacula.d/bacula-test1.conf
        wide links = Yes


[test1]
        path = /opt/bacula/bin
        valid users = bacula
        read only = No
        browseable = No

bacula@storage# smbclient //localhost/test1
Enter bacula's password:
Domain=[BACULA] OS=[Windows 6.1] Server=[Samba 4.2.3]
smb: \> ls
  .                                   D        0  Fri May  3 14:15:48 2013
  ..                                  D        0  Fri Jun 12 16:21:39 2015
  bacula                              A     1614  Fri May  3 14:15:46 2013
  bacula-dir                          A  2771768  Fri May  3 14:15:47 2013
  bacula-fd                           A   887111  Fri May  3 14:15:47 2013
  bacula-sd                           A  2169994  Fri May  3 14:15:48 2013
...
```

At this point, it should be possible to mount the network share “test1” from the Exchange server using the “bacula” account. This can be verified using Windows’ Explorer. As the created shares are not browseable, they will not appear automatically in an Explorer window showing the Storage Daemon host; instead, the full path to the share needs to be entered into the address bar.

To delete the network share the following command can be used:

```
bacula@storage# /opt/bacula/bin/smbadd --del --share test1
```

Note on SMB Password

The password used in the SMB share should not contain the following characters: &, ’ and “.

BWeb Management Suite GUI Notes

To use the BWeb Management Suite graphical GUI with the Exchange Single Item Restore option, it is currently necessary to install and configure BWeb Management Suite on the Storage Daemon where the relevant jobs are stored. If the Director is not on the same machine as the Storage Daemon, it should not be a problem, just remember that the administrator needs to connect to the correct BWeb Management Suite instance to use specific Exchange Single Item Restore screens.

Configuration

Storage Daemon Configuration

On the **Storage Daemon** host server, the `bconsole` program has to be configured properly to allow the “bacula” user to connect to the Director with the configuration file `/opt/bacula/etc/bconsole.conf`.

```
bacula@storage# /opt/bacula/bin/bconsole
Connecting to Director mydir-dir:9101
1000 OK: 10002 mydir-dir Version: 8.4.0 (31 October 2015)
Enter a period to cancel a command.
* version
mydir-dir Version: 8.4.0 (31 October 2015) x86_64-redhat-linux-gnu
* quit
```

The Exchange Single Item Restore package contains a script that enables testing the connection with the Director and testing if the system can mount the *Bacula Virtual File System* properly.

```
bacula@storage#  /opt/bacula/scripts/install-single-item-restore.sh check
I: Try to restart the script with sudo...
I: Found catalog MyCatalog
I: bacula-fused started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
I: bacula-fused (rw) started on /tmp/bee-bfuse.XXXXX
I: MyCatalog found
I: 10 Client(s) found
I: /tmp/bee-bfuse.XXXXX unmounted
OK: All tests are good.
```

The *Bacula Virtual File System* is not designed to be used by end users to browse or restore files directly. If you try to access and browse the mount point, you will not see any files.

bee-exchange Service

It is possible to configure the `bee-exchange` service on the Exchange server by creating a configuration file named `bee-exchange.conf` in the **Bacula Enterprise** installation directory on the Exchange server.

```
C:> type c:\Program Files\Bacula\bee-exchange.conf
###################################################################
# Bacula Enterprise Exchange Single Item Restore Configuration File
###################################################################
Port=8082
Hostname=*
SSL
```

The following parameters can be set:

Port Specify the TCP/IP port to use. Ex: *Port=8080*

Hostname Specify the TCP/IP interface to bind to. Ex: *Hostname=192.168.0.1*

SSL Use SSL. Ex: *SSL*

SSLName Specify SSL Certificate name to use. Ex: *SSLName=MyCert*

LogFile Specify a custom Log file.

DataDir Specify a custom Data directory.

Authentication Specify a Authentication scheme (Ntlm, Basic, IntegratedWindows, Anonymous). Ex: *Authentication=Anonymous*

The service must be restarted after creation or modification of the configuration file.

The service account requires `Read` and `Write` permissions to all files in the folder with the database and sufficient access rights including the “mailbox import export” role. These permissions can be granted using impersonation as described in https://msdn.microsoft.com/en-us/library/bb204095.aspx and https://technet.microsoft.com/en-us/library/ee633452%28v=exchg.141%29.aspx

For example, if the local “System” account is member of the “Server Management” security group

```
# New-ManagementRoleAssignment -Name "Import Export_Enterprise Support" -SecurityGroup
↪"Server Management" -Role "Mailbox Import Export"
```

Backup

This product uses backups that are created as described in the **Bacula Systems** documentation of the Exchange server functionality of the VSS plugin, which at this time is available as *Exchange VSS Plugin*.

In short, backups need to be done with a line like **Plugin = “vss:/@EXCHANGE/”** in the File Set used.

There are no additional considerations beyond what is detailed in the above mentioned white paper, which also implies that Exchange Single Item Restore can be added to an existing **Bacula Enterprise** infrastructure and can be used with all existing backups.

Restore Scenarios

With Text Console Interface

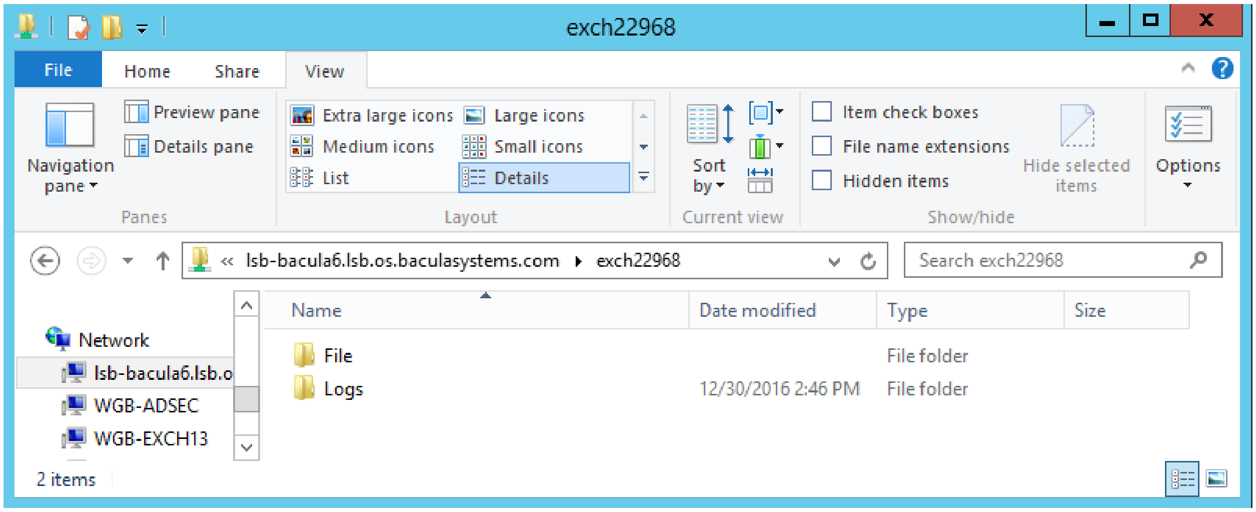
The Exchange Single Item Restore provides a console program that allows you to initialize the Exchange restore process.

```
[root@storage ~]# su - bacula -c '/opt/bacula/bin/mount-exchange'
Automatically Selected Catalog: MyCatalog

Client list:
1: bacula6-fd
2: wgb-exch13-fd
Select a Client: 2
Selected wgb-exch13-fd

Job list:
1: wgb-exch13.2016-12-20_13.09.55_18
2: wgb-exch13.2016-12-20_14.28.11_20
3: wgb-exch13.2016-12-20_14.28.53_21
4: wgb-exch13.2016-12-21_17.37.15_31
5: wgb-exch13.2016-12-21_17.42.36_32
6: wgb-exch13.2016-12-21_23.45.18_35
7: wgb-exch13.2016-12-27_13.09.24_06
Select a Job: 7
Selected wgb-exch13.2016-12-27_13.09.24_06
Automatically Selected Exchange Server: WGB-EXCH13

Exchange Database list:
1: MailboxDBA
2: MailboxDBStandalone
Select an Exchange Database: 1
Selected MailboxDBA

I: From the Exchange server web browser, you need to access the bee-echange
    service that uses the default URL: http://localhost:8081/?Share=exch22968

I: The Network share name is "exch22968"

I: Once the restore is done and the database is dismounted,
   press enter to finish and cleanup the session.
```

At this step, the Exchange data files are available locally on the Storage Daemon server, and a samba network share is configured.

For a verification of basic functionality, the Windows' Explorer can be used (remember that, as the share should not be browseable, the complete path needs to be entered in the address line). This is shown in figure *Verifying Access to a Created SMB Share*. Note the "File" folder without time stamp – this is represented by a symbolic link, and unless it can be opened in the Explorer window without problems, security restrictions are set too tightly in the `/etc/samba/smb.conf` file and the dynamically created share. In that case, reviewing and modifying those settings will be necessary.

Note that this access verification should only be useful during deployment and initial configuration of the Exchange Single Item Restore. Once things work out without further manual interaction, it is possible to directly use the actual restore functionality, described below.



« lsb-bacula6.lsb.os.baculasystems.com ▸ exch22968

Search exch22968

Name	Date modified	Type	Size
File		File folder	
Logs	12/30/2016 2:46 PM	File folder	

Fig. 3: Verifying Access to a Created SMB Share

To actually mount and browse the Exchange database, connect to the Exchange server and start a browser with the URL printed above.

The Bacula Enterprise Exchange service will handle all steps necessary to restore a mailbox (see chapter *Exchange Single Item Restore Screens*) through a Graphical User Interface.

To cleanup the restore session, just press "Enter" in the terminal session where the `mount-exchange` script was started.

With BWeb Management Suite Interface

The Exchange Single Item Restore option in BWeb Management Suite is a wizard allowing easy restoration of items from an Exchange database.

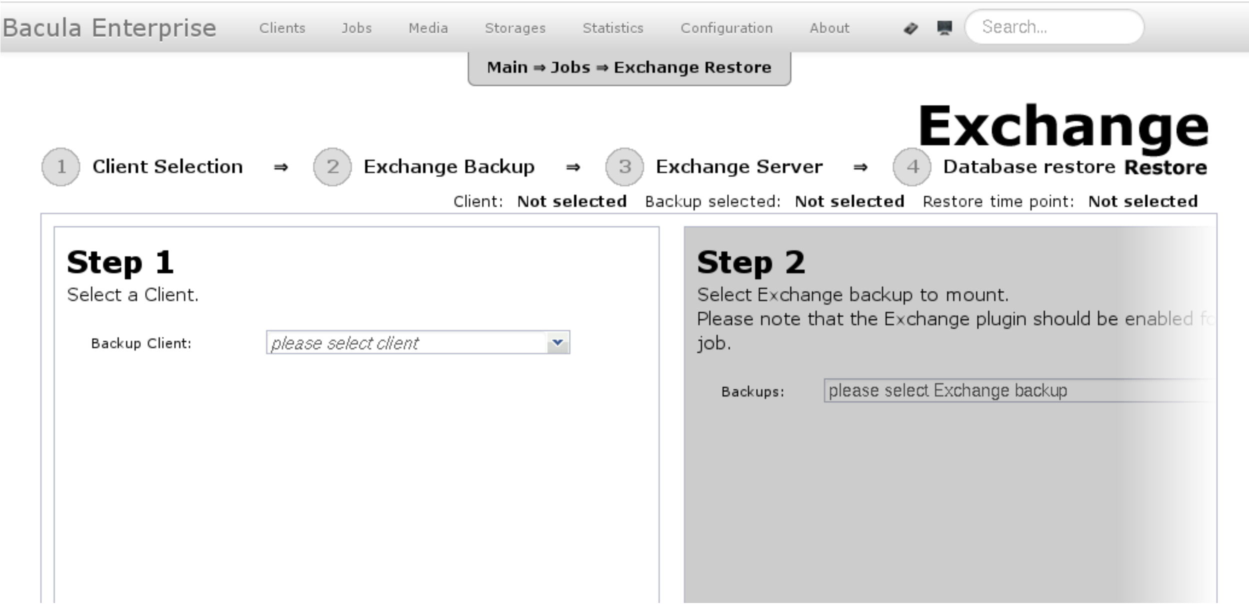
The first step is to select the Client where the VSS backup job was done (see fig. *Client Selection*).

Once the Client is selected, the administrator needs to select the Job (a Restore Point) to restore. (Fig. *Restore Point Selection*).

If the selected Job is a valid VSS backup job, the third step will display a list of all Exchange databases included. (Fig. *Exchange Database Selection*).

At this point, Bacula will create a network share with Samba, and the administrator needs to connect to the Exchange server and open the URL displayed in the wizard with a web browser. The Exchange Server address field is set to the Bacula Client address (see fig. *Connection to Exchange Single Item Restore Service*) automatically and can be changed if desired. Note that by default, the Exchange Single Item Restore windows service does **not** accept remote connections, so accessing it requires using a web browser started **on the Exchange server** console. (See chapter *Exchange Single Item Restore Screens*)

Once the restore is actually done, it is important to terminate the restore session to release resources (fig. *Terminate the Restore Session*).


Bacula Enterprise
Clients
Jobs
Media
Storages
Statistics
Configuration
About
Search...
Main ⇒ Jobs ⇒ Exchange Restore
Exchange Restore
1 Client Selection ⇒ 2 Exchange Backup ⇒ 3 Exchange Server ⇒ 4 Database restore

Client: **Not selected** Backup selected: **Not selected** Restore time point: **Not selected**

Step 1

Select a Client.

Backup Client: please select client

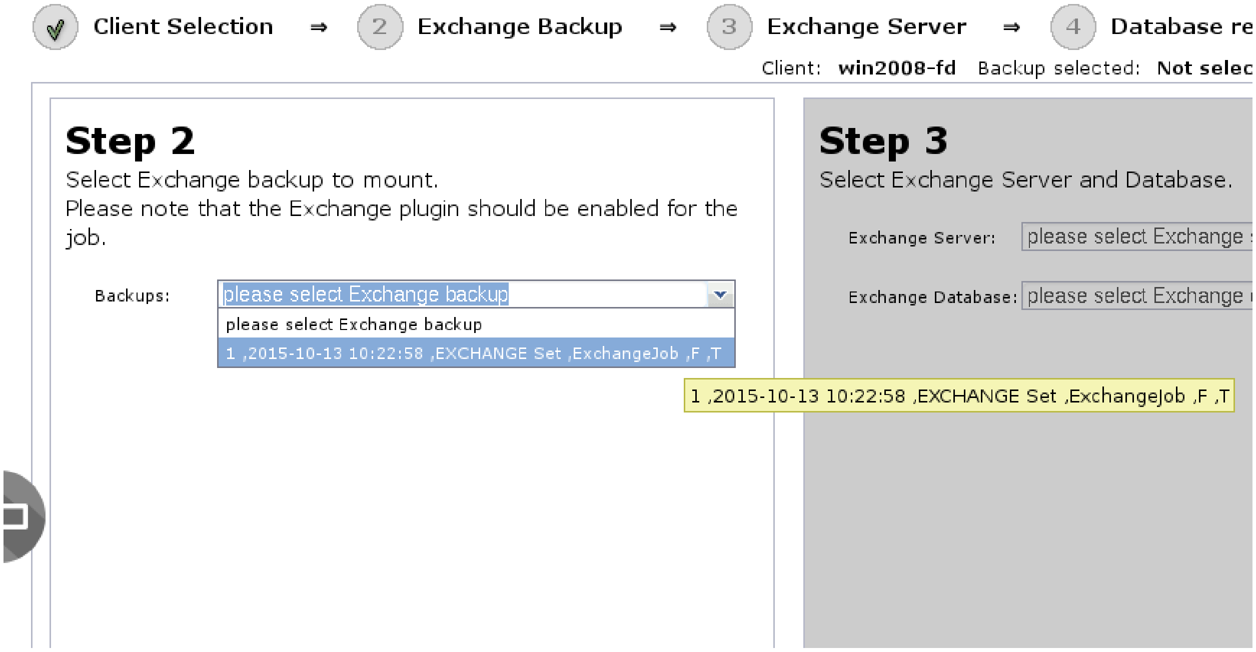
Step 2

Select Exchange backup to mount.

Please note that the Exchange plugin should be enabled fo job.

Backups: please select Exchange backup

Fig. 4: Client Selection


Client Selection ⇒ 2 Exchange Backup ⇒ 3 Exchange Server ⇒ 4 Database re

Client: **win2008-fd** Backup selected: **Not selec**

Step 2

Select Exchange backup to mount.

Please note that the Exchange plugin should be enabled for the job.

Backups: please select Exchange backup

please select Exchange backup

1 ,2015-10-13 10:22:58 ,EXCHANGE Set ,ExchangeJob ,F ,T

1 ,2015-10-13 10:22:58 ,EXCHANGE Set ,ExchangeJob ,F ,T

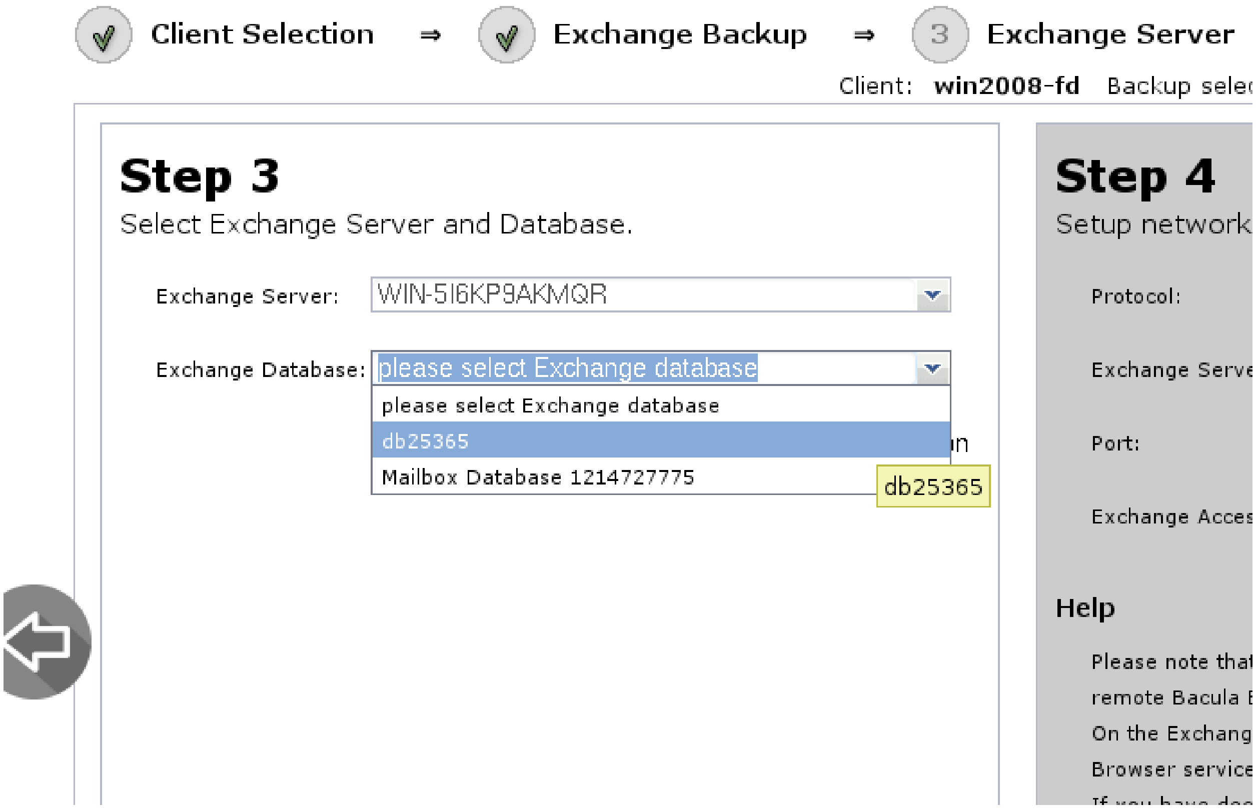
Step 3

Select Exchange Server and Database.

Exchange Server: please select Exchange s

Exchange Database: please select Exchange d

Fig. 5: Restore Point Selection



Client Selection ⇒ Exchange Backup ⇒ 3 Exchange Server
Client: win2008-fd Backup selec

Step 3

Select Exchange Server and Database.

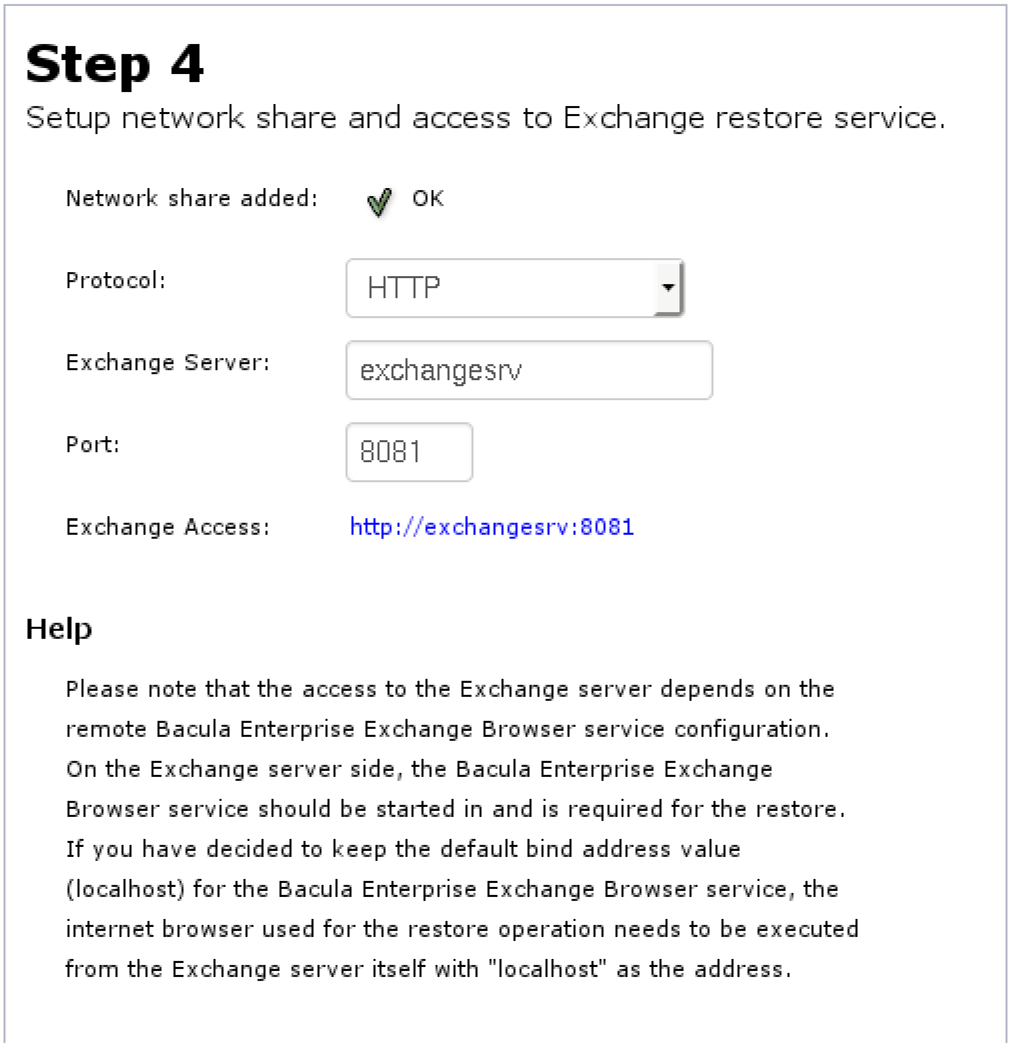
Exchange Server: WIN-5I6KP9AKMQR

Exchange Database: please select Exchange database

please select Exchange database
db25365
Mailbox Database 1214727775
db25365

Step 4
Setup network
Protocol:
Exchange Serve
Port:
Exchange Acces
Help
Please note that
remote Bacula E
On the Exchang
Browser service

Fig. 6: Exchange Database Selection



Step 4

Setup network share and access to Exchange restore service.

Network share added: ✓ OK

Protocol: HTTP

Exchange Server: exchangesrv

Port: 8081

Exchange Access: http://exchangesrv:8081

Help

Please note that the access to the Exchange server depends on the remote Bacula Enterprise Exchange Browser service configuration. On the Exchange server side, the Bacula Enterprise Exchange Browser service should be started in and is required for the restore. If you have decided to keep the default bind address value (localhost) for the Bacula Enterprise Exchange Browser service, the internet browser used for the restore operation needs to be executed from the Exchange server itself with "localhost" as the address.

Fig. 7: Connection to Exchange Single Item Restore Service

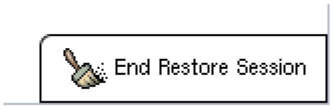
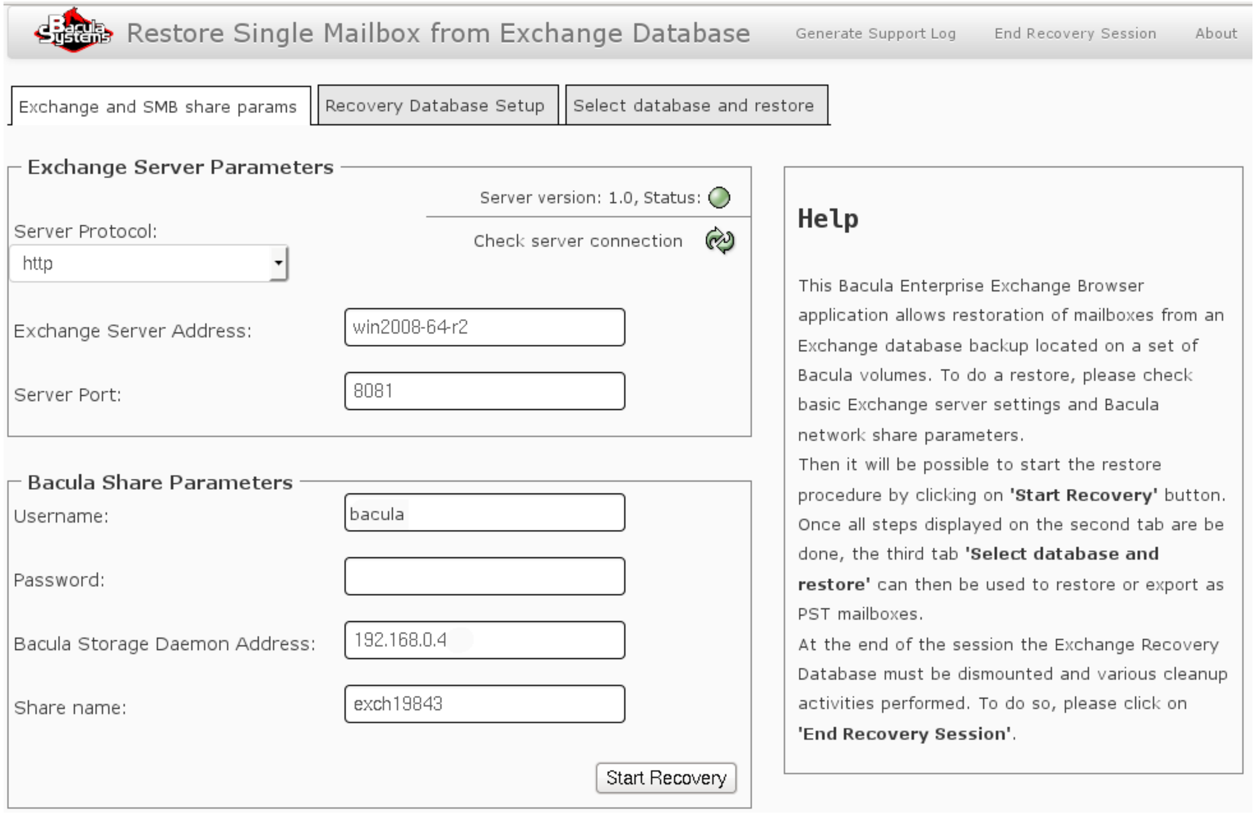


Fig. 8: Terminate the Restore Session

Exchange Single Item Restore Screens

The “Restore Single Mailbox from Exchange Database” page should be displayed as shown in figure *Bacula Exchange Single Item Restore GUI*. At this point, operations will take place directly on the Exchange server and will accomplish the following operations:

1. Mount the network share
2. Check the Exchange database status
3. Perform recovery if needed
4. Create a Recovery database
5. Mount the Recovery database
6. Browse and restore mailboxes
7. Dismount and cleanup



Exchange Server Parameters

Server version: 1.0, Status: ●

Server Protocol: http

Exchange Server Address: win2008-64-r2

Server Port: 8081

Bacula Share Parameters

Username: bacula

Password:

Bacula Storage Daemon Address: 192.168.0.4

Share name: exch19843

Help

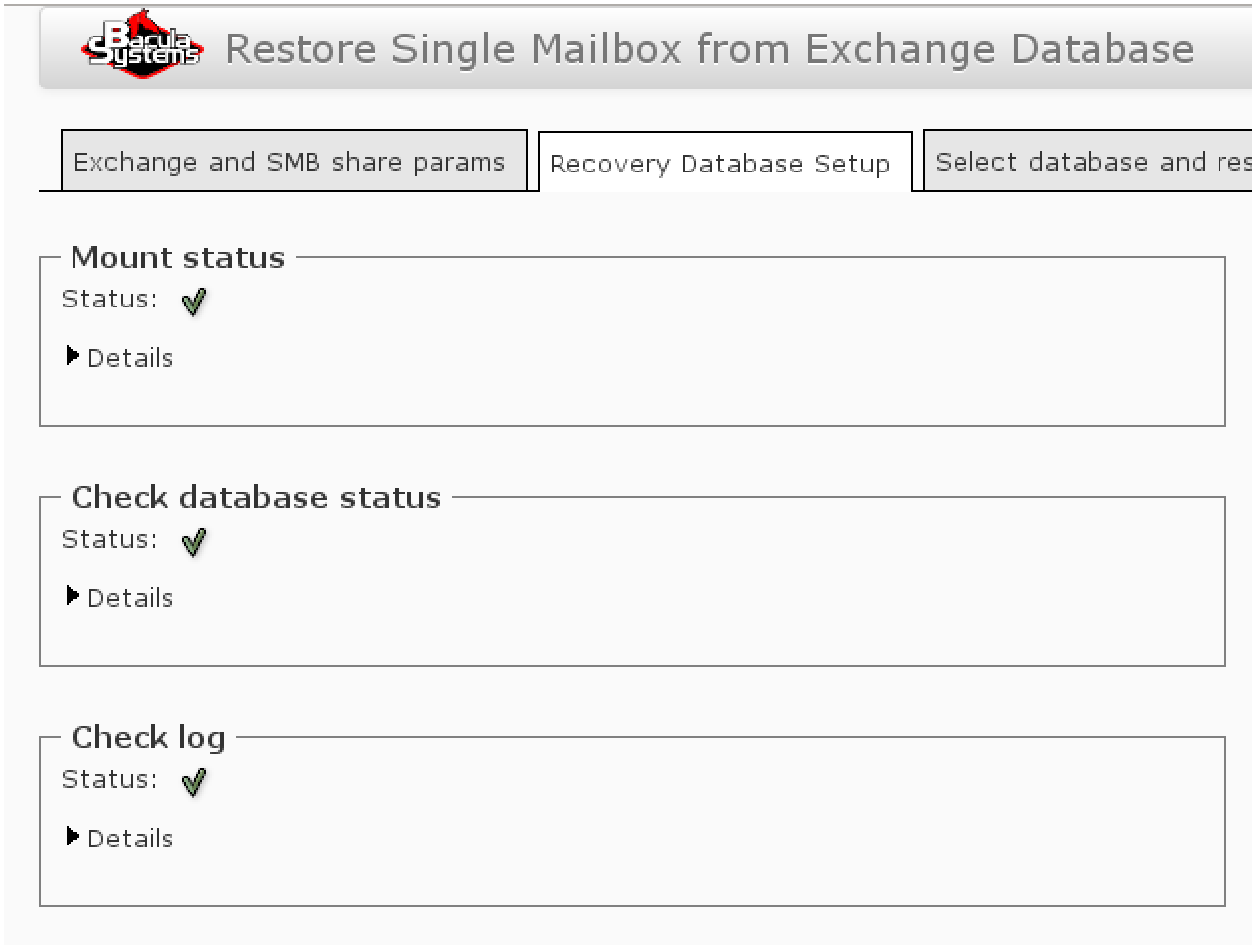
This Bacula Enterprise Exchange Browser application allows restoration of mailboxes from an Exchange database backup located on a set of Bacula volumes. To do a restore, please check basic Exchange server settings and Bacula network share parameters.
Then it will be possible to start the restore procedure by clicking on **'Start Recovery'** button. Once all steps displayed on the second tab are be done, the third tab **'Select database and restore'** can then be used to restore or export as PST mailboxes.
At the end of the session the Exchange Recovery Database must be dismounted and various cleanup activities performed. To do so, please click on **'End Recovery Session'**.

Fig. 9: Bacula Exchange Single Item Restore GUI

In order to mount the network share where the Exchange database is available, “Bacula Share Parameters” must be specified. Then press the “Start Recovery” button. All steps will then be executed one after the other (fig. *Recovery Database Setup*). Information about a particular step is available by clicking on “Details” (fig. *Details of an Operation*).

Once in the last tab, the recovery database will be mounted and available to the Exchange console and the Powershell interface. The list of all mailboxes will be displayed like shown in figure *Mailbox List*. It is possible to select a Mailbox and restore it to another database.

Once the restore is actually done, it is important to terminate the restore session to release resources (fig. *Dismount Recovery Database*) on the Exchange server, and then terminate the restore session in the “mount-exchange” terminal

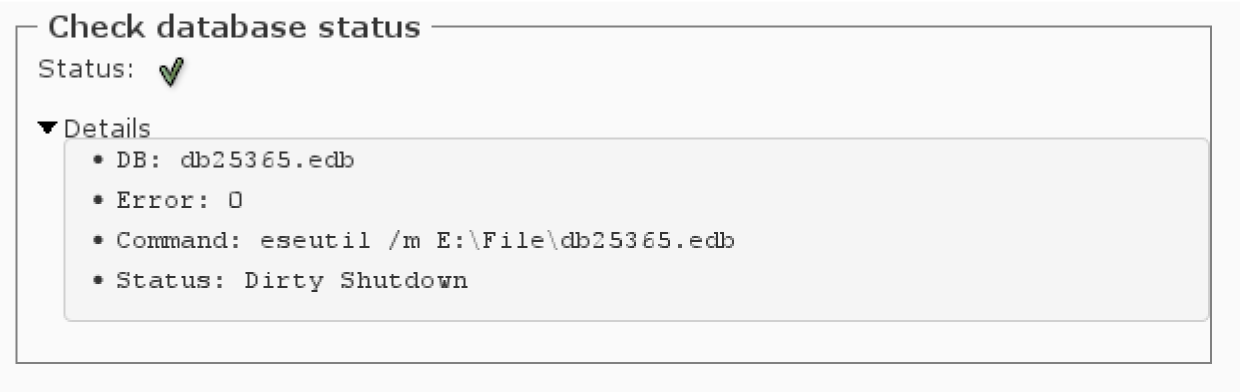


Status: ✓

Status: ✓

Status: ✓

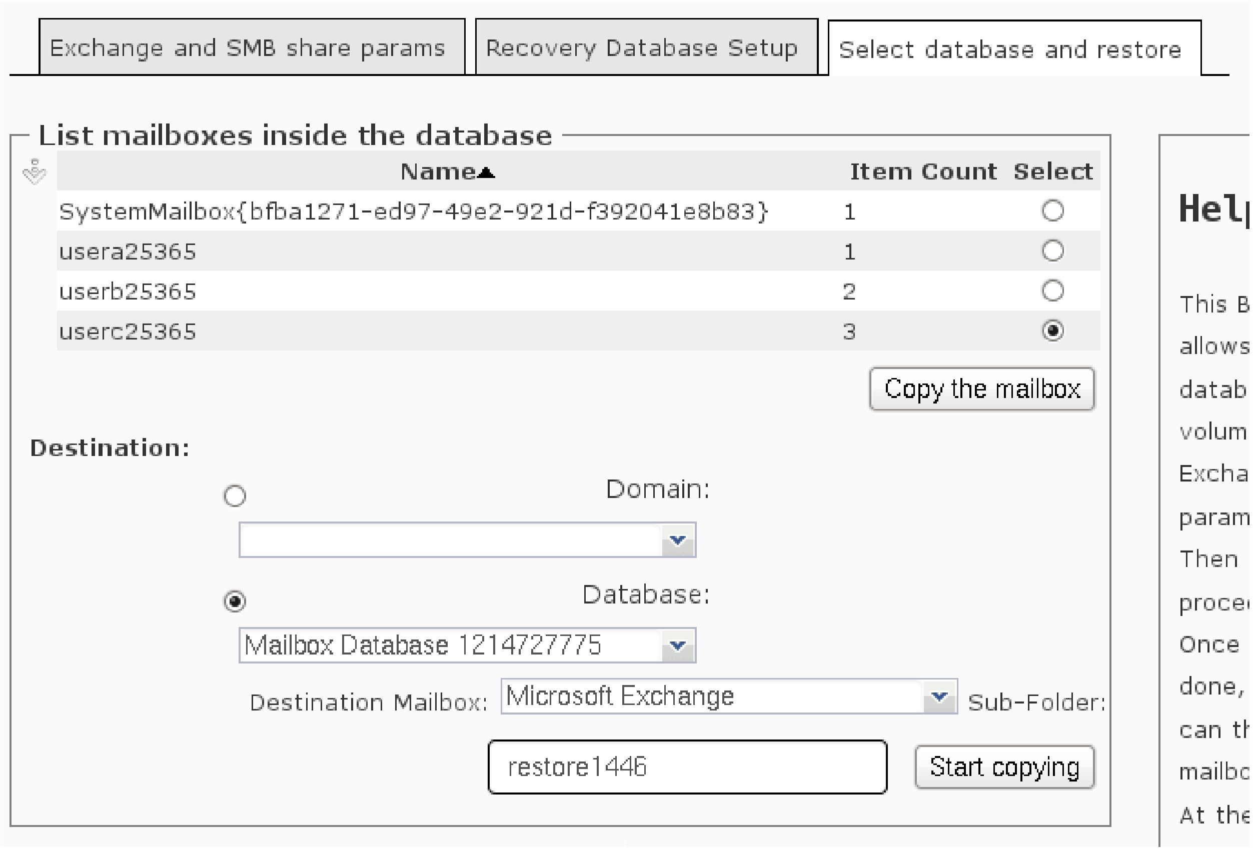
Fig. 10: Recovery Database Setup



Status: ✓

- DB: db25365.edb
- Error: 0
- Command: eseutil /m E:\File\db25365.edb
- Status: Dirty Shutdown

Fig. 11: Details of an Operation



List mailboxes inside the database

Name▲	Item Count	Select
SystemMailbox{bfba1271-ed97-49e2-921d-f392041e8b83}	1	○
usera25365	1	○
userb25365	2	○
userc25365	3	●

Destination:

○ Domain:

● Database: Mailbox Database 1214727775

Destination Mailbox: Microsoft Exchange

Sub-Folder: restore1446

Fig. 12: Mailbox List

session or in the BWeb page.

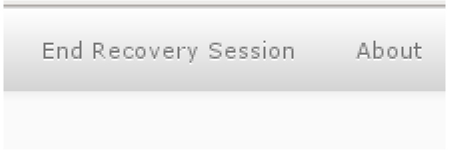


Fig. 13: Dismount Recovery Database

Notes

Cache Directory

To speed up future Exchange Single Item Restore sessions, some files that are generated during a restore session are kept in a cache directory.

```
storage# ls /opt/bacula/working/mount-cache
MyCatalog-2.idx  MyCatalog-5.idx  MyCatalog-8.idx
MyCatalog-4.idx  MyCatalog-6.idx  MyCatalog-9.idx
```

It is possible to remove files in the cache after some time. They are re-generated if needed.

Limitations

- The PST export is not yet implemented in the Bacula Exchange Single Item Restore service.
- The Bacula Exchange Single Item Restore Service is not multi-threaded and can serve only one request at a time.
- The Exchange Single Item Restore feature uses the Bacula BVFS interface to list files and directories. The Bacula BVFS interface is known to have some performance issues with MySQL catalog backends due to internal MySQL limitations with indexes on TEXT columns. For VMWare and Exchange Single Item Restore, it should not impact performance too much (the backup structure is usually quite small) but **Bacula Systems** advises using the PostgreSQL catalog backend for the best experience.
- The Exchange Single Item Restore performance may vary depending on various factors. For example, Bacula will have to read more data if the Volume was written with a large number of concurrent jobs.
- The Exchange Single Item Restore is not compatible with FileDaemon data encryption.
- The password used in the network share between the Bacula Storage Daemon and the Exchange server should not use : &, ’ and “
- The Exchange Single Item Restore is compatible with *file based* devices (cloud, dedup, aligned, file, etc..). Tape devices are not supported.

2.2 Introduction to Bacula

This chapter is mostly intended for readers without any familiarity with Bacula. We expect that many Windows administrators might decide to research how to back up their Exchange, as the required procedures have changed significantly over time.

Note that the functionality described here is all about traditional Exchange instances, *not* about Microsoft 365 Software-as-a-Service email functionality. For information about backing up and restoring that, refer to Microsoft 365 Plugin.

Bacula Enterprise provides commercial, enterprise-level support options and additional features compared to the community-supported open source Bacula software and can be one solution considered by many Exchange administrators. In particular, on top of the "normal" open source advantages provided by the community version, it provides a plugin to back up and recover Microsoft Exchange.

High Level Overview

Bacula, like most other enterprise-ready backup software, consists of several parts which work together through a network:

Director (Dir) is the component that controls all operations.

Storage daemon (SD) is the component that manages the final targets for backed up data.

File daemon (FD) is the *agent* that is run on all systems backed up.

Console refers to one of the various user interface programs. Terminal, graphical, and web interface programs are available.

There are more Bacula components, but we don't need to discuss them here. For now, it's sufficient to understand that Bacula File Daemons (**FD** s) are deployed to all your clients to back up and require minimal configuration and maintenance, Bacula Storage Daemons (**SD** s) require a bit more configuration, but normally few configuration changes over time, and all the important configuration exists in one single location, the Bacula Director program (**Dir**)'s configuration.

Configuration

All **Bacula** configuration is traditionally done in simple text files, using a text editor. This may seem a bit inconvenient to many Windows administrators used to graphical user interfaces, but it allows configuration under almost all circumstances, even from a handheld computer and through a low-bandwidth network connection.

A graphical, web-based user interface to operate and configure Bacula Enterprise is available for **Bacula Systems** customers: BWeb Management Suite. This Web GUI allows to configure and execute backing up of Exchange data, as well as most other specific application data.

Platforms

While **FD****s are available for a wide range of operating systems including Windows, the server components of **Bacula Enterprise** require a Linux or Solaris operating system. This may cause some slight culture shock in Windows-only organizations, but experience shows that for systems dedicated to a single purpose, which can essentially be viewed as a black box, this often poses no serious problems. (Actually, even seemingly Windows-only organizations often run a number of different operating systems, often even Linux, on network infrastructure and storage appliances.)

Bacula Systems still recommends that a **Bacula Enterprise** administrator should have some fundamental understanding of the operating system chosen as the platform, especially if disaster recovery is one of the reasons to create backups.

2.3 Exchange: An Overview

Microsoft Exchange is Microsoft's server application to provide email, groupware, and unified communication services. Echange runs on Microsoft Windows Servers and consists of a number of services. The actual list of features is quite flexible, as not all organizations will need all the features Exchange can offer.

Backup and restore of Exchange 2003 is not supported by the VSS plugins, so is not discussed in this white paper.

Exchange integrates tightly into Microsoft's Active Directory (AD) technology and can not be run outside of an AD domain. Both AD and Exchange can be run in clusters, providing better performance and higher reliability than single instance installations. All instances making up an Exchange cluster can be managed centrally.

Both the configuration and the data of an Exchange installation can be spread over a number of computers, but will usually be kept consistent between the involved Exchange instances.

However, several distinct Exchange sites may be closely linked to each other, even sharing some configuration information. For purposes of backup and recovery, these need to be handled independently while an Exchange cluster itself could be considered only one backup data source.

Microsoft provides tools and procedures to allow deployment and management of Exchange installations by non-trained administrators, but to get the best out of Exchange, a trained, dedicated administrator is required. Also, to design, implement, test, and maintain backup and recovery procedures an experienced Exchange administrator is helpful. However, **Bacula Systems** believes that, with the knowl edge presented in this White Paper, robust backup and recovery scenarios can be implemented even without that level of in-depth knowledge.

As for any application of a certain complexity, the set of data needed to re-create the full functionality of the application requires restoring both the configuration and the actual (user-) data. As Exchange keeps most of its data in a dedicated database, backing up and restoring that data poses the typical challenges of any database backup:

- Consistency of disk files is not guaranteed while the database engine is running normally, because many changes may still be in the computer RAM not yet written to disk;
- ensuring consistency of disk files requires database-specific tools and knowledge;
- and feeding restored data into the database requires database-specific procedures and knowledge.

The **Bacula Enterprise** VSS plugin handles most of those tasks automatically, only requiring minimal Exchange administrator intervention.

Regarding the configuration of an Exchange installation, backing up the configuration is a bit complicated, because the configuration information is stored in the Windows' Registry, and can be managed and distributed by AD. Accordingly, to allow the full recovery of an Exchange installation, one has to ensure availability or recovery of the base system's configuration including the Registry. The **Bacula Enterprise** VSS Plugin can handle that, as well as the AD information. In an environment with several AD servers, this requires Disaster Recovery (DR) procedures to ensure recovery of at least one AD server.

Naturally, the programs making up Exchange are also required to be running when restoring Exchange data. Ensuring this is in many cases best done by preparing for Bare-Metal Recovery of server machines hosting Exchange.

Microsoft Windows Disaster Recovery is described in more detail in the **Bacula Systems** White Paper "Windows Bare Metal Recovery", so we will not go into much detail here.

2.4 The Bacula Enterprise VSS Plugin

Volume Shadow Copy Service (VSS) is the name for Microsoft's snapshot technology, which not only creates snapshots of file systems, but also interfaces to applications to make on-disk data consistent (*freezing, quiescing*) to allow proper backups of application data.

In the simplest case, a vss-aware application's data can simply be backed up from disk, reading from the snapshot, not the original file system. This functionality is part of **Bacula** for a number of years.

However, in order to do online restores (i. e. restores while Windows is running rather than a Bare Metal Recovery) some applications require additional data processing, so using vss alone is not sufficient to back up and restore their data. In particular, understanding of database log files may be required, or, on restore, files that are always locked when Windows is running need to be replaced during a subsequent system boot.

The **Bacula Enterprise** VSS plugin handles the additional data processing needed to be able to do such online restores of many applications, among them Microsoft Exchange.

Please note, in this White Paper, we discuss only the **Bacula Enterprise** version 6.x VSS plugin and not the older Exchange plugin (exchange-fd.dll) that was previously supported by the Bacula project. To avoid all possible interference with the VSS plugin, we recommend that the older exchange-fd.dll should not exist in the plugin directory defined in the **FD** 's configuration file. If it does, Exchange backups and restores could possibly fail.

The VSS plugin needs to be explicitly included in a FileSet used for backups (the plugin takes care of files that would be backed up, making sure no unnecessary or even harmful files get stored).

In addition, the VSS plugin will also create and store some data that needs to be restored or is needed to control the restore process prior to the actual data files being restored. This data is not stored on the backup media, but in **Bacula**'s catalog. This allows restoring data in a different order than it is written, but it requires that all **the Bacula components: Bacula Director program, Bacula Storage Daemon and Bacula File Daemon must be the exact same versions** to ensure that all components know how to handle that additional job data. It also means that you can not restore Exchange data with only the volumes available, after losing the catalog.[1]

As backing up through the vss plugin uses the functionality provided in Microsoft application specific writers, some limitations exist when using this approach to backup. Essentially, **Bacula** can not do things the Microsoft VSS application writers do not provide.

Important Points

For Exchange, some important points are:

- In Exchange 2005, the Recovery Storage Groups (RSGs) exist, so in principle, the Exchange Administrator can enable restore to the RSG, and whe Bacula does a full restore it will go into the RSG rather than restore to the active Exchange database. Once the RSG is restored, the Windows Exchange Administrator can do individual mailbox restores. We have not yet tested this.
- In Exchange 2010 Microsoft eliminated the Recovery Storage Group code that was in previous versions of ex. The Recovery Storage Group allowed the Exchange Administrator to automatically direct a restore to the Recovery Storage Group. Thus other techniques such as use of the Recovery Database are necessary for individual mail box recovery.
- Exchange 2010 permits a single Recovery Database (RDB), which works somewhat similarly to the Recovery Storage Groups in prior versions. The difference is to restore to a Recovery Database, the user must explicitly specify the restore location. This feature is not implemented in the initial version 6.0 release of the VSS plugin but available as of **Bacula Enterprise** version 6.0.4.

[1] Regularly saving the catalog database dumps is a practice strongly recommended by anyway, so this should not be an issue in any production system.

- The Exchange VSS plugin is supplied in a separate **Bacula Enterprise** VSS plugin installer that must be executed after installing the File daemon.
- The Exchange VSS plugin described in this white paper may not work with the old **exchange-fd.dll** plugin installed in the PluginDirectory. Please make sure it is not present. How to check which plugins are loaded is described in the general VSS Plugin White paper provided by **Bacula Systems**.
- We have not fully tested the Differential backup feature, and recommend that you not use it, since Microsoft's concept of Differential backups is different from Bacula's. Doing a Differential backup after several Incremental backups, could possibly result in lost log files and thusunrecoverable databases.
- We recommend that you use Exchange 2010 SP2 since it corrects many of the problems in previous Exchange versions.
- When you do a restore, you must always do a full restore (item 5 on the restore prompt) that includes the Full backup as well as all the Incremental backups. If you attempt to select Jobs to restore by individual JobIds, the restore will fail. This is because in order for the Microsoft VSS writer to work correctly, it needs all the information about all the backup jobs that were made.
- If you unmount (dismount in Microsoft terminology) a database, its time and date stamp will be updated, and thus it and its log files will be backed up on the next Bacula backup. The same is true if you reboot your system. Backing up active Exchange data files without the VSS plugin and not using Accurate mode will cause incomplete data to be stored.
- Bacula's Accurate mode **must** be enabled in your Job resource. If it is not, you will get a warning message, but much worse, the files being backed up will not be the set of files that needs to be backed up (files may be missed or erroneously be backed up). Bacula needs Accurate mode enabled in order to optimize the backup.
- Do not use Bacula's **Accurate** mode to check MD5 (accurate flag '5') or SHA1 (accurate flag '1') signatures as they will produce a large number of error messages during backup operations. In fact, the default criteria for Accurate Mode, "mcs", are sufficient. Using a Signature directive with an MD5 or SHA1 is OK.
- We do not support and do not guarantee that the VSS plugin can properly backup and restore Exchange in a clustered environment. Databases that are part of A Database Availability Group, however, can usually be backed up and restored correctly.
- Due to the tight coupling between Exchange and Active Directory, we do not support backup of Exchange and restore to a machine that has a different Active Directory environment. It should work if Active Directory on both machines is identical, i. e. the machines are members of the same AD domain.

Fortunately, these are usually not serious restrictions, as Exchange by default provides settings to make partial restores unnecessary, so actually restoring Exchange data would only be needed in case of a disaster recovery, never during regular operations. This also implies that Exchange should always be run in a clustered environment (Database Availability Group or DAG), as Microsoft recommends.

2.5 Plugin Options

- **index** Creates additional internal parameters needed for Single Item Restore
- **cinclude=<glob>** Specifies the names of mailbox databases to backup. It is possible to specify the `cinclude` parameter multiple times on the plugin command line.
- **cexclude=<glob>** Specifies the names of mailbox databases to exclude from the backup. It is possible to specify the `cexclude` parameter multiple times on the plugin command line.

2.6 Backup Scenarios

Due to Microsoft Exchange's feature to not delete mailbox items, but keep them hidden from regular users when deleted, recovery operations to get individual items back should happen rarely, if at all.[2] Thus, backing up Exchange is mostly useful as part of a dr plan. Accordingly, **Bacula Systems**'s procedures focus on complete backup of Exchange and full restores to the original location.

Recovery operations are similarly focused on (Exchange-specific) DR procedures. Single object restores are discussed but should not be the main goal of deploying a backup and recovery tool for Exchange.

For the purposes of this White Paper, DR does not imply a full Bare Metal Restore of the underlying Operating System and all applications and data on it, but refers to the data `only`. In other words, we are concerned about Application-level Disaster Recovery. This has some consequences on the presented scenarios:

- We assume the version of remains the same, i. e. the backed up data was written by an exact identical version of Exchange as the one it is restored to,
- the operating version is the same for the backup source and backup target,
- and the target locations for the Exchange databases remain unchanged or are part of the same Exchange deployment.

To clarify the outlined backup and recovery scenario, the following table may be helpful:

Table 1: Backup and Recovery Scenario

Situation	Next step
Complete Mailbox Server lost	Set up Windows and Exchange with Mailbox Server role, or do Bare-Metal Recovery
All Mailbox Databases lost	Restore complete sequence of all backups to original location
One Mailbox Database lost	Restore complete sequence of this individual Mailbox Database to original location
One Mailbox Database corrupt	Restore complete sequence of this individual Mailbox Database to an alternate location and proceed as necessary
Single Mailbox or single Mailbox Item required	Restore to Recovery Database and pick out desired Items

Minimal Exchange

In our examples, we will initially use a single server running in an AD domain. The Exchange instance uses only one Mailbox storage database which is located on a disk drive with the drive letter `D:`, but the main ex installation is located on drive `C:`. Accordingly, to create a useful backup of the Exchange data, the disk drives lettered `C:` and `D:` need to be included in the backup[3]. The resulting FileSet is shown below.

```
File Set {
  Name = Exchange-CD
  Include {
    Options {
      Verify = pnugsi1
      Signature = SHA1
    }
    File = c:/backmeup
    File = d:/backmeup
```

(continues on next page)

[2] The Retention policy is defined in Exchange Management Console and can be applied per Database or per Mailbox. Read more about it at technet.microsoft.com/en-us/library/dd297955.aspx

[3] Starting with version 12.5, specifying the volumes is not mandatory anymore

(continued from previous page)

```
    Plugin = "vss:/@EXCHANGE/"
  }
}
```

Note the dummy files included on each drive where data is located.

To allow the VSS plugin of **Bacula Enterprise** to work correctly, backup jobs must be done in accurate mode. Accordingly, the job definition we use in our examples includes that option; the job definition we use is shown below.

```
Job {
  Name = Exchange
  Type = Backup
  Level = Incremental
  Accurate = Yes
  File Set = Exchange-CD
  Client = wsb-exch10-fd
  Storage = File
  Messages = Standard
  Pool = Tier1
  Priority = 10
  Write Bootstrap = "/var/lib/bacula/%n.bsr"
}
```

Backing Up

To create a backup of a data set, one needs to specify a FileSet to use the Exchange functionality of **Bacula Enterprise**'s VSS plugin as outlined in chapter *The Bacula Enterprise VSS Plugin*. One essential consideration is that at least one file from each drive that is used by Exchange **has** to be included explicitly.

On the Exchange side, databases that are to be backed up with bsee's VSS plugin must be set to **not do circular logging**. This has to be done for each database and can be achieved with the Database Properties panel in Exchange Management Console (see figure *Mailbox Database Maintenance Properties need to Allow Overwriting*, the last line of the checkboxes, below the arrow) or through ps.

Backup Levels

Backups with **Bacula** can be of three levels: Full, Differential, and Incremental. A Full backup is simple to understand; it just backs up everything included in its FileSet then allows Exchange to remove the log files that are no longer needed. A Differential level backup backs up everything changed since the latest Full level backup, but does not trigger log file removal. Incremental backup takes everything since the last backup job run and allow Exchange to remove any log files that are no longer needed.

Transaction Log Truncation

A transaction log as written by many database engines stores all database transactions done so they can be replayed in a recovery situation. To ensure that disk space used by those logs can be reclaimed, those logs should be truncated from time to time. When using the VSS Exchange plugin, log truncation is automatically done by the Windows Exchange writer when a Full or an Incremental backup succeeds.

Single Database Backup

A single database backup is possible, by excluding other databases from the file set and including the wanted database. Code below presents a fileset structure where 4 databases are present on volume G[4]. MDB03 will be backuped while MDB01, MDB02 and MDB04 will be excluded.

```
File = "G:/backmeup.txt" ## volume G: will be part of the snapshot

Plugin = "vss:/@EXCHANGE/
   cexclude=*/*/*/MDB01/*   ## to cexclude MDB01/Log or MDB01/File
   cexclude=*/*/*/MDB01      ## to cexclude MDB01 folder
   cexclude=*/*/*/*/MDB01/* ## extra level to cexclude MDB01/Log or MDB01/File when in a␣
↪Replica
   cexclude=*/*/*/*/MDB01   ## extra level to cexclude MDB01 folder when it's a Replica

## same for DB02
   cexclude=*/*/*/MDB02/* cexclude=*/*/*/MDB02 cexclude=*/*/*/*/MDB02/* cexclude=*/*/*/*/
↪MDB02
## same for DB04
   cexclude=*/*/*/MDB04/* cexclude=*/*/*/MDB04 cexclude=*/*/*/*/MDB04/* cexclude=*/*/*/*/
↪MDB04

## cinclude DB03 with the same logic
   cinclude=*/*/*/MDB03/* cinclude=*/*/*/MDB03 cinclude=*/*/*/*/MDB03/* cinclude=*/*/*/*/
↪MDB03
```

However, a VSS snapshot is always global to a single volume. In this example all databases present on volume G: will get their transaction logs truncated, although some are not backuped. To avoid this, consider separating the databases on different volumes or backup all databases present on the volume in 1 single job.

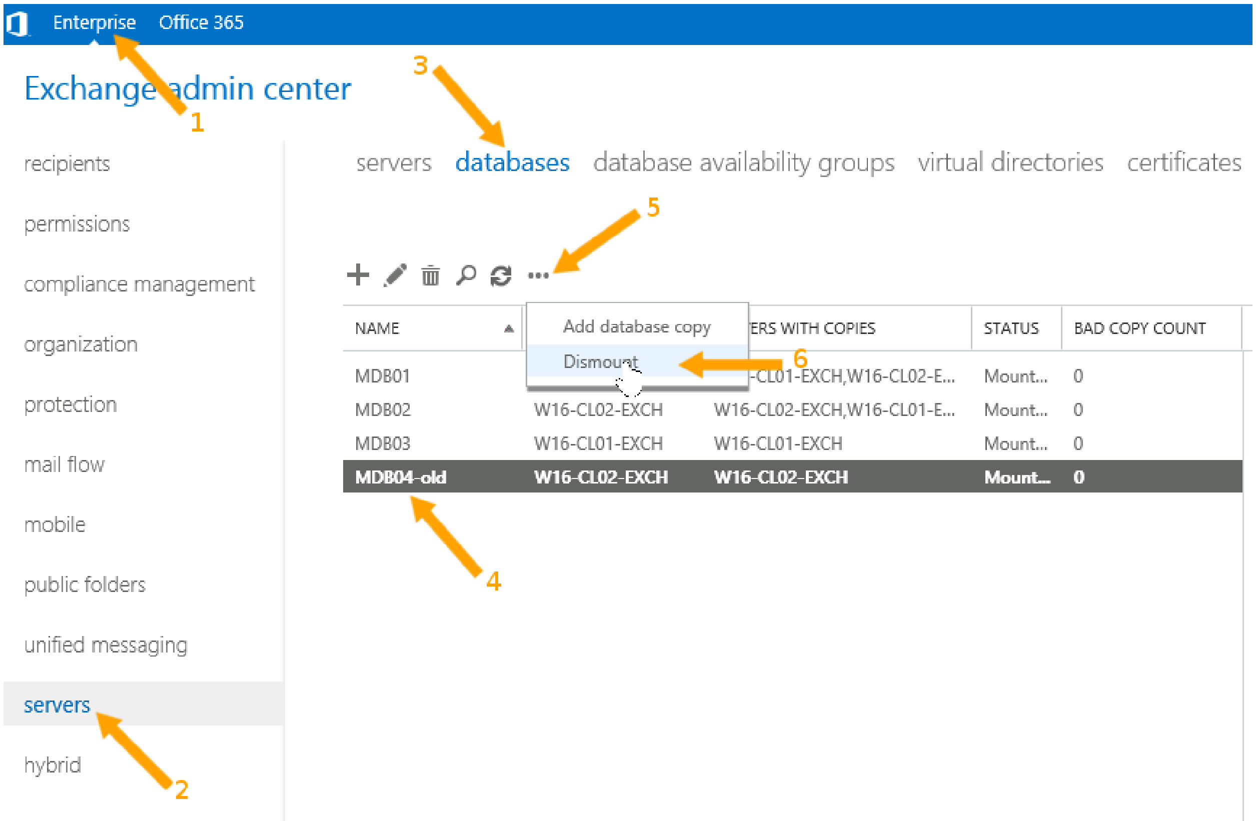
Differential Backups

We have not sufficiently tested Differential backups with Exchange, and suspect that due to a difference in concept of Differential backups between Exchange and Bacula, using Differential backups may result in lost log files. Thus we recommend that you do not use Differential backups with the Exchange plugin.

[4] Starting with version 12.5, specifying the volumes is not mandatory anymore

Recovery

The Mailbox Database needs to be unmounted and marked to allow overwriting to be restored. An example on how to unmount a Mailbox Database is shown below. In `Exchange Administrative Center`, navigate to the database to restore, select it, click on `...` and then `Dismount`:



NAME		ERS WITH COPIES	STATUS	BAD COPY COUNT
MDB01		-CL01-EXCH,W16-CL02-E...	Mount...	0
MDB02	W16-CL02-EXCH	W16-CL02-EXCH,W16-CL01-E...	Mount...	0
MDB03	W16-CL01-EXCH	W16-CL01-EXCH	Mount...	0
MDB04-old	W16-CL02-EXCH	W16-CL02-EXCH	Mount...	0

Fig. 14: Preparing an Exchange Database for Restore with the `Exchange Administrative Center`

Then edit the Database Mailbox Maintenance properties by clicking on the pencil icon and checking the box `This Database can be Overwritten by a Restore`:

Alternatively, these steps can also be done with **PowerShell** commands. Please refer to the documentation available with the Microsoft Exchange Management Shell.

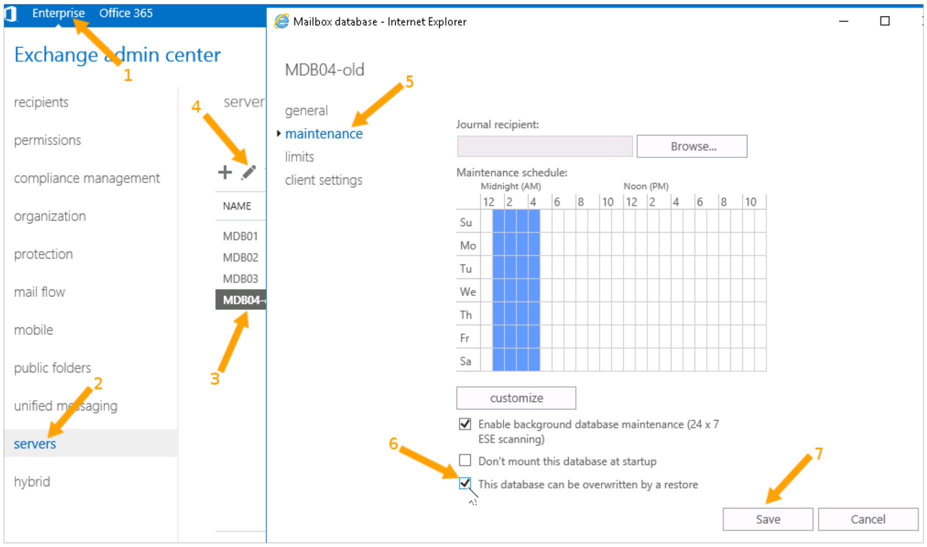
If the database to be restored exists, for example when testing, or if parts of its files still exist (which might be the case after a file system fault) it is best to remove the remaining parts before restoring.

To do this to just safely rename the whole database directory with Window's shell, `cmd`. The path can be found through `Exchange Administrative Center`, from the Database Properties panel, as shown in figure *Database Path in Exchange Administrative Center*, or with the **PowerShell**-based Exchange Management Shell. A way to do this, and the result, is shown in figure *Database Path in Exchange Management Shell*.

The full database path and name cannot be edited but can be selected and copied.

The path name is shown completely – remember that you need the whole directory, not only the `.edb` file.

The fault that can be observed when restoring with database files remaining is caused by out-of-sequence log files that are found by the Exchange-specific database recovery and is reported in the Job Report with lines like these:



MDB04-old

Journal recipient:

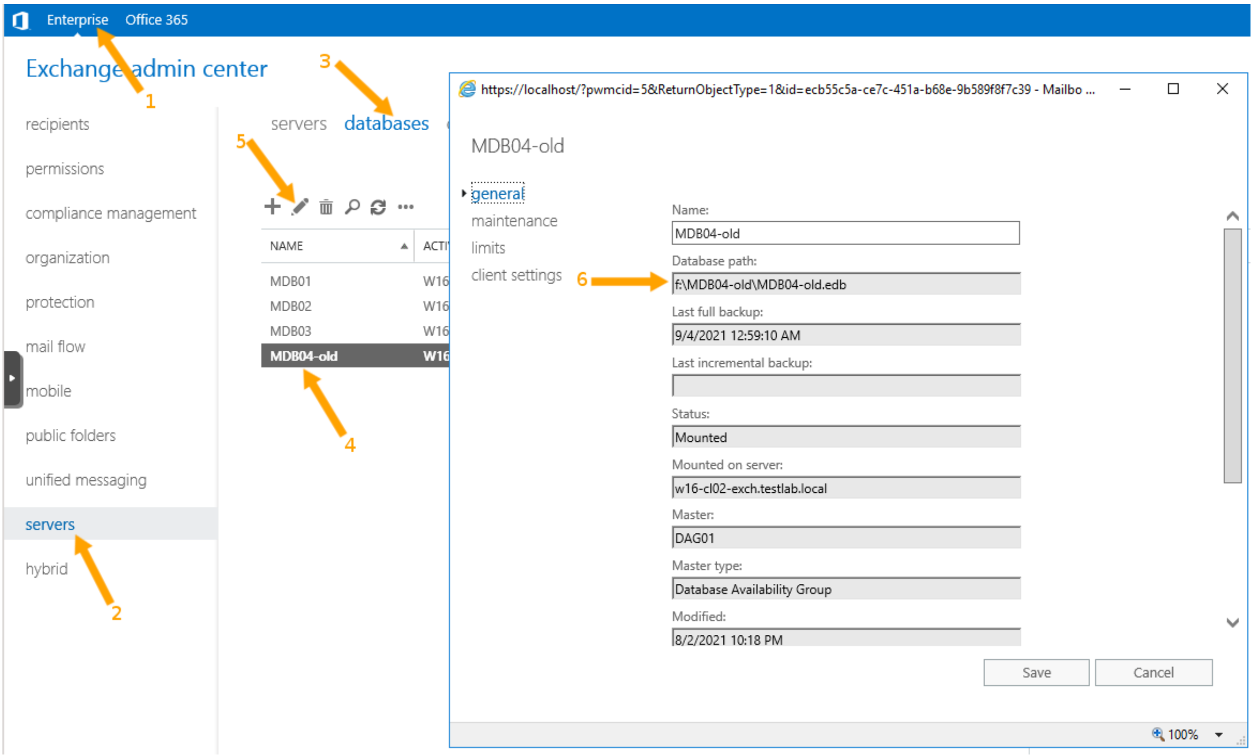
Maintenance schedule:

☑ Enable background database maintenance (24 x 7 ESE scanning)

□ Don't mount this database at startup

☑ This database can be overwritten by a restore

Fig. 15: Mailbox Database Maintenance Properties need to Allow Overwriting



MDB04-old

Name: MDB04-old

Database path: f:\MDB04-old\MDB04-old.edb

Last full backup: 9/4/2021 12:59:10 AM

Last incremental backup:

Status: Mounted

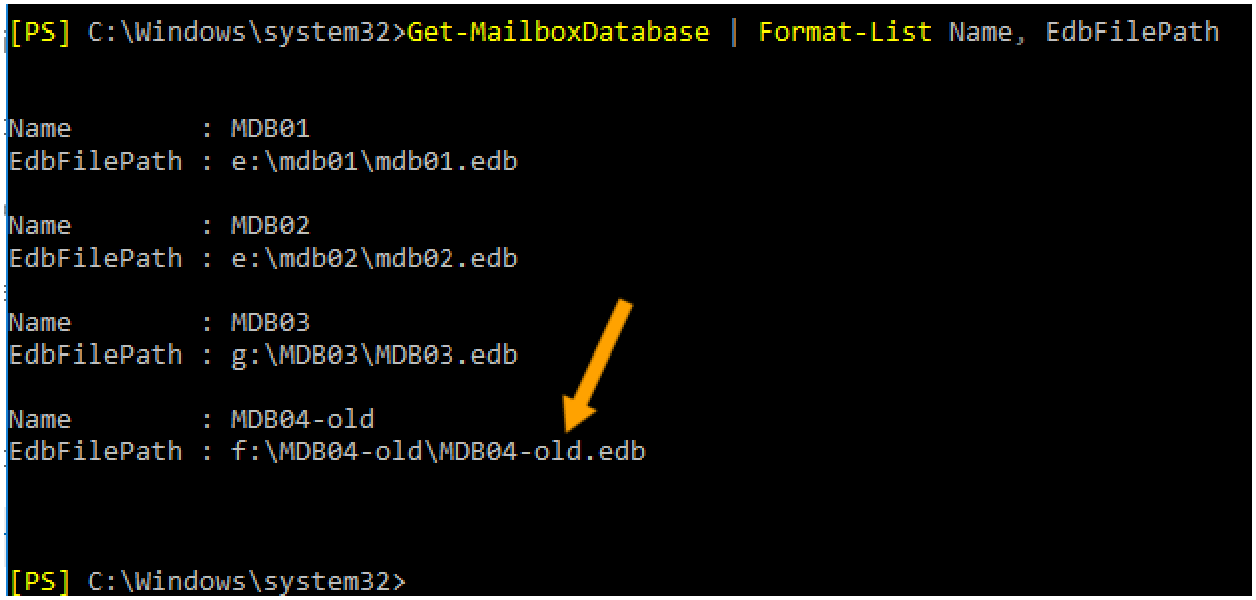
Mounted on server: w16-cl02-exch.testlab.local

Master: DAG01

Master type: Database Availability Group

Modified: 8/2/2021 10:18 PM

Fig. 16: Database Path in `Exchange Administrative Center`



```
[PS] C:\Windows\system32>Get-MailboxDatabase | Format-List Name, EdbFilePath


Name        : MDB01
EdbFilePath : e:\mdb01\mdb01.edb

Name        : MDB02
EdbFilePath : e:\mdb02\mdb02.edb

Name        : MDB03
EdbFilePath : g:\MDB03\MDB03.edb

Name        : MDB04-old
EdbFilePath : f:\MDB04-old\MDB04-old.edb



[PS] C:\Windows\system32>
```

Fig. 17: Database Path in `Exchange Management Shell`

```
Exchange VSS Writer failed restoring a backup with error code -515
when performing an integrity-check of the log files to be used for
database recovery after restore for 'Mailbox Database 1568811476'.
```

If you encounter those problems the safest way to proceed is to redo the whole restore process, this time removing the remaining database files before the actual restore job starts.

Restore procedure with Bat

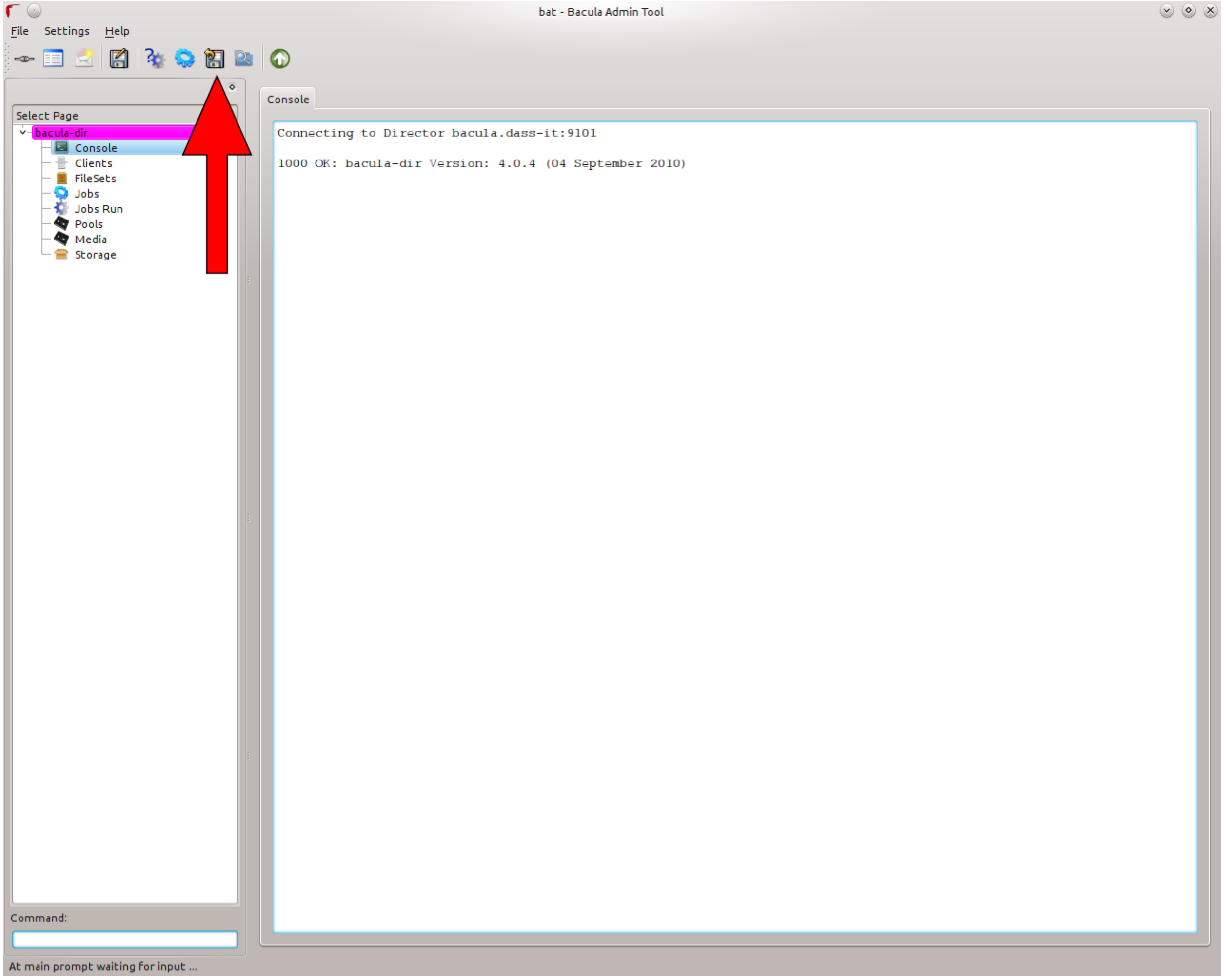
In `BAT` press the restore Button (see figure *Starting the Restore Procedure*).

Select the corresponding job for your exchange server and choose the "@EXCHANGE" folder presented in the directory overview (see figure *Restore Selection*)

It is important to remove the directive in the "Where" line as shown in figure *Restore Where*, otherwise the database may be restored to an alternate location, which is not supported without additional preparation. This is explained in chapter *Restoring to an Alternative Location*.

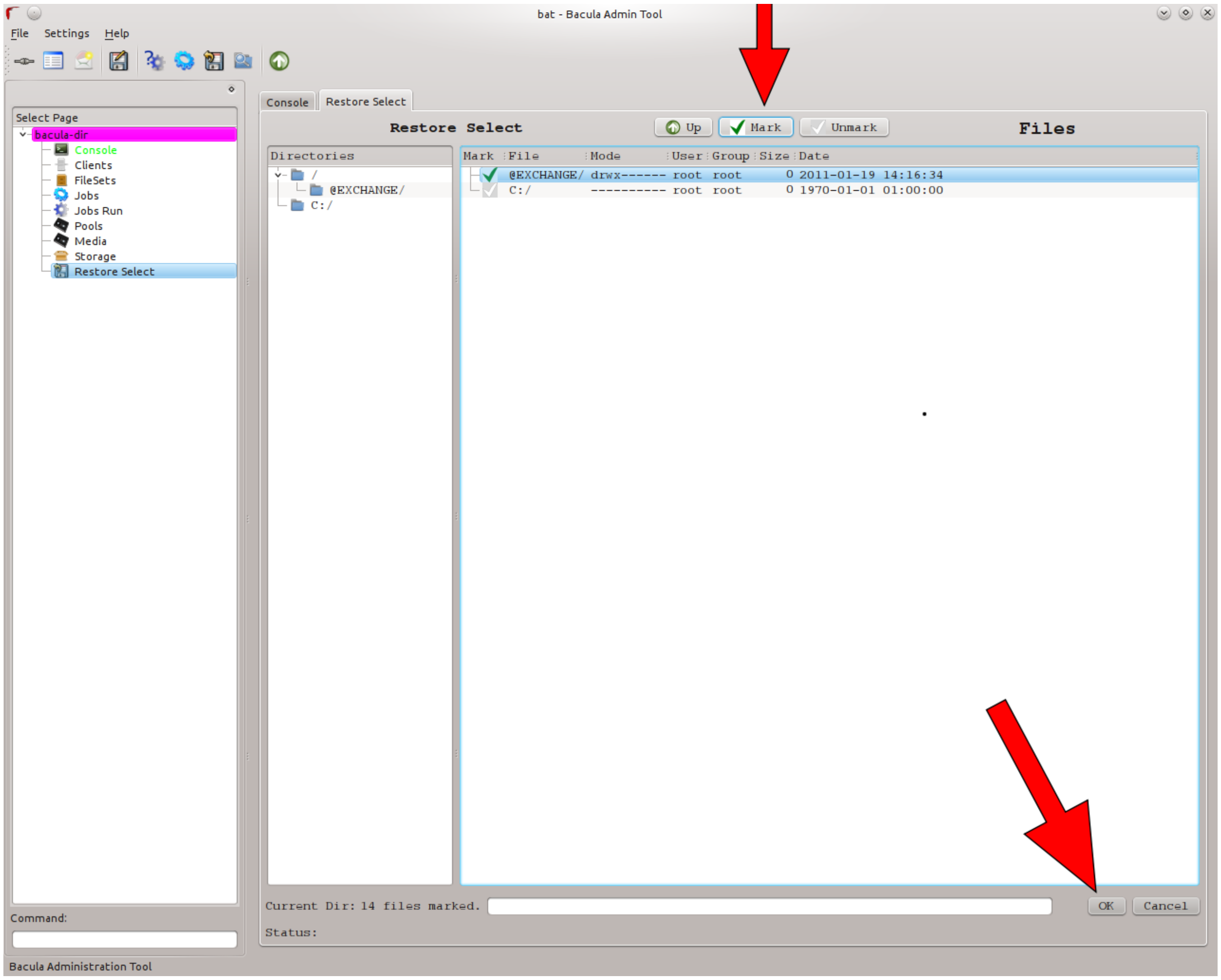
After the actual restore (which could also be initiated from a **Bacula** console) is finished the Exchange machine may need to be rebooted, during which some of the actual database files will be moved to their final locations. If the reboot is needed, Bacula will print a message to that effect in the Job report. After the reboot, or after the restore is done, the administrator will need to mount the restored database(s), which is done similar to dismounting, and the server will be functional again.

For reasons of safety, **Bacula Systems** recommends to verify that the setting to allow the database to be overwritten is turned off when mounting it after the server's restart.



```
Connecting to Director bacula.dass-it:9101

1000 OK: bacula-dir Version: 4.0.4 (04 September 2010)
```

Fig. 18: Starting the Restore Procedure

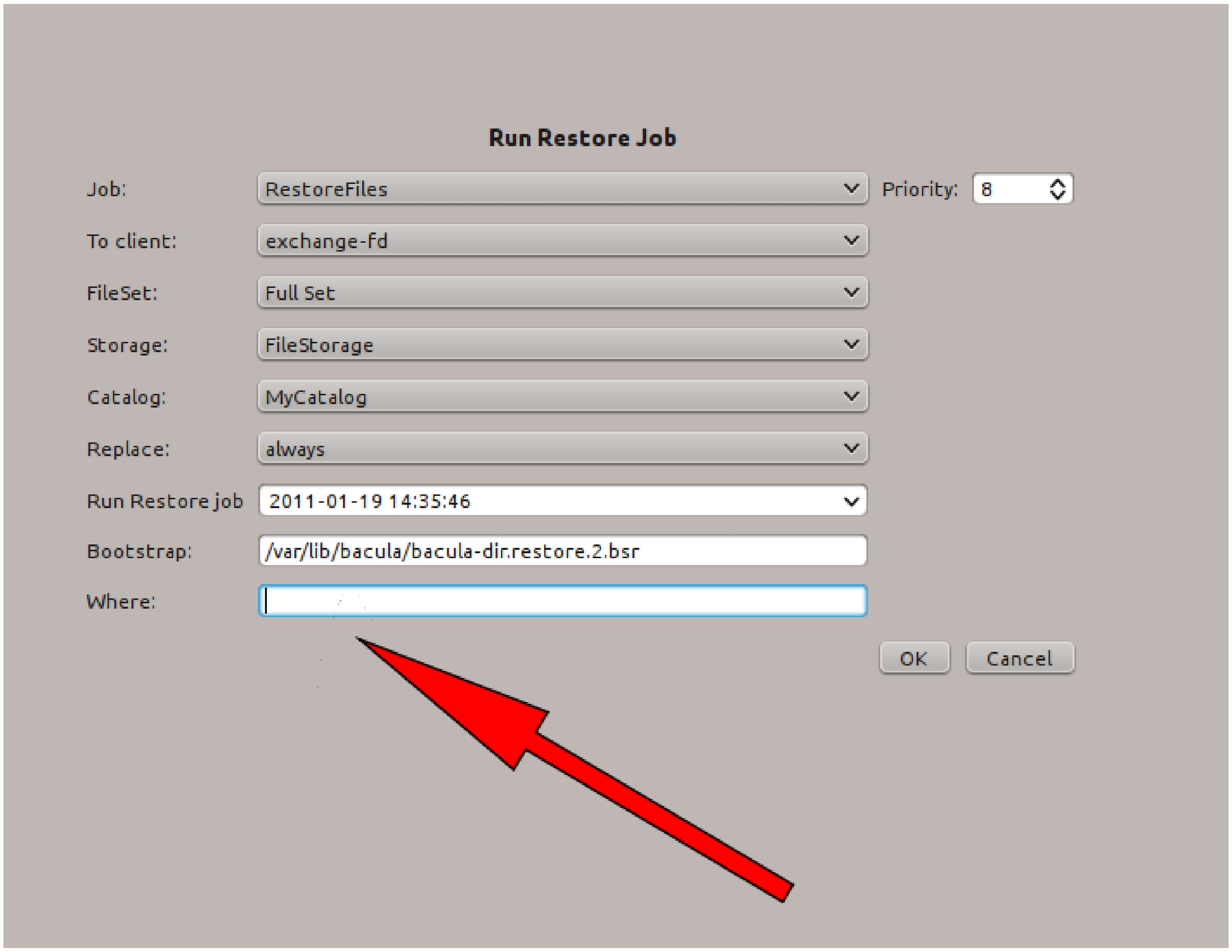


Mark	File	Mode	User	Group	Size	Date
✓	@EXCHANGE/	drwx------	root	root	0	2011-01-19 14:16:34
✓	C:/	----------	root	root	0	1970-01-01 01:00:00

Current Dir: 14 files marked.

Status:

Fig. 19: Restore Selection



Run Restore Job

Job: RestoreFiles

Priority: 8

To client: exchange-fd

FileSet: Full Set

Storage: FileStorage

Catalog: MyCatalog

Replace: always

Run Restore job 2011-01-19 14:35:46

Bootstrap: /var/lib/bacula/bacula-dir.restore.2.bsr

Where:

Fig. 20: Restore Where

Restoring a Single Database

By default, if you have multiple Exchange databases, they will all be restored when following the procedures given above. If you wish to restore only a single database, it is possible, but during the restore file selection procedure instead of marking the “@EXCHANGE” folder presented in the directory overview, you must descend several levels – usually four – into the “@EXCHANGE” directory, until you find the names of all the databases that you have. At that point you may mark only the database or databases that you want restored. An example of this, where only one database is selected for restore is given below.

```
  $ pwd
  cwd is: /@EXCHANGE/Microsoft Exchange Writer/Microsoft Exchange Server/Microsoft␣
↪Information Store/WSB-EXCH10/
  $ ls
  Mailbox A/
  Mailbox Database 1568811476/
  mark "Mailbox Database 1568811476"
  16 files marked.
  $
```

Database Availability Groups

Database Availability Groups (DAG) is one of the main new features of Exchange 10. The DAG provides automatic, database-level recovery from a database, server, or network outage. In order to use the full potential of the DAG groups, two or more Exchange Servers are required. DAG is based on continuous replication, enabling the Exchange environment to offer high availability and site resilience. In a DAG the recovery mechanism is done automatically. Hence if a mailbox in the Exchange cluster has a failure another Exchange Server (within the cluster) will automatically fix the database failure. In a scenario of a Database Availability Group, the database is replicated to all involved Servers. In this situation, the Backup of only one instance is necessary because all databases are kept synchronized.

Other Ways to Cluster Exchange

There are several other ways to cluster environments. In order to give a brief but sophisticated overview it will be focused on mainly two scenarios in order to cover a simple and a more advanced solution. The first solution is the easiest one. To cluster an Exchange Server a shared storage is used and a Heartbeat in order to monitor the two nodes. If one of the Exchange Servers now realizes a fail-over situation, the running node takes over all connections and work in order to still provide availability. In this scenario no separation between the database and server is implied.

The second scenario involves in total four Exchange Servers where three actives nodes are productive and the fourth is set in a passive mode in order to cover the fail-over case. In the background redundant, shared mass storage is used to store logs and database files. With this environment an effective usage of the hardware is possible as well as a high level of security in terms of availability and consistency of the database is provided. It has to be mentioned that, if this environment is used, the database files have to be saved from just one Exchange server, since data is stored on a central, shared storage device. The other Exchange server instances will automatically pick up database changes.

Restoring to an Alternative Location

Note: This procedure is available for Exchange 2010 and later, not with older versions.

In order to do restores of single mailboxes or single mailbox items, it is necessary that the administrator restores to a secondary database and then moves the interesting items to their final destination.

For this purpose, Microsoft provides special *Recovery Databases* which are used to (temporarily) store the recovered objects before they are moved to their final location. These Recovery Databases are managed quite differently than regular Mailbox Databases, as they are only intended as a temporary storage during recovery operations. See http://technet.microsoft.com/en-us/library/dd876954.aspx for details.

Creating and operating on Recovery Databases is done exclusively through Exchange Management Shell; **Bacula Systems** provides scripts making these tasks easier.

The high-level overview of the steps required to restore individual items looks like this:

1. If necessary, create a Recovery Database.
2. Unmount this database and enable restores to it (this is the same procedure as described in *Recovery*)
3. Restore data (usually restoring an individual database, as in section *Restoring a Single Database*). Use a `where=` setting to direct the restored data to the Recovery Database.
4. Move the items of interest to their final location using Microsoft Exchange Management Shell.
5. After checking the success of the restore, remove the Recovery Database.

All those steps can be done through Powershell commands with Microsoft Exchange Management Shell, and some of them can only be done this way.

Most of those steps have to be done on a Windows machine and need to work with the ex server data will be restored to. For stability reasons, **Bacula Systems** recommends that you run the actual restore to the ex server that physically hosts the recovery mailbox, and in this case it's most convenient to work on that server itself.

A typical restore session done the way **Bacula Systems** recommends accordingly looks like this:

1. Log on to a ex mailbox database server with credentials allowing to manage Microsoft Exchange.
2. Open the Microsoft Exchange Management Shell. In the start menu, this will look similar to figure *Microsoft Exchange Management Shell in the Start Menu*.
3. A command windows opens which will look similar to the one in figure *Microsoft Exchange Management Shell Startup Window*. You should know the Powershell essentials if you're an Administrator; in-depth information can be found on Microsoft's web site, for example at http://technet.microsoft.com/de-de/library/bb123778
4. Use appropriate commands to set up a Recovery Database and enable restores to it. **Bacula Systems** provides an example script which is also attached to this white paper in its pdf version. This script should, if stored in `C:\Users\Administrator\Documents`, be sourced with the``Powershell`` dot command, i. e. `."\C:\Users\Administrator\Documents\file.ps1"`. An example can be seen in figure *Calling the Preparation Script*, and sample output is shown in figure *Preparation Script Output*. In *Script to Prepare Recovery Database* we provide the listing of this script. The commands that are essential are `New-MailboxDatabase` and `Set-MailboxDatabase`.
5. Do the actual restore with **Bacula Enterprise**
    1. Set up the restore job as usual, using any convenient console. Make sure you restore a full sequence of backups beginning with a full level backup, and all jobs based on this one up to the point in time required.
    2. Select the files to restore – normally, a specific database will be selected for restore. See above in **ch:singledbrestore** for details.

   3. Make sure no file name mangling is done by adding a prefix or suffix.

   4. Set the restore location to the Recovery Database name. Do not use the path but the plain name!

   5. Let the job execute.

   An example restore session using `bconsole` is shown in figure *Restore Job Relocating a Single Mailbox Database*.

6. Move the relevant items from the Recovery Database to their final location. The Recovery Database has to be mounted for that purpose. The following steps can be used:

   1. Execute `Mount-Database Bacula EnterpriseRecovery` (substitute the Recovery Database name if necessary) in .

   2. In , you can use the following command to get a listing of all the mailboxes that exist in the Recovery Database (again, substitute the Recovery Database Name if necessary):

      ```
      Get-MailboxStatistics -Database Bacula EnterpriseRecovery | Format-List␣
      ↪DisplayName
      ```

   3. The following Powershell command in restores a complete mailbox from the Recovery Database to the regular user's database (which must exist) into a dedicated folder:

      ```
      Restore-Mailbox -Identity "DisplayName" -RecoveryDatabase Bacula␣
      ↪EnterpriseRecovery -TargetFolder "Recovered 2010-07-10" -RecoveryMailbox
      ↪"DisplayName"
      ```

      The `DisplayName` must be the displayed name of an existing mailbox – the source, i. e. the mailbox inside the Recovery Database is identified by the `RecoveryMailbox` parameter.

   4. The user can now arrange (or delete) recovered items as needed. It should be noted that items are restored without any permissions set, so that the user may have to give herself permissions to manage the restored folders.

      An example of the process and results of this recovery procedure is given in figures *Restoring a Mailbox with Microsoft Exchange Management Shell* and *Restored Mailbox and Folder Permissions*.

7. To clean up after all data is moved to its final location, unmount the Recovery Database, remove it from ex, and delete the data files on disk. The exs commands `Dismount-Database`, `Remove-MailboxDatabase` and `Remove-Item -Recurse` are suitable to achieve that.

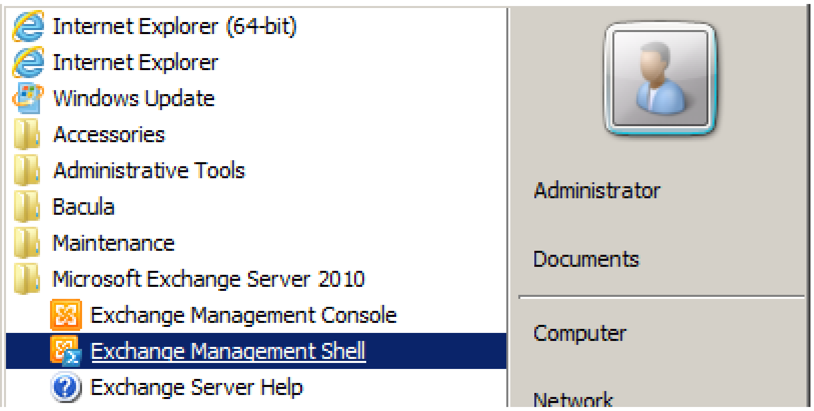
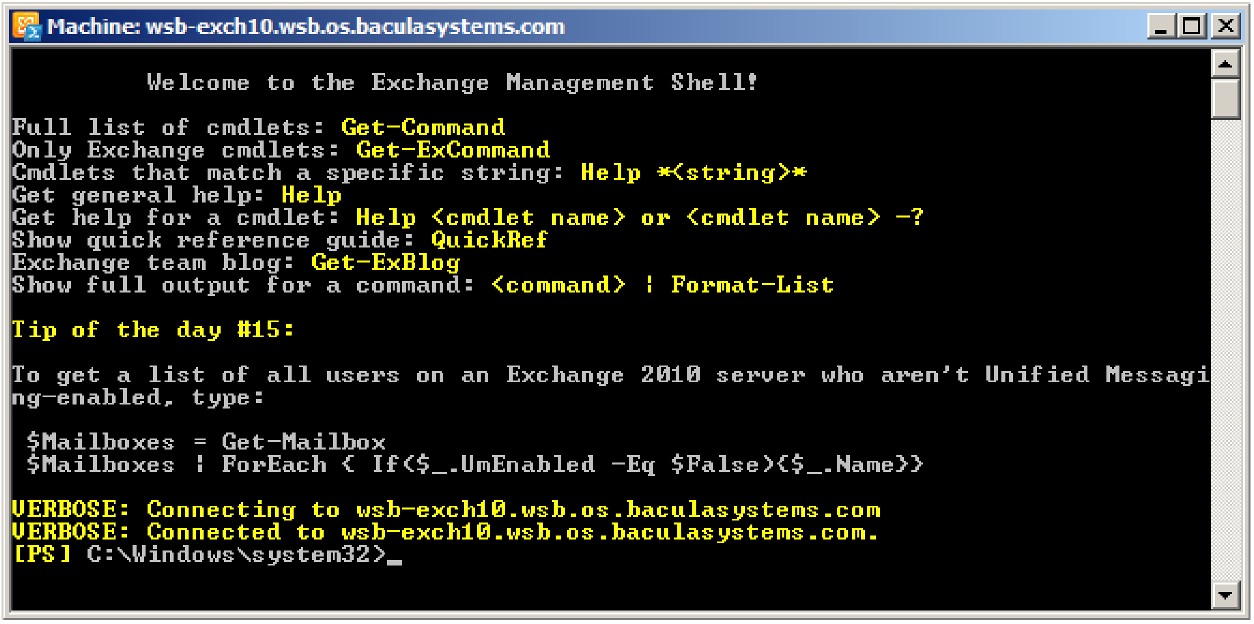
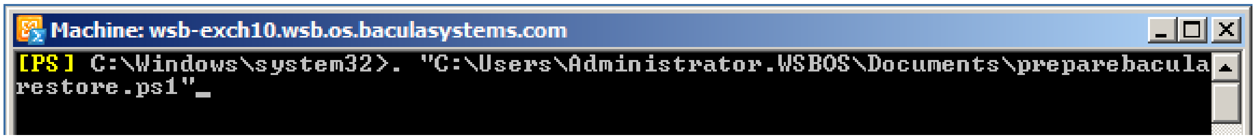


Fig. 21: Microsoft Exchange Management Shell in the Start Menu



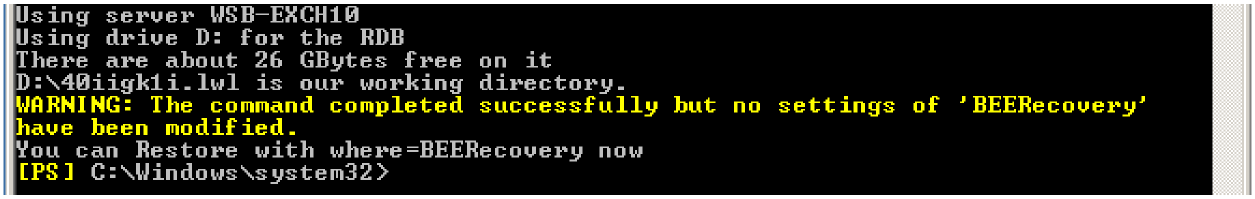
```
         Welcome to the Exchange Management Shell!

Full list of cmdlets: Get-Command
Only Exchange cmdlets: Get-ExCommand
Cmdlets that match a specific string: Help *<string>*
Get general help: Help
Get help for a cmdlet: Help <cmdlet name> or <cmdlet name> -?
Show quick reference guide: QuickRef
Exchange team blog: Get-ExBlog
Show full output for a command: <command> | Format-List

Tip of the day #15:

To get a list of all users on an Exchange 2010 server who aren't Unified Messagi
ng-enabled, type:

 $Mailboxes = Get-Mailbox
 $Mailboxes | ForEach { If($_.UmEnabled -Eq $False){$_.Name}}

VERBOSE: Connecting to wsb-exch10.wsb.os.baculasystems.com
VERBOSE: Connected to wsb-exch10.wsb.os.baculasystems.com.
[PS] C:\Windows\system32>_
```

Fig. 22: Microsoft Exchange Management Shell Startup Window



```
[PS] C:\Windows\system32>. "C:\Users\Administrator.WSBOS\Documents\preparebacula
restore.ps1"_
```

Fig. 23: Calling the Preparation Script



```
Using server WSB-EXCH10
Using drive D: for the RDB
There are about 26 GBytes free on it
D:\40iigk1i.lwl is our working directory.
WARNING: The command completed successfully but no settings of 'BEERecovery'
have been modified.
You can Restore with where=BEERecovery now
[PS] C:\Windows\system32>
```

Fig. 24: Preparation Script Output

Listing 33: Restore Job Relocating a Single Mailbox Database

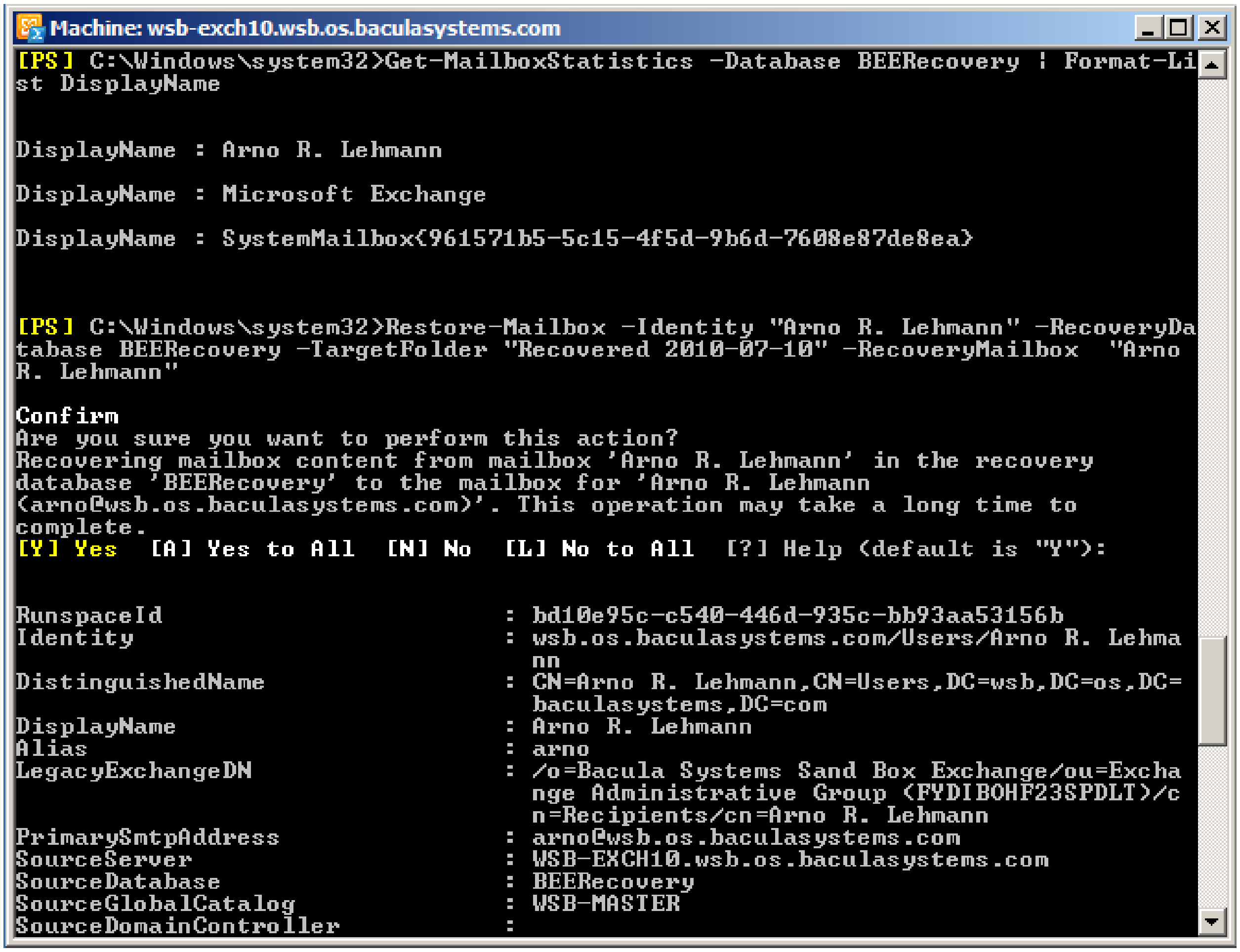
```
   *restore client=wsb-exch10-fd before="2012-07-09 23:59:00" where="Bacula␣
↪EnterpriseRecovery"
    restoreclient=wsb-exch10-fd

   First you select one or more JobIds that contain files
   to be restored. You will be presented several methods
   of specifying the JobIds. Then you will be allowed to
   select which files from those JobIds are to be restored.

   To select the JobIds, you have the following choices:
   ...
        5: Select the most recent backup for a client
   ...
       13: Cancel
   Select item:  (1-13): 5
   The defined FileSet resources are:
        1: Exchange-CD
        2: Exchange-SS
        3: WindowsUserC
   Select FileSet resource (1-3): 1
   +-------+-------+----------+-------------+---------------------+------------+
   | jobid | level | jobfiles | jobbytes    | starttime           | volumename |
   +-------+-------+----------+-------------+---------------------+------------+
   |   362 | F     |      258 | 397,610,415 | 2012-07-09 00:26:41 | F-0007     |
   |   363 | I     |       23 |   6,294,004 | 2012-07-09 00:32:58 | F-0007     |
   +-------+-------+----------+-------------+---------------------+------------+
   You have selected the following JobIds: 362,363

   Building directory tree for JobId(s) 362,363 ...  +++++++++++++++++++++++++++
   +++++++++++++++
   257 files inserted into the tree.

   You are now entering file selection mode where you add (mark) and
   remove (unmark) files to be restored. No files are initially added, unless
   you used the "all" keyword on the command line.
   Enter "done" to leave this mode.

   cwd is: /
   $ cd @EXCHANGE
   cwd is: /@EXCHANGE/
   $ cd "Microsoft Exchange Writer/"
   cwd is: /@EXCHANGE/Microsoft Exchange Writer/
   $ cd "Microsoft Exchange Server/"
   cwd is: /@EXCHANGE/Microsoft Exchange Writer/Microsoft Exchange Server/
   $ cd "Microsoft Information Store/"
   cwd is: /@EXCHANGE/Microsoft Exchange Writer/Microsoft Exchange Server/Microsoft
    Information Store/
   $ cd WSB-EXCH10/
   cwd is: /@EXCHANGE/Microsoft Exchange Writer/Microsoft Exchange Server/Microsoft
    Information Store/WSB-EXCH10/
```

(continues on next page)

(continued from previous page)

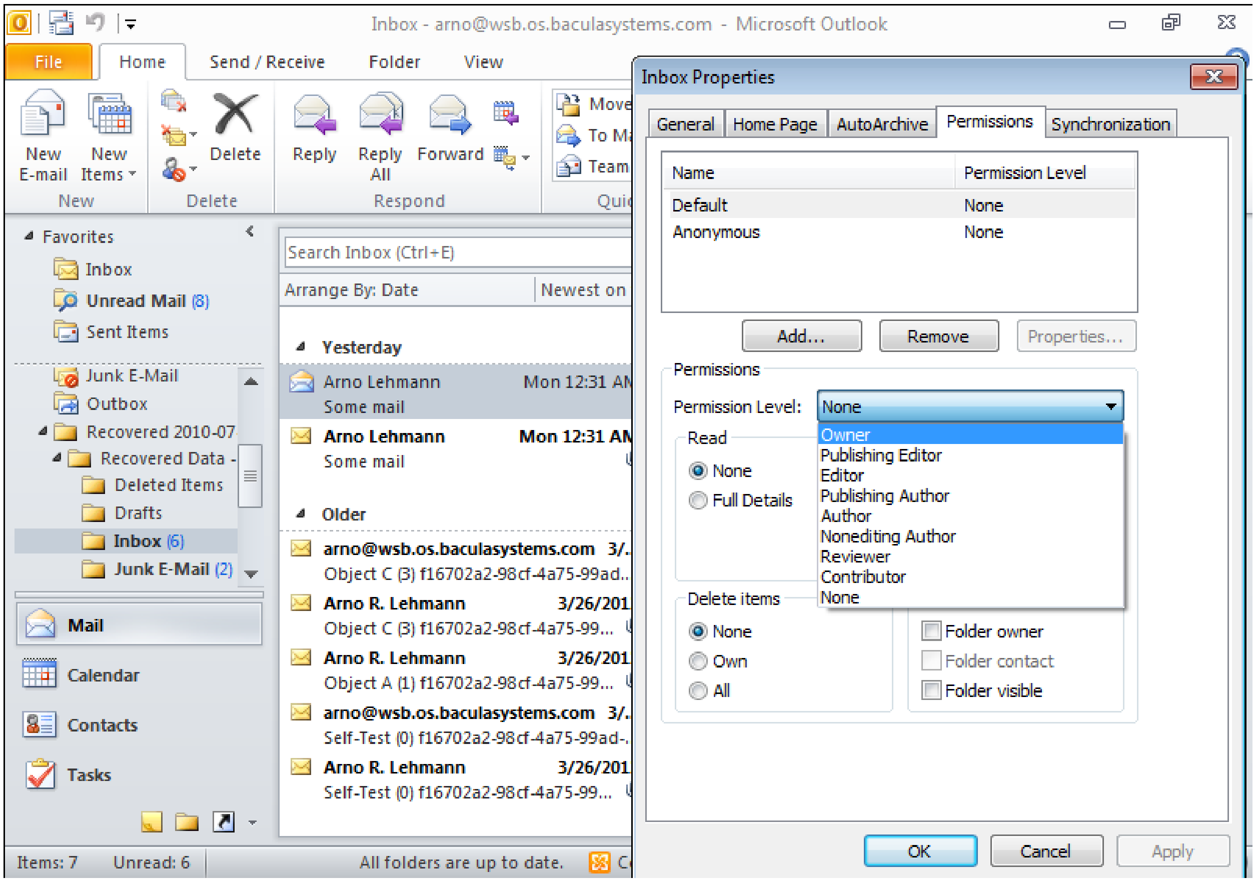
```
$ ls
Mailbox A/
Mailbox Database 1568811476/
$ mark "Mailbox Database 1568811476"
131 files marked.
$ done

Bootstrap records written to /var/lib/bacula/s-1-dir.restore.10.bsr

The job will require the following
   Volume(s)                 Storage(s)                SD Device(s)
===========================================================================

    F-0007                    File                      FileStorage

Volumes marked with "*" are online.

133 files selected to be restored.

Run Restore job
JobName:         RestoreFiles
Bootstrap:       /var/lib/bacula/s-1-dir.restore.10.bsr
Where:           Bacula EnterpriseRecovery
Replace:         always
FileSet:         Test Set
Backup Client:   wsb-exch10-fd
Restore Client:  wsb-exch10-fd
Storage:         File
When:            2012-07-09 00:38:03
Catalog:         MyCatalog
Priority:        10
Plugin Options:  *None*
OK to run? (yes/mod/no): yes
Job queued. JobId=364
*
```

Permission level “Owner” allows the user to manage and delete restored folders.



```
[PS] C:\Windows\system32>Get-MailboxStatistics -Database BEERecovery ¦ Format-Li
st DisplayName


DisplayName : Arno R. Lehmann

DisplayName : Microsoft Exchange

DisplayName : SystemMailbox{961571b5-5c15-4f5d-9b6d-7608e87de8ea}



[PS] C:\Windows\system32>Restore-Mailbox -Identity "Arno R. Lehmann" -RecoveryDa
tabase BEERecovery -TargetFolder "Recovered 2010-07-10" -RecoveryMailbox  "Arno
R. Lehmann"

Confirm
Are you sure you want to perform this action?
Recovering mailbox content from mailbox 'Arno R. Lehmann' in the recovery
database 'BEERecovery' to the mailbox for 'Arno R. Lehmann
(arno@wsb.os.baculasystems.com)'. This operation may take a long time to
complete.
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [?] Help (default is "Y"):


RunspaceId                     : bd10e95c-c540-446d-935c-bb93aa53156b
Identity                       : wsb.os.baculasystems.com/Users/Arno R. Lehma
                                 nn
DistinguishedName              : CN=Arno R. Lehmann,CN=Users,DC=wsb,DC=os,DC=
                                 baculasystems,DC=com
DisplayName                    : Arno R. Lehmann
Alias                          : arno
LegacyExchangeDN               : /o=Bacula Systems Sand Box Exchange/ou=Excha
                                 nge Administrative Group (FYDIBOHF23SPDLT)/c
                                 n=Recipients/cn=Arno R. Lehmann
PrimarySmtpAddress             : arno@wsb.os.baculasystems.com
SourceServer                   : WSB-EXCH10.wsb.os.baculasystems.com
SourceDatabase                 : BEERecovery
SourceGlobalCatalog            : WSB-MASTER
SourceDomainController         :
```

Fig. 25: Restoring a Mailbox with Microsoft Exchange Management Shell



Inbox - arno@wsb.os.baculasystems.com - Microsoft Outlook

Yesterday

Arno Lehmann Mon 12:31 AM
Some mail

Arno Lehmann Mon 12:31 AM
Some mail

Older

arno@wsb.os.baculasystems.com 3/...
Object C (3) f16702a2-98cf-4a75-99ad...

Arno R. Lehmann 3/26/201
Object C (3) f16702a2-98cf-4a75-99...

Arno R. Lehmann 3/26/201
Object A (1) f16702a2-98cf-4a75-99...

arno@wsb.os.baculasystems.com 3/...
Self-Test (0) f16702a2-98cf-4a75-99ad-...

Arno R. Lehmann 3/26/201
Self-Test (0) f16702a2-98cf-4a75-99...

Items: 7 Unread: 6 All folders are up to date.

Inbox Properties

Name	Permission Level
Default	None
Anonymous	None

Permissions

Permission Level: None

Read
- ● None
- ○ Full Details

Delete items
- ● None
- ○ Own
- ○ All

- □ Folder owner
- □ Folder contact
- □ Folder visible

Fig. 26: Restored Mailbox and Folder Permissions

2.7 Troubleshooting

As with any complex application, problems during backup or recovery may happen with Microsoft Exchange. In this chapter, we try to guide you through resolving some problems.

Backup

The most common problem during backup is that not all required file systems are included in the snapshot set; in these cases, the VSS writer responsible for handling ex will refuse the snapshot's creation. The solution is to explicitly include the required drive by adding a dummy file entry to the file set used.

If the backup itself works, but no files or not all required files are being backed up, this is often caused by some of the required infrastructure not being functional. For example, in a DAG configuration, if the *Microsoft Exchange Replication* service is not running because of a cluster environment issue, backups may be incomplete.

Restore

In most cases, when **Bacula Enterprise** reports a restore as successful, but actual Mailbox access shows no restore took place, the cause of the problem can be found in Windows' Event log. In most cases, detailed error messages will also be included in the Job report. For those Events that we encountered, we have a listing prepared below, explaining how to proceed.

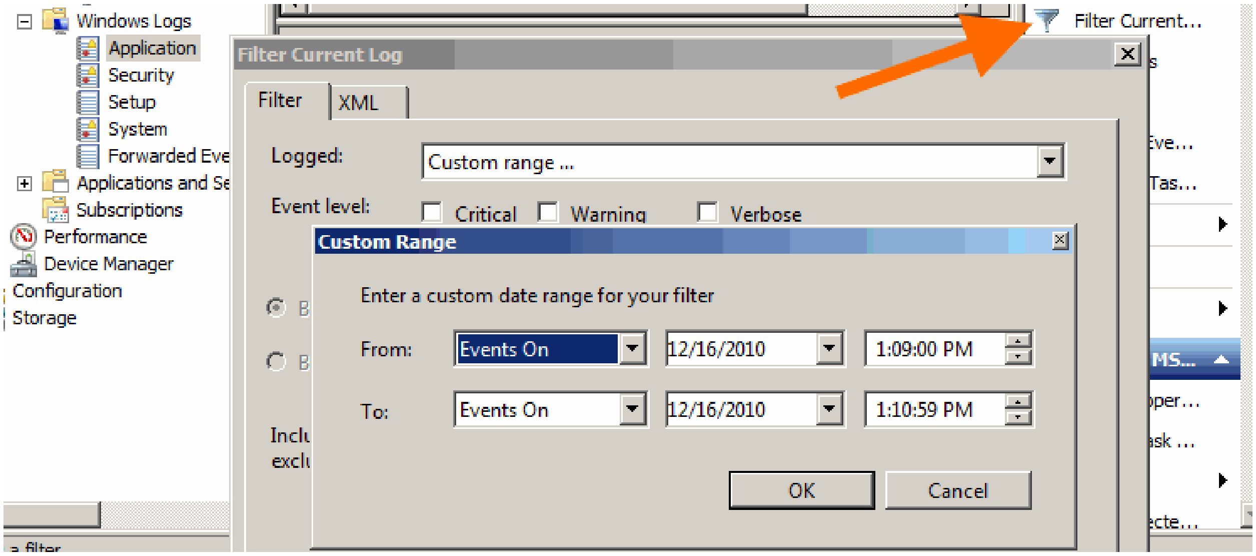
Windows Events

This is a list of Events that you may find in the Windows Application Event log after a restore operation, which is reported as successful by **Bacula Enterprise**. When looking for problem causes, always start with the `earliest` Event related to the current restore attempt (this is most easily done by filtering the Event log by date and time, using the start and end dates of the restore job). We provide an example in figures *Windows' Event Log Filtered, Showing Cause for Failed Restore* and *Job Report is OK, but actual Restore Failed*. In figure *Exchange Message after Database Mount Failure* the corresponding message from ex when trying to mount the database is shown, and figure *Application Log Messages after Database Mount Failure* shows an overview of the corresponding Windows Event Log messages. Note that, in most cases, there will be many failures reported in Windows' Event Log, but normally the first Error is the one you are interested in. With the examples we provide it should be easy to correlate the different reports and drill down to the underlying problem.

Listing 34: Job Report is OK, but actual Restore Failed

```
...
20-Mar 00:31 wsb-exch10-fd JobId 20: Exchange VSS Writer successfully
 restored the backup set to database ...
20-Mar 00:31 wsb-exch10-fd JobId 20: Exchange VSS Writer will perform
 database recovery on database ...
...
20-Mar 00:31 wsb-exch10-fd JobId 20: Exchange VSS Writer failed restoring
 a backup with error code -528 when processing post-restore tasks ...
20-Mar 00:31 wsb-exch10-fd JobId 20: Exchange VSS Writer failed with error
 code -528 when processing the post-restore event.

If any databases were restored, they are likely in a dirty-shutdown state.
20-Mar 00:31 wsb-exch10-fd JobId 20: A VSS writer has rejected an event with
 error 0x00000000, The operation completed successfully.
. Changes that the writer made to the writer components while handling the
```

(continues on next page)



Logged: Custom range ...

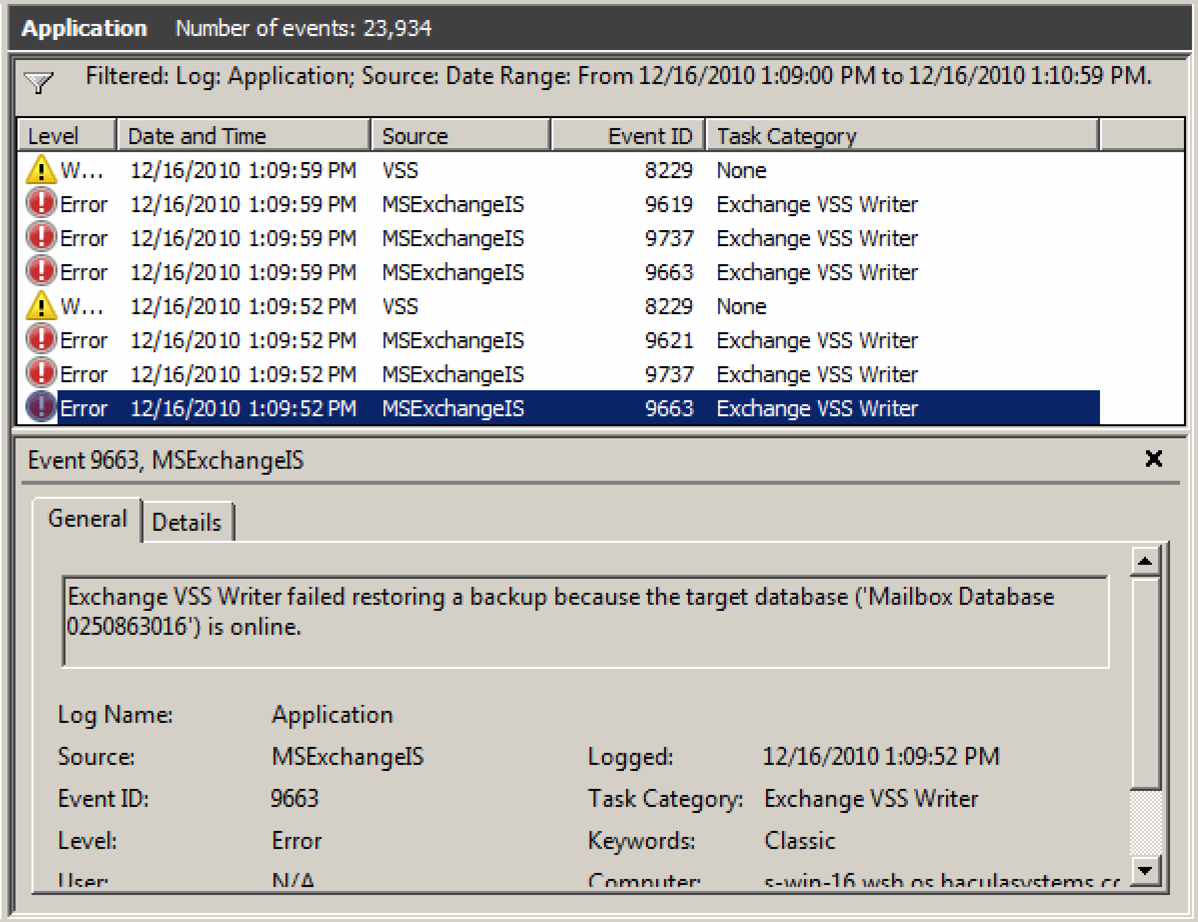
Event level: □ Critical □ Warning □ Verbose

Enter a custom date range for your filter

From: Events On 12/16/2010 1:09:00 PM

To: Events On 12/16/2010 1:10:59 PM

Fig. 27: Windows' Event Log Filtering



Application Number of events: 23,934

Filtered: Log: Application; Source: Date Range: From 12/16/2010 1:09:00 PM to 12/16/2010 1:10:59 PM.

Level	Date and Time	Source	Event ID	Task Category
W...	12/16/2010 1:09:59 PM	VSS	8229	None
Error	12/16/2010 1:09:59 PM	MSExchangeIS	9619	Exchange VSS Writer
Error	12/16/2010 1:09:59 PM	MSExchangeIS	9737	Exchange VSS Writer
Error	12/16/2010 1:09:59 PM	MSExchangeIS	9663	Exchange VSS Writer
W...	12/16/2010 1:09:52 PM	VSS	8229	None
Error	12/16/2010 1:09:52 PM	MSExchangeIS	9621	Exchange VSS Writer
Error	12/16/2010 1:09:52 PM	MSExchangeIS	9737	Exchange VSS Writer
Error	12/16/2010 1:09:52 PM	MSExchangeIS	9663	Exchange VSS Writer

Event 9663, MSExchangeIS

Exchange VSS Writer failed restoring a backup because the target database ('Mailbox Database 0250863016') is online.

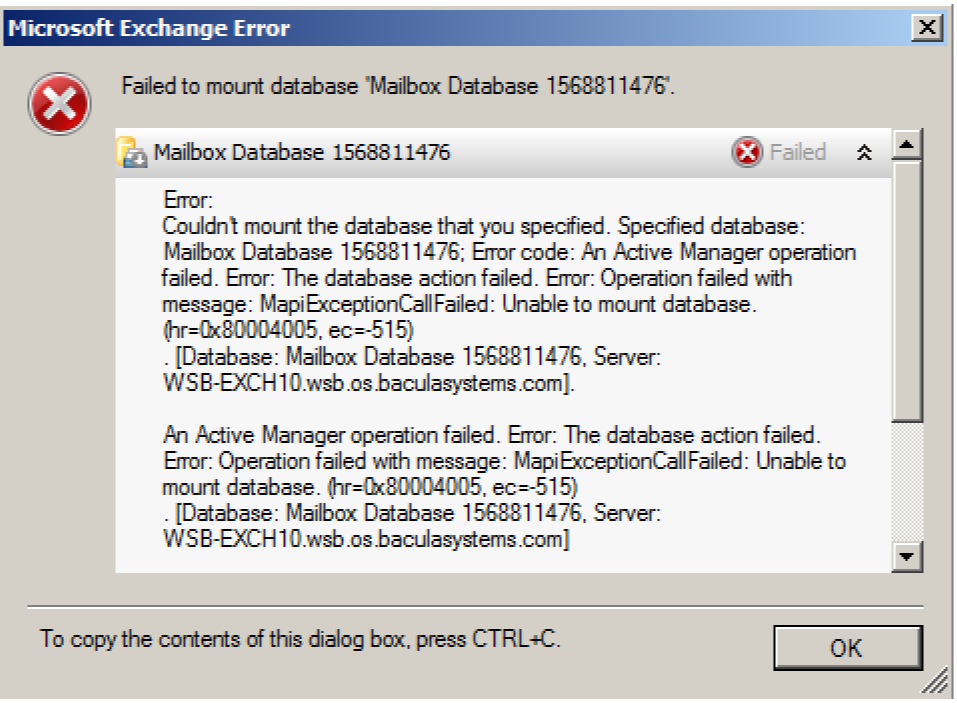
Log Name:	Application		
Source:	MSExchangeIS	Logged:	12/16/2010 1:09:52 PM
Event ID:	9663	Task Category:	Exchange VSS Writer
Level:	Error	Keywords:	Classic
User:	N/A	Computer:	s-win-16.wsb.os.baculasystems.cc

Fig. 28: Windows' Event Log Filtered, Showing Cause for Failed Restore

(continued from previous page)

```
 event will not be available to the requester.
Check the event log for related events from the application hosting the VSS writer.
...
20-Mar 00:31 s-1-dir JobId 20: Bacula s-1-dir 6.0.0.5 (06Mar12):
  Build OS:               x86_64-unknown-linux-gnu suse 12.1
  JobId:                  20
  Job:                    RestoreFiles.2012-03-20_00.30.23_01
  Restore Client:         wsb-exch10-fd
  Start time:             20-Mar-2012 00:30:25
  End time:               20-Mar-2012 00:31:58
  Files Expected:         17
  Files Restored:         17
  Bytes Restored:         150,022,642
  Rate:                   1613.1 KB/s
  FD Errors:              0
  FD termination status:  OK
  SD termination status:  OK
  Termination:            Restore OK
```

Note that the actual Job report contains more detailed information!



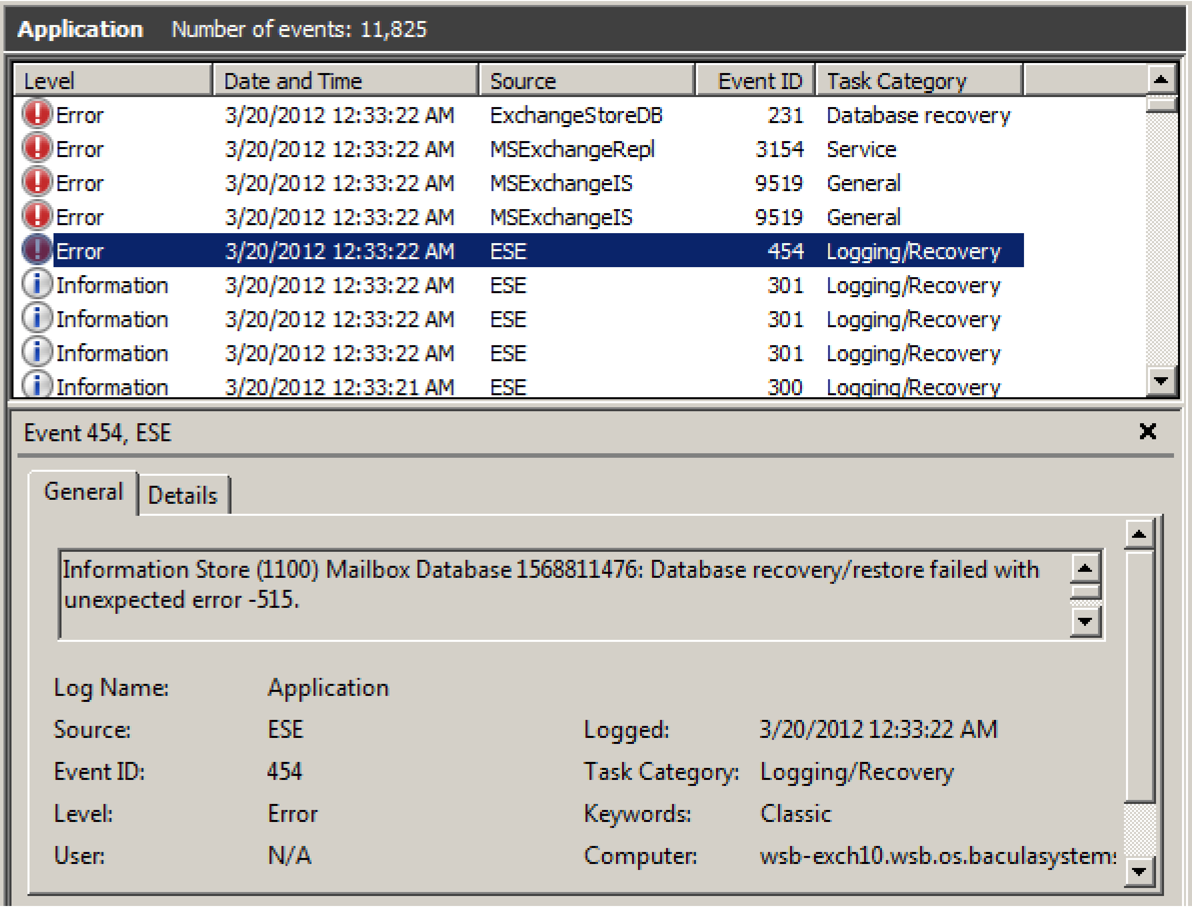
Failed to mount database 'Mailbox Database 1568811476'.

Error:
Couldn't mount the database that you specified. Specified database: Mailbox Database 1568811476; Error code: An Active Manager operation failed. Error: The database action failed. Error: Operation failed with message: MapiExceptionCallFailed: Unable to mount database. (hr=0x80004005, ec=-515)
. [Database: Mailbox Database 1568811476, Server: WSB-EXCH10.wsb.os.baculasystems.com].

An Active Manager operation failed. Error: The database action failed. Error: Operation failed with message: MapiExceptionCallFailed: Unable to mount database. (hr=0x80004005, ec=-515)
. [Database: Mailbox Database 1568811476, Server: WSB-EXCH10.wsb.os.baculasystems.com]

To copy the contents of this dialog box, press CTRL+C.

Fig. 29: Exchange Message after Database Mount Failure



Application Number of events: 11,825

Level	Date and Time	Source	Event ID	Task Category
Error	3/20/2012 12:33:22 AM	ExchangeStoreDB	231	Database recovery
Error	3/20/2012 12:33:22 AM	MSExchangeRepl	3154	Service
Error	3/20/2012 12:33:22 AM	MSExchangeIS	9519	General
Error	3/20/2012 12:33:22 AM	MSExchangeIS	9519	General
Error	3/20/2012 12:33:22 AM	ESE	454	Logging/Recovery
Information	3/20/2012 12:33:22 AM	ESE	301	Logging/Recovery
Information	3/20/2012 12:33:22 AM	ESE	301	Logging/Recovery
Information	3/20/2012 12:33:22 AM	ESE	301	Logging/Recovery
Information	3/20/2012 12:33:21 AM	ESE	300	Logging/Recovery

Event 454, ESE

Information Store (1100) Mailbox Database 1568811476: Database recovery/restore failed with unexpected error -515.

Log Name:	Application		
Source:	ESE	Logged:	3/20/2012 12:33:22 AM
Event ID:	454	Task Category:	Logging/Recovery
Level:	Error	Keywords:	Classic
User:	N/A	Computer:	wsb-exch10.wsb.os.baculasystem:

Fig. 30: Application Log Messages after Database Mount Failure

Table 2: Problem Resolution by Windows Event

Source	Event ID	Task Category	Resolution
MSExchangeRepl	3154	Service	Mount the database through Exchange Management Console
MSExchangeIS	9663	Exchange VSS Writer	Dismount the Database prior to restoring
MSExchangeIS	9751	Exchange VSS Writer	Try the following: Move the `*.env`-file out of the Mailbox Database directory. Run `eseutil /r <Three-char logfile base name>` in the Mailbox Database directory. Re-run the restore as usual. The database should now be restored correctly.
ESE	440	Logging/Recovery	Remove the existing database log files from the data directory prior to restoring
ESE	518	Logging/Recovery	see above
ESE	454	Logging/Recovery	see above

2.8 Script to Prepare Recovery Database

```
# $Id: preparebacularestore-E10.ps1 1387 2014-01-26 13:57:57Z arno $
# (C) 2012 Bacula Systems SA
#
# This script is an EXAMPLE of how to prepare Exchange 2010 for an alternate location
# restore of data backed up with the vss plugin's @EXCHANGE module
#
# Use at your own risk, make sure to audit before using!
#
# Configuration is done right at the top.
#
# NULL means detect server automatically. Fails if there's more than one
# To hard-code the server to use, try
#  $server = "<nodename>"  with a hard-coded nodename or
#  $server = hostname to use the local hostname as reported by the hostname command
$server = $null


if ($server -eq $null) {
  $mbs = @(Get-ExchangeServer | where {$_.IsMailboxServer} | ForEach-Object {$_.Name})
#  $mbs += @(Get-ExchangeServer | where {$_.IsMailboxServer} | ForEach-Object {$_.Name})
  if ($mbs.Count -gt 1) {
        "More than one Mailbox host. Define explicitly, please!"
        Exit
  } else {
        $server = $mbs.Get(0)
  }
```

(continues on next page)

(continued from previous page)

```
}

$me = hostname
if ([string]::Compare($me, $server, $true)) {
  "The Mailbox server we would use is not the local machine... we do not support that!"
  Exit
}
"Using server $server"

$mdbs = @(Get-MailboxDatabase | where { $_.Revovery -or $_.AllowFileRestore})

if ($mdbs.Count -gt 0) {
  "The following Mailbox Databases are Recovery Databases or set to allow Recovery. This␣
↪may cause problems, please fix and retry!"
  $mdbs | Format-Table -AutoSize Server,Identity,AllowFileRestore,Recovery
  Exit
}

$tgtdrive = (Get-WMIObject Win32_logicaldisk | Sort-Object -Property FreeSpace -
↪Descending).Get(0)

"Using drive $($tgtdrive.DeviceId) for the RDB"
"There are about $("{0:N0}" -f ($tgtdrive.FreeSpace / 1024 / 1024 /1024) ) GBytes free␣
↪on it"


do {
  $tgtdir = $tgtdrive.DeviceId + "\" + [System.IO.Path]::GetRandomFileName()
} while (Test-Path $tgtdir)

"$tgtdir is our working directory."

New-Item -Path $tgtdir -Type "directory" | Out-Null

$rdbn = "Bacula EnterpriseRecovery"
$suff = $null

while (($rdb = Get-MailboxDatabase -Identity ($rdbn + $suff) -WarningAction␣
↪SilentlyContinue -ErrorAction SilentlyContinue) -and ($rdb.Count -gt 0)) {
  "MailboxDatabase " + $rdbn + $suff + "exists... trying next one"
  $suff++
}

$rdbn += $suff

New-MailboxDatabase -Recovery -Name $rdbn -Server $server -EdbFilePath ($tgtdir + "\
↪RecoveryDB.edb") -LogFolderPath $tgtdir | Out-Null

Set-MailboxDatabase -Identity $rdbn -AllowFileRestore $true -WarningAction␣
↪SilentlyContinue -ErrorAction SilentlyContinue | Out-Null

"You can Restore with where=" + $rdbn + " now"
```

2.9 Limitations

- The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

In order to have the best protection level for a Microsoft Exchange on-premise instance, Bacula Systems recommends using a combined strategy of the two plugins. Exchange EWS Plugin can be used to restore small pieces of lost information at a user level. However, in case of suffering a disaster where the whole Exchange deployment has been impacted, the recommended protection will be given by the Exchange Plugin, based on VSS, where the speed to recover a full database will be superior.