opentext™

GroupWise
Utilities Reference

October 2023

Legal Notices

Contents

About This Guide

This *GroupWise Utilities Reference* presents the content of the GroupWise Linux man pages in guide format as a reference.

- grpwise-ma(1)
- gwpoa(1)
- gwmta(1)
- gwia(1)
- gwdva(1)
- gwmon(1)

Audience

This guide is intended for network administrators who install and administer GroupWise.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the Comment feature 💬 from the page toolbar or at the bottom of each page of the online documentation.

Additional Documentation

For additional GroupWise documentation, see the GroupWise Documentation website.

1 GroupWise Administration Utility

Reference the following section for GroupWise Administration Utility commands:

gwadminutil(1) (page 8)

gwadminutil(1)

Name

`gwadminutil` - GroupWise Administration Utility

Syntax

`gwadminutil` *option* [*arguments*]

Description

The GroupWise Administration Utility (GWAdminUtil) includes the following capabilities:

- Command-line alternatives to the domain and post office database maintenance tasks available in the GroupWise Admin console.
- Domain database maintenance tasks that are not available in the Admin console because they require simultaneous direct file access to two domain databases.
- Security tasks for managing passwords and certificates.
- Installation of agents on GroupWise servers where no domain or post office is located.
- Management of agents similar to the `grpwise` script.

Options

Database Usage Options That Are Also Available in the Admin Console:

`validate`

Checks a database to determine if there are any physical problems with the database.

See gwadminutil-validate(1) for detailed syntax and examples.

`recover`

Corrects physical problems in a domain or post office database, but does not update incorrect information. After you perform a Recover Database operation, you can update information by synchronizing the database.

See gwadminutil-recover(1) for detailed syntax and examples.

`rebuild`

Corrects physical problems in a domain or post office database, and updates incorrect information. It is useful to rebuild a database when you have many user deletions or when GroupWise information has been lost.

See gwadminutil-rebuild(1) for detailed syntax and examples.

reclaim

Reclaims unused database space. As you add information to a GroupWise system, the databases increase in size. If you delete information, the space that was created in the databases for the deleted information is not immediately released. When you reclaim the disk space, it can reduce the size of the database.

See gwadminutil-reclaim(1) for detailed syntax and examples.

reindex

Rebuilds the listing indexes in the database that are used for sorting users in the Admin console.

See gwadminutil-reindex(1) for detailed syntax and examples.

Database Usage Options That Are Not Available in the Admin Console:

sync

Updates the primary domain database from a secondary domain database.

This process usually takes place automatically. However, if the link between the two domains has been down, you might want to run the synchronization manually. Any records owned by the secondary domain, such as post offices or users, are replicated from the secondary domain database to the primary domain database.

See gwadminutil-sync(1) for detailed syntax and examples.

convert

Promotes a secondary domain to a primary domain in your GroupWise system. When you do this, the current primary domain is demoted to a secondary domain. The MTA and POA must be running to perform this conversion.

See gwadminutil-convert(1) for detailed syntax and examples.

release

Removes a secondary domain from the current GroupWise system and creates a new GroupWise system. The released domain becomes the new primary domain in its own single-domain GroupWise system.

See gwadminutil-release(1) for detailed syntax and examples.

merge

Merges an external domain into the local GroupWise system as a new secondary domain. The external domain must be the only domain in the other GroupWise system.

See gwadminutil-merge(1) for detailed syntax and examples.

upgrade

Upgrades the version of a domain or post office database.

See gwadminutil-upgrade(1) for detailed syntax and examples.

Security Usage Options

`setadmin`

Changes the user name and password for the GroupWise Super Admin.

See gwadminutil-setadmin(1) for detailed syntax and examples.

`installcfg`

Sets the authentication mode for the GroupWise Installation console.

See gwadminutil-installcfg(1) for detailed syntax and examples.

`ca`

Interacts with the GroupWise Admin Service certificate authority (CA) to generate, issue, list, and revoke certificates for use with GroupWise domains, post offices, and agents.

See gwadminutil-ca(1) for detailed syntax and examples.

`certinst`

Contacts the primary domain Admin Service certificate authority (CA) and installs a new certificate for the Admin Service on the local domain or post office server.

See gwadminutil-certinst(1) for detailed syntax and examples.

Other Usage Options

`services`

Manages the GroupWise agent services. (Some functionality overlaps with the `grpwise` script.)

See gwadminutil-services(1) for detailed syntax and examples.

`config`

Provides configuration alternatives for the following aspects of the GroupWise Admin Service:

- Support in the GroupWise Admin console for clustering of the GroupWise agents
- Default port for the GroupWise Admin console

See gwadminutil-config(1) for detailed syntax and examples.

`dbinfo`

Displays information about a GroupWise domain or post office database.

See gwadminutil-dbinfo(1) for detailed syntax and examples.

Help Options:

`--help, -?`

Displays the help information and exits.

Files

`/opt/novell/groupwise/admin/gwadminutil`

Script file that runs GWAdminUtil.

See Also

gwadminutil-ca(1), gwadminutil-certinst(1), gwadminutil-convert(1), gwadminutil-dbinfo(1), gwadminutil-installcfg(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-services(1), gwadminutil-setadmin(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

2 GroupWise Administration Utility — Database Options

Reference the sections below for database command options in the GroupWise Administrator Utility.

gwadminutil-convert(1)

Name

`gwadminutil-convert` - The GroupWise Administration Utility database convert option.

Syntax

`gwadminutil convert -d` *path* `-p` *path*

Description

Promotes a secondary domain to a primary domain in your GroupWise system. When you do this, the current primary domain is demoted to a secondary domain. The MTA and POA must be running to perform this conversion.

This task requires direct file access to both databases. For that reason, it is not available in the Admin console, which provides direct file access to one domain database through the Admin Service for that domain.

Options

Usage Options:

-d */path_to_secondary_domain*

Specifies the path to the secondary domain that you want to promote to the primary domain.

-p */path_to_primary_domain*

Specifies the path to the primary domain that you want to convert into a secondary domain.

Help Options:

--help, -?

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

gwadminutil -convert -d /gwsystem/provo2 -p /gwsystem/provo1

Promotes the Provo2 secondary domain to be the primary domain and converts the original Provo1 primary domain into a secondary domain.

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

gwadminutil-merge(1)

Name

`gwadminutil-merge` - The GroupWise Administration Utility system merge option

Syntax

```
gwadminutil merge [-mergesync] -d path -p path
```

Description

Merges an external domain into the local GroupWise system as a new secondary domain. The external domain must be the only domain in the other GroupWise system.

This task requires direct file access to both databases. For that reason, it is not available in the Admin console, which provides direct file access to one domain database through the Admin Service for that domain.

After you merge an external domain into your GroupWise system, you must perform the following additional steps in order to complete the setup of the new secondary domain and its post offices:

1 On the secondary domain server, install a new server certificate so that the local Admin Service can communicate with the primary domain Admin Service:

```
gwadminutil certinst -d /path_to_secondary_domain
          -ca ip_address_of_primary_domain_server:9710 -a
admin_user_name -p
```

2 Wait for replication of the domain information to the post office.

On each post office server, you can use the following command to view the owning domain name in the post office database:

```
gwadminutil dbinfo /path_to_post_office
```

3 On each post office server, install a new server certificate so that the local Admin Service can communicate with the primary domain Admin Service:

```
gwadminutil certinst -d /path_to_post_office
          -ca ip_address_of_primary_domain_server:9710 -a
admin_user_name -p
```

Options

Usage Options:

-d, --domain */path_to_external_domain*

Specifies the path to the external domain that you want to merge into the local GroupWise system.

-mergesync

Merges external system synchronization information from the external domain into the local GroupWise system.

-p, --primary */path_to_primary_domain*

Specifies the path to the primary domain of your local GroupWise system.

Help Options:

--help, -?

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

Script file that runs GWAdminUtil.

Examples

gwadminutil merge -p /gwsystem/provo1 -d /gwmail/orem4

Merges the Orem4 external domain into the local GroupWise system where Provo1 is the primary domain.

gwadminutil merge -mergesync -p /gwsystem/provo1 -d /gwmail/orem4

Merges the Orem4 external domain into the local GroupWise system were Provo1 is the primary domain, and establishes external system synchronization between the local GroupWise system and any other external systems that were syncing with the Orem4 domain.

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

gwadminutil-rebuild(1)

Name

`gwadminutil-rebuild` - The GroupWise Administration Utility database rebuild option

Syntax

`gwadminutil rebuild -d `*`path`*` -n `*`name`*` -o path [-cd]`

Description

Corrects physical problems in a domain or post office database, and updates incorrect information. It is useful to rebuild a database when you have many user deletions or when GroupWise information has been lost.

When you rebuild a domain database, you must be connected to the primary domain in order to rebuild the secondary domain. Rebuilding the primary domain database is not recommended, but can be necessary in a single-domain system.

When you rebuild a post office database, you must be connected to the domain that owns the post office.

This task requires exclusive access to the owning domain database. Stop the agents and the Admin Service for the domain before rebuilding the database.

Options

Usage Options:

`-cd, --create-directory`

Creates a new domain or post office folder structure, along with the rebuilt database and the accompanying dictionary (`.dc`) files.

`-d, --domain `*`/path_to_domain_database`*

Specifies the path to the domain database from which the domain or post office database will be rebuilt.

`-n, --name `*`object`*

Specifies the name of the domain or post office that is being rebuilt. For a post office, use the `domain.post_office` format.

`-o `*`/path_to_output_folder`*

Specifies the location for the rebuilt database. After the database has been rebuilt, copy it into the original domain or post office folder.

Help Options:

--help, -?

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

gwadminutil rebuild -d /gwsystem/provo1 -n provo2 -o /tmp

Rebuilds the Provo2 secondary domain database, based on the Provo1 primary domain database in the `/gwsystem/provo1` folder, and places the rebuilt secondary domain database in the `/tmp` folder.

gwadminutil rebuild -d /gwsystem/provo1 -n provo1 -o /tmp

Rebuilds the Provo1 primary domain database, and places the rebuilt primary domain database in the `/tmp` folder. Rebuilding the primary domain database is not recommended. If the primary domain database is sufficiently damaged, the rebuilt version not viable. Regular backups of the primary domain database in a single-domain system are critical.

gwadminutil rebuild -d /gwsystem/provo2 -n provo2.sales -o /tmp

Rebuilds the Sales post office that belongs to the Provo2 domain, and places the rebuilt post office database in the `/tmp` folder.

gwadminutil rebuild -d /gwsystem/provo1 -n provo2 -o /tmp -cd

Rebuilds the Provo2 domain, places the rebuilt domain database in the `/tmp` folder, and provides all the supporting folder structure and files to create a new domain directory.

gwadminutil rebuild -d /gwsystem/provo2 -n provo2.sales -o /tmp -cd

Rebuilds the Sales post office that belongs to the Provo2 domain, places the rebuilt post office database in the `/tmp` folder, and provides all the supporting folder structure and files to create a new post office directory. This process does not rebuild user databases (user*xxx*.db) and message databases (msg*nnn*.db).

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

gwadminutil-reclaim(1)

Name

`gwadminutil-reclaim` - The GroupWise Administration Utility database reclaim option

Syntax

`gwadminutil reclaim` *path*

Description

Reclaims unused database space. As you add information to a GroupWise system, the databases increase in size. If you delete information, the space that was created in the databases for the deleted information is not immediately released. When you reclaim the disk space, it can reduce the size of the database.

Options

Usage Options:

/path_to_database

Removes unused space from the database at the specified location.

Help Options:

--help, -?

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

Script file that runs GWAdminUtil.

Examples

This program normally runs as `root.`

gwadminutil reclaim /gwsystem/provo1

Removes unused space from the domain database for the Provo1 domain.

gwadminutil reclaim /gwsystem/engineering

Removes unused space from the post office database for the Engineering post office.

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

gwadminutil-recover(1)

Name

`gwadminutil-recover` - The GroupWise Administration Utility database recover option

Syntax

`gwadminutil recover` *path*

Description

Corrects physical problems in a domain or post office database, but does not update incorrect information. After you perform a Recover Database operation, you can update information by synchronizing the database.

Options

Usage Options:

/path_to_database

> Specifies the path to the database to recover.

Help Options:

--help, -?

> Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

> Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

gwadminutil recover /gwsystem/provo1

> Recovers the domain database for the Provo1 domain.

gwadminutil recover /gwsystem/engineering

> Recovers the post office database for the Engineering domain.

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

gwadminutil-reindex(1)

Name

`gwadminutil-reindex` - The GroupWise Administration Utility database reindex option

Syntax

`gwadminutil reindex path`

Description

Rebuilds the listing indexes in the database that are used for sorting users in the Admin console.

There is a listing index for all users in your GroupWise system, for all users in each domain, and for all users in each post office. If you see differences in user listings when you are connected to different domains in the Admin console, you should rebuild the listing indexes. When you rebuild the listing indexes, the current system index is used.

Options

Usage Options:

/path_to_database
> Specifies the path to the database to reindex.

Help Options:

--help, -?
> Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil
> Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

gwadminutil reindex /gwsystem/provo1
> Rebuilds the listing index for the Provo1 domain.

gwadminutil reindex /gwsystem/engineering
> Rebuilds the listing index for the Engineering post office.

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

gwadminutil-release(1)

Name

`gwadminutil-release` - The GroupWise Administration Utility database release option

Syntax

```
gwadminutil release -d path -p path -n name
```

Description

Removes a secondary domain from the current GroupWise system and creates a new GroupWise system. The released domain becomes the new primary domain in its own single-domain GroupWise system.

After you release a domain, you must perform the following additional steps in order to complete the setup of the new GroupWise system:

1 On the server where you performed the release, set up the GroupWise Super Admin user for the new GroupWise system:

    ```
    gwadminutil setadmin /path_to_primary_domain_database -a
    admin_user_name -p
    ```

2 On the new primary domain server, set up the GroupWise certificate authority for the new GroupWise system:

    ```
    gwadminutil ca -d /path_to_primary_domain_database -g
    ```

3 Wait for replication of the new GroupWise system information from the primary domain to the post office.

    On each post office server, you can use the following command to view the system name in the post office database:

    ```
    gwadminutil dbinfo /path_to_post_office
    ```

4 On each post office server, install a new server certificate so that the local Admin Service can communicate with the primary domain Admin Service:

    ```
    gwadminutil certinst -d /path_to_post_office
                -ca ip_address_of_primary_domain_server:9710 -a
    admin_user_name -p
    ```

Options

Usage Options:

-d, --domain /path_to_domain_to_release

Specifies the path to the secondary domain that you want to release from one GroupWise system so that it can be merged into another GroupWise system.

-n, --name *name_of_new_system*

    Specifies the name of the new GroupWise system that is created when the domain is released.

-p, --primary */path_to_new_primary_domain*

    Specifies the path to the new primary domain that is created when you release the secondary domain from the original GroupWise system.

Help Options:

--help, -?

    Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

    Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

gwadminutil release -p /gwsystem/provo1 -d /gwmail/provo4 -n orem1

    Releases the Provo4 domain from its GroupWise system and creates a new, single-domain GroupWise system named Orem1.

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

gwadminutil-sync(1)

Name

`gwadminutil-sync` - The GroupWise Administration Utility database synchronization option

Syntax

`gwadminutil sync -d `*`path`*` -p `*`path`*

Description

Updates the primary domain database from a secondary domain database.

Options

Usage Options:

`-d, --domain `*`/path_to_secondary_domain_database`*

Specifies the path to the secondary domain database from which you want to synchronize records up to the primary domain.

`-p, --primary `*`/path_to_primary_domain_database`*

Specifies the path to the primary domain database that needs to be updated with information from the secondary domain.

Help Options:

`--help, -?`

Displays the help information and exits.

Files

`/opt/novell/groupwise/admin/gwadminutil`

Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

`gwadminutil sync -p /gwsystem/provo1 -d /gwsystem/provo2`

Updates the Provo1 primary domain database with records from the Provo2 secondary domain database.

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

gwadminutil-upgrade(1)

Name

`gwadminutil-upgrade` - The GroupWise Administration Utility database upgrade option.

Syntax

`gwadminutil upgrade` *path*

Description

Upgrades the version of a domain or post office database.

The upgrade process copies the data dictionary (`.dc`) files that match the version of GWAdminUtil into the domain or post office directory, performs a Recover on the database, and then completes the upgrade of the database. In order for the version upgrade to succeed, the following conditions must be met:

- ◆ The owning domain must already be upgraded to the GWAdminUtil version.
- ◆ The version information from the owning domain must already be replicated into the target database.

This command is the equivalent of using the Installation console to upgrade a domain or post office.

Options

Usage Options:

`/path_to_database`
   Specifies the path of the database to upgrade to the current GroupWise version.

Help Options:

`--help, -?`
   Displays the help information and exits.

Files

`/opt/novell/groupwise/admin/gwadminutil`
   Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

```
gwadminutil upgrade /gwsystem/provo2
```
    Upgrades the Provo2 domain to the same GroupWise version that the primary domain, Provo1, has been upgraded to.

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

gwadminutil-validate(1)

Name

`gwadminutil-validate` - The GroupWise Administration Utility database validate option

Syntax

`gwadminutil validate` *path*

Description

Checks a database to determine if there are any physical problems with the database.

If there are any physical problems, you can rebuild or recover the database to correct them. You should validate the data in databases on a regular basis. This process takes just a few minutes and does not interrupt normal operation. The frequency can vary depending on the size of your system and the number of changes you make to users, resources, and groups.

Options

Usage Options:

`/path_to_database`
> Specifies the path to the domain database to validate.

Help Options:

`--help, -?`
> Displays the help information and exits.

Files

`/opt/novell/groupwise/admin/gwadminutil`
> Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

`gwadminutil validate /gwsystem/provo1`
> Validates the Provo1 domain database.

`gwadminutil validate /gwsystem/engineering`
> Validates the Engineering post office database.

See Also

gwadminutil(1), gwadminutil-convert(1), gwadminutil-merge(1), gwadminutil-rebuild(1), gwadminutil-reclaim(1), gwadminutil-recover(1), gwadminutil-reindex(1), gwadminutil-release(1), gwadminutil-sync(1), gwadminutil-upgrade(1), gwadminutil-validate(1)

3 GroupWise Administration Utility - Security Options

Reference the sections below for security command options in the GroupWise Administrator Utility.

gwadminutil-ca(1)

Name

`gwadminutil-ca` - The GroupWise Administration Utility certificate authority (CA) option

Syntax

```
gwadminutil ca -d path [-g] [-f] [-l] [-i path] [-n name]
                       [-p serial_number] [-r serial_number][-cr path]
```

Description

Interacts with the GroupWise Admin Service certificate authority (CA) to generate, issue, list, and revoke certificates for use with GroupWise domains, post offices, and agents.

The GroupWise Admin Service CA issues self-signed server certificates for domain and post office servers. The CA certificate for the primary domain server signs the other server certificates. The CA can regenerate itself if necessary.

Options

Usage Options:

-cr, --certroot */path_to_certificates*

   Specifies an optional path to the root certificates directory.

-d, --domain */primary_domain_folder*

   Specifies the path to the primary domain database.

-f, --force

   By default, if you specify an existing output file name, you are notified that the certificate file already exists. Use this option to overwrite the existing certificate file.

-g, --generate

   Generates a certificate and private key for the primary domain.

-i, --issue */path_to_database*

   Creates a new certificate for the Admin Service that is running for the domain or post office associated with the specified database.

-l, --list

   Lists the certificates issued by the GroupWise CA. The list includes the serial number for each certificate.

-n, --name *poa_name*

   Specifies the object name of the POA when multiple POAs run for a single post office.

-p, --print *serial_number*

Displays the information about the certificate that has the specified serial number.

-r, --revoke *serial_number*

Revokes the certificate with the specified serial number.

Help Options:

--help, -?

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

Script file that runs GWAdminUtil.

/opt/novell/groupwise/certificates/*system_guid***/ca.***

Files created when a certificate is issued.

ca.crt

CA certificate file.

ca.key

CA private key file.

ca.srl

Certificate serial number list.

ca.crl

Certificate revocation list.

/opt/novell/groupwise/certificates/*system_guid***/issued**

Folder for storing copies of all issued certificates.

/opt/novell/groupwise/certificates/*system_guid***/revoked**

Folder for storing copies of all revoked certificates. The folder is not created until a certificate is revoked.

Examples

This program normally runs as `root`.

gwadminutil ca -g -d /gwsystem/provo1 -f

Generates a new CA certificate and private key for the primary domain. Overwrites the existing CA certificate.

gwadminutil ca -i /gwsystem/provo2

Generates a new server certificate for a secondary domain server.

`gwadminutil ca -i /gwsystem/sales`

Generates a new server certificate for a post office server.

`gwadminutil ca -l`

Lists all of the certificates that the GroupWise CA has issued, along with their serial numbers.

`gwadminutil ca -p 5266AC7B000A`

Displays detailed information about the certificate whose serial number you have specified.

`gwadminutil ca -r 5266AC7B000A`

Revokes the specified certificate.

See Also

gwadminutil(1), gwadminutil-certinst(1), gwadminutil-dbinfo(1), gwadminutil-installcfg(1), gwadminutil-setadmin(1)

gwadminutil-certinst(1)

Name

`gwadminutil-certinst` - The GroupWise Administration Utility certificate installation option

Syntax

`gwadminutil certinst -db` *`path`* `-ca` *`address`* `-a` *`name`* `-p [`*`password`*`] [-n name]`

Description

Contacts the primary domain Admin Service certificate authority (CA) and installs a new certificate for the Admin Service on the local domain or post office server.

In order to contact the primary domain Admin Service certificate authority, you must provide the path to the primary domain database, the GroupWise Super Admin user name and password, and the IP address and port number of the Admin Service.

Options

Usage Options:

`-a, --admin name`

   Specifies the user name of the GroupWise Super Admin.

`-ca` *`ip_address:port`*

   Specifies the IP address and port number of the Admin Service certificate authority (CA) on the primary domain server.

`-db, --database /`*`path_to_database`*

   Specifies the path to the domain or post office database for which the local Admin Service is running.

`-n, --name` *`poa_name`*

   Specifies the object name of the POA when multiple POAs run for a single post office.

`-p, --password [`*`password`*`]`

   Provides the password for the GroupWise Super Admin. Do not specify the password on the command line if you want to be prompted for it.

Help Options:

`--help, -?`

   Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

    Script file that runs GWAdminUtil.

/opt/novell/groupwise/certificates/*system_guid*/admin.*.*

    Certificate and key files created for domain servers and post office servers.

admin.*domain*.crt

    Server certificate file for a domain/MTA.

admin.*domain*.key

    Private key file for a domain server certificate.

admin.*domain.post_office*.crt

    Server certificate file for a post office/POA.

admin.*domain.post_office*.key

    Private key file for a post office server certificate.

Examples

This program normally runs as `root`.

gwadminutil certinst -db /gwsystem/provo1 -ca 172.16.5.18:9710 -a admin -p

    Contacts the Admin Service CA on the primary domain server and installs a new certificate on the local server.

See Also

gwadminutil(1), gwadminutil-certinst(1), gwadminutil-dbinfo(1), gwadminutil-installcfg(1), gwadminutil-setadmin(1)

gwadminutil-installcfg(1)

Name

`gwadminutil-installcfg` - The GroupWise Administration Utility installation configuration option.

Syntax

```
gwadminutil installcfg --mode user -u user_name -p [password]
gwadminutil installcfg --mode disabled | token
```

Description

Sets the authentication mode for the GroupWise Installation console.

By default, the GroupWise Installation Wizard passes a token to the Installation console to allow access. The token is valid for one hour. If the installation process is interrupted and more than an hour passes, your session in the Installation console is terminated. You can start a new session by reusing the Configure option in the Installation Wizard to generate a new token.

As an alternative, you can configure the GroupWise server so that a user name and password can be used to access the Installation console, instead of the token generated by the Installation Wizard. You can also prevent further installations on the GroupWise server until you establish a different authentication mode.

Options

Usage Options:

-m, --mode token|user|disabled

Specifies the authentication mode that is required to access the Installation console on the server.

-p, --password [*password*]

When the authentication mode is set to `user`, specifies the password that must be given in order to access the Installation console. Do not specify the password on the command line if you want to be prompted for it.

-u, --username *user_name*

When the authentication mode is set to `user`, specifies the user name that must be given in order to access the Installation console.

Help Options:

--help, -?

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

> Script file that runs GWAdminUtil.

Examples

This program normally runs as `root.`

gwadminutil installcfg --mode user -u gsmith -p

> Changes from token authentication to user authentication, and specifies the user name that is required to access the Installation console. Because no password is specified with the `-p` option, you are prompted to provide the password in order to put the authentication method into affect.

gwadminutil installcfg --disabled

> Prevents future access to the Installation console.

gwadminutil installcfg --token

> Returns to the original authentication method for the Installation console, so that you must use the Configure option in the Installation Wizard in order to access the Installation console.

See Also

gwadminutil(1), gwadminutil-certinst(1), gwadminutil-dbinfo(1), gwadminutil-installcfg(1), gwadminutil-setadmin(1)

gwadminutil-setadmin(1)

Name

`gwadminutil-setadmin` - The GroupWise Administration Utility Super Admin user option

Syntax

`gwadminutil setadmin -d` *path* `-p [`*password*`] -a` *user_name*

Description

Changes the user name and password for the GroupWise Super Admin.

Options

Usage Options:

-a, --admin *user_name*

Specifies the new user name for the GroupWise Super Admin for your GroupWise system.

-d, --domain */path_to_domain_database*

Specifies the path to the primary domain database.

-p, --password [password**]**

Provides the password for the GroupWise Super Admin. Do not specify the password on the command line if you want to be prompted for it.

Help Options:

--help, -?

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

gwadminutil setadmin -d /gwsystem/provo1 -a supergw -p

Specifies the primary domain of the GroupWise system, sets the user name of the GroupWise Super Admin to `supergw`, and prompts for the password for the new `supergw` user.

See Also

gwadminutil(1), gwadminutil-certinst(1), gwadminutil-dbinfo(1), gwadminutil-installcfg(1), gwadminutil-setadmin(1)

4 GroupWise Administration Utility - Other Options

Reference the sections below for other command options in the GroupWise Administrator Utility.

- "gwadminutil-config(1)" on page 46
- "gwadminutil-dbinfo(1)" on page 48
- "gwadminutil-services(1)" on page 50

gwadminutil-config(1)

Name

`gwadminutil-config` - The GroupWise Administration Utility configuration option.

Syntax

`gwadminutil config [-cluster enable] [-defaultport port_number] []`

Description

Provides configuration alternatives for the following aspects of the GroupWise Admin Service:

- Support in the GroupWise Admin console for clustering of the GroupWise agents
- Default port for the GroupWise Admin console

Options

Usage Options:

`-cluster enable`

Enables the GroupWise Admin Service to run in a clustering environment. You must run this command on each node in the cluster where a domain or post office is located.

This option causes certificate files to be stored under a `certificates` folder that is a peer to the domain or post office folder (instead of being stored under `/opt/novell/groupwise/certificates`). It also causes agent log files to be stored under a `log` folder that is a peer to the domain or post office folder (instead of being stored in `/var/log/novell/groupwise`). For example, the folder structure for a cluster resource on an NSS volume would be:

```
/media/nss/VOLUME/domain_or_post_office
/media/nss/VOLUME/groupwise/certificates
/media/nss/VOLUME/groupwise/agents/log
```

`-defaultport admin_service_port_number`

Establishes the default port number for the GroupWise Admin console, when the GroupWise Admin Service is installed but is not configured in the Installation console.

Run the command with no options to display the current configuration settings.

Help Options:

`--help, -?`

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

gwadminutil config

Displays the current configuration settings.

gwadminutil config -cluster enable

Enables the GroupWise Admin console to manage domains and post offices that are cluster resources located on multiple nodes in a cluster.

gwadminutil config -defaultport 9720

Establishes the default port number for an unconfigured GroupWise Admin Service that is located on a server where the standard default port of 9710 is already in use.

gwadminutil config -cluster []

Disables clustering functionality in the Admin console.

gwadminutil config -defaultport []

Resets the default Admin Service port back to the default of 9710.

See Also

gwadminutil(1)

gwadminutil-dbinfo(1)

Name

`gwadminutil-dbinfo` - The GroupWise Administration Utility database information option.

Syntax

`gwadminutil dbinfo` *path* [`-n` *name*]

Description

Displays information about a GroupWise domain or post office database.

The following information is displayed about the database:

- GroupWise system name and GUID
- Name of the domain or post office
- Database version
- GroupWise Admin Service IP address and port

Options

Usage Options:

/path_to_database

Specifies the full path to the domain or post office database.

-n, --name *poa_name*

Specifies the object name of the POA when multiple POAs run for a single post office.

Help Options:

--help, -?

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

Script file that runs GWAdminUtil.

Examples

This program normally runs as `root.`

`gwadminutil dbinfo /gwsystem/provo1`

    Displays database information about the Provo1 domain.

`/gwadminutil dbinfo /gwsystem/engineering`

    Displays database information about the Engineering post office.

See Also

gwadminutil(1)

gwadminutil-services(1)

Name

`gwadminutil-services` - The GroupWise Administration Utility agent services option.

Syntax

```
gwadminutil services [-i path] [-dva [name] [path]] [-s name]]
                     [-l [types] [-q [name]]
                      [-start [name]] [-stop [name]] [-n name] [-show]
```

Description

Manages the GroupWise agent services. (Some functionality overlaps with the `grpwise` script.)

- Installs the DVA service on a server where no post office is located.
- Installs the POA service on a server where no post office is located.
- Lists agent service information as it appears in the `gwha.conf` file.
- Starts and stops the agent services, and lists agent status, similar to the functionality of the `grpwise` script.
- Starts the agents with a user interface on the server console. An agent that runs with a user interface cannot be managed in the GroupWise Admin console because it is not running as a service.

Options

Usage Options:

-dva [*object_name*] [*/path_to_startup_file*]

Installs the DVA service on a server where no post office is located. Use with the `-i` option. If there are multiple DVAs for the post office, specify the DVA object name. If the DVA startup file is not local with the running DVA, such as in a clustering environment where the DVA fails over with the POA, specify the path to the DVA startup file. Specify a unique startup file name to accommodate multiple DVAs on a single cluster node.

-i, --install */path_to_database*

Installs the POA service for a post office database. Installs the MTA service for a domain database. Installs the GWIA software for a GWIA database. Add the `-dva` option to install the DVA service. You must use the GroupWise Installation Wizard to install the GroupWise Server software on the server, and start the Admin Service, before you can install agent services.

-l, --list *types*

Lists agent service information as it appears in the `gwha.conf` file. Optionally, include one or more agent types in a comma-separated list (POA,MTA,GWIA,DVA).

-n, --name *poa_name*

Specifies the object name of the POA when multiple POAs run for a single post office.

-q, --query *name*

Displays current status information for all agents on the server. Optionally, specify a service name, partial service name, or wildcard pattern. This is similar to `grpwise status`.

-s, --service *name*

Specifies an agent service name. Use with the `-u` option to uninstall a service by its name rather than by its database location.

-show

Starts an agent with a user interface. Use with the `-start` option. An agent that runs with a user interface cannot be managed in the GroupWise Admin console because it is not running as a service.

-start *agent_name*

Starts the specified agent service. You can specify a full agent name, a partial agent name, or a wildcard pattern. This is similar to `grwpise start` *object*, but is more flexible.

-stop *agent_name*

Starts the specified agent service. You can specify a full agent name, a partial agent name, or a wildcard pattern. This is similar to `grwpise start` *object*, but is more flexible.

-u, --uninstall */path_to_database* | **-s** *agent_service*

Uninstalls the agent service associated with the specified database or service name. This removes the agent service information from the `gwha.conf` file. Do this after you have deleted the domain or post office that the agent serviced.

-upgrade

Upgrades Windows service definitions for the agents. Not applicable on Linux

Help Options:

--help, -?

Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwadminutil

Script file that runs GWAdminUtil.

Examples

This program normally runs as `root`.

gwadminutil services -l

Lists all agent information in the `gwha.conf` file.

`gwadminutil services -l mta,poa`

> Lists MTA and POA information in the `gwha.conf` file, but not the information for other agents on the server.

`gwadminutil services -i -dva`

> Installs the DVA service on a server where no post office is located. You must use the GroupWise Installation Wizard to install the GroupWise Server software on the server, and start the Admin Service, before you can install agent services.

`gwadminutil services -i -dva /media/nss/VOL1/dva1.dva`

> Installs a DVA service named `dva1` in a Novell Cluster Services cluster where the startup file is located in the `/media/nss/VOL1` folder. You must use the GroupWise Installation Wizard to install the GroupWise Server software on the server, and start the Admin Service, before you can install agent services.

`gwadminutil services -i /gwsystem/engineering`

> Installs the POA service for the Engineering post office. You must use the GroupWise Installation Wizard to install the GroupWise Server software on the server, and start the Admin Service, before you can install agent services.

> If you are installing a second POA service on a different server from where the post office is located, you must be able to specify the post office database location across a file system mount.

`gwadminutil services -i /gwsystem/provo1`

> Installs the MTA service for the Provo1 domain. You must use the GroupWise Installation Wizard to install the GroupWise Server software on the server, and start the Admin Service, before you can install agent services.

`gwadminutil services -i /gwsystem/provo1/wpgate/gwia`

> Installs the GWIA service for the Provo1 domain. You must use the GroupWise Installation Wizard to install the GroupWise Server software on the server, and start the Admin Service, before you can install agent services.

`gwadminutil services -q`

> Displays current status information for all agents on the server. This is similar to `grpwise status`.

`gwadminutil services -q poa`

> Displays current status information for the POA. This is similar to `grpwise status engineering.provo1`.

`gwadminutil services -start provo1`

> Starts the MTA for the Provo1 domain. This is similar to `grpwise start provo1`.

`gwadminutil services -stop engineering`

> Stops the POA for the Engineering post office. This is similar to `grpwise stop provo1.engineering`.

`gwadminutil services -u /gwsystem/provo1`

> Removes the Provo1 MTA information from the `gwha.conf` file. Do this after you have deleted the Provo1 domain.

`gwadminutil services -u -s gwia`

Removes the GWIA information from the `gwha.conf` file.

See Also

gwadminutil(1)

5 GroupWise Service Control Utility

Reference the following section for GroupWise Service Control Utility commands:

gwsc(1) (page 56)

gwsc(1)

Name

`gwsc` - The GroupWise Service Control Utility

Syntax

`gwsc` *option* [*arguments*]

Description

Provides a short-hand version of the `gwadminutil services` command.

Options

Usage Options:

option [arguments]

　　Accepts all of the same options and arguments as the `gwadminutil services` command.

Help Options:

--help, -?

　　Displays the help information and exits.

Files

/opt/novell/groupwise/admin/gwsc

　　Script file that runs GroupWise Service Control Utility.

Examples

This program normally runs as `root`.

gwsc -l

　　Lists all agent information in the `gwha.conf` file. Equivalent to `gwadminutil services -l`.

gwsc -i -dva

　　Installs the DVA service on a server where no post office is located. Equivalent to `gwadminutil services -i -dva`.

gwsc -q

　　Displays current status information for all agents on the server. Equivalent to `gwadminutil services -q`.

gwsc -start provo1

>Starts the MTA for the Provo1 domain. Equivalent to `gwadminutil services -start provo1`.

gwsc -stop engineering

>Stops the POA for the Engineering post office. Equivalent to `gwadminutil services -stop engineering`.

See Also

gwadminutil(1), gwadminutil-services(1)

6 GroupWise Database Utilities

Reference the sections below for the GroupWise Database Utilities commands.

gwcheck(1)

Name

`gwcheck` - GroupWise Check Utility

Syntax

```
gwcheck [--opt=[/path_name/]options_file [--batch] ]
                                         [--po=post_office_folder]
                                         [--pa=archive_folder]
                                         [--pr=remote_mailbox_folder]
```

Description

GroupWise Check (GWCheck) is a tool provided for GroupWise to check and repair GroupWise user, message, library, and resource databases without needing the Administration Console. In addition to checking and repairing databases in the post office, it also checks and repairs users' remote, caching, and archive databases on user workstations or other personal locations.

GWCheck provides a user interface similar to the Mailbox/Library Maintenance feature in the GroupWise Admin console. The list below illustrates the types of database repair that you can perform using GWCheck. Click **Help** in GWCheck for more information about the database repair options that are available.

Analyze/Fix Databases

```
Structure
     Index Check
Contents
     Collect Statistics
     Attachment File Check
Fix Problems
     Update User Disk Space Totals
```

Expire/Reduce Messages

```
Reduce Only
Expire and Reduce
     Items Older Than
     Downloaded Items Older Than
     Items Larger Than
     Trash Older Than
     Reduce Mailbox To
     Reduce Mailbox to Limited Size
Include
     Received Items
     Sent Items
     Calendar Items
     Only Backed-Up Items
     Only Retained Items
```

Mailbox Statistics

```
Mailbox Statistics
     Box Limit
Expire Statistics
     (same as the Expire and Reduce options)
```

Archive/Delete Documents

(no options)

Delete Activity Logs

```
Delete Logs Older Than nn Days
```

Analyze/Fix Library

```
Verify Library
Fix Document/Version/Element
Verify Document Files
Validate All Document Security
Synchronize User Name
Remove Deleted Storage Areas
     Move Documents First
Reassign Orphaned Documents
     New Author
Reset Word Lists
```

Audit Report

```
Log Accounts without Activity for Previous nn Days
```

GWCheck Options File

If you repeatedly use the same set of database repair options, you can save them in an options file. To create an options file, start GWCheck, select database repair options as needed, then click **Save**. By default, the options file is named `gwcheck.opt` and is saved in your home folder. You can change the file name and location as needed.

You can retrieve the options file from the user interface or you can specify the options file on the command line. The options file is created in XML format.

After you have created one or more options files, you can run GWCheck as a script, rather than using the user interface.

Options

Usage Options:

`--opt=`*`options_file`*

Specifies a database maintenance options file created in a GWCheck session. This starts GWCheck with the same options settings as the session in which the options file was created.

By default, this file is saved in your home folder. If you want to save it to another folder, specify a path relative to your home folder or specify an absolute path.

`--batch`

Runs GWCheck in the background, without a user interface. Use an options file to specify the database repair options.

`--po=`*`post_office_folder`*

Specifies the path to the post office folder.

`--pa=`*`archive_folder`*

Specifies the path to the archive folder.

`--pr=`*`remote_mailbox_folder`*

Specifies the path to the remote mailbox folder.

Help Options:

`--help, -?`

Displays the help information and exits.

Files

`[/`*`path_name`*`/]gwcheck.opt`

Options file. Create this file by setting database repair options in GWCheck, then click **Save**. The options file is created in XML format.

Examples

This program normally runs as `root`.

`/opt/novell/groupwise/admin/gwcheck.sh`

Starts the GUI GWCheck.

`/opt/novell/groupwise/admin/gwcheck.sh --opt=gwcheck.opt --batch`

Starts GWCheck with an options file that provides default database repair options and runs it in batch mode so that there is no user interface.

`/opt/novell/groupwise/admin/gwcheck.sh --opt=gwcheck.opt --pa=/home/gsmith/gwarchive`

Runs GWCheck for an archive mailbox.

See Also

gwcheckt(1)

gwcheckt(1)

Name

`gwcheckt` - Text-Based GroupWise Check Utility

Syntax

`gwcheckt` *`options_file`* `[--dump]`

Description

Text-Based GroupWise Check (GWCheckT) is a tool provided for GroupWise to check and repair GroupWise user, message, library, and resource databases without needing the GroupWise Admin console or any GUI environment. Text-Based GWCheckT can perform all of the same types of database repair as the GUI GWCheck, but in a text-only environment.

Text-Based GWCheckT has no user interface. You must create the required options file by using the GUI GroupWise Check (GWCheck), or by using the Mailbox/Library Maintenance feature in the Admin console.

To create the required options file, start the GUI GWCheck in an environment where the X Window System is available, select database repair options as needed, then click **Save**. By default, the options file is named `gwcheck.opt` and is saved in your home folder. You can change the file name and location as needed.

Options

Usage Options:

`[/path_name/]options_file`

Specifies the name of the options file. By default, this file is saved by the GUI GWCheck in your home folder. If you saved it to another folder, specify a path relative to your home folder or specify an absolute path.

`--dump`

Displays the contents of the options file so that you can see what database repair options have been selected.

Help Options:

`--help, -?`

Displays the help information and exits.

Files

options_file

(Required) Use the GUI GWCheck to select and save database repair options in the `gwcheck.opt` file, then run the Text-Based GWCheckT with the options file to execute the database repair options.

Examples

This program normally runs as `root`.

/opt/novell/groupwise/gwcheck/bin/gwcheckt gwcheck.opt

Runs GWCheckT in a text only environment, based on the database repair options provided in the options file created in the GUI GWCheck.

/opt/novell/groupwise/gwcheck/bin/gwcheckt gwcheck.opt --dump

Lists the settings of the database repair options in the specified options file.

See Also

gwcheck(1)

dbcopy(1)

Name

`dbcopy` - GroupWise Database Copy Utility

Syntax

```
dbcopy /source_folder /destination_folder
dbcopy [-b] [-i] [-t] [-v] [-w] /source_folder /destination_folder
dbcopy -m [-d] [-p] [-f] [-s] [-a] [-u] [-k] [-o] [-l]
                                   /source_folder /destination_folder
```

Description

The GroupWise Database Copy utility (DBCopy) copies files from a live GroupWise post office or domain to a static location for backup. During the copy process, DBCopy prevents the files from being modified, using the same locking mechanism used by other GroupWise programs that access databases. This ensures that the backed-up versions are consistent with the originals even when large databases take a substantial amount of time to copy.

In addition to backups, the DBCopy utility is used in conjunction with the GroupWise Server Migration Utility (http://www.novell.com/documentation/groupwise2014), which helps you migrate GroupWise data from NetWare or Windows to Linux. Some startup switches can be used for both backups and migration. Other startup switches are used only for migration.

Options

General Usage Options:

source_folder

Specifies the full path to the domain folder, post office folder, or remote document storage area folder.

destination_folder

Specifies the full path to the folder where you want to copy the domain, post office, or remote document storage area.

Usage Option for Backups Only:

-i mm-dd-yyyy, -increment

Specifies the date of the previous copy of the data. This causes DBCopy to copy only files that have the specified date or newer, such as an incremental backup. There is no default date; you must specify a date or an increment backward from today. Valid increments are -1 to -31.

-j

> Raises the priority of DBCopy processing. By default, if DBCopy detects that a POA is running, it lowers its own priority so that it does not interfere with POA processing. If DBCopy runs at night, when GroupWise users are not active, use the -j switch so that DBCopy does not lower its own priority. This speeds up DBCopy processing.

Usage Options for Backups and Migration:

-b

> Indicates that DBCopy is copying a document storage area, which includes BLOB (binary large object) files. Use this switch only when you need to copy BLOB files.

-t *threads*

> Specifies the number of threads for DBCopy to start for copying data. The default number of threads is 5. Valid values range from 1 to 10.

-v

> Turns on verbose logging, which provides more detail than the default of normal logging. DBCopy creates a log file named *mmddgwbk.nnn*. The first four characters represent the date. A three-digit extension allows for multiple log files created on the same day. The log file is created at the root of the destination domain, post office, or document storage area folder. In addition to status and error messages, it lists any remote document storage areas associated with a post office.

-w

> Turns on continuous logging to the screen.

Usage Options for Migration Only:

-m

> Copies all directories and files associated with a domain, post office, or document storage area as part of a migration; also ensures that all folder names and file names are in lowercase.

-d

> Indicates migration of a domain.

-p

> Indicates migration of a post office.

-f

> Indicates that this is the first pass of the migration process.

-s

> Indicates that this is the second pass of the migration process.

-a *ip_address*

> Specifies the IP address to bind to for the migration process. The default is all available IP addresses.

-u *port_number*

Specifies the TCP port number for status requests during the migration.

-k

Skips collecting database size information during the migration.

-o

Skips the second copy of the post office `offiles` folder during the migration.

-l */source_folder*

Performs the GroupWise Check function of `storelowercase` on the migrated GroupWise databases. Its purpose is to do an "in-place" conversion of files and directories to lowercase, rather than as part of a copy operation. For a post office, it also updates the guardian database with the new, lowercase names.

For example, you could use this functionality if you have a domain or post office located on a SAN that was mounted for access by the GroupWise NetWare agents, but you now want to run the GroupWise Linux agents for the domain or post office, because NetWare is not supported as of GroupWise 2012.

Do not use other switches with the -l switch.

Help Options:

--help, -?

Displays the help information and exits.

Files

/path_name/mmddgwbk.nnn

Log file created at the root of the destination folder.

Examples

This program normally runs as `root`.

Backup Examples:

`/opt/novell/groupwise/agents/bin/dbcopy /gwsystem/acct /backups/acct`

Copies a post office to a backup location.

`/opt/novell/groupwise/agents/bin/dbcopy --b /gwsystem/acct_library /backups/acct_library`

Copies a library to a backup location.

Migration Examples:

```
/opt/novell/groupwise/agents/bin/dbcopy -m -f /post_office_folder /
destination_folder
```

Performs the first migration pass for a post office.

```
/opt/novell/groupwise/agents/bin/dbcopy -m -s /post_office_folder /
destination_folder
```

Performs the second migration pass for a post office.

```
/opt/novell/groupwise/agents/bin/dbcopy -m /domain_folder /
destination_folder
```

Migrates a domain.

```
/opt/novell/groupwise/agents/bin/dbcopy -m -b /storage_area_folder /
destination_folder
```

Migrates a document storage area.

See Also

gwtmstmp(1)

gwtmstmp(1)

Name

`gwtmstmp` - GroupWise Time Stamp Utility

Syntax

```
gwtmstmp --postpath post_office_folder [--set] [--get] [--clear]
         [--backup] [--restore] [--retention]
         [--date mm/dd/yyyy] [--time hh:mm[am|pm]] [--gmttime seconds]
         [--userid groupwise_user_name] [--userdb user_database]
```

Description

The GroupWise Backup Time Stamp utility (GWTMSTMP) places date and time information on user databases (`userxxx.db`) in order to support message backup, restore, and retention. The time stamp indicates the last time the database was backed up.

If a user deletes an item from his or her mailbox and purges it from the Trash, the item is removed from the user's database only if the time stamp shows that the item has already been backed up. Otherwise, the item remains in the user's database until the database is backed up, at which time it is purged from the database.

You can run GWTMSTMP on all user databases in a post office or on a single user database. No other databases are affected.

Backup

To ensure thorough user database backups, you can ensure that deleted items are not purged from users' databases until they have been backed up. Two conditions must be met in order to provide this level of protection against loss of deleted items:

- The **Do Not Purge Items Until They Are Backed Up** option must be selected for the post office in the Administration Console (**Post Office** object > **Client Options** > **Environment** > **Cleanup**).
- User databases (`userxxx.db`) must be time-stamped every time a backup is performed so that items can be purged only after they are backed up.

Restore

You can use GWTMSTMP to manually add the restore time stamp to the database. The restore time stamp is not required for any GroupWise feature to work properly. Its primary purpose is informational.

Retention

If you use a message retention application, the application should automatically add the retention time stamp after retaining the database's messages. Any messages with dates that are newer than the retention time stamp cannot be purged from the database. You can use GWTMSTMP to manually add a retention time stamp.

Modified Retention

If you use a message retention application, you might need to retain items more than once if you want to capture changes to personal subjects and personal attachments on items. You can use GWTMSTMP to manually update the retention time stamp on modified items, so that they are retained again.

Options

Usage Options:

`--postpath /post_office_folder, -p /post_office_folder`

    (Required) Specifies the full path to the post office folder where the user databases to time-stamp are located.

`--set, -s`

    Sets the current date and time (of backup, restore, or retention) on user databases.

`--get, -g`

    Lists existing backup, restore, and retention time stamp information for user databases. If no time stamps are set, no times are displayed. If no other operational switch is used, --get is assumed.

`--clear, -c`

    Removes time stamps (of backup, restore, or retention) from user databases.

`--backup, --restore, --retention, --modifiedretention, -b, -r, -n, -mn`

    Specifies the type of time stamp (backup, restore, retention, or modified retention) on which to perform the get or set operation. If no time stamp type is specified, the operation is performed on the backup time stamp. Multiple time stamp types can be specified.

`--date mm/dd/yyyy, -d mm/dd/yyyy`

    Specifies the date that you want placed on user databases. If no date is specified, the current date is used.

    Use the format *mm/dd/yyyy* (for example, 05/18/2014 for May 18, 2014). If your locale does not use the *mm/dd/yyyy* format, use `gwtmstmp --help` to determine the format for your locale.

`--time hh:mm[am|pm], -t hh:mm[am|pm]`

    Specifies the time that you want placed on user databases. If no time is specified, 00:00 is used.

    Use the format *hh:mm*, expressed in a 24-hour format (for example, 20:45) or expressed with the am or pm option (for example, 8:45pm).

`--gmttime` *`seconds`*, **`-m`** *`seconds`*

> Specifies the number of seconds since midnight on January 1, 1970 Greenwich Mean Time (GMT), that you want placed on the user databases.

`--userid` *`user_name`*, **`-u`** *`user_name`*

> Provides a specific GroupWise user name so that an individual user database can be time-stamped.

`--userdb` *`user_database`*, **`-e`** *user_database*

> Provides a specific user database (`userxxx.db`) so that an individual user database can be time-stamped.

Help Options:

`--help, -?`
> Displays the help information and exits.

Examples

This program normally runs as `root`.

`/opt/novell/groupwise/agents/bin/gwtmstmp -p /gwsystem/acct`
> Checks the existing time stamp on all GroupWise user databases in a post office.

`/opt/novell/groupwise/agents/bin/gwtmstmp -p /gwsystem/acct --set`
> Sets a current time stamp on all user databases in a post office.

See Also

dbcopy(1)

7 Linux GroupWise Agent Commands

Reference the sections below for the GroupWise Agent commands.

grpwise(1)

Name

`grpwise` - GroupWise Script

Syntax

```
grpwise start [object]
grpwise status [object]
grpwise stop [object]
grpwise restart [object]
grpwise restartall
grpwise print [object]
```

The objects are:

```
Admin Service: gwadminservice
POA:           post_office_name.domain_name
MTA:           domain_name
GWIA:          gwia.domain_name
DVA:           gwdva.post_office_name
```

NOTE: The Monitor Agent is not controlled by the `grpwise` script. It has its own script named `grpwise-ma`, which works essentially the same as the `grpwise` script.

Description

The `grpwise` script is created during installation and can control any agents installed on the local server. It is created in the `/etc/init.d` folder. It works in conjunction with the `gwha.conf` file, which is created in the `/etc/opt/novell/groupwise` folder. The `gwha.conf` file lists the agents installed on the local server so that the `grpwise` script knows which agents to start.

The GroupWise agents are Run Control compliant. During installation, a symbolic link is created from `/etc/init.d/grpwise` to `/usr/sbin/rcgrpwise`. Typically, `/usr/sbin` is already on your path. This enables you to run `rcgrpwise` from any folder, rather than changing to `/etc/init.d` in order to run the `grpwise` script.

Options

Usage Options:

start

Starts all GroupWise agents on the local server or the specific GroupWise agent specified in the `grpwise` command. A green "done" indicates that an agent is running. If a green "done" does not appear, the agent was unable to start.

status

Displays status for all GroupWise agents on the local server or the specific GroupWise agent specified in the `grpwise` command. A green "running" indicates that an agent is running. A black "unused" indicates that an agent is not running.

stop

Stops all GroupWise agents on the local server or the specific GroupWise agent specified in the `grpwise` command. A green "done" indicates that an agent is stopped. A black "unused" indicates that an agent is not running and therefore cannot be stopped.

restart

Restarts all GroupWise agents on the local server or the specific GroupWise agent specified in the `grpwise` command. A green "done" indicates that an agent is stopped. A black "unused" indicates that an agent is not running and therefore cannot be stopped. The agents then start again with the standard start status indicators.

print

Lists settings for all GroupWise agents in the `gwha.conf` file or for the GroupWise agent specified in the `grpwise` command.

post_office_name.domain_name

Specifies the POA to start or stop. The syntax allows for multiple POAs running on the same server and specifies which POA to start or stop.

domain_name

Specifies the MTA to start or stop. The syntax allows for multiple MTAs running on the same server and specifies which MTA to start or stop.

gwia.domain_name

Specifies the GWIA to start or stop. The syntax allows for multiple GWIAs running on the same server and specifies which GWIA to start or stop.

gwdva.post_office_name

Specifies the DVA to start or stop. The syntax allows for multiple DVAs running on the server. Running multiple DVAs on a server is not recommended.

Help Options:

--help, -?

Displays the help information and exits.

Files

/etc/opt/novell/groupwise/gwha.conf

GroupWise High Availability Service (gwha) configuration file. This configuration file lists all information necessary to start and stop each GroupWise agent that is installed on the server. For example, an entry for an MTA might look similar to this:

```
server    = /opt/novell/groupwise/agents/bin/gwmta
command   = /etc/init.d/grpwise
startup   = /gwsystem/provo1/provo1.mta
delay     = 2
wait      = 10
```

The `server` = setting specifies the full path to the agent executable.

The `command` = setting specifies the full path to the `grpwise` script.

The `startup` = setting specifies the name of the agent startup file. Agent startup files are found in the following locations:

```
/post_office_folder/post_office_name.poa
/domain_folder/domain_name.mta
/domain_folder/wpgate/gwia/gwia.cfg
/post_office_folder/post_office_name.dva
```

The `delay` = setting controls the length of time between when the script issues the command to start an agent and when the script displays a message indicating that the agent has started. The default delay time is 2 seconds.

The `wait` = setting controls the length of time between when the script issues the command to stop an agent and when the script kills the agent if the agent has not yet stopped. The default wait time is 10 seconds.

By default, the `grpwise` script starts the agents as services with no user interface. If you want a user interface, you can add a `show = yes` line for each agent that you want to have a user interface. An agent that runs with a user interface cannot be managed in the GroupWise Admin console because it is not running as a service.

Examples

This script must be run as `root`. If you have configured the GroupWise agents to run as a non-`root` user, they switch over to that user after they have started.

In all of the examples below, replace `/etc/init.d/grpwise` with `rcgrpwise` to run the script from any folder.

/etc/init.d/grpwise start

Starts all GroupWise agents that are installed on the server.

/etc/init.d/grpwise status development.provo1

Shows the status of the Development post office POA. The Development post office belongs to the Provo1 domain.

/etc/init.d/grpwise stop gwia.provo1

Stops the GWIA that belongs to the Provo1 domain.

Diagnostics

If a GroupWise agent fails to start when using the `grpwise` script, try starting the agent manually, as described in the man page for each agent. You might receive an error message indicating why the agent is failing to start.

If you have changed the non-`root` user that you want the agent to run by modifying the `/etc/opt/novell/groupwise/agents/uid.conf` file, you might have forgotten to delete the corresponding `uid.run` file in the domain or post office folder that the agent services. Delete the `uid.run` file, then use the `grpwise` script to try to start the agent again.

See Also

gwpoa(1), gwmta(1), gwia(1), gwdav(1)

grpwise-ma(1)

Name

`grpwise-ma` - GroupWise Monitor Script

Syntax

`grpwise-ma [start] [stop] [status] [restart]`

Description

The `grpwise-ma` script is created during installation and controls the Monitor Agent. (The `grpwise` script controls the other four GroupWise agents.) The `grpwise-ma` script is created in the `/etc/init.d` folder. It works in conjunction with the `monitor.xml` file, which is created in the `/opt/novell/groupwise/agents/share` folder. The `monitor.xml` file points the Monitor Agent to a domain database (`wpdomain.db`) where it can access a list of agents to monitor in your GroupWise system.

The `grpwise-ma` script includes an `MA_OPTIONS` variable where you can specify Monitor Agent startup switches. This is convenient because the Monitor Agent does not use a startup file as the other GroupWise agents do.

The Monitor Agent is Run Control compliant. During installation, a symbolic link is created from `/etc/init.d/grpwise-ma` to `/usr/sbin/rcgrpwise-ma`. Typically, `/usr/sbin` is already on your path. This enables you to run `rcgrpwise-ma` from any folder, rather than changing to `/etc/init.d` in order to run the `grpwise-ma` script.

Options

Usage Options:

start

> Starts the Monitor Agent as a service on the local server. A green "done" indicates that an agent is running. If a green "done" does not appear, the agent was unable to start.

status

> Displays status for the Monitor Agent. A green "running" indicates that an agent is running. A black "unused" indicates that an agent is not running.

stop

> Stops the Monitor Agent. A green "done" indicates that an agent is stopped. A black "unused" indicates that an agent is not running and therefore cannot be stopped.

restart

> Stops the Monitor Agent. A green "done" indicates that an agent is stopped. A black "unused" indicates that an agent is not running and therefore cannot be stopped. The Monitor Agent then starts again with the standard start status indicators.

Help Options:

--help, -?

> Displays the help information and exits.

Files

/opt/novell/groupwise/agents/share/monitor.xml

> Monitor Agent configuration file. Provides a domain folder where the Monitor Agent can determine what GroupWise agents to monitor. The `monitor.xml` file is created during installation.

/var/opt/novell/groupwise/monitor/gwmonitor.cfg

> Monitor Application configuration file. Provides integration information for passing agent status information gathered by the Monitor Agent to your web browser for display in the main Monitor web console.

Examples

This script does not need to be run as `root`.

In the example below, `/etc/init.d/grpwise-ma` can be replaced with `rcgrpwise-ma` to run the script from any folder.

/etc/init.d/grpwise-ma start

> Starts the Monitor Agent.
>
> You can view the Monitor Agent console at the following URL:
>
> `http://localhost:8200`
>
> You can view the Monitor console at the following URL:
>
> `http://localhost/gwmon/gwmonitor`

See Also

gwmon(1)

gwpoa(1)

Name

`gwpoa` - GroupWise Post Office Agent

Syntax

```
gwpoa [--show] --home /post_office_folder
gwpoa [--show] @/post_office_folder/post_office_name.poa
```

Description

A post office serves as an administrative unit for a group of users and mailboxes. Messages are delivered to mailboxes by the Post Office Agent (POA).

The Post Office Agent (POA) responds to GroupWise client requests for mailbox data and delivers messages between users' mailboxes in a post office. The POA also communicates with the MTA to route messages to and from the post office. In addition, the POA performs a variety of maintenance tasks in the post office.

You can provide POA startup switches on the command line or in a startup file. By default, the startup file is named after the post office and has a `.poa` extension. It is located in the post office folder.

Options

Usage Options:

--home /post_office_folder

Specifies the post office folder, where the POA can access message and user databases. There is no default location. You must use this switch in order to start the POA.

The Installation Wizard automatically sets the --home switch to the post office folder in the POA startup file.

@/post_offce_folder/post_office_name.poa

Specifies the location of the POA startup file. The POA startup file is created in the post office folder and is named after the post office, with a `.poa` extension. The POA startup file includes the --home switch.

For information about agent startup switches, see the *GroupWise Administration Guide*.

--show

Starts the POA with a server console user interface. The agent user interface requires that the X Window System and Open Motif are running on the Linux server.

By default, no user interface is provided for the agents on Linux. An agent that runs with a user interface cannot be managed in the GroupWise Admin console.

The --show startup switch can be used on the command line or in the `gwha.conf` file used by the GroupWise High Availability Service. It cannot be placed in the agent startup file.

Help Options:

--help, -?

Displays the help information and exits.

Files

***/post_office_folder/post_office_name*`.poa`**

Startup file where startup switches are specified.

`/var/log/novell/groupwise/`*post_office*`.poa`*/mmddpoa.nnn*`

Log file where error messages are written.

Examples

By default, this program runs as `root`, but it is preferable to run it as a non-`root` user. For information, see "Running the Linux GroupWise Agents as a Non-root User" in the *GroupWise Installation Guide*.

`/opt/novell/groupwise/agents/bin/gwpoa --home /gwsystem/acct`

Starts the POA by specifying the post office folder.

`/opt/novell/groupwise/agents/bin/gwpoa @/gwsystem/acct/acct.poa`

Starts the POA by specifying the POA startup file.

`/opt/novell/groupwise/agents/bin/gwpoa --show @/gwsystem/acct/acct.poa`

Starts the POA with a user interface. An agent that runs with a user interface cannot be managed in the GroupWise Admin console because it is not running as a service.

Diagnostics

By default, error messages are written to the POA log file located in the `/var/log/novell/groupwise/`*post_office_name*`.poa` folder. You can view agent log files in the agent console.

Typically, you find multiple log files in this folder. The first four characters of each file name represent the date. The next three characters identify the agent. A three-digit extension allows for multiple log files created on the same day. For example, a log file named `0518poa.001` indicates that it is a POA log file created on May 18. If you restart the POA on the same day, a new log file is created, named `0518poa.002`.

See Also

grpwise(1)

gwmta(1)

Name

`gwmta` - GroupWise Message Transfer Agent

Syntax

```
gwmta [--show] --home /domain_folder
gwmta [--show] @/domain_folder/domain_name.mta
```

Description

A domain organizes post offices into a logical grouping for administration and routing purposes in your GroupWise system. Messages are transferred between post offices and domains by the Message Transfer Agent (MTA). Messages are transferred to and from your GroupWise system by the Internet Agent (GWIA).

You can provide MTA startup switches on the command line or in a startup file. By default, the startup file is named after the domain and has a `.mta` extension. It is located in the domain folder.

Options

Usage Options:

`--home /domain_folder`

Specifies the domain folder, where the MTA can access the domain database (`wpdomain.db`). There is no default location. You must use this switch in order to start the MTA.

The Installation Wizard automatically sets the --home switch to the domain folder in the MTA startup file.

`@/domain_folder/domain_name.mta`

Specifies the location of the MTA startup file. The MTA startup file is created in the domain folder and is named after the domain, with a `.mta` extension. The startup file includes the --home switch.

For information about agent startup switches, see the *GroupWise Administration Guide*.

`--show`

Starts the MTA with a server console user interface.

By default, no user interface is provided for the agents on Linux. An agent that runs with a user interface cannot be managed in the GroupWise Administration Console.

The --show startup switch can be used on the command line or in the `gwha.conf` file used by the GroupWise High Availability Service. It cannot be placed in the agent startup file.

Help Options:

`--help, -?`

Displays the help information and exits.

Files

`/domain_folder/domain_name.mta`

Startup file where startup switches are specified.

`/var/log/novell/groupwise/domain.mta/mmddmta.nnn`

Log file where error messages are written.

Examples

By default, this program runs as `root`, but it is preferable to run it as a non-`root` user. See the *GroupWise Installation Guide* (http://www.novell.com/documentation/groupwise2014) to set up the agent to run as a non-`root` user.

`/opt/novell/groupwise/agents/bin/gwmta --home /gwsystem/provo1`

Starts the MTA by specifying the domain folder.

`/opt/novell/groupwise/agents/bin/gwmta @/gwsystem/provo1/provo1.mta`

Starts the MTA by specifying the MTA startup file.

`/opt/novell/groupwise/agents/bin/gwmta --show @/gwsystem/provo1/provo1.mta`

Starts the MTA with a user interface. An agent that runs with a user interface cannot be managed in the GroupWise Admin console because it is not running as a service.

Diagnostics

Error messages are written to the MTA log file located in the `/var/log/novell/groupwise/` `domain_name`.mta folder. You can view agent log files in the agent console.

Typically, you find multiple log files in this folder. The first four characters of each file name represent the date. The next three characters identify the agent. A three-digit extension allows for multiple log files created on the same day. For example, a log file named `0518mta.001` indicates that it is an MTA log file created on May 18. If you restart the MTA on the same day, a new log file is created, named `0518mta.002`.

See Also

grpwise(1)

gwia(1)

Name

`gwia` - GroupWise Internet Agent

Syntax

```
gwia [--show] --home /domain_folder/wpgate/gwia
gwia [--show] @/domain_folder/wpgate/gwia_folder/gwia.cfg
gwia [--show] --dhome /domain_folder/wpgate/smtp_folder
gwia [--show] --hn hostname
```

Description

The Internet Agent (GWIA) converts GroupWise messages into SMTP format and transfers them to the Internet SMTP hosts where recipients are located. In reverse, it receives SMTP messages from Internet SMTP hosts and converts them into GroupWise format. The GWIA can also provide access for POP3 and IMAP4 clients to access GroupWise mailboxes.

You can provide GWIA startup switches on the command line or in a startup file. At startup, the GWIA reads its startup file, `gwia.cfg`. It is located in the `/domain_folder/wpgate/gwia` folder.

Options

Usage Options:

--home `/domain_folder/wpgate/gwia`

Specifies the GWIA home folder (`/domain_folder/wpgate/gwia`), where the GWIA can find its databases, input/output queues, and configuration files. There is no default location. You must use this switch in order to start the GWIA.

The Installation Wizard automatically sets the --home switch to the domain folder in the MTA startup file.

@`/domain_folder/wpgate/gwia/gwia.cfg`

Specifies the location of the GWIA configuration file (`gwia.cfg`). The `gwia.cfg` file is created in the `/domain_folder/wpgate/gwia` folder. The `gwia.cfg` file includes the --home switch.

For information about agent startup switches, see the *GroupWise Administration Guide*.

--dhome `/domain_folder/wpgate/SMTP_folder`

Points to the SMTP service work area. This is normally the same as the GWIA folder (`/domain_folder/wpgate/gwia`).

--hn `hostname`

Specifies the hostname that is displayed when someone connects to your GWIA using a Telnet session. You should enter the hostname assigned to you by your Internet service provider.

`--show`

> Starts the GWIA with a server console user interface.
>
> By default, no user interface is provided for the agents on Linux. An agent that runs with a user interface cannot be managed in the GroupWise Administration Console.
>
> The --show startup switch can be used on the command line or in the `gwha.conf` file used by the GroupWise High Availability Service. It cannot be placed in the agent startup file.

Help Options:

`--help, -?`

> Displays the help information and exits.

Files

`/domain_folder/wpgate/gwia/gwia.cfg`

> Startup file where startup switches are specified.

`/var/log/novell/groupwise/gwia.domain_name/mmddgwia.nnn`

> Log file where error messages are written.

Examples

By default, this program runs as `root`, but it is preferable to run it as a non-`root` user. See the *GroupWise Installation Guide* (http://www.novell.com/documentation/groupwise2014) to set up the agent to run as a non-`root` user.

`/opt/novell/groupwise/agents/bin/gwia --home /gwsystem/provo1/wpgate/gwia`

> Starts the GWIA by specifying its GWIA folder in the domain.

`/opt/novell/groupwise/agents/bin/gwia @/gwsystem/provo1/wpgate/gwia/gwia.cfg`

> Starts the GWIA by specifying its startup file.

`/opt/novell/groupwise/agents/bin/gwia --show @/gwsystem/provo1/wpgate/gwia/gwia.cfg`

> Starts the GWIA with a user interface. An agent that runs with a user interface cannot be managed in the GroupWise Admin console because it is not running as a service.

Diagnostics

Error messages are written to the GWIA log file located in the `/var/log/novell/groupwise/domain_name.gwia` folder. You can view agent log files in the agent console.

Typically, you find multiple log files in this folder. The first four characters of each file name represent the date. The next three characters identify the agent. A three-digit extension allows for multiple log files created on the same day. For example, a log file named `0518gwia.001` indicates that it is a GWIA log file created on May 18. If you restart the GWIA on the same day, a new log file is created, named `0518gwia.002`.

See Also

grpwise(1)

grpwise(1)

gwdva(1)

Name

`gwdva` - GroupWise Document Viewer Agent

Syntax

`gwdva [@/`*`post_office_folder`*`/`*`post_office_name`*`.dva]`

Description

The Document Viewer Agent (DVA) converts attached document files from a wide variety of formats into HTML format for indexing by the POA and for viewing in GroupWise Web. You can run up to three DVAs to service conversion requests for a single instance of the POA. Each DVA must be installed on a different server.

The DVA consists of a master DVA process, and multiple worker DVA processes. The master DVA process communicates with the WebAccess Application and the POA to transfer documents back and forth for conversion. Each worker process converts a single document. Because some documents contain unexpected data, they cannot be successfully converted. Such a conversion failure terminates a worker process, but does not affect the stability of the master DVA process.

You can provide DVA startup switches on the command line or in a startup file. By default, the startup file is named after the post office and has a `.dva` extension. It is located in the `/opt/novell/groupwise/agents/share` folder.

Options

Usage Options:

`--home` *`folder`*

Specifies the path to the DVA working folder. The default location is `/var/opt/novell/groupwise/gwdva`.

`@`*`startup_file`*

Specifies the location of the DVA startup file. The DVA startup file is created in the `/opt/novell/groupwise/agents/share` folder and is named after the post office, with a `.dva` extension.

For information about agent startup switches, see the *GroupWise Administration Guide*.

Help Options:

`--help, -?`

Displays the help information and exits.

Files

/opt/novell/groupwise/share/*post_office_name*.dva

    Startup file where startup switches are specified.

/var/log/novell/groupwise/dva/*mmdd*dva.*nnn*

    Log file where error messages are written.

Examples

This program must run as `root`.

/opt/novell/groupwise/agents/bin/gwdva

    Starts the DVA using its default configuration.

/opt/novell/groupwise/agents/bin/gwdva --home /var/opt/novell/groupwise/gwdva

    Starts the DVA and specifies its working folder.

/opt/novell/groupwise/agents/bin/gwdva @gwdva.dva

    Starts the DVA by specifying its startup file.

Diagnostics

Error messages are written to the DVA log file located in the `/var/log/novell/groupwise/gwdva` folder. You can view agent log files in the agent console.

Typically, you find multiple log files in this folder. The first four characters of each file name represent the date. The next three characters identify the agent. A three-digit extension allows for multiple log files created on the same day. For example, a log file named `0518dva.001` indicates that it is a DVA log file created on May 18. If you restart the DVAA on the same day, a new log file is created, named `0518dva.002`.

See Also

[grpwise(1)](#)

gwmon(1)

Name

`gwmon` - GroupWise Monitor Agent

Syntax

`gwmon --home` *`domain_folder`*

Description

GroupWise Monitor is a monitoring and management tool that allows you to monitor GroupWise agents from any location where you are connected to the Internet and have access to a web browser. Some agent administration can also be performed from your web browser. Monitor provides more detail than is provided on the System page in the GroupWise Admin console.

The Monitor Agent continuously polls the POA, MTA, and GWIA; gathers status information from them; and displays the status information in the Monitor Agent console. The Monitor Agent also services requests for agent status information from the Monitor Application, which displays the Monitor Web console.

Monitor Agent configuration information is stored in the `monitor.xml` file in the `/opt/novell/groupwise/agents/share` folder.

Options

Usage Options:

`--home` *`domain_folder`*

    Specifies a domain folder where the Monitor Agent can access a domain database (`wpdomain.db`). From the domain database, the Monitor Agent can determine which agents to monitor, what user names and passwords are necessary to access them, and so on.

    For information about agent startup switches, see the *GroupWise Administration Guide*.

Help Options:

`--help, -?`

    Displays the help information and exits.

Files

`/opt/novell/groupwise/agents/share/monitor.xml`

    Monitor Agent configuration file. Provides a domain folder where the Monitor Agent can determine which GroupWise agents to monitor. The `monitor.xml` file is created during installation.

/var/log/novell/groupwise/gwmon/*mmdd*mon.*nnn*

　　Log file for Monitor Agent events.

/var/log/novell/groupwise/gwmon/*mmdd*hist.*nnn*

　　Log file for dumps of monitored agent MIB values.

Examples

This program can run as any user.

/opt/novell/groupwise/agents/bin/gwmon --home /gwsystem/provo1

　　Starts the Monitor Agent by accessing the domain database in the specified domain folder.

Diagnostics

The Monitor Agent writes to two different types of log files. Event log files record error messages, status messages, and other types of event-related messages. History log files record dumps of all MIB values gathered during each poll cycle of monitored agents. By default, both types of log files are created in the `/var/log/novell/groupwise/gwmon` folder.

Typically, you find multiple log files in this folder. The first four characters of each file name represent the date. The following characters are `mon` for event log files and `hist` for history log files. A three-digit extension allows for multiple log files created on the same day. For example, a log file named `0518mon.001` indicates that it is a Monitor Agent event log file created on May 18. If you restart the Monitor Agent on the same day, a new event log file is created, named `0518mon.002`.

See Also

grpwise-ma(1)