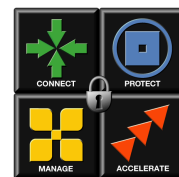




# The Check Point™ Merge Utilities **cp\_merge Tool** for NG with Application Intelligence



---

## In This Document

*Introduction to the cp\_merge Tool* page 1

*Section 1 - Export and Import of policy packages* page 2

*Section 2 - Merging Objects* page 7

## Introduction to the cp\_merge Tool

A Policy Package is a collection of all or some of the following policies, which are all installed together: Security, Quality of Service, Desktop Security.

The merge utility `cp_merge` allows two major functionalities:

- Export and import of policy packages as described under “Section 1 - Export and Import of policy packages” on page 2 below.
- Merge of objects from a given file into SmartCenter database as described under “Section 2 - Merging Objects” on page 7.

## Check Point Products Version

This utility is supported from version NG FP3.

## Installation

The package includes an executable file: `cp_merge`. This file should be copied to a machine where Check Point NG with Application Intelligence is installed.

## Usage

Usage is printed by running the following command: `cp_merge -help`

---

## Section 1 - Export and Import of policy packages

In This Section

*Export command* page 2

*Import/Restore Command* page 3

*Delete command* page 4

*List command* page 6

The `cp_merge` tool enables:

- the exporting of a policy from the policies repository, *as well as*,
- the importing of a saved policy into the policies repository, *and*
- the deletion of a policy from the policies repository.

The *export* action provides the options of leaving the policy package in the active repository, or deleting it as part of the export process. The default policy cannot be deleted during the export action.

The *import* action provides the options of overwriting an existing policy package with the same name, or preventing overwriting when the same policy name already exists.

The *delete* action provides the options of deleting an existing policy package. Note that the default policy can be deleted by delete action.

### Export command

`cp_merge export_policy`

#### Usage

```
cp_merge export_policy [-s <db server>] [-u <user> | -c <certificate file>] [-p <password>] [-n <policy package name> | -l <policy name>] [-d <output directory>] [-f <outputfile>] [-r]
```

#### Arguments

**TABLE 1** `cp_merge export_policy`

command	option	definition
-s	<server>	Specify the database server IP Address or DNS name. <sup>2</sup>
-c	<certificate file>	The path to the certificate file. <sup>1</sup>
-u	<user>	The database administrator's name. <sup>1</sup>
-p	<password>	The administrator's password. <sup>1</sup>
-d	<directory>	Specify the output directory. <sup>2</sup>

---

**TABLE 1** cp\_merge export\_policy (Continued)

<b>command</b>	<b>option</b>	<b>definition</b>
-n	<policy package name>	The policy package to export. <sup>2,3</sup>
-l	<policy name>	Export the policy package which encloses the policy name. <sup>2,3,4</sup>
-r	none	Remove the original policy from the repository. <sup>2</sup>
-f	<file name>	Specify the output file name (where the default file name is <policy name>.pol). <sup>2</sup>

- Note** - 1. Either use certificate file or user and password  
2. Optional  
3. If both -n and -l are omitted all policy packages are exported.  
4. If both -n and -l are present -l is ignored.

## Examples

```
cp_merge export_policy
```

Export all policy packages from the repository to the current directory.

```
cp_merge export_policy -n Standard -f StandardPolicyPackageBackup.pol -d C:\bak
```

Export policy package Standard to file C:\bak\StandardPolicyPackageBackup.pol

## Import/Restore Command

```
cp_merge import_policy|restore_policy
```

### Usage

```
cp_merge import_policy|restore_policy [-s <db server>] [-u <user> | -c  
<certificate file>] [-p <password>] [-n <package name>] [-d <input directory>]  
-f <input file> [-v]
```

---

## Arguments

**TABLE 1-1** cp\_merge import\_policy|restore\_policy

command	option	definition
-c	<certificate file>	The path to the certificate file. <sup>1</sup>
-s	<server>	Specify the database server IP Address or DNS name. <sup>2</sup>
-u	<user>	The administrator's name. <sup>1,2</sup>
-p	<password>	The administrator's password. <sup>1,2</sup>
-d	<directory>	Specify the input directory. <sup>2</sup>
-f	<file name>	Specify the input file name.
-v	none	Override an existing policy if found. <sup>2</sup>
-n	<policy package name>	Rename the policy package to <policy package name> when importing. <sup>2</sup>

**Note** - 1. Either use certificate file or user and password  
2. Optional

The cp\_merge restore\_policy works only locally on the SmartCenter Server and it will not work from remote machines.

## Examples

```
cp_merge import_policy -f Standard.pol -n StandardCopy
```

Import the policy package saved in file standard.pol into the repository and rename it to StandardCopy.

---

**Caution** - A FireWall-1 policy from <policy>.w file can be restored using this utility; however, important information may be lost when the policy is translated into .w format. This restoration should be used only if there is no other backup of the policy.

---

## Delete command

```
cp_merge delete_policy
```

---

## Usage

```
cp_merge delete_policy [-s <db server>] [-u <user> | -c <certificate file>] [-p  
<password>] -n <package name>
```

## Arguments

**TABLE 1-2** cp\_merge delete\_policy

command	option	definition
-s	<server>	Specify the database server IP Address or DNS name. <sup>2</sup>
-c	<certificate file>	The path to the certificate file. <sup>1,2</sup>
-u	<user>	The administrator's name. <sup>1,2</sup>
-p	<password>	The administrator's password. <sup>1,2</sup>
-n	<policy package name>	The policy package to delete.

**Note** - 1. Either use certificate file or user and password  
2. Optional

## Examples

```
cp_merge delete_policy -n Standard
```

Delete the policy package called **standard**.

---

## List command

cp\_merge list\_policy

### Usage

```
cp_merge list_policy [-s <db server>] [-u <user> | -c <certificate file>] [-p <password>]
```

### Arguments

**TABLE 1-3** cp\_merge list\_policy

command	option	definition
-s	<server>	Specify the database server IP Address or DNS name. <sup>2</sup>
-c	<certificate file>	The path to the certificate file. <sup>1,2</sup>
-u	<user>	The administrator's name. <sup>1,2</sup>
-p	<password>	The administrator's password. <sup>1,2</sup>

**Note** - 1. Either use certificate file or user and password  
2. Optional

### Examples

```
cp_merge list -s localhost
```

Lists all policy packages which reside in the specified repository.

### Considerations and Limitations

- Check Point Management GUI clients may interfere with the cp\_merge utility and prevent it from changing the repository.
- The import operation will fail, if the policy uses objects which were deleted after the policy was exported.
- Policy package names are case-sensitive.
- The policy package import is performed one policy package at a time.
- The default policy package may be deleted by the cp\_merge delete\_policy command without warning. If this happens use SmartDashboard to open another policy package.
- Restore action restores FireWall-1 policy only.
- Restore action may not restore all the initial FireWall-1 policy information.

---

## Section 2 - Merging Objects

The `cp_merge` tool merges objects from a given `objects_5_0.c` file to the NG with Application Intelligence SmartCenter Database. This tool eliminates the need to recreate the objects manually on the SmartCenter Database.

In This Section

*Configuration* page 7

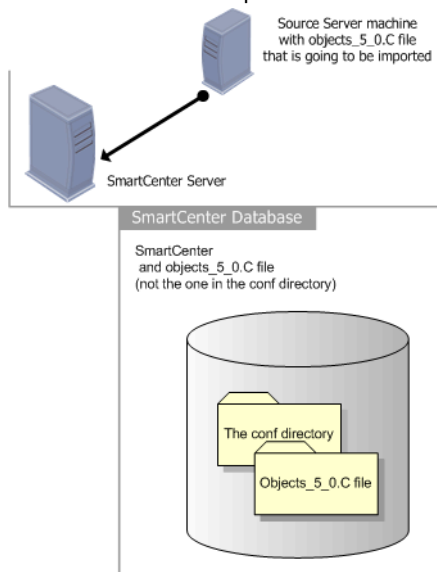
*Steps for merging objects using the cp\_merge tool* page 8

*Which Objects are Imported* page 9

*Considerations and Limitations* page 10

### Configuration

**FIGURE 2** The two possible scenarios for `cp_merge`



- In scenario one of FIGURE 2 above, a remote server machine with an `objects_5_0.c` file imports the necessary objects from its `objects_5_0.c` file into SmartCenter's Database.
- In scenario two of FIGURE 2 above, the `objects_5_0.c` file that you want to import is found in a different directory from the `conf` directory in SmartCenter.

For a full list of considerations to use before using the `cp_merge` tool see “Which Objects are Imported” on page 9.

---

## Steps for merging objects using the cp\_merge tool

- 1 Verify that your `objects_5_0.c` file is from NG FP3 or higher.
- 2 Verify that SmartCenter NG FP3 or higher is running.
- 3 Verify that there are no clients (such as SmartDashboard) connected to the SmartCenter NG FP3 or higher because the `cp_merge` tool locks the database.
- 4 Run the command in test mode (highly recommended) which does not import the objects, it just lists any conflicts or errors. Fix the errors before moving on to step 5. Some of items reported may need to be repaired manually after you complete the automatic import via `cp_merge`. For help and examples of some of these considerations see the section called “Considerations and Limitations” on page 10.  
**Note** - The original (SmartCenter) `objects_5_0.C` file is automatically backed up when `merge_objects` is run in anything but test mode. The name of the backed up file (in the SmartCenter conf directory) is: `objects_5_0.C_before_merge`.
- 5 From the command line run the command:

```
cp_merge merge_objects [-s <db server>] [-u <user> | -c <certificate file>][-p <password>] [-d <input directory>]
```

### Explanation of Flags:

**TABLE 2-1** cp\_merge arguments

command	option	definition
-t	none	Run the tool in test mode without importing objects to make sure that all errors and conflicts are either eliminated or can be fixed manually after merging the objects. <sup>2</sup>
-s	<server>	Specify the target database, server IP, and name. <sup>2</sup>
-c	<certificate file>	The path to the certificate file. <sup>1,2</sup>
-u	<user>	The administrator's name. <sup>1,2</sup>
-p	<password>	The administrator's password. <sup>1,2</sup>
-d	<directory>	Specify the working directory where the imported <code>objects_5_0.C</code> file is located. <sup>2</sup>

- Note** - 1. Either use certificate file or user and password  
2. Optional

- 6 The tool will prompt you to save your merged changes. If you agree to save, your new objects will be merged, otherwise the `cp_merge` tool will not save and your SmartCenter database will not be changed.

- 
- 7 To check if your objects imported correctly, launch SmartDashboard and look for the imported objects. Depending on what you imported, you should see them in Network Objects, Services, Servers or other places.

## After Performing the Merge

- 8 Re-establish SIC with the imported entities that need SIC.
- 9 In **SmartDashboard > Manage > Servers > Show LDAP Account Unit**, go through each imported Account Unit and from:
- the **Objects Management** tab select the correct server from the **Manage Objects On** drop down list.
  - the **Servers** tab select the correct server from the **Early Version Compatibility server**. If you imported Network Objects with pre-shared IKE pre-shared secrets, redefine the secrets.

In **SmartDashboard > Network Objects** open the Objects which contain secrets via the **VPN Manager** tab > click the **Traditional mode configuration** button > **Edit Secrets** and re-establish your secret.

## Which Objects are Imported

Only the following tables (all of which are from `objects_5_0.c`) are imported:

- accounting\_schemes
- ce\_properties
- communities
- credentials\_manager
- customers
- desktop\_profiles
- encryption
- ldap
- methods
- network\_objects
- opsec
- products
- qos
- resources
- securemote
- servers
- services
- sofaware\_gw\_types
- svn
- times
- trusts
- web\_authority\_URLs
- web\_authority\_must\_rules
- web\_authority\_allow\_rules
- web\_authority\_effect\_rules
- web\_sites

**Note** - Users and policies are not within the scope of this tool.

---

## Considerations and Limitations

- Using the `cp_merge` utility to merge a large number of objects (more than 10,000) from two SmartCenter Servers may not work. This is because at some point two main audit logs are generated. If you have a large number of objects, and you wish to perform the merge even though from some point the audit logs will not be generated, then do as follows:
  - Define the environment variable `FWM_ALLOW_AUDIT_FAILURE` from a shell.
  - Use the `cp_merge` command from the same shell.
- If there is a validation problem such as the name of the object is a reserved word, the object will not be imported and you will receive notification.
- If you had a primary management defined in the `objects_5_0.c` file it turns into a secondary management after the import.

## Example of Possible Errors

```
The following objects will not be imported:

July
Gateway_A

Do you want to see an expanded report (y/n) ? y

The following object will not be imported because one of their
attributes' values is not valid. Use the Check Point Database Tool to
correct the objects on the source Management Server:

July : Failed to validate the name of 'July' - Token is a reserved word

Gateway_A : Error in field <firewall_setting> at object <Gateway_A>
--> Error in field 'fwver' at object <Gateway_A> --> '6.0' is not in
the list of valid values '{5.0,4.1,4.0,3.0}'

The following groups will be imported without their members. After the
merge, use SmartDashboard to add the members manually.

Marketing_Gateways
```

### Explanation of the Example

#### 1) The following objects will not be imported:

If you receive this error, you have a validation problem(s). You are offered the choice of seeing an expanded report which provides you with more information such as validation problems like the ones in the example that can be fixed by you.

- `July` can be fixed via SmartDashboard by choosing another name for the object that is not a reserved word.

- 
- Gateway\_A can be fixed by editing the source objects\_5\_0.c file and entering a valid value in the `fwver` field. The valid values are the values located inside the brackets `{}`.

**Note** - Some validation problems are better fixed by you manually after merging by redefining the object in the Server's new database.

- 2) **The following groups will be imported without members:** If you are trying to import a group that is a participant in a circular reference, `cp_merge` warns you that this group may be empty when it is imported. If you get this message while running in test mode, look at the name of the group, remember its members and re-establish it after the merge takes place.