

# Managing Extensions in Your Enterprise

Securely manage Chrome extensions at scale

# Table of contents

## **Purpose of this guide**

## **Introduction**

### **Considerations for managing Chrome extensions**

- What are extension permissions?

- Deciding which extensions to allow

## **Manage extensions**

### **Option 1: Block extensions based on their permissions**

- Manage extensions by their permissions in the Google Admin console

- Manage extensions by their permissions in Group Policy

### **Option 2: Manage extensions by policy**

- Configure the extension policy using the Windows Registry

- Configure using a JSON string in Windows Group Policy Editor

- Additional setting: Prevent extensions from altering webpages

### **Option 3A: Allow or block extensions in the Google Admin console**

- Allow all extensions except those you want to block

- Block all extensions except those you want to allow

- Block or allow one extension

- Force-install an extension

- Force-install several extensions

- Force-install one extension

### **Option 3B: Allow or block extensions in Group Policy**

- Allow all extensions except those you want to block

- Block or allow one extension

- Force-install an extension

## **Create your own on-premises web store**

- Requirements

- Publishing your extension

- Publishing updates to your extension

- Distributing privately hosted extensions

## **Manage extensions using Chrome Browser Cloud Management**

## **Additional resources**

## Purpose of this guide

There are many useful extensions built for the Chrome Browser that empower workers and make workplaces more efficient. However, given the sheer number of extensions that might be running on users' computers at any given time, it can be daunting for IT administrators to monitor and control these extensions.

This guide is for IT administrators who are looking for best practices to manage Chrome Browser extensions in their organizations. It provides steps for managing extensions using both the Google Admin console and Windows Group Policies.

This guide is organized by the methods that you can use to manage extensions. You can:

1. Block extensions based on their permissions
2. Manage extensions by policy
3. Allow or block extensions in the Google Admin console or Windows Group Policy
4. Create your own on-premise web store (not recommended as a best practice)
5. Manage extensions using [Chrome Browser Cloud Management](#) (new in April 2019)

What's covered	Instructions, recommendations, and critical considerations for managing extensions for Chrome Browser in an enterprise
Primary audience	Microsoft® Windows® and Chrome Browser administrators
IT environment	Microsoft Windows 7 and later
Takeaways	Best practices for managing extensions with Chrome Browser

*Last updated:* April 7, 2019

*Published location:* <https://support.google.com/chrome/a/answer/9296680>

**Third-party products:** This document describes how Google products work with the Microsoft Windows operating systems and the configurations that Google recommends. Google does not provide technical support for configuring third-party products. Google accepts no responsibility for third-party products. Please consult the product's website for the latest configuration and support information. You may also contact Google Solutions Providers for consulting services.

©2019 Google LLC All rights reserved. Google and the Google logo are registered trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated. [EXTENSIONS-en-1.0]

## Introduction

Organizations want to protect their user data and more easily evaluate browser extensions that are safe and relevant to their enterprise. IT administrators want to:

1. Prevent bad apps and extensions from being installed.
2. Keep extensions that users need.
3. Provide limited access to user and company data.

The goal of this guide is to clarify how IT administrators manage extensions in their enterprise and provide a safe, productive experience for their users. There are multiple methods to manage extensions. This guide walks you through the different options and helps you pick the right method for your enterprise.

## Considerations for managing Chrome extensions

Your users need access to certain apps, sites, and extensions to do their jobs. As the IT administrator, you need to protect user and company data. An effective security strategy involves asking the right questions for your enterprise and how Chrome can fit your company's needs. Key questions to ask:

- What regulations and compliance measures do I need to adhere to?
- Do some extensions ask for overly broad permissions, which could go against my company's data security policies?
- How much user or corporate data is stored on my user's machines?

As you make these decisions, Chrome Browser provides granular policies that allow you to:

- Block or allow extensions on users computers based on your data protection policies.
- Force-install extensions on your users machines so they have tools that they need to be productive.
- Whitelist or blacklist extensions to allow the least amount of rights needed for your users to work.

The traditional model to manage extensions has been to whitelist and blacklist specific extensions. However, Chrome also allows you to manage permissions requested by extensions. Using this other model, you can decide which rights and permissions you want to allow extensions to use on your machines, and then enforce a global policy that will allow or block extensions that meet your requirements.

## What are extension permissions?

Extensions can require rights to make changes on a machine or a web page to run properly. These rights are called permissions. Developers must list what rights and access their extensions require. There are 2 main categories, but many extensions have both:

- Site permissions require the extension to list sites it may view or modify.  
Examples: Modify a webpage, access cookies, modify tabs
- Device permissions are the rights needed by an extension on the machine where it's running.  
Examples: Access to USB port / storage / viewing screen; talking to native programs

## Deciding which extensions to allow

To help decide which extensions to allow in your organization:

1. Assemble a list of which extensions employees need on their computers.
2. Test the extensions in a test environment to diagnose any compatibility issues with internal apps.
3. Determine which permissions are required for these extensions to run.

**Testing process:** Before allowing specific permissions (such as site access) in security-conscious organizations, you can look at the web app manifest JSON file in the code of the Chrome web extension. Other organizations might wait for users to request to install specific extensions and validate them before approving them in the organization. Take these steps to see what rights the extension needs:

1. Install the extension from the [Chrome Web Store](#).
2. Test the extension and learn how it works in your enterprise.
3. Review the permissions that the extension requires by navigating to **chrome://extensions**.

After you take these steps, decide whether to allow or block an extension. For example, the Legacy Browser Support extension at [chrome://extensions](#) requests the permissions “Read your browsing history” and “Communicate with cooperating native applications.” Weigh the usefulness of this extension against the level of permissions it requests. After you approve an extension for your organization, manage it using the tools below.

## Manage extensions

Most organizations should manage extensions by their permissions and what websites they have access to. This method is more secure, easier to manage, and is scalable for large organizations. You must use 3 or 4 of the following policies. Link directly to the relevant section in this guide:

- [Blocked/allowed permissions](#)
- [Runtime block hosts](#)
- [Force installed extensions](#)
- [Whitelist/blacklist \(if required\)](#)

Using this method saves you time because you only need to set these once. The days of managing long whitelists and blacklists are gone. You can still include a small blacklist of extensions that should not be installed. And with the run-time hosts policy, your most important sites will be protected. To manage extensions in your organization:

1. Find out which extensions are installed on your users’ computers.
  - **Method 1: Survey:** Ask your coworkers and their managers about what extensions they use regularly. Build a list of the extensions that users need for their jobs.
  - **Method 2: Admin console:** Use [Chrome Browser Cloud Management](#) to see which extensions users have downloaded. Go to the section for browser extensions. For desktop computers, it shows: the version, user- or admin-installed, permissions required, number of installs, and status (active or disabled).

2. Choose which sites you need to be more secure:
  - Find out which sensitive internal websites or domains you need to block extensions from making changes or reading data.
  - Prevent access to these sites by blocking the API calls when the extension is run. These include blocking web requests, reading cookies, JavaScript injection, XHR, etc.
3. Identify which permissions pose potential risks to your users:
  - Audit the extensions you users have installed and see what permissions these require.
  - Some of the permissions that extensions use can be vague. For business critical apps, you can reach out to the app developer or vendor directly to get more information about the extension or look at the code. They should be able to detail the changes that the extension can make on machines and websites.
  - Review the [Declare Permissions list](#), which lists all permissions an extension can use. From this list, you can decide which permissions you want to allow in your organization.
4. Create a master list from the data you collected, including:
  - **Required extensions:** This list could be broken down by department, office location, or other relevant information.
  - **Whitelist:** Required extensions with permissions that would be blocked but will be allowed to run. These extensions are needed by your users or are determined to not be a risk through conversations with the vendor.
  - **Blacklist:** Extensions that are blocked from installation. This list includes the permissions that aren't allowed to run. Also include the core sites and domains to be kept secure and not allowed extension access. Afterward, you can compare this blacklist to others you already have in place. You might find that you can relax your current blacklist policies.
5. Present your master list to your stakeholders and IT team to get buy in.
6. Test out the new policy in your lab or with a small pilot in your organization.
7. Roll out these new sets of policies to employees in phases.
8. Review feedback from your users.
9. Repeat and fine-tune the process monthly, quarterly, or yearly.

With your baseline of allowed permissions enforced and sensitive corporate sites protected, you can provide your enterprise with more security while providing a better experience for users. Employees might install extensions that they couldn't before, but not run them on sensitive business sites.

## Option 1: Block extensions based on their permissions

You can control what extensions your users can install by the permissions. If an installed extension needs a permission that is blocked, it just won't run. The extension isn't removed, just disabled. These steps cover Windows only. For other platforms, see: [CHROME OS DEVICES](#) | [MAC](#) | [LINUX](#)

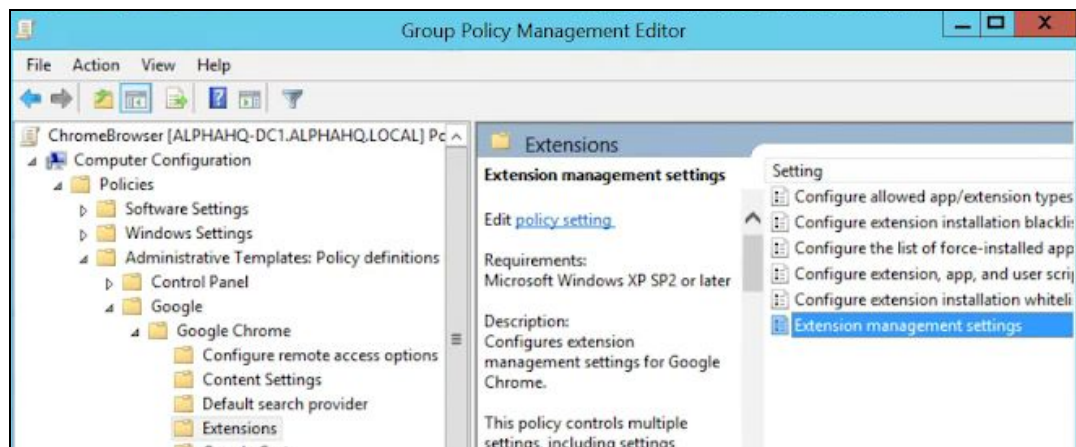
## Manage extensions by their permissions in the Google Admin console

You can block your users from running extensions that need permissions which aren't allowed. For example, you could block an extension that connects to your users USB devices or prevents access to reading cookies.

1. In your Admin console, go to **Devices > Chrome management > User settings**.
2. Select the organizational unit with the users you want to allow extensions for.  
For details, see [Set a Chrome policy for multiple apps](#).
3. Next to Block extensions by permission, select the option to either block or allow the extensions that need the permissions you have chosen.
4. Check each permission to block or allow.  
For complete details, see this [list of permissions](#).
5. Click **Save**.

## Manage extensions by their permissions in Group Policy

1. Browse to the Google Policy object (created in Before you begin - Setting up Google Chrome Policies in GPO section) in the Microsoft Management console.
2. Right-click > Click **Edit**.
3. In the group policy management editor, browse to **Policies > Administrative Templates > Google Chrome > Extensions > Extensions management settings**.



Configure extension management settings path

4. Enable the policy, then enter the permissions that you want allowed or blocked, compressing it to a single JSON string.

Format according to this example JSON data. (Here, you block any extension that needs the use of USB.)

```
{
  "*": {
    "blocked_permissions": ["usb"]
  }
}
```

Compact JSON data:

```
{"*":{"blocked_permissions":["usb"]}}
```

Note:

- If you can specify one extension ID, the policy will only apply to that extension. You can block more than one, but they need to be separated into their own entries.
- To block all extensions that use that permission, use an asterisk for the extension ID.

## Option 2: Manage extensions by policy

Windows offers multiple ways to manage extensions. A common way is to set multiple policies in one place with a JSON string or in the Windows Registry using the [extensions settings policy](#).

This policy can control settings such as Update URL, where the extension will be downloaded from for initial install, and Blocked permissions, which permissions are not allowed to run. Read the [Extension settings full description](#).

You decide if you want to set all extension management settings here or set these controls through other individual policies.

- The Runtime allowed/blocked hosts setting can only be set within the extension settings policy.
- The extension settings policy can overwrite other policies that you have elsewhere in group policy, including:
  - [ExtensionAllowedTypes](#)
  - [ExtensionInstallBlacklist](#)
  - [ExtensionInstallForcelist](#)
  - [ExtensionInstallSources](#)
  - [ExtensionInstallWhitelist](#)

The setting is set by one of these 2 methods:

- [Windows Registry](#)
- [JSON string in Windows Group Policy Editor](#)

Tip: Correctly formatting a JSON string can be tricky. Use a JSON checker before implementing the policy.

## Configure the extension policy using the Windows Registry

The ExtensionSettings policy should be written to the registry under:

HKLM\Software\Policies\Google\Chrome\ExtensionSettings\

- It's possible to use HKCU instead of HKLM. The equivalent path can be configured with GPO.
- The keys can be created with your chosen method on your user's machine.

For Chrome, all settings will start under this key:

HKEY\_LOCAL\_MACHINE\SOFTWARE\Policies\Google\Chrome\ExtensionSettings\



The next key that you will create is either the Extension ID for individual scope or an asterisk for the Default Scope. For example, use the following location for settings that apply to Google Hangouts:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome\ExtensionSettings\nckgahadag
oajjgafhacjanaoiihapd
```

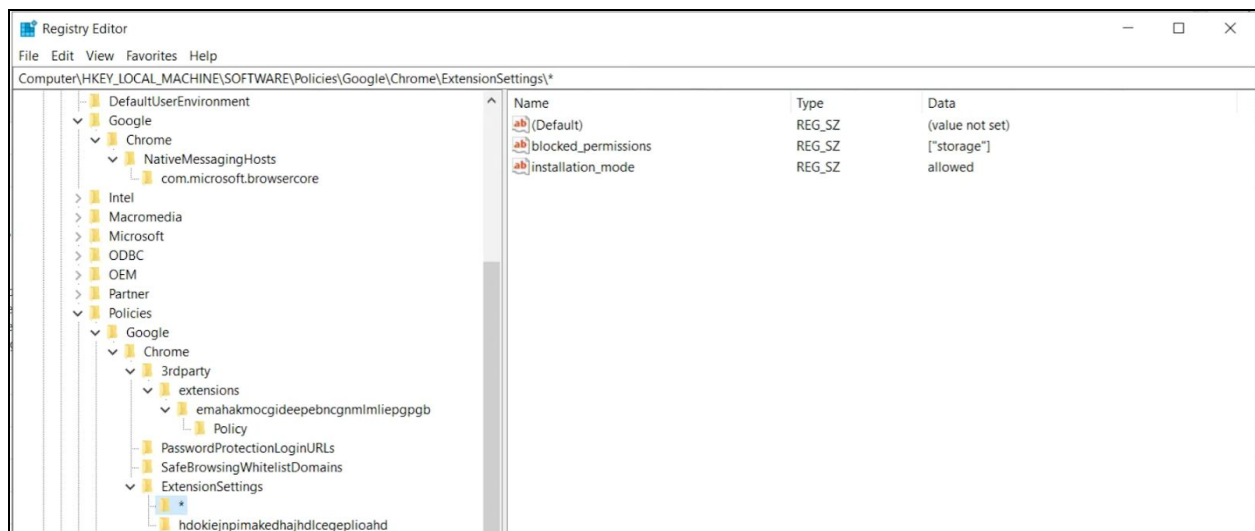
For settings that apply to the Default Scope, use this location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Google\Chrome\ExtensionSettings\*
```

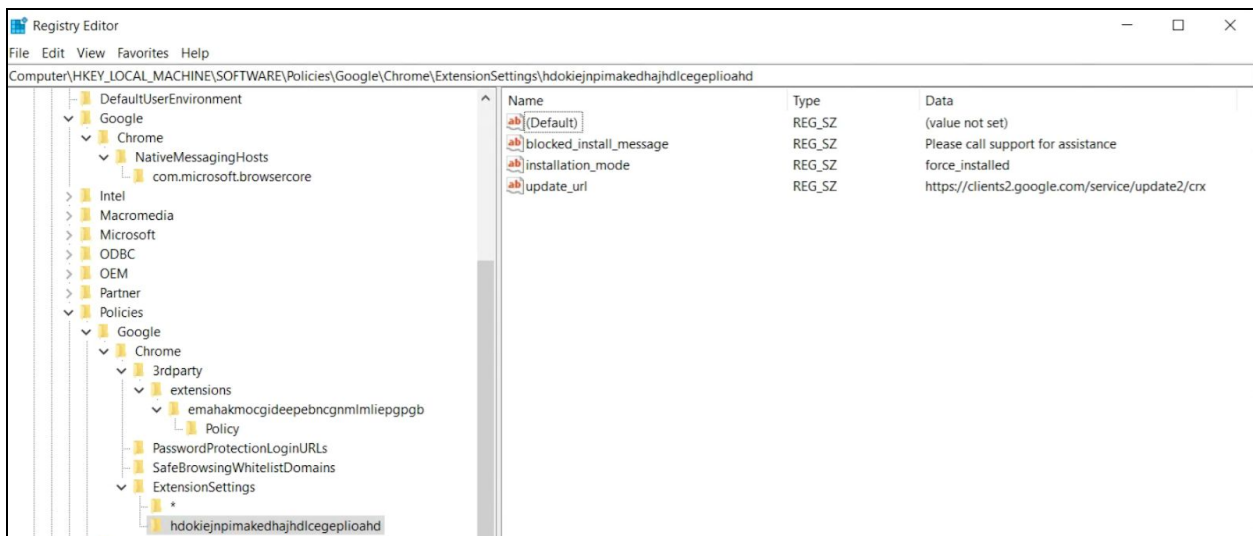
Different settings will require different formats, depending on whether they are a string or an array of strings. Array values require [ " value " ]. String values can be entered as is. The list of which settings are arrays or strings:

- Installation\_mode = String
- update\_url = String
- blocked\_permissions = Array of strings
- allowed\_permissions = Array of Strings
- minimum\_version\_required = String
- runtime\_blocked\_hosts = Array of strings
- runtime\_allowed\_hosts = Array of Strings
- blocked\_install\_message = String

Examples of what the keys look like within the registry:



The default (\*) scope key and its values



An individual scope and its values

Here, the keys set in the registry are converted to JSON with the policy:

Chrome policies

Applies to	Level	Source	Policy name
Machine	Mandatory	Platform	<a href="#">DefaultBrowserSettingEnabled</a>
Machine	Mandatory	Platform	<a href="#">ExtensionSettings</a>

```

{
  "": {
    "blocked_permissions": [ "storage" ],
    "installation_mode": "allowed"
  },
  "hdokiejnpimakedhajhdiceplioahd": {
    "blocked_install_message": "Please call support for assistance",
    "installation_mode": "force_installed",
    "update_url": "https://clients2.google.com/service/update2/crx"
  }
}

```

## Configure using a JSON string in Windows Group Policy Editor

The steps to use the extension settings policy using GPO assume you have already imported in the [ADM/ADMX for Chrome Policies](#).

For other OS platforms, check: [Mac](#) | [Linux](#)

1. Within GPO management editor, go to **Google Chrome > Extensions > Extensions management setting policy**.
2. Enable the policy and enter its compact JavaScript Object Notation (JSON) data in the text box as a single line with no line breaks.  
To validate policies and compact them into a single line (example JSON data below), use this [third-party JSON compression tool](#).

**Properly format JSON for the extension settings policy:** To use this method, you need to understand the 2 parts to this policy—the **default** and the **individual** scope. The default scope is a catch-all for extensions without their own scope. The individual scope is applied to that extension only.

The default scope is identified by the asterisk (\*). This example defines a default scope and a single individual extension scope:

```
{
  "*": {},
  "nckgahadagoaajjgafhacjanaoiihapd": {}
}
```

An extension will only get its settings from one scope. If there's an individual extension scope for that extension, those will be the settings that apply to that extension. If no individual extension scope exists, then it will use the default scope.

Here is an example JSON that blocks any extension from running on .example.com and blocks any extension that requires the permission "USB":

```
{
  "*": {
    "runtime_blocked_hosts": ["*://*.example.com"],
    "blocked_permissions": ["usb"]
  }
}
```

Compact JSON data:

```
{"*":{"runtime_blocked_hosts":["*://*.example.com"],"blocked_permissions":["usb"]}}
```

### Reference examples with example values:

- "allowed" (default)  
Your user can install the extension from the Chrome Web Store.  
Example JSON:  

```
{ "*": {"installation_mode": "allowed" } }
```
- "blocked"  
Your user can't install the extension from the Chrome Web Store.  
Example JSON:  

```
{ "*": {"installation_mode": "blocked" } }
```
- "Blocked\_install\_message"  
Here you can specify a custom message to display when installation is blocked.  
Example JSON - blocked\_install\_message:  

```
{ "*": {"blocked_install_message": ["Call IT(408 - 555 - 1234) for an exception"] } }
```
- "force\_installed"
  - The extension is automatically installed without your user's interaction.
  - Your user can't disable or remove the extension.

- "Normal\_installed"  
The extension is automatically installed without your user interaction, but they can disable the extension.

If an extension is "normal" or "force" installed, another field "update\_url" must also be defined, pointing to where the extension can be installed from.

- If the extension you're downloading is hosted on the Chrome Web Store, use ["https://clients2.google.com/service/update2/crx"](https://clients2.google.com/service/update2/crx).
- If you're hosting the extension on your own server, put the URL where Chrome can download the packed extension (.crx file).  
Example JSON - force\_installed extension with update\_url:  

```
{ "nckgahadagoaajjgafhacjanaoiihapd": { "installation_mode":  
  "force_installed", "update_url":  
  "https://clients2.google.com/service/update2/crx" } }
```

## Additional setting: Prevent extensions from altering webpages

This setting prevents extensions from changing and reading data from your most sensitive websites and domains.

Do this through blocking injecting scripts into your websites, reading the cookies, or making web-request modifications. This setting doesn't prevent your users from installing or removing extensions. It only prevents those extensions from altering the websites that you specify in the policy. These instructions are for managing this GPO on windows machines. For other platforms, check: [Admin console](#) | [Mac](#) | [Linux](#)

Within the Extension Settings policy, you can set the following settings to prevent (or allow) alterations of websites or domains:

- Runtime\_blocked\_hosts  
This setting blocks extensions from making changes or reading data from your chosen websites.
- Runtime\_allowed\_hosts  
This setting allows extensions to make changes or read data from your chosen websites. The format for specifying your site(s) in the JSON string in the policy:

```
[http|https|ftp|*]://[subdomain|*].[hostname|*].[eTLD|*]  
[http|https|ftp|*],
```

Note: [hostname|\*], and [eTLD|\*] sections are required, but [subdomain|\*] section is optional.

Examples of valid host patterns and matching patterns:

Valid host patterns	Matches	Doesn't match
*://*.example.*	http://example.com https://test.example.co.uk	https://example.google.com http://example.google.co.uk
http://example.*	http://example.com http://example.ly	https://example.com http://test.example.com
http://example.com	http://example.com	https://example.com http://test.example.co.uk
*://*	All Urls	

Here is a sample of a JSON string that blocks access for a single extension. This string prevents an extension from augmenting a specific site:

```
{
  "aapbdbdomjkkjkaonfhkkikfgjllcleb": {
    "runtime_blocked_hosts": ["*://*.importantwebsite"]
  }
}
```

Compact JSON data:

```
{"aapbdbdomjkkjkaonfhkkikfgjllcleb": {
  "runtime_blocked_hosts": ["*://*.importantwebsite"]}}
```

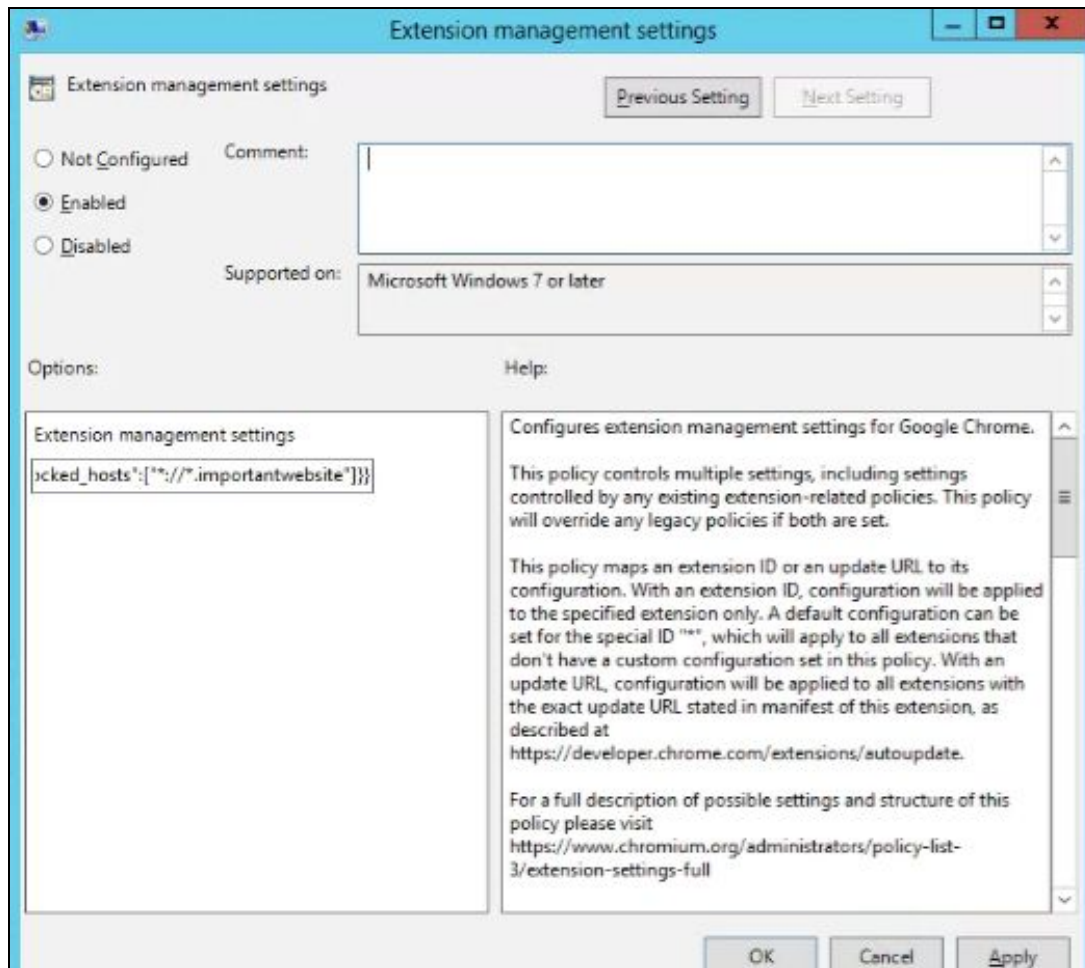
Separate multiple entries into an entry for each app ID that you want to block. Here's an example of how to block 2 extensions from running on the same domain:

```
{
  "aapbdbdomjkkjkaonfhkkikfgjllcleb": {
    "runtime_blocked_hosts": ["*://*.importantwebsite"]
  },
  "bfbmjmiodbnnpllbbbfblcplfjjepjdn": {
    "runtime_blocked_hosts": ["*://*.importantwebsite"]
  }
}
```

Compact JSON data:

```
{"aapbdbdomjkkjkaonfhkkikfgjllcleb": {"runtime_blocked_hosts":
["*://*.importantwebsite"]}, "bfbmjmiodbnnpllbbbfblcplfjjepjdn":
{"runtime_blocked_hosts": ["*://*.importantwebsite"]}}
```

And here is the JSON string entered in the **Google Chrome > Extensions > Extensions management setting** policy:



## Option 3A: Allow or block extensions in the Google Admin console

In addition to the methods described above, you can control which extensions your users can install on their devices by creating whitelists and blacklists. You can allow users to install any app or extension. Or, you can set policies which block or allow apps for your whole organization or certain groups of employees.

**Before you begin:** To manage extensions for users, you need to turn on their Chrome Web Store service in your Admin console. You can find Additional Google Services in your Admin console under Apps. For steps, see [Turn Additional Google Services on or off](#).

The following steps assume you're familiar with changing settings in your Admin console.

### Allow all extensions except those you want to block

1. In your Admin console, go to **Devices > Chrome management > User settings**.
2. Select the organization containing the users you want to block extensions for.

3. Next to Allow or Block All Apps and Extensions, select the option to allow all applications and extensions except ones you block.
4. Next to Allowed Apps and Extensions, click **Manage**.
5. Select each extension you want to block.
6. Click **Save**.

### Block all extensions except those you want to allow

1. In your Admin console, go to **Devices > Chrome management > User settings**.
2. Select the organization containing the users you want to allow extensions for.  
For complete details, see [Set a Chrome policy for multiple apps](#).
3. Next to Allow or Block All Apps and Extensions, select the option to block all applications and extensions except ones you allow.
4. Next to Allowed Apps and Extensions, click **Manage**.
5. Select each extension you want to allow.
6. Click **Save**.

### Block or allow one extension

1. In your Admin console, go to **Devices > Chrome management > App management**.
2. Select the extension you want to block or allow.
3. Select a type of setting, such as User settings or Public session settings.
4. Select the organization with the users you want to allow or block the extension for.  
For complete details, see [Set Chrome policies for one app](#).
5. Under Allow installation, click to either block or allow the extension.  
Initially, an organization inherits the settings of its parent.
6. If you're changing a setting for a child organization:
  - To override an inherited value, click **Override** and then change the setting.
  - To return an overridden setting to the value of its parent, click **Inherit**.
7. Click **Save**.

### Force-install an extension

If you know that a user requires an extension for them to do their job, you can automatically install it for them. Keep in mind that if you force install an extension, it will grant all of the permissions it needs to run automatically.

### Force-install several extensions

1. In your Admin console, go to **Devices > Chrome management > User settings**.
2. Select the organization containing the users you want to force install extensions for.  
For complete details, see [Set a Chrome policy for multiple apps](#).
3. In the Force-installed Apps and Extensions section, click **Manage force-installed apps**.  
Tip: Use the search bar to quickly find the section.
4. Select the extension to force-install and click **Save**.

## Force-install one extension

1. In your Admin console, go to **Devices > Chrome management > App management**.
2. Select the extension you want to block or allow.
3. Select a type of setting, such as User settings or Public session settings.
4. Select the organization containing the users you want to allow or block the extension for.  
For complete details, see [Set Chrome policies for one app](#).
5. Under Force Installation, turn the setting on.  
Initially, an organization inherits the settings of its parent.
6. If you're changing a setting for a child organization:
  - To override an inherited value, click **Override** and then change the setting.
  - To return an overridden setting to the value of its parent, click **Inherit**.
7. Click **Save**.

## Option 3B: Allow or block extensions in Group Policy

**Before you begin:** The following steps assume that you already have Chrome managed for your users. For more on how to deploy Chrome on Windows, refer to [Chrome Browser Deployment Guide \(Windows\)](#). For Mac® deployment and policy management, go to [Set up Chrome Browser on Mac](#).

For Windows, there are 2 types of policy templates: an ADM and ADMX template. Ensure that you verify which type you can use on your network. The templates show which registry keys you can set to configure Chrome and what the acceptable values are. Chrome looks at the values set in these registry keys to determine how to act.

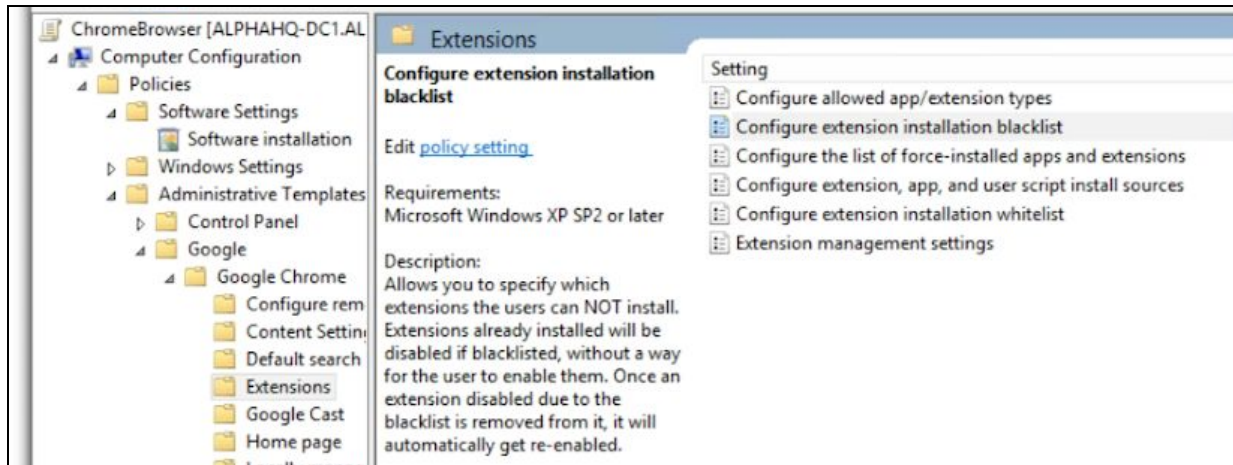
1. Download Chrome policy templates.  
Windows templates, as well as common policy documentation for all operating systems, can be found in this [zip file of Google Chrome templates and documentation](#).
2. Open the ADM or ADMX template you downloaded:
  - a. Go to **Start > Run: gpedit.msc**. (Or, run gpedit.msc in your terminal.)
  - b. Go to **Local Computer Policy > Computer Configuration > Administrative Templates**.
  - c. Right-click **Administrative Templates** and select **Add/Remove Templates**.
  - d. Add the chrome.adm template through the dialog.

Afterward, if it's not already there, a Google or Google Chrome folder will appear under Administrative Templates. If you added the ADM template on Windows 7 or 10, it will appear under Classic Administrative Templates / Google / Google Chrome.

## Allow all extensions except those you want to block

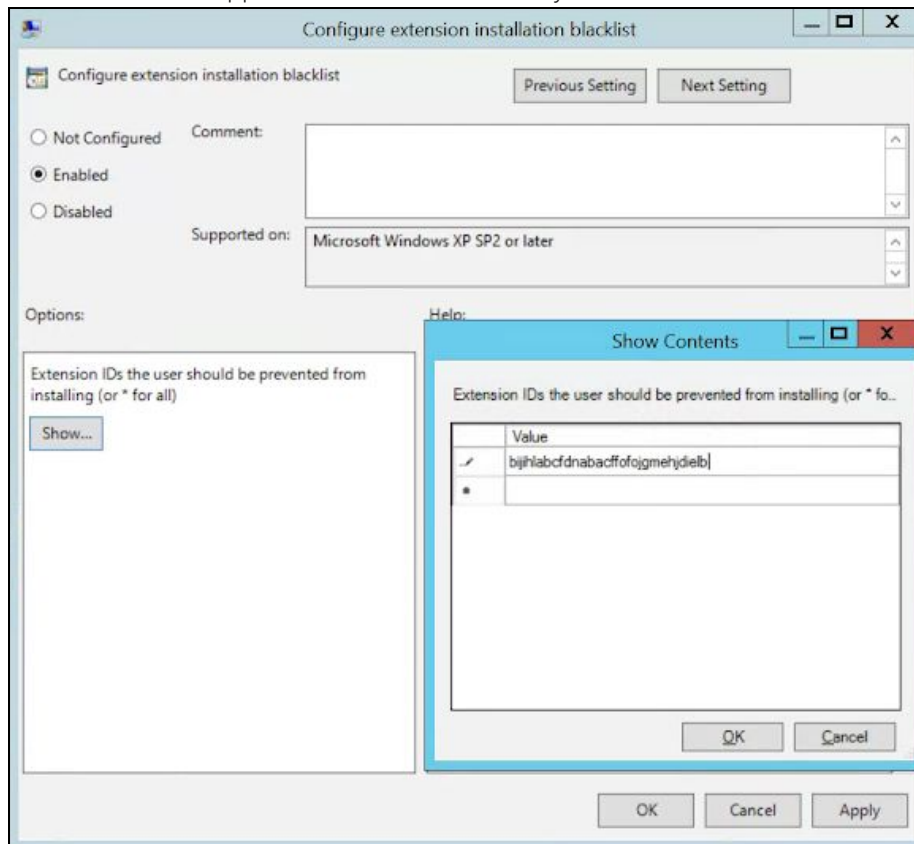
1. In the Group Policy Editor, open the template you just added.
2. Browse to **Google > Google Chrome > Extensions > Configure Extension installation blacklist**.





Path to Extension management policies


2. In the setting, select **Enabled**.
3. Click **Show**.
4. Enter the app ID of the extensions that you want to block.



Configure extension installation blacklist

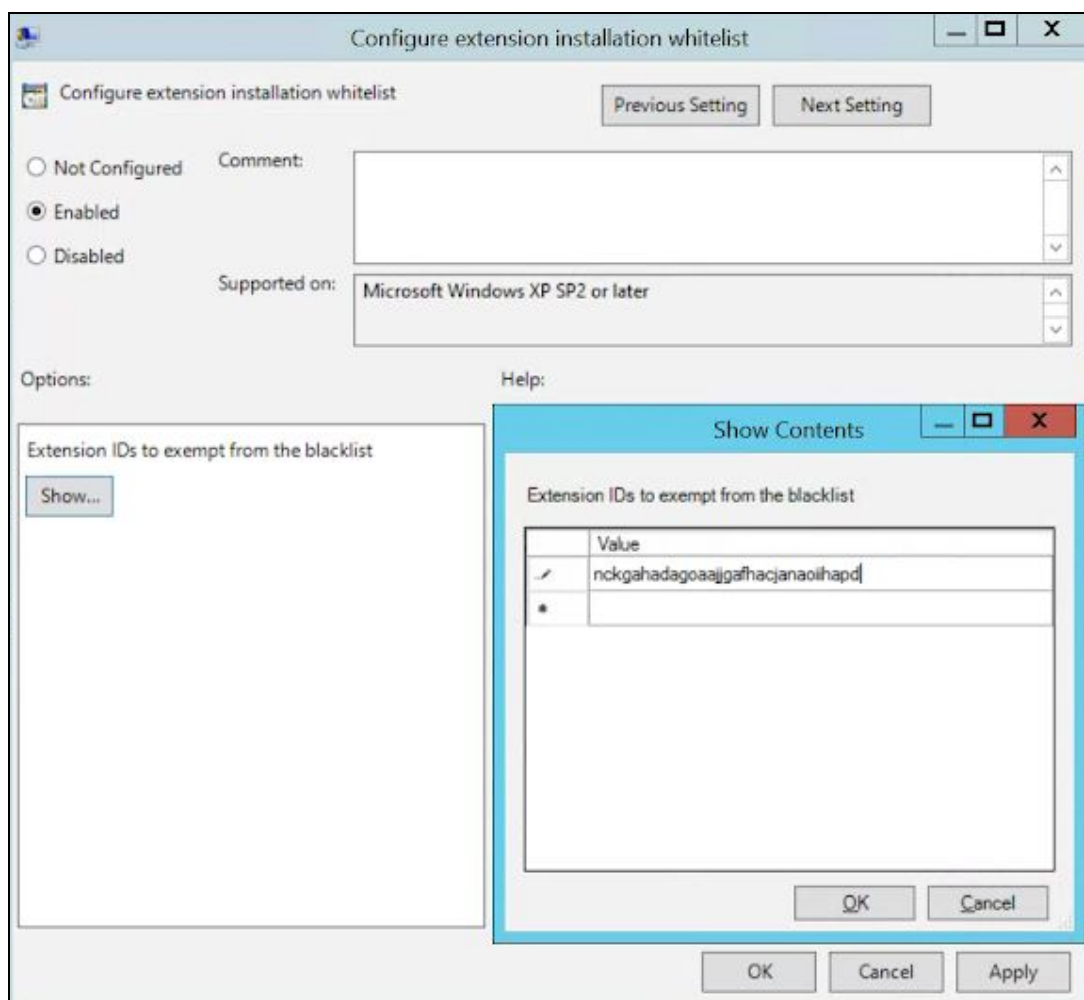
Notes:

- If you can't find the app ID of an extension, view it in the Chrome Web Store. There, find the specific extension and you will see the app ID at the end of the URL in the Chrome omnibox:



App ID example located after google-hangouts/

- Enter \* into the policy to prevent any extensions from being installed. You can use this with the Configure extension white list policy. This way you only allow certain extensions to be installed by your users.
- You can add an extension to the blacklist that is already installed on a user's machine. It will disable the extension and prevent the user from re-enabling it. It will not be uninstalled, just disabled.



Configure extension installation whitelist

## Block or allow one extension

To block a single extension, add the app ID of the extension you want blocked to the configure extension installation blacklist policy. All of your other extensions will be allowed to be installed.

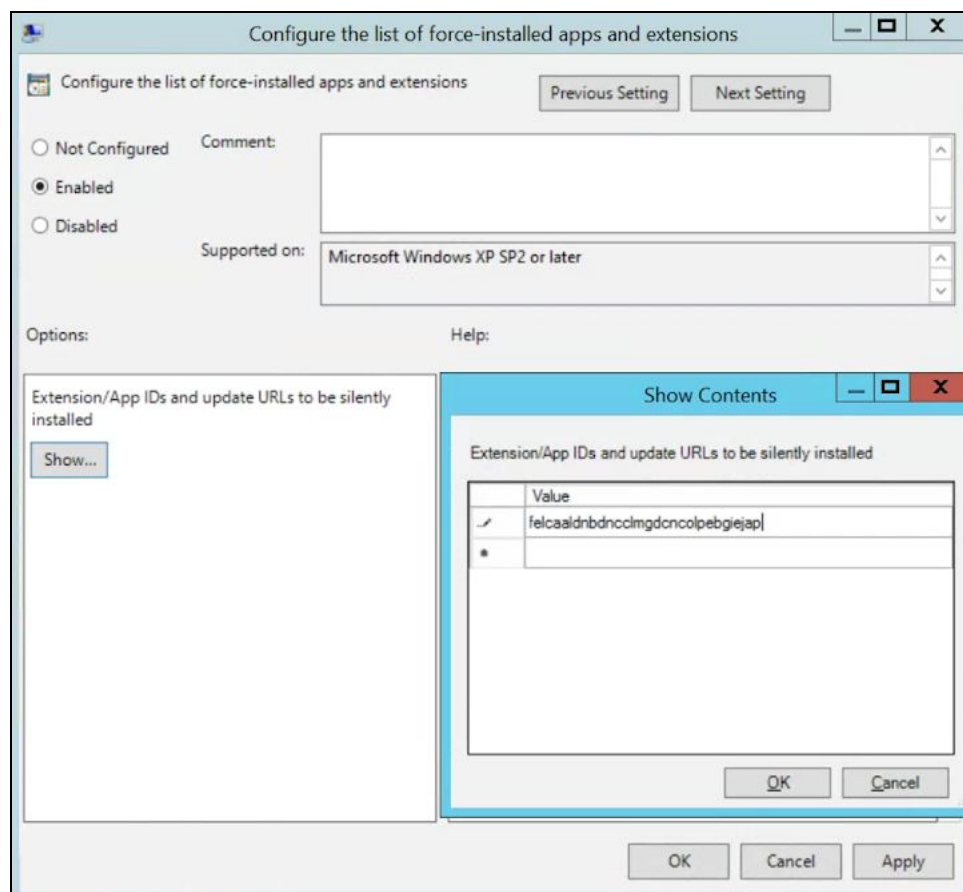
To allow only one extension:

1. In the content section in the Configure extension installation blacklist policy, enter \*. This will blacklist all extensions from being installed.
2. Add the app ID of the allowed extension to the Configure extension installation whitelist policy.

## Force-install an extension

1. In the Group Policy Editor, browse to **Google > Google Chrome > Extensions > Configure the list of force-installed apps and extensions**.
2. Select **Enabled**.
3. Click **Show**.
4. Enter the app ID or IDs of the extension or extensions you want to force-install.

The extension will be installed silently with no need for a user to interact. The user also won't be able to uninstall or disable the extension. This setting will overwrite over any blacklist policy that you might have enabled.



Configure the list of force-installed apps and extensions

## Create your own on-premises web store

The [Chrome Web Store](#) host extensions and provides a number of security features, such as automated and manual code scans to prevent malicious code from being sent to your users. There's an option to host your extensions in your own web store, but it's not recommended. The self-hosting method requires significant amount of work, in terms of validating the security of your extensions and keeping them updated.

If you choose to host your own store, this section tells you how. It covers how to package an extension and host it without using the Chrome Web Store. It also includes instructions on how to deploy these extensions to your devices and users.

As an alternative to creating your own web store, consider marking internal extensions on the Chrome Web Store as private. Here are the different options to [Publish in the Chrome Web Store](#).

### Requirements

To host your own extension, you will need to provide your own web hosting services for the extension and its manifest file. This hosting location shouldn't require authentication. It needs to be accessible by devices wherever they might be used. Keep this in mind if you want to host the file on your internal repository.

The steps assume that you've already created your extension, have some experience with XML files, and have some knowledge about group policy and using the windows registry.

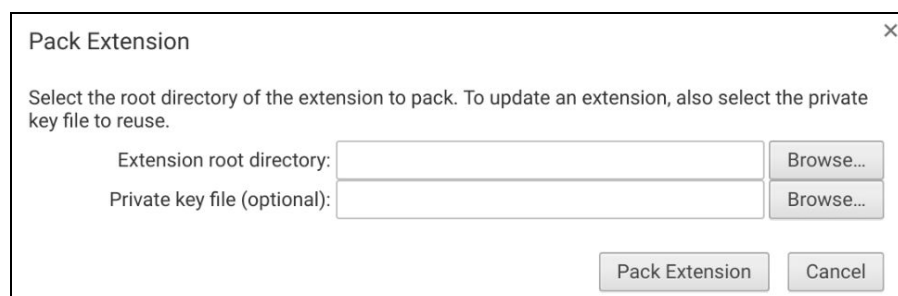
### Publishing your extension

Extensions first need to be packed into a CRX file. If the extension isn't packed as a CRX file, here are the steps:

1. Go to **chrome://extensions** in the Chrome address bar and check the box for **Developer mode**.



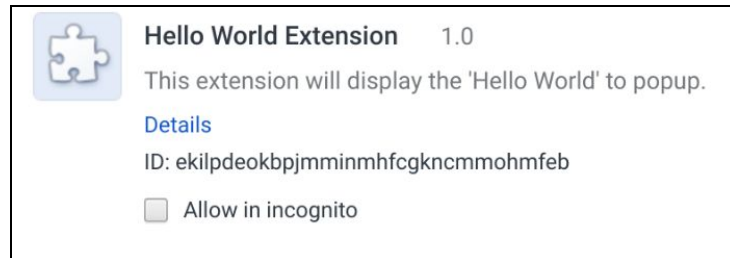
2. After you're in developer mode, create the CRX file by clicking **Pack Extension**.
3. Select the directory where your source is at.  
This will create your CRX file, along with a PEM file.



## Pack Extension root directory selector

Tip: Keep the PEM file securely stored, because this is the key to your extension. You'll need it for future updates.

4. Drag the CRX in to your extensions window and make sure that it loads.
5. Test the extension and take note of the ID field and version number.  
These will be important later on.



Extension details

5. Place the CRX file in the host location where your users or devices will download it from.
6. Note the URL of where the file is uploaded.  
This will be important for the manifest XML file.
7. To create a manifest XML file with the app/extension ID, download URL, and version, define these 3 fields:
  - **appid** (the extension ID from step 3)
  - **codebase** (the download location for the CRX file from step 4)
  - **version** (the version of the app/extension, which should match step 3)

Example XML manifest file:

```
<?xml version='1.0' encoding='UTF-8'?>
<gupdate xmlns='http://www.google.com/update2/response' protocol='2.0'>
  <app appid='ekilpdeokbpjmmminmhfcgkncmmohmfef'>
    <updatecheck
      codebase='https://app.somecompany.com/chrome/helloworld.crx'
      version='1.0' />
  </app>
</gupdate>
```

8. Upload the completed XML file to a location from where your users or devices can download it, while noting the URL.

## Publishing updates to your extension

Make sure that you've made the required changes to your extension and tested it. To publish updates:

1. Change the version number in your extension's manifest JSON file to a higher number.  
Example:  
`"version": "versionString"`  
 If the `"version": "1.0"`, then you can update to `"version": "1.1"` or any number higher than `"1.0"`.
2. Update the `"version"` of `<updatecheck>` in the XML file to match the number that you put in the manifest file in the last step.

Another example:

```
<updatecheck codebase='https://app.somecompany.com/chrome/helloworld.crx'  
version='1.1' />
```

3. Recreate a CRX file which includes the new changes:
  - a. Go to **chrome://extensions** in the Chrome address bar.
  - b. Check the box for **Developer mode**.
4. Create the CRX file by clicking **Pack Extension** and selecting the directory where your source is at. Note: For the PEM file, use the same file which was generated and saved during the first time the CRX file was created.
5. Drag the CRX in to your extensions window and make sure that it loads.
6. Test the extension.
7. Replace the old CRX file and XML file with the new file.  
This needs to be at the same host location from where the users or devices downloaded the files before.

The changes will be picked up during the next policy sync cycle.

Reference URLs:

- [Autoupdating](#)
- [Update URL](#)
- [Update manifest](#)



## Distributing privately hosted extensions

**In the Google Admin console:** These are the steps you can take to have Chrome to download your hosted extension.

1. Launch your Admin console from [admin.google.com](https://admin.google.com).
2. Go to the Device management section, then to Chrome management.
3. Depending on the use case for your extension, select one of these Settings groups:
  - **User Settings - for Extensions to be used by Logged in Users managed in that domain**
  - **Public Session Settings - for Extensions to be used by Public Session kiosks**
4. Make sure that you select the relevant organizational unit to limit the scope of the installation.
5. Select **Specify a Custom App**.
6. Fill out your extensions ID and the XML manifest URL and click **Add**.

## Force-installed Apps and Extensions

The selected apps and extensions will be automatically installed.



Specify a Custom App

Total to force install: 0

You must supply both the extension id and the url where the extension is hosted.

ID

URL

ADD

Force install Apps and Extensions

- Make sure that you click **Save**.  
During your user's next policy refresh, the extension will be installed on the targeted devices.

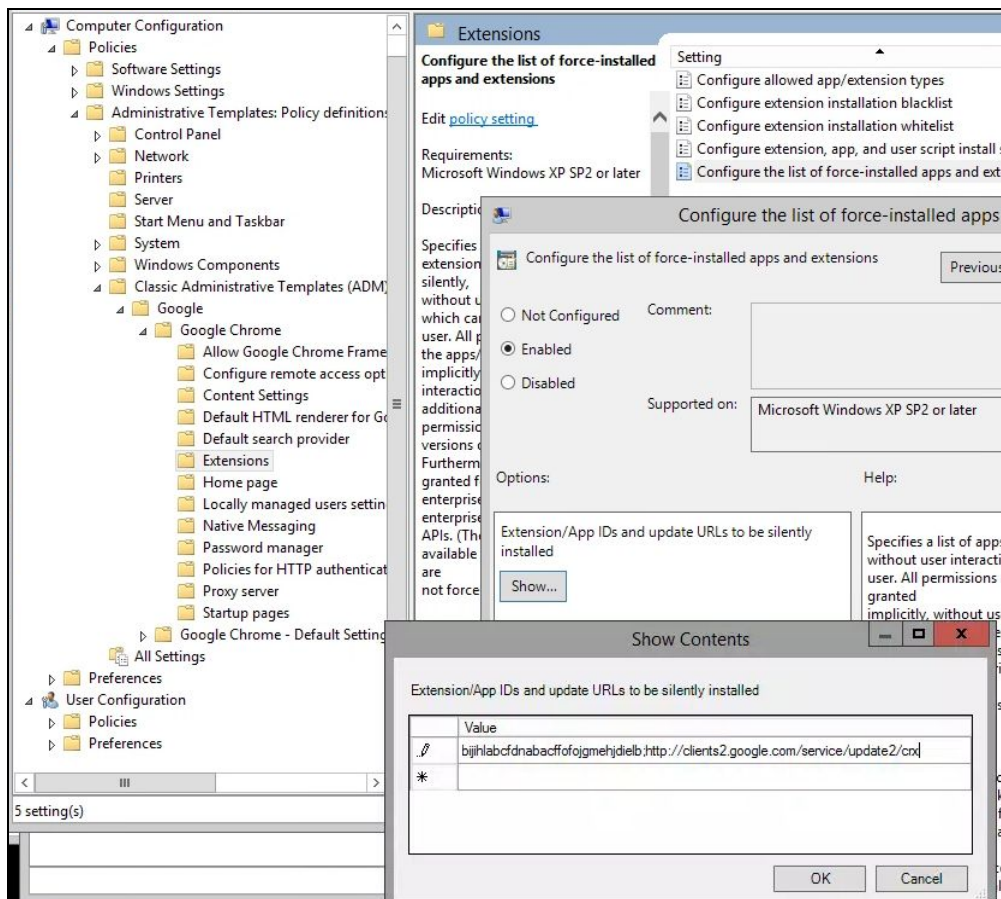
**In Group Policy:** If you aren't using the Admin console, you can use the policy called "Configure the list of force installed apps and extensions" to force-install an extension on your user's device.

For privately hosted apps (not in the Chrome Web Store), use a string such as:

```
pckdojakecnhhplcgfflhndiffaohfah;https://sites.google.com/site/pushcrx/privatewebstore/extension_info.xml
```

The URL is specified to the **internal app's update.xml**, rather than the public-facing `clients2.google.com` URL.





GPO Policy "Configure the list of force-installed apps/extensions" (Show Contents)

The policies can then be applied to your chosen users, machines, or both. It can take some time for the policy to take effect. Speed things up by running "gpupdate" on your user's machine.

## Manage extensions using Chrome Browser Cloud Management


Manage Chrome Browser for your Windows, Mac, and Linux machines all in one place, and get an in-depth view of the state of Chrome Browser in your environment. Chrome Browser Cloud Management is a new console for managing Chrome Browser settings. With the console, you can quickly get insights on the:

- Current Chrome Browser versions deployed across your fleet
- Extensions installed on each browser
- Policies being applied to each browser

You can also take quick actions in the console, such as blocking a suspicious extension on all of your machines. Manage extensions from any of the 3 subpages in the Admin console. To access them:

1. In the Admin console, go to **Devices > Chrome management**.
2. Click **Managed Browsers**.
3. Click any of the following subpages to manage extensions:



- **Installed apps & extensions:** On this page, you can view the installed extensions, their stats, how it was installed, the version and release channel, and what user profile it's installed on. This console gives you more control over managing extensions and seeing what apps are installed. By clicking More  on each extension, you will see 2 actions:
  - **Block:** Restrict the extension from being run.
  - **Force install:** Require and autoinstall the selected extension.
- **Device details:** On this page, you can view a managed machine's name, OS version, user details, architecture (32 or 64 bit), the enrollment date, and how many policies are applied.
- **Browser & Profiles:** Here you can view the browser version and release channel (Stable, Dev, Beta or Canary), as well as which profiles the Chrome Browser is linked to.

## Additional resources

Here are more resources to help you with managing the Chrome Browser in your organization:

- [Chrome Browser Deployment Guide \(Windows\)](#)
- [Chrome Policy list](#)
- [Chrome Enterprise release notes](#)
- [Chrome Enterprise Help Center](#)
- [Make Chrome default browser \(Windows 10\)](#)