/\* chrome version: 103.0.5058.0 \*/

Google\_Chrome.pfm\_title = "Google Chrome";

Google\_Chrome.pfm\_description = "Google Chrome preferences";

AbusiveExperienceInterventionEnforce.pfm\_title = "Abusive Experience Intervention Enforce";

AbusiveExperienceInterventionEnforce.pfm\_description = "If SafeBrowsingEnabled is not Disabled, then setting AbusiveExperienceInterventionEnforce to Enabled or leaving it unset prevents sites with abusive experiences from opening new windows or tabs.\n\nSetting SafeBrowsingEnabled to Disabled or AbusiveExperienceInterventionEnforce to Disabled lets sites with abusive experiences open new windows or tabs.";

AccessCodeCastDeviceDuration.pfm\_title = "Specifies how long (in seconds) a cast device selected with an access code or QR code stays in the Google Cast menu's list of cast devices.";

AccessCodeCastDeviceDuration.pfm\_description = "This policy specifies how long (in seconds) a cast device that was previously selected via an access code or QR code can be seen within the Google Cast menu of cast devices.\nThe lifetime of an entry starts at the time the access code was first entered or the QR code was first scanned.\nDuring this period the cast device will appear in the Google Cast menu's list of cast devices.\nAfter this period, in order to use the cast device again the access code must be reentered or the QR code must be rescanned.\nBy default, the period is zero seconds, so cast devices will not stay in the Google Cast menu, and so the access code must be reentered, or the QR code rescanned, in order to initiate a new casting session.\nNote that this policy only affects how long a cast devices appears in the Google Cast menu, and has no effect on any ongoing cast session which will continue even if the period expires.\nThis policy has no effect unless the AccessCodeCastEnabled policy is Enabled.";

AccessCodeCastEnabled.pfm\_title = "Allow users to select cast devices with an access code or QR code from within the Google Cast menu.";

AccessCodeCastEnabled.pfm\_description = "This policy controls whether a user will be presented with an option, within the Google Cast menu which allows them to cast to cast devices that do not appear in the Google Cast menu, using either the access code or QR code displayed on the cast devices's screen.\nBy default, a user must reenter the access code or rescan the QR code in order to initiate a subsequent casting session, but if the AccessCodeCastDeviceDuration policy has been set to a non-zero value (the default is zero), then the cast device will remain in the list of available cast devices until the specified period of time has expired.\nWhen this policy is set to Enabled, users will be presented with the option to select cast devices by using an access code or by scanning a QR code.\nWhen this policy is set to Disabled or not set, users will not be given the option to select cast devices by using an access code or by scanning a QR code.";

AccessibilityImageLabelsEnabled.pfm\_title = "Enable Get Image Descriptions from Google.";

AccessibilityImageLabelsEnabled.pfm\_description = "The Get Image Descriptions from Google\naccessibility feature enables visually-impaired screen reader users to\nget descriptions of unlabeled images on the web. Users who choose to enable it\nwill have the option of using an anonymous Google service to provide\nautomatic descriptions for unlabeled images they encounter on the web.\n\nIf this feature is enabled, the content of images will be sent to Google\nservers in order to generate a description. No cookies or other user\ndata is sent, and Google does not save or log any image content.\n\nIf this policy is set to Enabled, the\nGet Image Descriptions from Google\nfeature will be enabled, though it will only affect users who are using a\nscreen reader or other similar assistive technology.\n\nIf this policy is set to Disabled, users will not have the option of enabling\nthe feature.\n\nIf this policy is not set, user can choose to use this feature or not.\n";

AdditionalDnsQueryTypesEnabled.pfm\_title = "Allow DNS queries for additional DNS record types";

AdditionalDnsQueryTypesEnabled.pfm\_description = "This policy controls whether Google Chrome may query additional DNS record types when making insecure DNS requests. This policy has no effect on DNS queries made via Secure DNS, which may always query additional DNS types.\n\nIf this policy is unset or set to Enabled, additional types such as HTTPS (DNS type 65) may be queried in addition to A (DNS type 1) and AAAA (DNS type 28).\n\nIf this policy is set to Disabled, DNS will only be queried for A (DNS type 1) and/or AAAA (DNS type 28).\n\nThis policy is a temporary measure and will be removed in future versions of Google Chrome. After removal of the policy, Google Chrome will always be able to query additional DNS types.";

AdsSettingForIntrusiveAdsSites.pfm\_title = "Ads setting for sites with intrusive ads";

AdsSettingForIntrusiveAdsSites.pfm\_description = "1 - Allow ads on all sites\n2 - Do not allow ads on sites with intrusive ads\nUnless SafeBrowsingEnabled is set to False, then setting AdsSettingForIntrusiveAdsSites to 1 or leaving it unset allows ads on all sites.\n\nSetting the policy to 2 blocks ads on sites with intrusive ads.";

AdvancedProtectionAllowed.pfm\_title = "Enable additional protections for users enrolled in the Advanced Protection program";

AdvancedProtectionAllowed.pfm\_description = "This policy controls whether users enrolled in the Advanced Protection program receive extra protections. Some of these features may involve the sharing of data with Google (for example, Advanced Protection users will be able to send their downloads to Google for malware scanning). If set to True or not set, enrolled users will receive extra protections. If set to False, Advanced Protection users will receive only the standard consumer features.";

AllHttpAuthSchemesAllowedForOrigins.pfm\_title = "List of origins allowing all HTTP authentication";

AllHttpAuthSchemesAllowedForOrigins.pfm\_description = "Setting the policy specifies for which origins to allow all the HTTP authentication schemes Google Chrome supports regardless of the AuthSchemes policy.\n\nFormat the origin pattern according to this format (https://www.chromium.org/administrators/url-blocklist-filter-format). Up to 1,000 exceptions can be defined in AllHttpAuthSchemesAllowedForOrigins.\nWildcards are allowed for the whole origin or parts of the origin, either the scheme, host, port.";

AllowCrossOriginAuthPrompt.pfm\_title = "Cross-origin HTTP Authentication prompts";

AllowCrossOriginAuthPrompt.pfm\_description = "Setting the policy to Enabled allows third-party images on a page to show an authentication prompt.\n\n Setting the policy to Disabled or leaving it unset renders third-party images unable to show an authentication prompt.\n\nTypically, this policy is Disabled as a phishing defense.";

AllowDeletingBrowserHistory.pfm\_title = "Enable deleting browser and download history";

AllowDeletingBrowserHistory.pfm\_description = "Setting the policy to Enabled or leaving it unset means browser history and download history can be deleted in Chrome, and users can't change this setting.\n\nSetting the policy to Disabled means browser history and download history can't be deleted. Even with this policy off, the browsing and download history are not guaranteed to be retained. Users may be able to edit or delete the history database files directly, and the browser itself may expire or archive any or all history items at any time.";

AllowDinosaurEasterEgg.pfm\_title = "Allow Dinosaur Easter Egg Game";

AllowDinosaurEasterEgg.pfm\_description = "Setting the policy to True allows users to play the dinosaur game. Setting the policy to False means users can't play the dinosaur easter egg game when device is offline.\n\nLeaving the policy unset means users can't play the game on enrolled Google Chrome OS, but can under other circumstances.";

AllowFileSelectionDialogs.pfm\_title = "Allow invocation of file selection dialogs";

AllowFileSelectionDialogs.pfm\_description = "Setting the policy to Enabled or leaving it unset means Chrome can display, and users can open, file selection dialogs.\n\nSetting the policy to Disabled means that whenever users perform actions provoking a file selection dialog, such as importing bookmarks, uploading files, and saving links, a message appears instead. The user is assumed to have clicked Cancel on the file selection dialog.";

AllowedDomainsForApps.pfm\_title = "Define domains allowed to access Google Workspace";

AllowedDomainsForApps.pfm\_description = "Setting the policy turns on Chrome's restricted sign-in feature in Google Workspace and prevents users from changing this setting. Users can only access Google tools using accounts from the specified domains (to allow gmail or googlemail accounts, add consumer\_accounts to the list of domains). This setting prevents users from signing in and adding a Secondary Account on a managed device that requires Google authentication, if that account doesn't belong to one of the explicitly allowed domains.\n\nLeaving this setting empty or unset means users can access Google Workspace with any account.\n\nUsers cannot change or override this setting.\n\nNote: This policy causes the X-GoogApps-Allowed-Domains header to be appended to all HTTP and HTTPS requests to all google.com domains, as described in https://support.google.com/a/answer/1668854.";

AlternateErrorPagesEnabled.pfm\_title = "Enable alternate error pages";

AlternateErrorPagesEnabled.pfm\_description = "Setting the policy to True means Google Chrome uses alternate error pages built into (such as \"page not found\"). Setting the policy to False means Google Chrome never uses alternate error pages.\n\nIf you set the policy, users can't change it. If not set, the policy is on, but users can change this setting.";

AlternativeBrowserParameters.pfm\_title = "Command-line parameters for the alternative browser.";

AlternativeBrowserParameters.pfm\_description = "Setting the policy to a list of strings means each string is passed to the alternative browser as separate command-line parameters. On Microsoft® Windows®, the parameters are joined with spaces. On macOS and Linux®, a parameter can have spaces and still be treated as a single parameter.\n\nIf an parameter contains ${url}, ${url} is replaced with the URL of the page to open. If no parameter contains ${url}, the URL is appended at the end of the command line.\n\nEnvironment variables are expanded. On Microsoft® Windows®, %ABC% is replaced with the value of the ABC environment variable. On macOS and Linux®, ${ABC} is replaced with the value of the ABC environment variable.\n\nLeaving the policy unset means only the URL is passed as a command-line parameter.";

AlternativeBrowserPath.pfm\_title = "Alternative browser to launch for configured websites.";

AlternativeBrowserPath.pfm\_description = "Setting the policy controls which command to use to open URLs in an alternative browser. The policy can be set to one of ${ie}, ${firefox}, ${safari}, ${opera}, ${edge} or a file path. When this policy is set to a file path, that file is used as an executable file. ${ie} is only available on Microsoft® Windows®. ${safari} and ${edge} are only available on Microsoft® Windows® and macOS.\n\nLeaving the policy unset puts a platform-specific default in use: Internet Explorer® for Microsoft® Windows®, or Safari® for macOS. On Linux®, launching an alternative browser will fail.";

AlwaysOpenPdfExternally.pfm\_title = "Always Open PDF files externally";

AlwaysOpenPdfExternally.pfm\_description = "Setting the policy to Enabled turns the internal PDF viewer off in Google Chrome, treats PDF files as a download, and lets users open PDFs with the default application.\n\nSetting the policy to Disabled means that unless users turns off the PDF plugin, it will open PDF files.\n\nIf you set the policy, users can't change it in Google Chrome. If not set, users can choose whether to open PDF externally or not.";

AmbientAuthenticationInPrivateModesEnabled.pfm\_title = "Enable Ambient Authentication for profile types.";

AmbientAuthenticationInPrivateModesEnabled.pfm\_description = "0 - Enable ambient authentication in regular sessions only.\n1 - Enable ambient authentication in incognito and regular sessions.\n2 - Enable ambient authentication in guest and regular sessions.\n3 - Enable ambient authentication in regular, incognito and guest sessions.\nConfiguring this policy will allow/disallow ambient authentication for Incognito and Guest profiles in Google Chrome.\n\nAmbient Authentication is http authentication with default credentials if explicit credentials are not provided via NTLM/Kerberos/Negotiate challenge/response schemes.\n\nSetting the RegularOnly (value 0), allows ambient authentication for Regular sessions only. Incognito and Guest sessions wouldn't be allowed to ambiently authenticate.\n\nSetting the IncognitoAndRegular (value 1), allows ambient authentication for Incognito and Regular sessions. Guest sessions wouldn't be allowed to ambiently authenticate.\n\nSetting the GuestAndRegular (value 2), allows ambient authentication for Guest and Regular sessions. Incognito sessions wouldn't be allowed to ambiently authenticate.\n\nSetting the All (value 3), allows ambient authentication for all sessions.\n\nNote that, ambient authentication is always allowed on regular profiles.\n\nIn Google Chrome version 81 and later, if the policy is left not set, ambient authentication will be enabled in regular sessions only.";

AudioCaptureAllowed.pfm\_title = "Allow or deny audio capture";

AudioCaptureAllowed.pfm\_description = "Setting the policy to Enabled or leaving it unset means that, with the exception of URLs set in the AudioCaptureAllowedUrls list, users get prompted for audio capture access.\n\nSetting the policy to Disabled turns off prompts, and audio capture is only available to URLs set in the AudioCaptureAllowedUrls list.\n\nNote: The policy affects all audio input (not just the built-in microphone).";

AudioCaptureAllowedUrls.pfm\_title = "URLs that will be granted access to audio capture devices without prompt";

AudioCaptureAllowedUrls.pfm\_description = "Setting the policy means you specify the URL list whose patterns get matched to the security origin of the requesting URL. A match grants access to audio capture devices without prompt\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns.";

AudioSandboxEnabled.pfm\_title = "Allow the audio sandbox to run";

AudioSandboxEnabled.pfm\_description = "This policy controls the audio process sandbox.\nIf this policy is enabled, the audio process will run sandboxed.\nIf this policy is disabled, the audio process will run unsandboxed and the WebRTC audio-processing module will run in the renderer process.\nThis leaves users open to security risks related to running the audio subsystem unsandboxed.\nIf this policy is not set, the default configuration for the audio sandbox will be used, which may differ per platform.\nThis policy is intended to give enterprises flexibility to disable the audio sandbox if they use security software setups that interfere with the sandbox.";

AuthNegotiateDelegateAllowlist.pfm\_title = "Kerberos delegation server allowlist";

AuthNegotiateDelegateAllowlist.pfm\_description = "Setting the policy assigns servers that Google Chrome may delegate to. Separate multiple server names with commas. Wildcards, \*, are allowed.\n\nLeaving the policy unset means Google Chrome won't delegate user credentials, even if a server is detected as intranet.";

AuthNegotiateDelegateByKdcPolicy.pfm\_title = "Use KDC policy to delegate credentials.";

AuthNegotiateDelegateByKdcPolicy.pfm\_description = "Setting the policy to Enabled means HTTP authentication respects approval by KDC policy. In other words, Google Chrome delegates user credentials to the service being accessed if the KDC sets OK-AS-DELEGATE on the service ticket. See RFC 5896 ( https://tools.ietf.org/html/rfc5896.html ). The service should also be allowed by AuthNegotiateDelegateAllowlist.\n\nSetting the policy to Disabled or leaving it unset means KDC policy is ignored on supported platforms and only AuthNegotiateDelegateAllowlist is respected.\n\nOn Microsoft® Windows®, KDC policy is always respected.";

AuthSchemes.pfm\_title = "Supported authentication schemes";

AuthSchemes.pfm\_description = "Setting the policy specifies which HTTP authentication schemes Google Chrome supports.\n\nLeaving the policy unset employs all 4 schemes.\n\nValid values:\n\n\* basic\n\n\* digest\n\n\* ntlm\n\n\* negotiate\n\nNote: Separate multiple values with commas.";

AuthServerAllowlist.pfm\_title = "Authentication server allowlist";

AuthServerAllowlist.pfm\_description = "Setting the policy specifies which servers should be allowed for integrated authentication. Integrated authentication is only on when Google Chrome gets an authentication challenge from a proxy or from a server in this permitted list.\n\nLeaving the policy unset means Google Chrome tries to detect if a server is on the intranet. Only then will it respond to IWA requests. If a server is detected as internet, then Google Chrome ignores IWA requests from it.\n\nNote: Separate multiple server names with commas. Wildcards, \*, are allowed.";

AutoLaunchProtocolsFromOrigins.pfm\_title = "Define a list of protocols that can launch an external application from listed origins without prompting the user";

AutoLaunchProtocolsFromOrigins.pfm\_description = "Allows you to set a list of protocols, and for each protocol an associated list of allowed origin patterns, that can launch an external application without prompting the user. The trailing separator should not be included when listing the protocol, so list \"skype\" instead of \"skype:\" or \"skype://\".\n\nIf this policy is set, a protocol will only be permitted to launch an external application without prompting by policy if the protocol is listed, and the origin of the site trying to launch the protocol matches one of the origin patterns in that protocol's allowed\_origins list. If either condition is false the external protocol launch prompt will not be omitted by policy.\n\nIf this policy is not set, no protocols can launch without a prompt by default. Users may opt out of prompts on a per-protocol/per-site basis unless the ExternalProtocolDialogShowAlwaysOpenCheckbox policy is set to Disabled. This policy has no impact on per-protocol/per-site prompt exemptions set by users.\n\nThe origin matching patterns use a similar format to those for the 'URLBlocklist' policy, which are documented at http://www.chromium.org/administrators/url-blocklist-filter-format.\n\nHowever, origin matching patterns for this policy cannot contain \"/path\" or \"@query\" elements. Any pattern that does contain a \"/path\" or \"@query\" element will be ignored.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=AutoLaunchProtocolsFromOrigins for more information about schema and formatting.";

AutoOpenAllowedForURLs.pfm\_title = "URLs where AutoOpenFileTypes can apply";

AutoOpenAllowedForURLs.pfm\_description = "List of URLs specifying which urls AutoOpenFileTypes will apply to. This policy has no impact on automatically open values set by users.\n\nIf this policy is set, files will only automatically open by policy if the url is part of this set and the file type is listed in AutoOpenFileTypes. If either condition is false the download won't automatically open by policy.\n\nIf this policy isn't set, all downloads where the file type is in AutoOpenFileTypes will automatically open.\n\nA URL pattern has to be formatted according to https://www.chromium.org/administrators/url-blocklist-filter-format.";

AutoOpenFileTypes.pfm\_title = "List of file types that should be automatically opened on download";

AutoOpenFileTypes.pfm\_description = "List of file types that should be automatically opened on download. The leading separator should not be included when listing the file type, so list \"txt\" instead of \".txt\".\n\nFiles with types that should be automatically opened will still be subject to the enabled safe browsing checks and won't be opened if they fail those checks.\n\nIf this policy isn't set, only file types that a user has already specified to automatically be opened will do so when downloaded.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

AutoSelectCertificateForUrls.pfm\_title = "Automatically select client certificates for these sites";

AutoSelectCertificateForUrls.pfm\_description = "Setting the policy lets you make a list of URL patterns that specify sites for which Chrome can automatically select a client certificate. The value is an array of stringified JSON dictionaries, each with the form { \"pattern\": \"$URL\_PATTERN\", \"filter\" : $FILTER }, where $URL\_PATTERN is a content setting pattern. $FILTER restricts the client certificates the browser automatically selects from. Independent of the filter, only certificates that match the server's certificate request are selected.\n\nExamples for the usage of the $FILTER section:\n\n\* When $FILTER is set to { \"ISSUER\": { \"CN\": \"$ISSUER\_CN\" } }, only client certificates issued by a certificate with the CommonName $ISSUER\_CN are selected.\n\n\* When $FILTER contains both the \"ISSUER\" and the \"SUBJECT\" sections, only client certificates that satisfy both conditions are selected.\n\n\* When $FILTER contains a \"SUBJECT\" section with the \"O\" value, a certificate needs at least one organization matching the specified value to be selected.\n\n\* When $FILTER contains a \"SUBJECT\" section with a \"OU\" value, a certificate needs at least one organizational unit matching the specified value to be selected.\n\n\* When $FILTER is set to {}, the selection of client certificates is not additionally restricted. Note that filters provided by the web server still apply.\n\nLeaving the policy unset means there's no autoselection for any site.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=AutoSelectCertificateForUrls for more information about schema and formatting.";

AutofillAddressEnabled.pfm\_title = "Enable AutoFill for addresses";

AutofillAddressEnabled.pfm\_description = "Setting the policy to True or leaving it unset gives users control of Autofill for addresses in the UI.\n\nSetting the policy to False means Autofill never suggests or fills address information, nor does it save additional address information that users submit while browsing the web.";

AutofillCreditCardEnabled.pfm\_title = "Enable AutoFill for credit cards";

AutofillCreditCardEnabled.pfm\_description = "Setting the policy to True or leaving it unset means users can control autofill suggestions for credit cards in the UI.\n\nSetting the policy to False means autofill never suggests or fills credit card information, nor will it save additional credit card information that users might submit while browsing the web.";

AutoplayAllowed.pfm\_title = "Allow media autoplay";

AutoplayAllowed.pfm\_description = "Setting the policy to True lets Google Chrome autoplay media. Setting the policy to False stops Google Chrome from autoplaying media.\n\n By default, Google Chrome doesn't autoplay media. But, for certain URL patterns, you can use the AutoplayAllowlist policy to change this setting.\n\nIf this policy changes while Google Chrome is running, it only applies to newly opened tabs.";

AutoplayAllowlist.pfm\_title = "Allow media autoplay on a allowlist of URL patterns";

AutoplayAllowlist.pfm\_description = "Setting the policy lets videos play automatically (without user consent) with audio content in Google Chrome. If AutoplayAllowed policy is set to True, then this policy has no effect. If AutoplayAllowed is set to False, then any URL patterns set in this policy can still play. If this policy changes while Google Chrome is running, it only applies to newly opened tabs.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns.";

BasicAuthOverHttpEnabled.pfm\_title = "Allow Basic authentication for HTTP";

BasicAuthOverHttpEnabled.pfm\_description = "Setting the policy to Enabled or leaving it unset will allow Basic authentication challenges received over non-secure HTTP.\n\nSetting the policy to Disabled forbids non-secure HTTP requests from using the Basic authentication scheme; only secure HTTPS is allowed.\n\nThis policy setting is ignored (and Basic is always forbidden) if the AuthSchemes policy is set and does not include Basic.";

BlockExternalExtensions.pfm\_title = "Blocks external extensions from being installed";

BlockExternalExtensions.pfm\_description = "Controls external extensions installation.\n\nEnabling this setting blocks external extensions from being installed.\n\nDisabling this setting or leaving it unset allows external extensions to be installed.\n\nExternal extensions and their installation are documented at https://developer.chrome.com/apps/external\_extensions.\n";

BlockThirdPartyCookies.pfm\_title = "Block third party cookies";

BlockThirdPartyCookies.pfm\_description = "Setting the policy to Enabled prevents webpage elements that aren't from the domain that's in the browser's address bar from setting cookies. Setting the policy to Disabled lets those elements set cookies and prevents users from changing this setting.\n\nLeaving it unset turns third-party cookies on, but users can change this setting.";

BookmarkBarEnabled.pfm\_title = "Enable Bookmark Bar";

BookmarkBarEnabled.pfm\_description = "Setting the policy to True displays a bookmark bar in Google Chrome. Setting the policy to False means users never see the bookmark bar.\n\nIf you set the policy, users can't change it. If not set, users decide whether to use this function.";

BrowserAddPersonEnabled.pfm\_title = "Enable add person in user manager";

BrowserAddPersonEnabled.pfm\_description = "If this policy is set to true or not configured, Google Chrome and Lacros will allow to add a new person from the user manager.\n\nIf this policy is set to false, Google Chrome and Lacros will not allow adding a new person from the user manager.\n\nNote: If this policy is not configured or set to true, but LacrosSecondaryProfilesAllowed is set to false, Lacros will not allow adding a new person from the user manager.";

BrowserGuestModeEnabled.pfm\_title = "Enable guest mode in browser";

BrowserGuestModeEnabled.pfm\_description = "If this policy is set to true or not configured, Google Chrome and Lacros will enable guest logins. Guest logins are Google Chrome profiles where all windows are in incognito mode.\n\nIf this policy is set to false, Google Chrome and Lacros will not allow guest profiles to be started.\n\nNote: If this policy is not configured or set to true, but LacrosSecondaryProfilesAllowed is set to false, Lacros will not allow guest profiles to be started.";

BrowserGuestModeEnforced.pfm\_title = "Enforce browser guest mode";

BrowserGuestModeEnforced.pfm\_description = "Setting the policy to Enabled means Google Chrome enforces guest sessions and prevents profile sign-ins. Guest sign-ins are Google Chrome profiles where windows are in Incognito mode.\n\nSetting the policy to Disabled, leaving it unset, or disabling browser Guest mode (through BrowserGuestModeEnabled) allows the use of new and existing profiles.";

BrowserLabsEnabled.pfm\_title = "Browser experiments icon in toolbar";

BrowserLabsEnabled.pfm\_description = "Setting the policy to Enabled or leaving the policy unset means that users can access browser experimental features through an icon in the toolbar\n\nSetting the policy to Disabled removes the browser experimental features icon from the toolbar.\n\nchrome://flags and any other means of turning off and on browser features will still behave as expected regardless of whether this policy is Enabled or Disabled.";

BrowserNetworkTimeQueriesEnabled.pfm\_title = "Allow queries to a Google time service";

BrowserNetworkTimeQueriesEnabled.pfm\_description = "Setting the policy to Enabled or leaving it unset means Google Chrome send occasional queries to a Google server to retrieve an accurate timestamp.\n\nSetting the policy to Disabled stops Google Chrome from sending these queries.";

BrowserSignin.pfm\_title = "Browser sign in settings";

BrowserSignin.pfm\_description = "0 - Disable browser sign-in\n1 - Enable browser sign-in\n2 - Force users to sign-in to use the browser\nThis policy controls the sign-in behavior of the browser. It allows you to specify if the user can sign in to Google Chrome with their account and use account related services like Google Chrome Sync.\n\nIf the policy is set to \"Disable browser sign-in\" then the user cannot sign in to the browser and use account-based services. In this case browser-level features like Google Chrome Sync cannot be used and will be unavailable. On iOS, if the user was signed in and the policy is set to \"Disabled\" they will be signed out immediately. On other platforms, they will be signed out the next time they run Google Chrome. On all platforms, their local profile data like bookmarks, passwords etc. will be preserved and still usable. The user will still be able to sign into and use Google web services like Gmail.\n\nIf the policy is set to \"Enable browser sign-in,\" then the user is allowed to sign in to the browser. On all platforms except iOS, the user is automatically signed in to the browser when signed in to Google web services like Gmail. Being signed in to the browser means the user's account information will be kept by the browser. However, it does not mean that Google Chrome Sync will be turned on by default; the user must separately opt-in to use this feature. Enabling this policy will prevent the user from turning off the setting that allows browser sign-in. To control the availability of Google Chrome Sync, use the SyncDisabled policy.\n\nIf the policy is set to \"Force browser sign-in\" the user is presented with an account selection dialog and has to choose and sign in to an account to use the browser. This ensures that for managed accounts the policies associated with the account are applied and enforced. The default value of BrowserGuestModeEnabled will be set to disabled. Note that existing unsigned profiles will be locked and inaccessible after enabling this policy. For more information, see help center article: https://support.google.com/chrome/a/answer/7572556 . This option is not supported on Linux, Android or iOS. It will fall back to \"Enable browser sign-in\" if used.\n\nIf this policy is not set then the user can decide if they want to enable browser sign-in in the Google Chrome settings and use it as they see fit.";

BrowserSwitcherDelay.pfm\_title = "Delay before launching alternative browser (milliseconds)";

BrowserSwitcherDelay.pfm\_description = "Setting the policy to a number has Google Chrome show a message for that number of milliseconds, then it opens an alternative browser.\n\nLeaving the policy unset or set to 0 means navigating to a designated URL immediately opens it in an alternative browser.";

BrowserSwitcherEnabled.pfm\_title = "Enable the Legacy Browser Support feature.";

BrowserSwitcherEnabled.pfm\_description = "Setting the policy to Enabled means Google Chrome will try to launch some URLs in an alternate browser, such as Internet Explorer®. This feature is set using the policies in the Legacy Browser support group.\n\nSetting the policy to Disabled or leaving it unset means Google Chrome won't try to launch designated URLs in an alternate browser.";

BrowserSwitcherExternalGreylistUrl.pfm\_title = "URL of an XML file that contains URLs that should never trigger a browser switch.";

BrowserSwitcherExternalGreylistUrl.pfm\_description = "Setting the policy to a valid URL has Google Chrome download the site list from that URL and apply the rules as if they were set up with the BrowserSwitcherUrlGreylist policy. These policies prevent Google Chrome and the alternative browser from opening one another.\n\nLeaving it unset (or set to a invalid URL) means Google Chrome doesn't use the policy as a source of rules for not switching browsers.\n\nNote: This policy points to an XML file in the same format as Internet Explorer®'s SiteList policy. This loads rules from an XML file, without sharing those rules with Internet Explorer®. Read more on Internet Explorer®'s SiteList policy ( https://docs.microsoft.com/internet-explorer/ie11-deploy-guide/what-is-enterprise-mode )";

BrowserSwitcherExternalSitelistUrl.pfm\_title = "URL of an XML file that contains URLs to load in an alternative browser.";

BrowserSwitcherExternalSitelistUrl.pfm\_description = "Setting the policy to a valid URL has Google Chrome download the site list from that URL and apply the rules as if they were set up with the BrowserSwitcherUrlList policy.\n\nLeaving it unset (or set to a invalid URL) means Google Chrome doesn't use the policy as a source of rules for switching browsers.\n\nNote: This policy points to an XML file in the same format as Internet Explorer®'s SiteList policy. This loads rules from an XML file, without sharing those rules with Internet Explorer®. Read more on Internet Explorer®'s SiteList policy ( https://docs.microsoft.com/internet-explorer/ie11-deploy-guide/what-is-enterprise-mode)";

BrowserSwitcherKeepLastChromeTab.pfm\_title = "Keep last tab open in Chrome.";

BrowserSwitcherKeepLastChromeTab.pfm\_description = "Setting the policy to Enabled or leaving it unset has Google Chrome keep at least one tab open, after switching to an alternate browser.\n\nSetting the policy to Disabled has Google Chrome close the tab after switching to an alternate browser, even if it was the last tab. This causes Google Chrome to exit completely.";

BrowserSwitcherParsingMode.pfm\_title = "Sitelist parsing mode";

BrowserSwitcherParsingMode.pfm\_description = "0 - Default behavior for LBS.\n1 - More compatible with Microsoft IE/Edge enterprise mode sitelists.\nThis policy controls how Google Chrome interprets sitelist/greylist policies for the Legacy Browser Support feature. It affects the following policies: BrowserSwitcherUrlList, BrowserSwitcherUrlGreylist, BrowserSwitcherUseIeSitelist, BrowserSwitcherExternalSitelistUrl, and BrowserSwitcherExternalGreylistUrl.\n\nIf 'Default' (0) or unset, URL matching is less strict. Rules that do not contain \"/\" look for a substring anywhere in the URL's hostname. Matching the path component of a URL is case-sensitive.\n\nIf 'IESiteListMode' (1), URL matching is more strict. Rules that do not contain \"/\" only match at the end of the hostname. They must also be at a domain name boundary. Matching the path component of a URL is case-insensitive. This is more compatible with Microsoft® Internet Explorer® and Microsoft® Edge®.\n\nFor example, with the rules \"example.com\" and \"acme.com/abc\":\n\n\"http://example.com/\", \"http://subdomain.example.com/\" and \"http://acme.com/abc\" match regardless of parsing mode.\n\n\"http://notexample.com/\", \"http://example.com.invalid.com/\", \"http://example.comabc/\" only match in 'Default' mode.\n\n\"http://acme.com/ABC\" only matches in 'IESiteListMode'.";

BrowserSwitcherUrlGreylist.pfm\_title = "Websites that should never trigger a browser switch.";

BrowserSwitcherUrlGreylist.pfm\_description = "Setting the policy controls the list of websites that will never cause a browser switch. Each item is treated as a rule. Those rules that match won't open an alternative browser. Unlike the BrowserSwitcherUrlList policy, rules apply to both directions. When the Internet Explorer® add-in is on, it also controls whether Internet Explorer® should open these URLs in Google Chrome.\n\nLeaving the policy unset adds no websites to the list.\n\nNote: Elements can also be added to this list through the BrowserSwitcherExternalGreylistUrl policy.";

BrowserSwitcherUrlList.pfm\_title = "Websites to open in alternative browser";

BrowserSwitcherUrlList.pfm\_description = "Setting the policy controls the list of websites to open in an alternative browser. Each item is treated as a rule for something to open in an alternative browser. Google Chrome uses those rules when choosing if a URL should open in an alternative browser. When the Internet Explorer® add-in is on, Internet Explorer® switches back to Google Chrome when the rules don't match. If rules contradict each other, Google Chrome uses the most specific rule.\n\nLeaving the policy unset adds no websites to the list.\n\nNote: Elements can also be added to this list through the BrowserSwitcherUseIeSitelist and BrowserSwitcherExternalSitelistUrl policies.";

BrowserThemeColor.pfm\_title = "Configure the color of the browser's theme";

BrowserThemeColor.pfm\_description = "This policy allows admins to configure the color of Google Chrome's theme. The input string should be a valid hex color string matching the format \"#RRGGBB\".\n\nSetting the policy to a valid hex color causes a theme based on that color to be automatically generated and applied to the browser. Users won't be able to change the theme set by the policy.\n\nLeaving the policy unset lets users change their browser's theme as preferred.";

BrowsingDataLifetime.pfm\_title = "Browsing Data Lifetime Settings";

BrowsingDataLifetime.pfm\_description = "Configures browsing data lifetime settings for Google Chrome. This policy allows admins to configure (per data-type) when data is deleted by the browser. This is useful for customers that work with sensitive customer data. The policy will only take effect if SyncDisabled is set to true.\n\nThe available data types are 'browsing\_history', 'download\_history', 'cookies\_and\_other\_site\_data', 'cached\_images\_and\_files', 'password\_signin', 'autofill', 'site\_settings' and 'hosted\_app\_data'.\n\nThe browser will automatically remove data of selected types that is older than 'time\_to\_live\_in\_hours'. The minimum value that can be set is 1 hour.\n\nThe deletion of expired data will happen 15 seconds after the browser starts then every hour while the browser is running.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=BrowsingDataLifetime for more information about schema and formatting.";

BuiltInDnsClientEnabled.pfm\_title = "Use built-in DNS client";

BuiltInDnsClientEnabled.pfm\_description = "This policy controls which software stack is used to communicate with the DNS server: the Operating System DNS client, or Google Chrome's built-in DNS client. This policy does not affect which DNS servers are used: if, for example, the operating system is configured to use an enterprise DNS server, that same server would be used by the built-in DNS client. It also does not control if DNS-over-HTTPS is used; Google Chrome will always use the built-in resolver for DNS-over-HTTPS requests. Please see the DnsOverHttpsMode policy for information on controlling DNS-over-HTTPS.\n\nIf this policy is set to Enabled, the built-in DNS client will be used, if available.\n\nIf this policy is set to Disabled, the built-in DNS client will only be used when DNS-over-HTTPS is in use.\n\nIf this policy is left unset, the built-in DNS client will be enabled by default on macOS, Android (when neither Private DNS nor VPN are enabled) and Google Chrome OS.";

BuiltinCertificateVerifierEnabled.pfm\_title = "Determines whether the built-in certificate verifier will be used to verify server certificates";

BuiltinCertificateVerifierEnabled.pfm\_description = "When this setting is enabled, Google Chrome will perform verification of server certificates using the built-in certificate verifier.\nWhen this setting is disabled, Google Chrome will perform verification of server certificates using the legacy certificate verifier provided by the platform.\nWhen this setting is not set, the built-in or the legacy certificate verifier may be used.\n\nThis policy is planned to be removed in Google Chrome for macOS version 107, when support for the legacy certificate verifier on macOS is planned to be removed.\n";

CECPQ2Enabled.pfm\_title = "CECPQ2 post-quantum key-agreement enabled for TLS";

CECPQ2Enabled.pfm\_description = "If this policy is not configured, or is set to enabled, then Google Chrome will follow the default rollout process for CECPQ2, a post-quantum key-agreement algorithm in TLS.\n\nCECPQ2 results in larger TLS messages which, in very rare cases, can trigger bugs in some networking hardware. This policy can be set to False to disable CECPQ2 while networking issues are resolved.\n\nThis policy is a temporary measure and will be removed in future versions of Google Chrome.";

CORSNonWildcardRequestHeadersSupport.pfm\_title = "CORS non-wildcard request headers support";

CORSNonWildcardRequestHeadersSupport.pfm\_description = "Configures support of CORS non-wildcard request headers.\n\nGoogle Chrome version 97 introduces support for CORS non-wildcard request headers. When scripts make a cross-origin network request via fetch() and XMLHttpRequest with a script-added Authorization header, the header must be explicitly allowed by the Access-Control-Allow-Headers header in the CORS preflight response. \"Explicitly\" here means that the wild card symbol \"\*\" doesn't cover the Authorization header. See https://www.chromest atus.com/feature/5768642492891136 for more detail.\n\nIf this policy is not set, or set to True, Google Chrome will support the CORS non-wildcard request headers and behave as described above.\n\nWhen this policy is set to False, chrome will allow the wildcard symbol (\"\*\") in the Access-Control-Allow-Headers header in the CORS preflight response to cover the Authorization header.\n\nThis Enterprise policy is temporary; it's intended to be removed after Google Chrome version 103.";

CertificateTransparencyEnforcementDisabledForCas.pfm\_title = "Disable Certificate Transparency enforcement for a list of subjectPublicKeyInfo hashes";

CertificateTransparencyEnforcementDisabledForCas.pfm\_description = "Setting the policy turns off enforcement of Certificate Transparency disclosure requirements for a list of subjectPublicKeyInfo hashes. Enterprise hosts can keep using certificates that otherwise wouldn't be trusted (because they weren't properly publicly disclosed). To turn off enforcement, the hash must meet one of these conditions:\n\n\* It's of the server certificate's subjectPublicKeyInfo.\n\n\* It's of a subjectPublicKeyInfo that appears in a Certificate Authority (CA) certificate in the certificate chain. That CA certificate is constrained through the X.509v3 nameConstraints extension, one or more directoryName nameConstraints are present in the permittedSubtrees, and the directoryName has an organizationName attribute.\n\n\* It's of a subjectPublicKeyInfo that appears in a CA certificate in the certificate chain, the CA certificate has one or more organizationName attributes in the certificate Subject, and the server's certificate has the same number of organizationName attributes, in the same order, and with byte-for-byte identical values.\n\nSpecify a subjectPublicKeyInfo hash by linking the hash algorithm name, a slash, and the Base64 encoding of that hash algorithm applied to the DER-encoded subjectPublicKeyInfo of the specified certificate. Base64 encoding format matches that of an SPKI Fingerprint. The only recognized hash algorithm is sha256; others are ignored.\n\nLeaving the policy unset means that if certificates requiring disclosure through Certificate Transparency aren't disclosed, then Google Chrome doesn't trust those certificates.";

CertificateTransparencyEnforcementDisabledForLegacyCas.pfm\_title = "Disable Certificate Transparency enforcement for a list of Legacy Certificate Authorities";

CertificateTransparencyEnforcementDisabledForLegacyCas.pfm\_description = "Setting the policy turns off enforcement of Certificate Transparency disclosure requirements for a list of Legacy Certificate Authorities (CA) for certificate chains with a specified subjectPublicKeyInfo hash. Enterprise hosts can keep using certificates that otherwise wouldn't be trusted (because they weren't properly publicly disclosed). To turn off enforcement, the subjectPublicKeyInfo hash must appear in a CA certificate recognized as a Legacy CA. A Legacy CA is publicly trusted by one or more operating systems supported by Google Chrome, but not Android Open Source Project or Google Chrome OS.\n\nSpecify a subjectPublicKeyInfo hash by linking the hash algorithm name, a slash and the Base64 encoding of that hash algorithm applied to the DER-encoded subjectPublicKeyInfo of the specified certificate. Base64 encoding format matches that of an SPKI Fingerprint. The only recognized hash algorithm is sha256; others are ignored.\n\nLeaving the policy unset means that if certificates requiring disclosure through Certificate Transparency aren't disclosed, then Google Chrome doesn't trust those certificates.";

CertificateTransparencyEnforcementDisabledForUrls.pfm\_title = "Disable Certificate Transparency enforcement for a list of URLs";

CertificateTransparencyEnforcementDisabledForUrls.pfm\_description = "Setting the policy turns off Certificate Transparency disclosure requirements for the hostnames in the specified URLs. While making it harder to detect misissued certificates, hosts can keep using certificates that otherwise wouldn't be trusted (because they weren't properly publicly disclosed).\n\nLeaving the policy unset means that if certificates requiring disclosure through Certificate Transparency aren't disclosed, then Google Chrome doesn't trust those certificates.\n\nA URL pattern follows this format ( https://www.chromium.org/administrators/url-blocklist-filter-format ). However, because the validity of certificates for a given hostname is independent of the scheme, port, or path, Google Chrome only considers the hostname portion of the URL. Wildcard hosts aren't supported.";

ChromeVariations.pfm\_title = "Determine the availability of variations";

ChromeVariations.pfm\_description = "0 - Enable all variations\n1 - Enable variations concerning critical fixes only\n2 - Disable all variations\nConfiguring this policy allows to specify which variations are allowed to be applied in Google Chrome.\n\nVariations provide a means for offering modifications to Google Chrome without shipping a new version of the browser by selectively enabling or disabling already existing features. See https://support.google.com/chrome/a?p=Manage\_the\_Chrome\_variations\_framework for more information.\n\nSetting the VariationsEnabled (value 0), or leaving the policy not set allows all variations to be applied to the browser.\n\nSetting the CriticalFixesOnly (value 1), allows only variations considered critical security or stability fixes to be applied to Google Chrome.\n\nSetting the VariationsDisabled (value 2), prevent all variations from being applied to the browser. Please note that this mode can potentially prevent the Google Chrome developers from providing critical security fixes in a timely manner and is thus not recommended.";

ClearBrowsingDataOnExitList.pfm\_title = "Clear Browsing Data on Exit";

ClearBrowsingDataOnExitList.pfm\_description = "browsing\_history - Browsing history\ndownload\_history - Download history\ncookies\_and\_other\_site\_data - Cookies and other site data\ncached\_images\_and\_files - Cached images and files\npassword\_signin - Password signin\nautofill - Autofill\nsite\_settings - Site settings\nhosted\_app\_data - Hosted apps data\nConfigures a list of browsing data types that should be deleted when the user closes all browser windows. The available data types are browsing history (browsing\_history), download history (download\_history), cookies (cookies\_and\_other\_site\_data), cache(cached\_images\_and\_files), autofill (autofill), passwords (password\_signin), site settings (site\_settings) and hosted apps data (hosted\_app\_data). This policy does not take precedence over AllowDeletingBrowserHistory.\n\nThis policy requires the SyncDisabled policy to be set to true, otherwise it will be ignored. If this policy is set at platform level, Sync should be disabled at platform level. If this policy is set at user level, Sync should be disabled for that user in order for this policy to take effect.\n\nIf Google Chrome does not exit cleanly (for example, if the browser or the OS crashes), the browsing data will be cleared the next time the profile is loaded.";

ClickToCallEnabled.pfm\_title = "Enable the Click to Call Feature";

ClickToCallEnabled.pfm\_description = "Enable the Click to Call feature which allows users to send phone numbers from Chrome Desktops to an Android device when the user is Signed-in. For more information, see help center article: https://support.google.com/chrome/answer/9430554?hl=en.\n\nIf this policy is set to enabled, the capability of sending phone numbers to Android devices will be enabled for the Chrome user.\n\nIf this policy is set to disabled, the capability of sending phone numbers to Android devices will be disabled for the Chrome user.\n\nIf you set this policy, users cannot change or override it.\n\nIf this policy is left unset, the Click to Call feature is enabled by default.";

ClipboardAllowedForUrls.pfm\_title = "Allow clipboard on these sites";

ClipboardAllowedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify sites that can use the clipboard site permission. This does not include all clipboard operations on origins matching the patterns. For instance, users will still be able to paste using keyboard shortcuts as this isn't gated by the clipboard site permission.\n\n\nLeaving the policy unset means DefaultClipboardSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

ClipboardBlockedForUrls.pfm\_title = "Block clipboard on these sites";

ClipboardBlockedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify sites that can't use the clipboard site permission. This does not include all clipboard operations on origins matching the patterns. For instance, users will still be able to paste using keyboard shortcuts as this isn't gated by the clipboard site permission.\n\nLeaving the policy unset means DefaultClipboardSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

CloudManagementEnrollmentMandatory.pfm\_title = "Enable mandatory cloud management enrollment";

CloudManagementEnrollmentMandatory.pfm\_description = "Setting the policy to Enabled mandates Chrome Browser Cloud Management enrollment and blocks Google Chrome launch process if failed.\n\nSetting the policy to Disabled or leaving it unset renders Chrome Browser Cloud Management optional and doesn't block Google Chrome launch process if failed.\n\nMachine scope cloud policy enrollment on desktop uses this policy. See https://support.google.com/chrome/a/answer/9301891?ref\_topic=9301744 for details.";

CloudManagementEnrollmentToken.pfm\_title = "The enrollment token of cloud policy";

CloudManagementEnrollmentToken.pfm\_description = "Setting the policy means Google Chrome tries to register itself with Chrome Browser Cloud Management. The value of this policy is an enrollment token you can retrieve from the Google Admin console.\n\nSee https://support.google.com/chrome/a/answer/9301891?ref\_topic=9301744 for details.";

CloudPolicyOverridesPlatformPolicy.pfm\_title = "Google Chrome cloud policy overrides Platform policy.";

CloudPolicyOverridesPlatformPolicy.pfm\_description = "Setting the policy to Enabled means cloud policy takes precedence if it conflicts with platform policy.\n\nSetting the policy to Disabled or leaving it unset means platform policy takes precedence if it conflicts with cloud policy.\n\nThis mandatory policy affects machine scope cloud policies.";

CloudPrintProxyEnabled.pfm\_title = "Enable Google Cloud Print proxy";

CloudPrintProxyEnabled.pfm\_description = "Setting the policy to Enabled or leaving it unset lets Google Chrome act as a proxy between Google Cloud Print and legacy printers connected to the machine. Using their Google Account, users may turn on the cloud print proxy by authentication.\n\nSetting the policy to Disabled means users can't turn on the proxy, and the machine can't share its printers with Google Cloud Print.";

CloudUserPolicyMerge.pfm\_title = "Enables merging of user cloud policies into machine-level policies";

CloudUserPolicyMerge.pfm\_description = "Setting the policy to Enabled allows policies associated with a Google Workspace account to be merged into machine-level policies.\n\nOnly policies originating from secure users can be merged. A secure user is affiliated with the organization that manages their browser using Chrome Browser Cloud Management. All other user-level policies will always be ignored.\n\nPolicies that need to be merged also need to be set in either PolicyListMultipleSourceMergeList or PolicyDictionaryMultipleSourceMergeList. This policy will be ignored if neither of the two aforementioned policies is configured.\n\nLeaving the policy unset or setting it to Disabled prevents user-level cloud policies from being merged with policies from any other sources.";

CloudUserPolicyOverridesCloudMachinePolicy.pfm\_title = "Allow user cloud policies to override Chrome Browser Cloud Management policies.";

CloudUserPolicyOverridesCloudMachinePolicy.pfm\_description = "Setting the policy to Enabled allows policies associated with a Google Workspace account to take precedence if they conflict with Chrome Browser Cloud Management policies.\n\nOnly policies originating from secure users can take precedence. A secure user is affiliated with the organization that manages their browser using Chrome Browser Cloud Management. All other user-level policies will have default precedence.\n\nThe policy can be combined with CloudPolicyOverridesPlatformPolicy. If both policies are enabled, user cloud policies will also take precedence over conflicting platform policies.\n\nLeaving the policy unset or setting it to disabled causes user-level cloud policies to have default priority.";

CommandLineFlagSecurityWarningsEnabled.pfm\_title = "Enable security warnings for command-line flags";

CommandLineFlagSecurityWarningsEnabled.pfm\_description = "Setting the policy to Enabled or leaving it unset means security warnings appear when potentially dangerous command-line flags are used to launch Chrome.\n\nSetting the policy to Disabled prevents security warnings from appearing when Chrome is launched with potentially dangerous command-line flags.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

ComponentUpdatesEnabled.pfm\_title = "Enable component updates in Google Chrome";

ComponentUpdatesEnabled.pfm\_description = "Enables component updates for all components in Google Chrome when not set or set to enabled.\n\nIf set to disabled, updates to components are disabled. However, some components are exempt from this policy: updates to any component that does not contain executable code and is critical for the security of the browser will not be disabled.\nExamples of such components include the certificate revocation lists and subresource filters.";

CookiesAllowedForUrls.pfm\_title = "Allow cookies on these sites";

CookiesAllowedForUrls.pfm\_description = "Allows you to set a list of url patterns that specify sites which are allowed to set cookies.\n\nIf this policy is left not set the global default value will be used for all sites either from the DefaultCookiesSetting policy if it is set, or the user's personal configuration otherwise.\n\nSee also policies CookiesBlockedForUrls and CookiesSessionOnlyForUrls. Note that there must be no conflicting URL patterns between these three policies - it is unspecified which policy takes precedence.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

CookiesBlockedForUrls.pfm\_title = "Block cookies on these sites";

CookiesBlockedForUrls.pfm\_description = "Setting the policy lets you make a list of URL patterns that specify sites that can't set cookies.\n\nLeaving the policy unset results in the use of DefaultCookiesSetting for all sites, if it's set. If not, the user's personal setting applies.\n\nWhile no specific policy takes precedence, see CookiesAllowedForUrls and CookiesSessionOnlyForUrls. URL patterns among these 3 policies must not conflict.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

CookiesSessionOnlyForUrls.pfm\_title = "Limit cookies from matching URLs to the current session";

CookiesSessionOnlyForUrls.pfm\_description = "Unless the RestoreOnStartup policy is set to permanently restore URLs from previous sessions, then setting CookiesSessionOnlyForUrls lets you make a list of URL patterns that specify sites that can and can't set cookies for one session.\n\nLeaving the policy unset results in the use of DefaultCookiesSetting for all sites, if it's set. If not, the user's personal setting applies. URLs not covered by the patterns specified also result in the use of defaults.\n\nWhile no specific policy takes precedence, see CookiesBlockedForUrls and CookiesAllowedForUrls. URL patterns among these 3 policies must not conflict.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

DNSInterceptionChecksEnabled.pfm\_title = "DNS interception checks enabled";

DNSInterceptionChecksEnabled.pfm\_description = "This policy configures a local switch that can be used to disable DNS interception checks. The checks attempt to discover whether the browser is behind a proxy that redirects unknown host names.\n\nThis detection may not be necessary in an enterprise environment where the network configuration is known, since it causes some amount of DNS and HTTP traffic on start-up and each DNS configuration change.\n\nWhen this policy is not set, or is enabled, the DNS interception checks are performed. When explicitly disabled, they're not.";

DefaultBrowserSettingEnabled.pfm\_title = "Set Google Chrome as Default Browser";

DefaultBrowserSettingEnabled.pfm\_description = "Setting the policy to True has Google Chrome always check whether it's the default browser on startup and, if possible, automatically register itself. Setting the policy to False stops Google Chrome from ever checking if it's the default and turns user controls off for this option.\n\nLeaving the policy unset means Google Chrome lets users control whether it's the default and, if not, whether user notifications should appear.\n\nNote: For Microsoft®Windows® administrators, turning this setting on only works for machines running Windows 7. For later versions, you must deploy a \"default application associations\" file that makes Google Chrome the handler for the https and http protocols (and, optionally, the ftp protocol and other file formats). See Chrome Help ( https://support.google.com/chrome?p=make\_chrome\_default\_win ).";

DefaultClipboardSetting.pfm\_title = "Default clipboard setting";

DefaultClipboardSetting.pfm\_description = "2 - Do not allow any site to use the clipboard site permission\n3 - Allow sites to ask the user to grant the clipboard site permission\nSetting the policy to 2 blocks sites from using the clipboard site permission. Setting the policy to 3 or leaving it unset lets the user change the setting and decide if the clipboard APIs are available when a site wants to use one.\n\nThis policy can be overridden for specific URL patterns using the ClipboardAllowedForUrls and ClipboardBlockedForUrls policies.\n\nThis policy only affects clipboard operations controlled by the clipboard site permission, and does not affect sanitized clipboard writes or trusted copy and paste operations.";

DefaultCookiesSetting.pfm\_title = "Default cookies setting";

DefaultCookiesSetting.pfm\_description = "1 - Allow all sites to set local data\n2 - Do not allow any site to set local data\n4 - Keep cookies for the duration of the session\nUnless the RestoreOnStartup policy is set to permanently restore URLs from previous sessions, then setting CookiesSessionOnlyForUrls lets you make a list of URL patterns that specify sites that can and can't set cookies for one session.\n\nLeaving the policy unset results in the use of DefaultCookiesSetting for all sites, if it's set. If not, the user's personal setting applies. URLs not covered by the patterns specified also result in the use of defaults.\n\nWhile no specific policy takes precedence, see CookiesBlockedForUrls and CookiesAllowedForUrls. URL patterns among these 3 policies must not conflict.";

DefaultDownloadDirectory.pfm\_title = "Set default download directory";

DefaultDownloadDirectory.pfm\_description = "Setting the policy changes the default directory that Chrome downloads files to, but users can change the directory.\n\nLeaving the policy unset means Chrome uses its platform-specific default directory.\n\nThis policy has no effect if the policy DownloadDirectory is set.\n\nNote: See a list of variables you can use ( https://www.chromium.org/administrators/policy-list-3/user-data-directory-variables ).";

DefaultFileSystemReadGuardSetting.pfm\_title = "Control use of the File System API for reading";

DefaultFileSystemReadGuardSetting.pfm\_description = "2 - Do not allow any site to request read access to files and directories via the File System API\n3 - Allow sites to ask the user to grant read access to files and directories via the File System API\nSetting the policy to 3 lets websites ask for read access to files and directories in the host operating system's file system via the File System API. Setting the policy to 2 denies access.\n\nLeaving it unset lets websites ask for access, but users can change this setting.";

DefaultFileSystemWriteGuardSetting.pfm\_title = "Control use of the File System API for writing";

DefaultFileSystemWriteGuardSetting.pfm\_description = "2 - Do not allow any site to request write access to files and directories\n3 - Allow sites to ask the user to grant write access to files and directories\nSetting the policy to 3 lets websites ask for write access to files and directories in the host operating system's file system. Setting the policy to 2 denies access.\n\nLeaving it unset lets websites ask for access, but users can change this setting.";

DefaultGeolocationSetting.pfm\_title = "Default geolocation setting";

DefaultGeolocationSetting.pfm\_description = "1 - Allow sites to track the users' physical location\n2 - Do not allow any site to track the users' physical location\n3 - Ask whenever a site wants to track the users' physical location\nSetting the policy to 1 lets sites track the users' physical location as the default state. Setting the policy to 2 denies this tracking by default. You can set the policy to ask whenever a site wants to track the users' physical location.\n\nLeaving the policy unset means the AskGeolocation policy applies, but users can change this setting.";

DefaultImagesSetting.pfm\_title = "Default images setting";

DefaultImagesSetting.pfm\_description = "1 - Allow all sites to show all images\n2 - Do not allow any site to show images\nSetting the policy to 1 lets all websites display images. Setting the policy to 2 denies image display.\n\nLeaving it unset allows images, but users can change this setting.";

DefaultInsecureContentSetting.pfm\_title = "Control use of insecure content exceptions";

DefaultInsecureContentSetting.pfm\_description = "2 - Do not allow any site to load mixed content\n3 - Allow users to add exceptions to allow mixed content\nAllows you to set whether users can add exceptions to allow mixed content for specific sites.\n\nThis policy can be overridden for specific URL patterns using the 'InsecureContentAllowedForUrls' and 'InsecureContentBlockedForUrls' policies.\n\nIf this policy is left not set, users will be allowed to add exceptions to allow blockable mixed content and disable autoupgrades for optionally blockable mixed content.";

DefaultJavaScriptJitSetting.pfm\_title = "Control use of JavaScript JIT";

DefaultJavaScriptJitSetting.pfm\_description = "1 - Allow any site to run JavaScript JIT\n2 - Do not allow any site to run JavaScript JIT\nAllows you to set whether Google Chrome will run the v8 JavaScript engine with JIT (Just In Time) compiler enabled or not.\n\nDisabling the JavaScript JIT will mean that Google Chrome may render web content more slowly, and may also disable parts of JavaScript including WebAssembly. Disabling the JavaScript JIT may allow Google Chrome to render web content in a more secure configuration.\n\nThis policy can be overridden for specific URL patterns using the JavaScriptJitAllowedForSites and JavaScriptJitBlockedForSites policies.\n\nIf this policy is left not set, JavaScript JIT is enabled.";

DefaultJavaScriptSetting.pfm\_title = "Default JavaScript setting";

DefaultJavaScriptSetting.pfm\_description = "1 - Allow all sites to run JavaScript\n2 - Do not allow any site to run JavaScript\nSetting the policy to 1 lets websites run JavaScript. Setting the policy to 2 denies JavaScript.\n\nLeaving it unset allows JavaScript, but users can change this setting.";

DefaultLocalFontsSetting.pfm\_title = "Default Local Fonts permission setting";

DefaultLocalFontsSetting.pfm\_description = "2 - Denies the Local Fonts permission on all sites by default\n3 - Ask every time a site wants obtain the Local Fonts permission\nSetting the policy to BlockLocalFonts (value 2) automatically denies the local fonts permission to sites by default. This will limit the ability of sites to see information about local fonts.\n\nSetting the policy to AskLocalFonts (value 3) will prompt the user when the local fonts permission is requested by default. If users allow the permission, it will extend the ability of sites to see information about local fonts.\n\nLeaving the policy unset means the default behavior applies which is to prompt the user, but users can change this setting";

DefaultNotificationsSetting.pfm\_title = "Default notification setting";

DefaultNotificationsSetting.pfm\_description = "1 - Allow sites to show desktop notifications\n2 - Do not allow any site to show desktop notifications\n3 - Ask every time a site wants to show desktop notifications\nSetting the policy to 1 lets websites display desktop notifications. Setting the policy to 2 denies desktop notifications.\n\nLeaving it unset means AskNotifications applies, but users can change this setting.";

DefaultPopupsSetting.pfm\_title = "Default pop-ups setting";

DefaultPopupsSetting.pfm\_description = "1 - Allow all sites to show pop-ups\n2 - Do not allow any site to show pop-ups\nSetting the policy to 1 lets websites display pop-ups. Setting the policy to 2 denies pop-ups.\n\nLeaving it unset means BlockPopups applies, but users can change this setting.";

DefaultPrinterSelection.pfm\_title = "Default printer selection rules";

DefaultPrinterSelection.pfm\_description = "Setting the policy sets the rules for selecting the default printer in Google Chrome, overriding the default rules. Printer selection occurs the first time users try to print, when Google Chrome seeks a printer matching the specified attributes. In case of a less than perfect match, Google Chrome can be set to select any matching printer, depending on the order printers are discovered.\n\nLeaving the policy unset or set to attributes for which there's no match means the built-in PDF printer is the default. If there's no PDF printer, Google Chrome defaults to none.\n\nPrinters connected to Google Cloud Print are considered \"cloud\", the rest of the printers are classified as \"local\".\n\nNote: Omitting a field means all values match. For example, not specifying connectivity causes Print Preview to start discovery of all kinds of printers, \"local\" and \"cloud\". Regular expression patterns must follow the JavaScript RegExp syntax, and matches are case sensistive.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=DefaultPrinterSelection for more information about schema and formatting.";

DefaultSearchProviderAlternateURLs.pfm\_title = "List of alternate URLs for the default search provider";

DefaultSearchProviderAlternateURLs.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderAlternateURLs specifies a list of alternate URLs for extracting search terms from the search engine. The URLs should include the string '{searchTerms}'.\n\nLeaving DefaultSearchProviderAlternateURLs unset means no alternate URLs are used to extract search terms.";

DefaultSearchProviderContextMenuAccessAllowed.pfm\_title = "Allow default search provider context menu search access";

DefaultSearchProviderContextMenuAccessAllowed.pfm\_description = "Enables the use of a default search provider on the context menu.\n\nIf you set this policy to disabled the search context menu item that relies on your default search provider will not be available.\n\nIf this policy is set to enabled or not set, the context menu item for your default search provider will be available.\n\nThe policy value is only appled when the DefaultSearchProviderEnabled policy is enabled, and is not applicable otherwise.";

DefaultSearchProviderEnabled.pfm\_title = "Enable the default search provider";

DefaultSearchProviderEnabled.pfm\_description = "Setting the policy to Enabled means a default search is performed when a user enters non-URL text in the address bar. To specify the default search provider, set the rest of the default search policies. If you leave those policies empty, the user can choose the default provider. Setting the policy to Disabled means there's no search when the user enters non-URL text in the address bar.\n\nIf you set the policy, users can't change it in Google Chrome. If not set, the default search provider is on, and users can set the search provider list.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

DefaultSearchProviderEncodings.pfm\_title = "Default search provider encodings";

DefaultSearchProviderEncodings.pfm\_description = "If DefaultSearchProviderEnabled is on, setting DefaultSearchProviderEncodings specifies the character encodings supported by the search provider. Encodings are code page names such as UTF-8, GB2312, and ISO-8859-1. They're tried in the order provided.\n\nLeaving DefaultSearchProviderEncodings unset puts UTF-8 in use.";

DefaultSearchProviderIconURL.pfm\_title = "Default search provider icon";

DefaultSearchProviderIconURL.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderIconURL specifies the default search provider's favorite icon URL.\n\nLeaving DefaultSearchProviderIconURL unset means there's no icon for the search provider.";

DefaultSearchProviderImageURL.pfm\_title = "Parameter providing search-by-image feature for the default search provider";

DefaultSearchProviderImageURL.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderImageURL specifies the URL of the search engine used for image search. (If DefaultSearchProviderImageURLPostParams is set, then image search requests use the POST method instead.)\n\nLeaving DefaultSearchProviderImageURL unset means no image search is used.";

DefaultSearchProviderImageURLPostParams.pfm\_title = "Parameters for image URL which uses POST";

DefaultSearchProviderImageURLPostParams.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderImageURLPostParams specifies the parameters during image search with POST. It consists of comma-separated, name-value pairs. If a value is a template parameter, such as {imageThumbnail}, real image thumbnail data replaces it.\n\nLeaving DefaultSearchProviderImageURLPostParams unset means image search request is sent using the GET method.";

DefaultSearchProviderKeyword.pfm\_title = "Default search provider keyword";

DefaultSearchProviderKeyword.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderKeyword specifies the keyword or shortcut used in the address bar to trigger the search for this provider.\n\nLeaving DefaultSearchProviderKeyword unset means no keyword activates the search provider.";

DefaultSearchProviderName.pfm\_title = "Default search provider name";

DefaultSearchProviderName.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderName specifies the default search provider's name.\n\nLeaving DefaultSearchProviderName unset means the hostname specified by the search URL is used.";

DefaultSearchProviderNewTabURL.pfm\_title = "Default search provider new tab page URL";

DefaultSearchProviderNewTabURL.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderNewTabURL specifies the URL of the search engine used to provide a New Tab page.\n\nLeaving DefaultSearchProviderNewTabURL unset means no new tab page is provided.";

DefaultSearchProviderSearchURL.pfm\_title = "Default search provider search URL";

DefaultSearchProviderSearchURL.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderSearchURL specifies the URL of the search engine used during a default search. The URL should include the string '{searchTerms}', replaced in the query by the user's search terms.\n\nYou can specify Google's search URL as: '{google:baseURL}search?q={searchTerms}&{google:RLZ}{google:originalQueryForSuggestion}{google:assistedQueryStats}{google:searchFieldtrialParameter}{google:searchClient}{google:sourceId}ie={inputEncoding}'.";

DefaultSearchProviderSearchURLPostParams.pfm\_title = "Parameters for search URL which uses POST";

DefaultSearchProviderSearchURLPostParams.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderSearchURLPostParams specifies the parameters when searching a URL with POST. It consists of comma-separated, name-value pairs. If a value is a template parameter, such as '{searchTerms}', real search terms data replaces it.\n\nLeaving DefaultSearchProviderSearchURLPostParams unset means search requests are sent using the GET method.";

DefaultSearchProviderSuggestURL.pfm\_title = "Default search provider suggest URL";

DefaultSearchProviderSuggestURL.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderSuggestURL specifies the URL of the search engine to provide search suggestions. The URL should include the string '{searchTerms}', replaced in the query by the user's search terms.\n\nYou can specify Google's search URL as: '{google:baseURL}complete/search?output=chrome&q={searchTerms}'.";

DefaultSearchProviderSuggestURLPostParams.pfm\_title = "Parameters for suggest URL which uses POST";

DefaultSearchProviderSuggestURLPostParams.pfm\_description = "If DefaultSearchProviderEnabled is on, then setting DefaultSearchProviderSuggestURLPostParams specifies the parameters during suggestion search with POST. It consists of comma-separated, name-value pairs. If a value is a template parameter, such as '{searchTerms}', real search terms data replaces it.\n\nLeaving DefaultSearchProviderSuggestURLPostParams unset unset means suggest search requests are sent using the GET method.";

DefaultSensorsSetting.pfm\_title = "Default sensors setting";

DefaultSensorsSetting.pfm\_description = "1 - Allow sites to access sensors\n2 - Do not allow any site to access sensors\nSetting the policy to 1 lets websites access and use sensors such as motion and light. Setting the policy to 2 denies acess to sensors.\n\nLeaving it unset means AllowSensors applies, but users can change this setting.";

DefaultSerialGuardSetting.pfm\_title = "Control use of the Serial API";

DefaultSerialGuardSetting.pfm\_description = "2 - Do not allow any site to request access to serial ports via the Serial API\n3 - Allow sites to ask the user to grant access to a serial port\nSetting the policy to 3 lets websites ask for access to serial ports. Setting the policy to 2 denies access to serial ports.\n\nLeaving it unset lets websites ask for access, but users can change this setting.";

DefaultWebBluetoothGuardSetting.pfm\_title = "Control use of the Web Bluetooth API";

DefaultWebBluetoothGuardSetting.pfm\_description = "2 - Do not allow any site to request access to Bluetooth devices via the Web Bluetooth API\n3 - Allow sites to ask the user to grant access to a nearby Bluetooth device\nSetting the policy to 3 lets websites ask for access to nearby Bluetooth devices. Setting the policy to 2 denies access to nearby Bluetooth devices.\n\nLeaving the policy unset lets sites ask for access, but users can change this setting.";

DefaultWebHidGuardSetting.pfm\_title = "Control use of the WebHID API";

DefaultWebHidGuardSetting.pfm\_description = "2 - Do not allow any site to request access to HID devices via the WebHID API\n3 - Allow sites to ask the user to grant access to a HID device\nSetting the policy to 3 lets websites ask for access to HID devices. Setting the policy to 2 denies access to HID devices.\n\nLeaving it unset lets websites ask for access, but users can change this setting.\n\nThis policy can be overridden for specific url patterns using the WebHidAskForUrls and WebHidBlockedForUrls policies.";

DefaultWebUsbGuardSetting.pfm\_title = "Control use of the WebUSB API";

DefaultWebUsbGuardSetting.pfm\_description = "2 - Do not allow any site to request access to USB devices via the WebUSB API\n3 - Allow sites to ask the user to grant access to a connected USB device\nSetting the policy to 3 lets websites ask for access to connected USB devices. Setting the policy to 2 denies access to connected USB devices.\n\nLeaving it unset lets websites ask for access, but users can change this setting.";

DefaultWindowPlacementSetting.pfm\_title = "Default Window Placement permission setting";

DefaultWindowPlacementSetting.pfm\_description = "2 - Denies the Window Placement permission on all sites by default\n3 - Ask every time a site wants obtain the Window Placement permission\nSetting the policy to BlockWindowPlacement (value 2) automatically denies the window placement permission to sites by default. This will limit the ability of sites to see information about the device's screens and use that information to open and place windows or request fullscreen on specific screens.\n\nSetting the policy to AskWindowPlacement (value 3) will prompt the user when the window placement permission is requested by default. If users allow the permission, it will extend the ability of sites to see information about the device's screens and use that information to open and place windows or request fullscreen on specific screens.\n\nLeaving the policy unset means the AskWindowPlacement policy applies, but users can change this setting.";

DesktopSharingHubEnabled.pfm\_title = "Enable desktop sharing in the omnibox and 3-dot menu";

DesktopSharingHubEnabled.pfm\_description = "Setting the policy to True or leaving it unset lets users share or save the current webpage using actions provided by the desktop sharing hub. The sharing hub is accessed through either an omnibox icon or the 3-dot menu.\n\nSetting the policy to False removes the sharing icon from the omnibox and the entry from the 3-dot menu.";

DeveloperToolsAvailability.pfm\_title = "Control where Developer Tools can be used";

DeveloperToolsAvailability.pfm\_description = "0 - Disallow usage of the Developer Tools on extensions installed by enterprise policy, allow usage of the Developer Tools in other contexts\n1 - Allow usage of the Developer Tools\n2 - Disallow usage of the Developer Tools\nSetting the policy to 0 (the default) means you can access the developer tools and the JavaScript console, but not in the context of extensions installed by enterprise policy. Setting the policy to 1 means you can access the developer tools and the JavaScript console in all contexts, including that of extensions installed by enterprise policy. Setting the policy to 2 means you can't acess developer tools, and you can't inspect website elements.\n\nThis setting also turns off keyboard shortcuts and menu or context menu entries to open developer tools or the JavaScript console.\n\nAs of Google Chrome version 99, this setting also controls entry points for the 'View page source' feature. If you set this policy to 'DeveloperToolsDisallowed' (value 2), users cannot access source viewing via keyboard shortcut or the context menu. To fully block source viewing, you must also add 'view-source:\*' to the URLBlocklist policy.";

Disable3DAPIs.pfm\_title = "Disable support for 3D graphics APIs";

Disable3DAPIs.pfm\_description = "Setting the policy to True (or setting HardwareAccelerationModeEnabled to False) prevents webpages from accessing the WebGL API, and plugins can't use the Pepper 3D API.\n\nSetting the policy to False or leaving it unset lets webpages use the WebGL API and plugins use the Pepper 3D API, but the browser's default settings might still require command line arguments to use these APIs.";

DisableAuthNegotiateCnameLookup.pfm\_title = "Disable CNAME lookup when negotiating Kerberos authentication";

DisableAuthNegotiateCnameLookup.pfm\_description = "Setting the policy to Enabled skips CNAME lookup. The server name is used as entered when generating the Kerberos SPN.\n\nSetting the policy to Disabled or leaving it unset means CNAME lookup determines the canonical name of the server when generating the Kerberos SPN.";

DisablePrintPreview.pfm\_title = "Disable Print Preview";

DisablePrintPreview.pfm\_description = "Setting the policy to Enabled has Google Chrome open the system print dialog instead of the built-in print preview when users request a printout.\n\nSetting the policy to Disabled or leaving it unset has print commands trigger the print preview screen.";

DisableSafeBrowsingProceedAnyway.pfm\_title = "Disable proceeding from the Safe Browsing warning page";

DisableSafeBrowsingProceedAnyway.pfm\_description = "Setting the policy to Enabled prevents users from proceeding past the warning page the Safe Browsing service shows to the malicious site. This policy only prevents users from proceeding on Safe Browsing warnings such as malware and phishing, not for SSL certificate-related issues such as invalid or expired certificates.\n\nSetting the policy to Disabled or leaving it unset means users can choose to proceed to the flagged site after the warning appears.\n\nSee more about Safe Browsing ( https://developers.google.com/safe-browsing ).";

DisableScreenshots.pfm\_title = "Disable taking screenshots";

DisableScreenshots.pfm\_description = "Setting the policy to True disallows screenshots taken with keyboard shortcuts or extension APIs. Setting the policy to False allows screenshots.";

DiskCacheDir.pfm\_title = "Set disk cache directory";

DiskCacheDir.pfm\_description = "Setting the policy has Google Chrome use the directory you provide for storing cached files on the disk—whether or not users specify the --disk-cache-dir flag.\n\nIf not set, Google Chrome uses the default cache directory, but users can change that setting with the --disk-cache-dir command line flag.\n\nGoogle Chrome manages the contents of a volume's root directory. So to avoid data loss or other errors, do not set this policy to the root directory or any directory used for other purposes. See the variables you can use ( https://www.chromium.org/administrators/policy-list-3/user-data-directory-variables ).";

DiskCacheSize.pfm\_title = "Set disk cache size";

DiskCacheSize.pfm\_description = "Setting the policy to None has Google Chrome use the default cache size for storing cached files on the disk. Users can't change it.\n\nIf you set the policy, Google Chrome uses the cache size you provide—whether or not users specify the --disk-cache-size flag. (Values below a few megabytes are rounded up.)\n\nIf not set, Google Chrome uses the default size. Users can change that setting using the --disk-cache-size flag.\n\nNote: The value specified in this policy is used as a hint to various cache subsystems in the browser. Therefore the actual total disk consumption of all caches will be higher but within the same order of magnitude as the value specified.";

DisplayCapturePermissionsPolicyEnabled.pfm\_title = "Specifies whether the display-capture permissions-policy is checked or skipped.";

DisplayCapturePermissionsPolicyEnabled.pfm\_description = "\nThe display-capture permissions-policy gates access to getDisplayMedia(), as per this spec: https://www.w3.org/TR/screen-capture/#feature-policy-integration. However, if this policy is Disabled, this requirement is not enforced, and getDisplayMedia() is allowed from contexts that would otherwise be forbidden. This Enterprise policy is temporary; it's intended to be removed after Google Chrome version 100. It is intended to unblock Enterprise users whose application is non-spec compliant, but needs time to be fixed.\n\nWhen enabled or not set, sites can only call getDisplayMedia() from contexts which are allowlisted by the display-capture permissions-policy.\n\nWhen disabled, sites can call getDisplayMedia() even from contexts which are not allowlisted by the display-capture permissions policy. Note that other restrictions may still apply.";

DnsOverHttpsMode.pfm\_title = "Controls the mode of DNS-over-HTTPS";

DnsOverHttpsMode.pfm\_description = "off - Disable DNS-over-HTTPS\nautomatic - Enable DNS-over-HTTPS with insecure fallback\nsecure - Enable DNS-over-HTTPS without insecure fallback\nControls the mode of the DNS-over-HTTPS resolver. Please note that this policy will only set the default mode for each query. The mode may be overridden for special types of queries such as requests to resolve a DNS-over-HTTPS server hostname.\n\nThe \"off\" mode will disable DNS-over-HTTPS.\n\nThe \"automatic\" mode will send DNS-over-HTTPS queries first if a DNS-over-HTTPS server is available and may fallback to sending insecure queries on error.\n\nThe \"secure\" mode will only send DNS-over-HTTPS queries and will fail to resolve on error.\n\nOn Android Pie and above, if DNS-over-TLS is active, Google Chrome will not send insecure DNS requests.\n\nIf this policy is unset the browser may send DNS-over-HTTPS requests to a resolver associated with the user's configured system resolver.";

DnsOverHttpsTemplates.pfm\_title = "Specify URI template of desired DNS-over-HTTPS resolver";

DnsOverHttpsTemplates.pfm\_description = "The URI template of the desired DNS-over-HTTPS resolver. To specify multiple DNS-over-HTTPS resolvers, separate the corresponding URI templates with spaces.\n\nIf the DnsOverHttpsMode is set to \"secure\" then this policy must be set and not empty.\n\nIf the DnsOverHttpsMode is set to \"automatic\" and this policy is set then the URI templates specified will be used; if this policy is unset then hardcoded mappings will be used to attempt to upgrade the user's current DNS resolver to a DoH resolver operated by the same provider.\n\nIf the URI template contains a dns variable, requests to the resolver will use GET; otherwise requests will use POST.\n\nIncorrectly formatted templates will be ignored.";

DownloadBubbleEnabled.pfm\_title = "Enable download bubble UI";

DownloadBubbleEnabled.pfm\_description = "Setting the policy to Enabled or leaving it unset shows the new download bubble UI in Google Chrome.\n\nSetting the policy to Disabled means Google Chrome keeps showing the old download shelf UI.";

DownloadDirectory.pfm\_title = "Set download directory";

DownloadDirectory.pfm\_description = "Setting the policy sets up the directory Chrome uses for downloading files. It uses the provided directory, whether or not users specify one or turned on the flag to be prompted for download location every time.\n\nThis policy overrides the DefaultDownloadDirectory policy.\n\nLeaving the policy unset means Chrome uses the default download directory, and users can change it.\n\nNote: See a list of variables you can use ( https://www.chromium.org/administrators/policy-list-3/user-data-directory-variables ).";

DownloadRestrictions.pfm\_title = "Download restrictions";

DownloadRestrictions.pfm\_description = "0 - No special restrictions. Default.\n1 - Block malicious downloads and dangerous file types.\n2 - Block malicious downloads, uncommon or unwanted downloads and dangerous file types.\n3 - Block all downloads.\n4 - Block malicious downloads. Recommended.\nSetting the policy means users can't bypass download security decisions.\n\nThere are many types of download warnings within Chrome, which roughly break down into these categories (learn more about Safe Browsing verdicts https://support.google.com/chrome/?p=ib\_download\_blocked):\n\n\* Malicious, as flagged by the Safe Browsing server\n\* Uncommon or unwanted, as flagged by the Safe Browsing server\n\* A dangerous file type (e.g. all SWF downloads and many EXE downloads)\n\nSetting the policy blocks different subsets of these, depending on it's value:\n\n0: No special restrictions. Default.\n\n1: Blocks malicious files flagged by the Safe Browsing server AND Blocks all dangerous file types. Only recommended for OUs/browsers/users that have a high tolerance for False Positives.\n\n2: Blocks malicious files flagged by the Safe Browsing server AND Blocks uncommon or unwanted files flagged by the Safe Browsing server AND Blocks all dangerous file types. Only recommended for OUs/browsers/users that have a high tolerance for False Positives.\n\n3: Blocks all downloads. Not recommended, except for special use cases.\n\n4: Blocks malicious files flagged by the Safe Browsing server, does not block dangerous file types. Recommended.\n\nNote: These restrictions apply to downloads triggered from webpage content, as well as the Download link... menu option. They don't apply to the download of the currently displayed page or to saving as PDF from the printing options. Read more about Safe Browsing ( https://developers.google.com/safe-browsing ).";

EditBookmarksEnabled.pfm\_title = "Enable or disable bookmark editing";

EditBookmarksEnabled.pfm\_description = "Setting the policy to True or leaving it unset lets users add, remove, or modify bookmarks.\n\nSetting the policy to False means users can't add, remove, or modify bookmarks. They can still use existing bookmarks.";

EnableAuthNegotiatePort.pfm\_title = "Include non-standard port in Kerberos SPN";

EnableAuthNegotiatePort.pfm\_description = "Setting the policy to Enabled and entering a nonstandard port (in other words, a port other than 80 or 443) includes it in the generated Kerberos SPN.\n\nSetting the policy to Disabled or leaving it unset means the generated Kerberos SPN won't include a port.";

EnableExperimentalPolicies.pfm\_title = "Enables experimental policies";

EnableExperimentalPolicies.pfm\_description = "Allows Google Chrome to load experimental policies.\n\nWARNING: Experimental policies are unsupported and subject to change or be removed without notice in future version of the browser!\n\nAn experimental policy may not be finished or still have known or unknown defects. It may be changed or even removed without any notification. By enabling experimental policies, you could lose browser data or compromise your security or privacy.\n\nIf a policy is not in the list and it's not officially released, its value will be ignored on Beta and Stable channel.\n\nIf a policy is in the list and it's not officially released, its value will be applied.\n\nThis policy has no effect on already released policies.";

EnableMediaRouter.pfm\_title = "Enable Google Cast";

EnableMediaRouter.pfm\_description = "Setting the policy to Enabled or leaving it unset turns on Google Cast, which users can launch from the app menu, page context menus, media controls on Cast-enabled websites, and (if shown) the Cast toolbar icon.\n\nSetting the policy to Disabled turns off Google Cast.";

EnableOnlineRevocationChecks.pfm\_title = "Enable online OCSP/CRL checks";

EnableOnlineRevocationChecks.pfm\_description = "Setting the policy to True means online OCSP/CRL checks are performed.\n\nSetting the policy to False or leaving it unset means Google Chrome won't perform online revocation checks in Google Chrome 19 and later.\n\nNote: OCSP/CRL checks provide no effective security benefit.";

EnterpriseHardwarePlatformAPIEnabled.pfm\_title = "Enables managed extensions to use the Enterprise Hardware Platform API";

EnterpriseHardwarePlatformAPIEnabled.pfm\_description = "Setting the policy to True lets extensions installed by enterprise policy use the Enterprise Hardware Platform API.\n\nSetting the policy to False or leaving it unset prevents extensions from using this API.\n\nNote: This policy also applies to component extensions, such as the Hangout Services extension.";

ExemptDomainFileTypePairsFromFileTypeDownloadWarnings.pfm\_title = "Disable download file type extension-based warnings for specified file types on domains";

ExemptDomainFileTypePairsFromFileTypeDownloadWarnings.pfm\_description = "You can enable this policy to create a dictionary of file type extensions with a corresponding list of domains that will be exempted from file type extension-based download warnings. This lets enterprise administrators block file type extension-based download warnings for files that are associated with a listed domain. For example, if the \"jnlp\" extension is associated with \"website1.com\", users would not see a warning when downloading \"jnlp\" files from \"website1.com\", but see a download warning when downloading \"jnlp\" files from \"website2.com\".\n\nFiles with file type extensions specified for domains identified by this policy will still be subject to non-file type extension-based security warnings such as mixed-content download warnings and Safe Browsing warnings.\n\nIf you disable this policy or don't configure it, file types that trigger extension-based download warnings will show warnings to the user.\n\nIf you enable this policy:\n\n\* The URL pattern should be formatted according to https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns.\n\* The file type extension entered must be in lower-cased ASCII. The leading separator should not be included when listing the file type extension, so list \"jnlp\" should be used instead of \".jnlp\".\n\nExample:\n\nThe following example value would prevent file type extension-based download warnings on swf, exe, and jnlp extensions for \*.example.com domains. It will show the user a file type extension-based download warning on any other domain for exe and jnlp files, but not for swf files.\n\n[\n { \"file\_extension\": \"jnlp\", \"domains\": [\"example.com\"] },\n { \"file\_extension\": \"exe\", \"domains\": [\"example.com\"] },\n { \"file\_extension\": \"swf\", \"domains\": [\"\*\"] }\n]\n\nNote that while the preceding example shows the suppression of file type extension-based download warnings for \"swf\" files for all domains, applying suppression of such warnings for all domains for any dangerous file type extension is not recommended due to security concerns. It is shown in the example merely to demonstrate the ability to do so.\n\nIf this policy is enabled alongside DownloadRestrictions and DownloadRestrictions is set to block dangerous file types, download blocks determined by DownloadRestrictions take precedence. For example, if this policy is set to enable \"exe\" extension downloads from \"website1.com\", and DownloadRestrictions is set to block malicious downloads and dangerous file types, then \"exe\" extension downloads will still be blocked in all domains. If DownloadRestrictions is not set to block dangerous file types, then file types specified in this policy will be exempted from file-type extension-based download warnings in the specified domains. Read more about DownloadRestrictions (https://chromeenterprise.google/policies/?policy=DownloadRestrictions).\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=ExemptDomainFileTypePairsFromFileTypeDownloadWarnings for more information about schema and formatting.";

ExplicitlyAllowedNetworkPorts.pfm\_title = "Explicitly allowed network ports";

ExplicitlyAllowedNetworkPorts.pfm\_description = "554 - port 554 (can be unblocked until 2021/10/15)\n10080 - port 10080 (can be unblocked until 2022/04/01)\n6566 - port 6566 (can be unblocked until 2021/10/15)\n989 - port 989 (can be unblocked until 2022/02/01)\n990 - port 990 (can be unblocked until 2022/02/01)\nThere is a list of restricted ports built into Google Chrome. Connections to these ports will fail. This setting permits bypassing that list. The value is a comma-separated list of zero or more ports that outgoing connections will be permitted on.\n\nPorts are restricted to prevent Google Chrome being used as a vector to exploit various network vulnerabilities. Setting this policy may expose your network to attacks. This policy is intended as a temporary workaround for errors with code \"ERR\_UNSAFE\_PORT\" while migrating a service running on a blocked port to a standard port (ie. port 80 or 443).\n\nMalicious websites can easily detect that this policy is set, and for what ports, and use that information to target attacks.\n\nEach port here is labelled with a date that it can be unblocked until. After that date the port will be restricted regardless of this setting.\n\nLeaving the value empty or unset means that all restricted ports will be blocked. If there is a mixture of valid and invalid values, the valid ones will be applied.\n\nThis policy overrides the \"--explicitly-allowed-ports\" command-line option.";

ExtensionAllowedTypes.pfm\_title = "Types of extensions/apps that are allowed to be installed";

ExtensionAllowedTypes.pfm\_description = "extension - Extension\ntheme - Theme\nuser\_script - User script\nhosted\_app - Hosted app\nlegacy\_packaged\_app - Legacy packaged app\nplatform\_app - Platform app\nSetting the policy controls which apps and extensions may be installed in Google Chrome, which hosts they can interact with, and limits runtime access.\n\nLeaving the policy unset results in no restrictions on the acceptable extension and app types.\n\nExtensions and apps which have a type that's not on the list won't be installed. Each value should be one of these strings:\n\n\* \"extension\"\n\n\* \"theme\"\n\n\* \"user\_script\"\n\n\* \"hosted\_app\"\n\n\* \"legacy\_packaged\_app\"\n\n\* \"platform\_app\"\n\nSee the Google Chrome extensions documentation for more information on these types.\n\nVersions earlier than 75 that use multiple comma separated extension IDs aren't supported and are skipped. The rest of the policy applies.\n\nNote: This policy also affects extensions and apps to be force-installed using ExtensionInstallForcelist.";

ExtensionInstallAllowlist.pfm\_title = "Extension IDs to exempt from the blocklist";

ExtensionInstallAllowlist.pfm\_description = "Setting the policy specifies which extensions are not subject to the blocklist.\n\nA blocklist value of \* means all extensions are blocked and users can only install extensions listed in the allow list.\n\nBy default, all extensions are allowed. But, if you prohibited extensions by policy, use the list of allowed extensions to change that policy.";

ExtensionInstallBlocklist.pfm\_title = "Extension IDs the user should be prevented from installing (or \* for all)";

ExtensionInstallBlocklist.pfm\_description = "Allows you to specify which extensions the users can NOT install. Extensions already installed will be disabled if blocked, without a way for the user to enable them. Once an extension disabled due to the blocklist is removed from it, it will automatically get re-enabled.\n\nA blocklist value of '\*' means all extensions are blocked unless they are explicitly listed in the allowlist.\n\nIf this policy is left not set the user can install any extension in Google Chrome.";

ExtensionInstallForcelist.pfm\_title = "Extension/App IDs and update URLs to be silently installed";

ExtensionInstallForcelist.pfm\_description = "Setting the policy specifies a list of apps and extensions that install silently, without user interaction, and which users can't uninstall or turn off. Permissions are granted implicitly, including for the enterprise.deviceAttributes and enterprise.platformKeys extension APIs. (These 2 APIs aren't available to apps and extensions that aren't force-installed.)\n\nLeaving the policy unset means no apps or extensions are autoinstalled, and users can uninstall any app or extension in Google Chrome.\n\nThis policy superseeds ExtensionInstallBlocklist policy. If a previously force-installed app or extension is removed from this list, Google Chrome automatically uninstalls it.\n\nOn Microsoft® Windows® instances, apps and extensions from outside the Chrome Web Store can only be forced installed if the instance is joined to a Microsoft® Active Directory® domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management.\n\nOn macOS instances, apps and extensions from outside the Chrome Web Store can only be force installed if the instance is managed via MDM, or joined to a domain via MCX.\n\nThe source code of any extension may be altered by users through developer tools, potentially rendering the extension dysfunctional. If this is a concern, set the DeveloperToolsDisabled policy.\n\nEach list item of the policy is a string that contains an extension ID and, optionally, an \"update\" URL separated by a semicolon (;). The extension ID is the 32-letter string found, for example, on chrome://extensions when in Developer mode. If specified, the \"update\" URL should point to an Update Manifest XML document ( https://developer.chrome.com/extensions/autoupdate ). By default, the Chrome Web Store's update URL is used. The \"update\" URL set in this policy is only used for the initial installation; subsequent updates of the extension use the update URL in the extension's manifest.\n\n Note: This policy doesn't apply to Incognito mode. Read about hosting extensions ( https://developer.chrome.com/extensions/hosting ).";

ExtensionInstallSources.pfm\_title = "URL patterns to allow extension, app, and user script installs from";

ExtensionInstallSources.pfm\_description = "Setting the policy specifies which URLs may install extensions, apps, and themes. Before Google Chrome 21, users could click on a link to a \*.crx file, and Google Chrome would offer to install the file after a few warnings. Afterwards, such files must be downloaded and dragged to the Google Chrome settings page. This setting allows specific URLs to have the old, easier installation flow.\n\nEach item in this list is an extension-style match pattern (see https://developer.chrome.com/extensions/match\_patterns). Users can easily install items from any URL that matches an item in this list. Both the location of the \*.crx file and the page where the download is started from (the referrer) must be allowed by these patterns.\n\nExtensionInstallBlocklist takes precedence over this policy. That is, an extension on the blocklist won't be installed, even if it happens from a site on this list.";

ExtensionSettings.pfm\_title = "Extension management settings";

ExtensionSettings.pfm\_description = "Setting the policy controls extension management settings for Google Chrome, including any controlled by existing extension-related policies. The policy supersedes any legacy policies that might be set.\n\nThis policy maps an extension ID or an update URL to its specific setting only. A default configuration can be set for the special ID \"\*\", which applies to all extensions without a custom configuration in this policy. With an update URL, configuration applies to extensions with the exact update URL stated in the extension manifest ( http://support.google.com/chrome/a?p=Configure\_ExtensionSettings\_policy ). If the 'override\_update\_url' flag is set to true, the extension is installed and updated using the \"update\" URL specified in the ExtensionInstallForcelist policy or in 'update\_url' field in this policy. The flag 'override\_update\_url' is ignored if the 'update\_url' is a Chrome Web Store url.\n\nNote: For Microsoft® Windows® instances not joined to a Microsoft® Active Directory® domain and macOS instances not managed via MDM or joined to a domain via MCX, forced installation is limited to apps and extensions listed in the Chrome Web Store.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=ExtensionSettings for more information about schema and formatting.";

ExternalProtocolDialogShowAlwaysOpenCheckbox.pfm\_title = "Show an \"Always open\" checkbox in external protocol dialog.";

ExternalProtocolDialogShowAlwaysOpenCheckbox.pfm\_description = "This policy controls whether or not the \"Always open\" checkbox is shown on external protocol launch confirmation prompts.\n\n If this policy is set to True or not set, when an external protocol confirmation is shown, the user can select \"Always allow\" to skip all future confirmation prompts for the protocol on this site.\n\n If this policy is set to False, the \"Always allow\" checkbox is not displayed and the user will be prompted each time an external protocol is invoked.";

FetchKeepaliveDurationSecondsOnShutdown.pfm\_title = "Fetch keepalive duration on Shutdown";

FetchKeepaliveDurationSecondsOnShutdown.pfm\_description = "Controls the duration (in seconds) allowed for keepalive requests on browser shutdown.\n\nWhen specified, browser shutdown can be blocked up to the specified seconds,\nto process keepalive (https://fetch.spec.whatwg.org/#request-keepalive-flag) requests.\n\nThe default value (0) means this feature is disabled.";

FileSystemReadAskForUrls.pfm\_title = "Allow read access via the File System API on these sites";

FileSystemReadAskForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can ask users to grant them read access to files or directories in the host operating system's file system via the File System API.\n\nLeaving the policy unset means DefaultFileSystemReadGuardSetting applies for all sites, if it's set. If not, users' personal settings apply.\n\nURL patterns must not conflict with FileSystemReadBlockedForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

FileSystemReadBlockedForUrls.pfm\_title = "Block read access via the File System API on these sites";

FileSystemReadBlockedForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can't ask users to grant them read access to files or directories in the host operating system's file system via the File System API.\n\nLeaving the policy unset means DefaultFileSystemReadGuardSetting applies for all sites, if it's set. If not, users' personal settings apply.\n\nURL patterns can't conflict with FileSystemReadAskForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

FileSystemWriteAskForUrls.pfm\_title = "Allow write access to files and directories on these sites";

FileSystemWriteAskForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can ask users to grant them write access to files or directories in the host operating system's file system.\n\nLeaving the policy unset means DefaultFileSystemWriteGuardSetting applies for all sites, if it's set. If not, users' personal settings apply.\n\nURL patterns must not conflict with FileSystemWriteBlockedForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

FileSystemWriteBlockedForUrls.pfm\_title = "Block write access to files and directories on these sites";

FileSystemWriteBlockedForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can't ask users to grant them write access to files or directories in the host operating system's file system.\n\nLeaving the policy unset means DefaultFileSystemWriteGuardSetting applies for all sites, if it's set. If not, users' personal settings apply.\n\nURL patterns can't conflict with FileSystemWriteAskForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

ForceEphemeralProfiles.pfm\_title = "Ephemeral profile";

ForceEphemeralProfiles.pfm\_description = "If set to enabled this policy forces the profile to be switched to ephemeral mode. If this policy is specified as an OS policy (e.g. GPO on Windows) it will apply to every profile on the system; if the policy is set as a Cloud policy it will apply only to a profile signed in with a managed account.\n\nIn this mode the profile data is persisted on disk only for the length of the user session. Features like browser history, extensions and their data, web data like cookies and web databases are not preserved after the browser is closed. However this does not prevent the user from downloading any data to disk manually, save pages or print them.\n\nIf the user has enabled sync all this data is preserved in their sync profile just like with regular profiles. Incognito mode is also available if not explicitly disabled by policy.\n\nIf the policy is set to disabled or left not set signing in leads to regular profiles.";

ForceGoogleSafeSearch.pfm\_title = "Force Google SafeSearch";

ForceGoogleSafeSearch.pfm\_description = "Setting the policy to Enabled means SafeSearch in Google Search is always active, and users can't change this setting.\n\nSetting the policy to Disabled or leaving it unset means SafeSearch in Google Search is not enforced.";

ForceMajorVersionToMinorPositionInUserAgent.pfm\_title = "Freeze User-Agent string major version at 99";

ForceMajorVersionToMinorPositionInUserAgent.pfm\_description = "0 - Default to browser settings for User-Agent string version.\n1 - The User-Agent string will not freeze the major version.\n2 - The User-Agent string will freeze the major version as 99 and include the browser's major version in the minor position.\nThis policy controls whether the User-Agent string major\nversion should be frozen at 99.\n\nThe User-Agent request header lets websites identify the application,\noperating system, vendor, and/or version of the requesting user agent.\nSome websites make assumptions about how this header is formatted and may\nencounter issues with version strings that include three digits in the\nmajor position (e.g. 100.0.0.0).\n\nSetting the policy to 'Default' or leaving it unset will default to\nbrowser settings for the User-Agent string major version.\nIf set to 'ForceDisabled', the User-Agent string will not freeze the\nmajor version.\nIf set to 'ForceEnabled', the User-Agent string will always report the\nmajor version as 99 and include the browser's major version in the minor\nposition. For example, browser version 101.0.0.0 would send a User-Agent\nrequest header that reports version 99.101.0.0.\n\nThis policy is temporary and will be deprecated in the future. Note that\nif this policy and\nUser-Agent Reduction are\nboth enabled, the User-Agent version string will always be 99.0.0.0.";

ForceYouTubeRestrict.pfm\_title = "Force minimum YouTube Restricted Mode";

ForceYouTubeRestrict.pfm\_description = "0 - Do not enforce Restricted Mode on YouTube\n1 - Enforce at least Moderate Restricted Mode on YouTube\n2 - Enforce Strict Restricted Mode for YouTube\nSetting the policy enforces a minimum Restricted mode on YouTube and prevents users from picking a less restricted mode. If you set it to:\n\n\* Strict, Strict Restricted mode on YouTube is always active.\n\n\* Moderate, the user may only pick Moderate Restricted mode and Strict Restricted mode on YouTube, but can't turn off Restricted mode.\n\n\* Off or if no value is set, Restricted mode on YouTube isn't enforced by Chrome. External policies such as YouTube policies might still enforce Restricted mode.";

ForcedLanguages.pfm\_title = "Configure the content and order of preferred languages";

ForcedLanguages.pfm\_description = "This policy allows admins to configure the order of the preferred languages in Google Chrome's settings.\n\nThe order of the list will appear in the same order under the \"Order languages based on your preference\" section in chrome://settings/languages. Users won't be able to remove or reorder languages set by the policy, but will be able to add languages underneath those set by the policy. Users will also have full control over the browser's UI language and translation/spell check settings, unless enforced by other policies.\n\nLeaving the policy unset lets users manipulate the entire list of preferred languages.";

GloballyScopeHTTPAuthCacheEnabled.pfm\_title = "Enable globally scoped HTTP auth cache";

GloballyScopeHTTPAuthCacheEnabled.pfm\_description = "This policy configures a single global per profile cache with HTTP server authentication credentials.\n\nIf this policy is unset or disabled, the browser will use the default behavior of cross-site auth, which as of version 80, will be to scope HTTP server authentication credentials by top-level site, so if two sites use resources from the same authenticating domain, credentials will need to be provided independently in the context of both sites. Cached proxy credentials will be reused across sites.\n\nIf the policy is enabled, HTTP auth credentials entered in the context of one site will automatically be used in the context of another.\n\nEnabling this policy leaves sites open to some types of cross-site attacks, and allows users to be tracked across sites even without cookies by adding entries to the HTTP auth cache using credentials embedded in URLs.\n\nThis policy is intended to give enterprises depending on the legacy behavior a chance to update their login procedures, and will be removed in the future.";

HSTSPolicyBypassList.pfm\_title = "List of names that will bypass the HSTS policy check";

HSTSPolicyBypassList.pfm\_description = "Setting the policy specifies a list of hostnames that bypass preloaded HSTS upgrades from http to https.\n\nOnly single-label hostnames are allowed in this policy, and this policy only applies to \"static\" HSTS-preloaded entries (for instance, \"app\", \"new\", \"search\", \"play\"). This policy does not prevent HSTS upgrades for servers that have \"dynamically\" requested HSTS upgrades using a Strict-Transport-Security response header.\n\nSupplied hostnames must be canonicalized: Any IDNs must be converted to their A-label format, and all ASCII letters must be lowercase. This policy only applies to the specific single-label hostnames specified, not to subdomains of those names.";

HardwareAccelerationModeEnabled.pfm\_title = "Use hardware acceleration when available";

HardwareAccelerationModeEnabled.pfm\_description = "Setting the policy to Enabled or leaving it unset turns on hardware acceleration, if available.\n\nSetting the policy to Disabled turns off hardware acceleration.";

HeadlessMode.pfm\_title = "Control use of the Headless Mode";

HeadlessMode.pfm\_description = "1 - Allow use of the Headless Mode\n2 - Do not allow use of the Headless Mode\nSetting this policy to Enabled or leaving the policy unset allows use of the headless mode. Setting this policy to Disabled denies use of the headless mode.";

HideWebStoreIcon.pfm\_title = "Hide the web store from the New Tab Page and app launcher";

HideWebStoreIcon.pfm\_description = "Hide the Chrome Web Store app and footer link from the New Tab Page and Google Chrome OS app launcher.\n\nWhen this policy is set to true, the icons are hidden.\n\nWhen this policy is set to false or is not configured, the icons are visible.";

HistoryClustersVisible.pfm\_title = "Show Journeys on the Chrome history page";

HistoryClustersVisible.pfm\_description = "This policy controls the visibility of Journeys on the Chrome history page.\n\nIf the policy is set to Enabled, Journeys will be visible at chrome://history/journeys.\n\nIf the policy is set to Disabled, Journeys will not be visible at chrome://history/journeys.\n\nIf the policy is left unset, Journeys will be visible at chrome://history/journeys by default and users can change the visibility of Journeys.\n\nPlease note, if ComponentUpdatesEnabled policy is set to Disabled, but HistoryClustersVisible is set to Enabled or unset, Journeys will still be available at chrome://history/journeys, but may be absent from the omnibox, and less relevant to the user.\n";

HomepageIsNewTabPage.pfm\_title = "Use New Tab Page as homepage";

HomepageIsNewTabPage.pfm\_description = "Setting the policy to Enabled makes the New Tab page the user's homepage, ignoring any homepage URL location. Setting the policy to Disabled means that their homepage is never the New Tab page, unless the user's homepage URL is set to chrome://newtab.\n\nIf you set the policy, users can't change their homepage type in Google Chrome. If not set, the user decides whether or not the New Tab page is their homepage.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

HomepageLocation.pfm\_title = "Home page URL";

HomepageLocation.pfm\_description = "Setting the policy sets the default homepage URL in Google Chrome. You open the homepage using the Home button. On desktop, the RestoreOnStartup policies control the pages that open on startup.\n\nIf the homepage is set to the New Tab Page, by the user or HomepageIsNewTabPage, this policy has no effect.\n\n The URL needs a standard scheme, such as http://example.com or https://example.com. When this policy is set, users can't change their homepage URL in Google Chrome.\n\nLeaving both HomepageLocation and HomepageIsNewTabPage unset lets users choose their homepage.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

HttpsOnlyMode.pfm\_title = "Allow HTTPS-Only Mode to be enabled";

HttpsOnlyMode.pfm\_description = "allowed - Allow users to enable HTTPS-Only Mode\ndisallowed - Do not allow users to enable HTTPS-Only Mode\nforce\_enabled - Force enable HTTPS-Only Mode (not supported yet)\nThis policy controls whether users can enable HTTPS-Only Mode in Settings. HTTPS-Only Mode upgrades all navigations to HTTPS.\nIf this setting is not set or set to allowed, users will be allowed to enable HTTPS-Only Mode.\nIf this setting is set to disallowed, users will not be allowed to enable HTTPS-Only Mode.\nForce enabling HTTPS-Only Mode is not currently supported.";

ImagesAllowedForUrls.pfm\_title = "Allow images on these sites";

ImagesAllowedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify sites that may display images.\n\nLeaving the policy unset means DefaultImagesSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.\n\nNote that previously this policy was erroneously enabled on Android, but this functionality has never been fully supported on Android.";

ImagesBlockedForUrls.pfm\_title = "Block images on these sites";

ImagesBlockedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify sites that can't display images.\n\nLeaving the policy unset means DefaultImagesSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\n For detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.\n\n Note that previously this policy was erroneously enabled on Android, but this functionality has never been fully supported on Android.";

ImportAutofillFormData.pfm\_title = "Import autofill form data from default browser on first run";

ImportAutofillFormData.pfm\_description = "Setting the policy to Enabled imports autofill form data from the previous default browser on first run. Setting the policy to Disabled or leaving it unset means no autofill form data is imported on first run.\n\nUsers can trigger an import dialog and the autofill form data checkbox will be checked or unchecked to match this policy's value.";

ImportBookmarks.pfm\_title = "Import bookmarks from default browser on first run";

ImportBookmarks.pfm\_description = "Setting the policy to Enabled imports bookmarks from the previous default browser on first run. Setting the policy to Disabled or leaving it unset means no bookmarks are imported on first run.\n\nUsers can trigger an import dialog and the bookmarks checkbox will be checked or unchecked to match this policy's value.";

ImportHistory.pfm\_title = "Import browsing history from default browser on first run";

ImportHistory.pfm\_description = "Setting the policy to Enabled imports browsing history from the previous default browser on first run. Setting the policy to Disabled or leaving it unset means no browsing history is imported on first run.\n\nUsers can trigger an import dialog and the browsing history checkbox will be checked or unchecked to match this policy's value.";

ImportHomepage.pfm\_title = "Import of homepage from default browser on first run";

ImportHomepage.pfm\_description = "Setting the policy to Enabled imports the homepage from the previous default browser on first run. Setting the policy to Disabled or leaving it unset means the homepage isn't imported on first run.\n\nUsers can trigger an import dialog and the homepage checkbox will be checked or unchecked to match this policy's value.";

ImportSavedPasswords.pfm\_title = "Import saved passwords from default browser on first run";

ImportSavedPasswords.pfm\_description = "Setting the policy to Enabled imports saved passwords from the previous default browser on first run. Setting the policy to Disabled or leaving it unset means no saved passwords are imported on first run.\n\nUsers can trigger an import dialog and the saved passwords checkbox will be checked or unchecked to match this policy's value.";

ImportSearchEngine.pfm\_title = "Import search engines from default browser on first run";

ImportSearchEngine.pfm\_description = "Setting the policy to Enabled imports the default search engine from the previous default browser on first run. Setting the policy to Disabled or leaving it unset means the default search engine isn't imported on first run.\n\nUsers can trigger an import dialog and the default search engine checkbox will be checked or unchecked to match this policy's value.";

IncognitoModeAvailability.pfm\_title = "Incognito mode availability";

IncognitoModeAvailability.pfm\_description = "0 - Incognito mode available\n1 - Incognito mode disabled\n2 - Incognito mode forced\nSpecifies whether the user may open pages in Incognito mode in Google Chrome.\n\nIf 'Enabled' is selected or the policy is left unset, pages may be opened in Incognito mode.\n\nIf 'Disabled' is selected, pages may not be opened in Incognito mode.\n\nIf 'Forced' is selected, pages may be opened ONLY in Incognito mode. Note that 'Forced' does not work for Android-on-Chrome\n\nNote: On iOS, if the policy is changed during a session, it will only take effect on relaunch.";

InsecureContentAllowedForUrls.pfm\_title = "Allow insecure content on these sites";

InsecureContentAllowedForUrls.pfm\_description = "Allows you to set a list of url patterns that specify sites which are allowed to display blockable (i.e. active) mixed content (i.e. HTTP content on HTTPS sites) and for which optionally blockable mixed content upgrades will be disabled.\n\nIf this policy is left not set blockable mixed content will be blocked and optionally blockable mixed content will be upgraded, and users will be allowed to set exceptions to allow it for specific sites.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

InsecureContentBlockedForUrls.pfm\_title = "Block insecure content on these sites";

InsecureContentBlockedForUrls.pfm\_description = "Allows you to set a list of url patterns that specify sites which are not allowed to display blockable (i.e. active) mixed content (i.e. HTTP content on HTTPS sites), and for which optionally blockable (i.e. passive) mixed content will be upgraded.\n\nIf this policy is left not set blockable mixed content will be blocked and optionally blockable mixed content will be upgraded, but users will be allowed to set exceptions to allow it for specific sites.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

InsecureFormsWarningsEnabled.pfm\_title = "Enable warnings for insecure forms";

InsecureFormsWarningsEnabled.pfm\_description = "This policy controls the treatment for insecure forms (forms that submit over HTTP) embedded in secure (HTTPS) sites in the browser.\nIf the policy is enabled or unset, a full page warning will be shown when an insecure form is submitted. Additionally, a warning bubble will be shown next to the form fields when they are focused, and autofill will be disabled for those forms.\nIf the policy is disabled, warnings will not be shown for insecure forms, and autofill will work normally.";

InsecurePrivateNetworkRequestsAllowed.pfm\_title = "Specifies whether to allow websites to make requests to more-private network endpoints in an insecure manner";

InsecurePrivateNetworkRequestsAllowed.pfm\_description = "Controls whether websites are allowed to make requests to more-private network endpoints in an insecure manner.\n\nWhen this policy is set to true, all Private Network Access checks are disabled for all origins. This may allow attackers to perform CSRF attacks on private network servers.\n\nWhen this policy is either not set or set to false, the default behavior for requests to more-private network endpoints will depend on the user's personal configuration for the BlockInsecurePrivateNetworkRequests, PrivateNetworkAccessSendPreflights, and PrivateNetworkAccessRespectPreflightResults feature flags, which may be set by field trials or on the command line.\n\nThis policy relates to the Private Network Access specification. See https://wicg.github.io/private-network-access/ for more details.\n\nA network endpoint is more private than another if:\n1) Its IP address is localhost and the other is not.\n2) Its IP address is private and the other is public.\nIn the future, depending on spec evolution, this policy might apply to all cross-origin requests directed at private IPs or localhost.\n\nWhen this policy is set to true, websites are allowed to make requests to any network endpoint, subject to other cross-origin checks.";

InsecurePrivateNetworkRequestsAllowedForUrls.pfm\_title = "Allow the listed sites to make requests to more-private network endpoints in an insecure manner.";

InsecurePrivateNetworkRequestsAllowedForUrls.pfm\_description = "List of URL patterns. Requests initiated from websites served by matching origins are not subject to Private Network Access checks.\n\nIf unset, this policy behaves as if set to the empty list.\n\nFor origins not covered by the patterns specified here, the global default value will be used either from the InsecurePrivateNetworkRequestsAllowed policy, if it is set, or the user's personal configuration otherwise.\n\nFor detailed information on valid URL patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns.";

IntensiveWakeUpThrottlingEnabled.pfm\_title = "Control the IntensiveWakeUpThrottling feature.";

IntensiveWakeUpThrottlingEnabled.pfm\_description = "When enabled the IntensiveWakeUpThrottling feature causes Javascript timers in background tabs to be aggressively throttled and coalesced, running no more than once per minute after a page has been backgrounded for 5 minutes or more.\n\nThis is a web standards compliant feature, but it may break functionality\non some websites by causing certain actions to be delayed by up to a\nminute. However, it results in significant CPU and battery savings when\nenabled. See https://bit.ly/30b1XR4 for more details.\n\nIf this policy is set to enabled then the feature will be force enabled, and\nusers will not be able to override this.\n\nIf this policy is set to disabled then the feature will be force disabled, and\nusers will not be able to override this.\n\nIf this policy is left unset then the feature will be controlled by its\nown internal logic, which can be manually configured by users.\n\nNote that the policy is applied per renderer process, with the most recent\nvalue of the policy setting in force when a renderer process starts. A full\nrestart is required to ensure that all loaded tabs receive a consistent\npolicy setting. It is harmless for processes to be running with different\nvalues of this policy.\n";

IntranetRedirectBehavior.pfm\_title = "Intranet Redirection Behavior";

IntranetRedirectBehavior.pfm\_description = "0 - Use default browser behavior.\n1 - Disable DNS interception checks and did-you-mean \"http://intranetsite/\" infobars.\n2 - Disable DNS interception checks; allow did-you-mean \"http://intranetsite/\" infobars.\n3 - Allow DNS interception checks and did-you-mean \"http://intranetsite/\" infobars.\nThis policy configures behavior for intranet redirection via DNS interception checks. The checks attempt to discover whether the browser is behind a proxy that redirects unknown host names.\n\nIf this policy is not set, the browser will use the default behavior of DNS interception checks and intranet redirect suggestions. In M88, they are enabled by default but will be disabled by default in the future release.\n\nDNSInterceptionChecksEnabled is a related policy that may also disable DNS interception checks; this policy is a more flexible version which may separately control intranet redirection infobars and may be expanded in the future.\nIf either DNSInterceptionChecksEnabled or this policy requests to disable interception checks, the checks will be disabled.";

IsolateOrigins.pfm\_title = "Enable Site Isolation for specified origins";

IsolateOrigins.pfm\_description = "Setting the policy means each of the named origins in a comma-separated list runs in a dedicated process. Each named origin's process will only be allowed to contain documents from that origin and its subdomains. For example, specifying https://a1.example.com/ allows https://a2.a1.example.com/ in the same process, but not https://example.com or https://b.example.com.\n\nSince Google Chrome 77, you can also specify a range of origins to isolate using a wildcard. For example, specifying https://[\*.]corp.example.com will give every origin underneath https://corp.example.com its own dedicated process, including https://corp.example.com itself, https://a1.corp.example.com, and https://a2.a1.corp.example.com.\n\nNote that all sites (i.e., scheme plus eTLD+1, such as https://example.com) are already isolated by default on Desktop platforms, as noted in the SitePerProcess policy. This IsolateOrigins policy is useful to isolate specific origins at a finer granularity (e.g., https://a.example.com).\n\nAlso note that origins isolated by this policy will be unable to script other origins in the same site, which is otherwise possible if two same-site documents modify their document.domain values to match. Administrators should confirm this uncommon behavior is not used on an origin before isolating it.\n\nSetting the policy to off or leaving it unset lets users change this setting.\n\nNote: For Android, use the IsolateOriginsAndroid policy instead.";

JavaScriptAllowedForUrls.pfm\_title = "Allow JavaScript on these sites";

JavaScriptAllowedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify the sites that can run JavaScript.\n\nLeaving the policy unset means DefaultJavaScriptSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

JavaScriptBlockedForUrls.pfm\_title = "Block JavaScript on these sites";

JavaScriptBlockedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify the sites that can't run JavaScript.\n\nLeaving the policy unset means DefaultJavaScriptSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

JavaScriptJitAllowedForSites.pfm\_title = "Allow JavaScript to use JIT on these sites";

JavaScriptJitAllowedForSites.pfm\_description = "Allows you to set a list of site url patterns that specify sites which are allowed to run JavaScript with JIT (Just In Time) compiler enabled.\n\nFor detailed information on valid site url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.\n\nJavaScript JIT policy exceptions will only be enforced at a site granularity (eTLD+1). A policy set for only subdomain.site.com will not correctly apply to site.com or subdomain.site.com since they both resolve to the same eTLD+1 (site.com) for which there is no policy. In this case, policy must be set on site.com to apply correctly for both site.com and subdomain.site.com.\n\nThis policy applies on a frame-by-frame basis and not based on top level origin url alone, so e.g. if site-one.com is listed in the JavaScriptJitAllowedForSites policy but site-one.com loads a frame containing site-two.com then site-one.com will have JavaScript JIT enabled, but site-two.com will use the policy from DefaultJavaScriptJitSetting, if set, or default to JavaScript JIT enabled.\n\nIf this policy is not set for a site then the policy from DefaultJavaScriptJitSetting applies to the site, if set, otherwise Javascript JIT is enabled for the site.";

JavaScriptJitBlockedForSites.pfm\_title = "Block JavaScript from using JIT on these sites";

JavaScriptJitBlockedForSites.pfm\_description = "Allows you to set a list of site url patterns that specify sites which are not allowed to run JavaScript JIT (Just In Time) compiler enabled.\n\nDisabling the JavaScript JIT will mean that Google Chrome may render web content more slowly, and may also disable parts of JavaScript including WebAssembly. Disabling the JavaScript JIT may allow Google Chrome to render web content in a more secure configuration.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.\n\nJavaScript JIT policy exceptions will only be enforced at a site granularity (eTLD+1). A policy set for only subdomain.site.com will not correctly apply to site.com or subdomain.site.com since they both resolve to the same eTLD+1 (site.com) for which there is no policy. In this case, policy must be set on site.com to apply correctly for both site.com and subdomain.site.com.\n\nThis policy applies on a frame-by-frame basis and not based on top level origin url alone, so e.g. if site-one.com is listed in the JavaScriptJitBlockedForSites policy but site-one.com loads a frame containing site-two.com then site-one.com will have JavaScript JIT disabled, but site-two.com will use the policy from DefaultJavaScriptJitSetting, if set, or default to JavaScript JIT enabled.\n\nIf this policy is not set for a site then the policy from DefaultJavaScriptJitSetting applies to the site, if set, otherwise JavaScript JIT is enabled for the site.";

LegacySameSiteCookieBehaviorEnabledForDomainList.pfm\_title = "Revert to legacy SameSite behavior for cookies on these sites";

LegacySameSiteCookieBehaviorEnabledForDomainList.pfm\_description = "Cookies set for domains matching these patterns will revert to legacy SameSite behavior. Reverting to legacy behavior causes cookies that don't specify a SameSite attribute to be treated as if they were \"SameSite=None\", removes the requirement for \"SameSite=None\" cookies to carry the \"Secure\" attribute, and skips the scheme comparison when evaluating if two sites are same-site. See https://www.chromium.org/administrators/policy-list-3/cookie-legacy-samesite-policies for full description.\n\nFor cookies on domains not covered by the patterns specified here, or for all cookies if this policy is not set, the global default value will be the user's personal configuration.\n\nFor detailed information on valid patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns.\n\nNote that patterns you list here are treated as domains, not URLs, so you should not specify a scheme or port.";

LensRegionSearchEnabled.pfm\_title = "Allow Google Lens region search menu item to be shown in context menu if supported.";

LensRegionSearchEnabled.pfm\_description = "Leaving the policy unset or setting it to Enabled allows users to view and use the Google Lens region search menu item in the context menu. Setting the policy to Disabled means users will not see the Google Lens region search menu item in the context menu when Google Lens region search is supported.";

LocalFontsAllowedForUrls.pfm\_title = "Allow Local Fonts permission on these sites";

LocalFontsAllowedForUrls.pfm\_description = "Sets a list of site url patterns that specify sites which will automatically grant the local fonts permission. This will extend the ability of sites to see information about local fonts.\n\nFor detailed information on valid site url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. Wildcards, \*, are allowed. This policy only matches based on origin, so any path in the URL pattern is ignored.\n\nIf this policy is not set for a site then the policy from DefaultLocalFontsSetting applies to the site, if set, otherwise the permission will follow the browser's defaults and allow users to choose this permission per site.";

LocalFontsBlockedForUrls.pfm\_title = "Block Local Fonts permission on these sites";

LocalFontsBlockedForUrls.pfm\_description = "Sets a list of site url patterns that specify sites which will automatically deny the local fonts permission. This will limit the ability of sites to see information about local fonts.\n\nFor detailed information on valid site url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. Wildcards, \*, are allowed. This policy only matches based on origin, so any path in the URL pattern is ignored.\n\nIf this policy is not set for a site then the policy from DefaultLocalFontsSetting applies to the site, if set, otherwise the permission will follow the browser's defaults and allow users to choose this permission per site.";

LookalikeWarningAllowlistDomains.pfm\_title = "Suppress lookalike domain warnings on domains";

LookalikeWarningAllowlistDomains.pfm\_description = "This policy prevents the display of lookalike URL warnings on the sites listed. These warnings are typically shown on sites that Google Chrome believes might be trying to spoof another site the user is familiar with.\n\nIf the policy is enabled and set to one or more domains, no lookalike warnings pages will be shown when the user visits pages on that domain.\n\nIf the policy is not set, or set to an empty list, warnings may appear on any site the user visits.\n\nA hostname can be allowed with a complete host match, or any domain match. For example, a URL like \"https://foo.example.com/bar\" may have warnings suppressed if this list includes either \"foo.example.com\" or \"example.com\".";

ManagedAccountsSigninRestriction.pfm\_title = "Add restrictions on managed accounts";

ManagedAccountsSigninRestriction.pfm\_description = "primary\_account - A Managed account must be a primary account and importing existing browsing data is allowed at the time of profile creation\nprimary\_account\_strict - A Managed account must be a primary account and have no secondary accounts and importing existing browsing data is allowed at the time of profile creation\nnone - No restrictions on managed accounts\nprimary\_account\_keep\_existing\_data - A Managed account must be a primary account and the user can import existing data at the time of its creation\nprimary\_account\_strict\_keep\_existing\_data - A Managed account must be a primary account and have no secondary accounts and the user can import existing data at the time of its creation\n\n If this policy is set to 'primary\_account', after a signin into an a managed account subjected to this policy, the user be asked to create a new profile for the account.\n If this policy is set to 'primary\_account\_keep\_existing\_data', after a signin into an a managed account subjected to this policy, the user be asked to create a new profile for the account with an option to keep any existing browsing data. This option is supported on Chrome 102 and higher version.\n\n If this policy is set to 'primary\_account\_strict', after a signin into an a managed account subjected to this policy, the user be asked to create a new profile for the account. This profile will not allow any secondary accounts.\n If this policy is set to 'primary\_account\_strict\_keep\_existing\_data' after a signin into an a managed account subjected to this policy, the user be asked to create a new profile for the account with an option to keep any existing browsing data. This profile will not allow any secondary accounts. This option is supported on Chrome 102 and higher version.\n\n If this policy is set to 'none' or not set, managed accounts have no restrictions. This may result in a managed account being a secondary account, which disables its ability to receive policies set on the account by the admin.\n\n If this policy is set at the device level, all accounts in the browser are subjected to the policy.\n If this policy is set at an account level, only that account is affected in the browser.\n";

ManagedBookmarks.pfm\_title = "Managed Bookmarks";

ManagedBookmarks.pfm\_description = "Setting the policy sets up a list of bookmarks where each one is a dictionary with the keys \"name\" and \"url\". These keys hold the bookmark's name and target. Admins can set up a subfolder by defining a bookmark without a \"url\" key, but with an additional \"children\" key. This key also has a list of bookmarks, some of which can also be folders. Chrome amends incomplete URLs as if they were submitted through the address bar. For example, \"google.com\" becomes \"https://google.com/\".\n\nUsers can't change the folders the bookmarks are placed in (though they can hide it from the bookmark bar). The default folder name for managed bookmarks is \"Managed bookmarks\" but it can be changed by adding a new sub-dictionary to the policy with a single key named \"toplevel\_name\" with the desired folder name as its value. Managed bookmarks are not synced to the user account and extensions can't modify them.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=ManagedBookmarks for more information about schema and formatting.";

ManagedConfigurationPerOrigin.pfm\_title = "Sets managed configuration values to websites to specific origins";

ManagedConfigurationPerOrigin.pfm\_description = "Setting the policy defines the return value of Managed Configuration API for given origin.\n\n Managed configuration API is a key-value configuration that can be accessed via navigator.managed.getManagedConfiguration() javascript call. This API is only available to origins which correspond to force-installed web applications via WebAppInstallForceList.\n\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=ManagedConfigurationPerOrigin for more information about schema and formatting.";

MaxConnectionsPerProxy.pfm\_title = "Maximal number of concurrent connections to the proxy server";

MaxConnectionsPerProxy.pfm\_description = "Setting the policy specifies the maximal number of simultaneous connections to the proxy server. Some proxy servers can't handle a high number of concurrent connections per client, which is solved by setting this policy to a lower value. The value should be lower than 100 and higher than 6. Some web apps are known to consume many connections with hanging GETs, so setting a value below 32 may lead to browser networking hangs if there are too many web apps with hanging connections open. Lower below the default at your own risk.\n\nLeaving the policy unset means a default of 32 is used.";

MaxInvalidationFetchDelay.pfm\_title = "Maximum fetch delay after a policy invalidation";

MaxInvalidationFetchDelay.pfm\_description = "Setting the policy specifies the maximum delay in milliseconds between receiving a policy invalidation and fetching the new policy from the device management service. Valid values range from 1,000 (1 second) to 300,000 (5 minutes). Values outside this range will be clamped to the respective boundary.\n\nLeaving the policy unset means Google Chrome uses the default value of 10 seconds.";

MediaRecommendationsEnabled.pfm\_title = "Enable Media Recommendations";

MediaRecommendationsEnabled.pfm\_description = "By default the browser will show media recommendations that are personalized to the user. Setting this policy to Disabled will result in these recommendations being hidden from the user. Setting this policy to Enabled or leaving it unset will result in the media recommendations being shown to the user.";

MediaRouterCastAllowAllIPs.pfm\_title = "Allow Google Cast to connect to Cast devices on all IP addresses.";

MediaRouterCastAllowAllIPs.pfm\_description = "Unless EnableMediaRouter is set to Disabled, setting MediaRouterCastAllowAllIPs to Enabled connects Google Cast to Cast devices on all IP addresses, not just RFC1918/RFC4193 private addresses.\n\nSetting the policy to Disabled connects Google Cast to Cast devices only on RFC1918/RFC4193.\n\nLeaving the policy unset connects Google Cast to Cast devices only on RFC1918/RFC4193, unless the CastAllowAllIPs feature is turned on.";

MetricsReportingEnabled.pfm\_title = "Enable reporting of usage and crash-related data";

MetricsReportingEnabled.pfm\_description = "When this policy is enabled, anonymous reporting of usage and crash-related data about Chrome to Google is enabled by default. Users will still be able to change this setting in the Chrome settings.\n\nWhen this policy is disabled, anonymous reporting is disabled and no usage or crash data is sent to Google. Users won't be able to change this setting.\n\nWhen this policy isn't set, users can choose the anonymous reporting behavior at installation or first run, and can later change the setting in the Chrome settings.\n\nThis policy is available only on Windows instances that are joined to a Microsoft® Active Directory® domain or Windows 10 Pro or Enterprise instances that are enrolled for device management, and macOS instances that are managed via MDM or joined to a domain via MCX.\n\n(For Google Chrome OS, see DeviceMetricsReportingEnabled.)";

NTPCardsVisible.pfm\_title = "Show cards on the New Tab Page";

NTPCardsVisible.pfm\_description = "This policy controls the visibility of cards on the New Tab Page. Cards surface entry points to launch common user journeys based on the user's browsing behavior.\n\nIf the policy is set to Enabled, the New Tab Page will show cards if content is available.\n\nIf the policy is set to Disabled, the New Tab Page won't show cards.\n\nIf the policy is not set, the user can control the card visibility. The default is visible.\n";

NTPCustomBackgroundEnabled.pfm\_title = "Allow users to customize the background on the New Tab page";

NTPCustomBackgroundEnabled.pfm\_description = "If the policy is set to false, the New Tab page won't allow users to customize the background. Any existing custom background will be permanently removed even if the policy is set to true later.\n\nIf the policy is set to true or unset, users can customize the background on the New Tab page.";

NTPMiddleSlotAnnouncementVisible.pfm\_title = "Show the middle slot announcement on the New Tab Page";

NTPMiddleSlotAnnouncementVisible.pfm\_description = "This policy controls the visibility of the middle slot announcement on the New Tab Page.\n\nIf the policy is set to Enabled, the New Tab Page will show the middle slot announcement if it is available.\n\nIf the policy is set to Disabled, the New Tab Page will not show the middle slot announcement even if it is available.\n";

NativeMessagingAllowlist.pfm\_title = "Names of the native messaging hosts to exempt from the blocklist";

NativeMessagingAllowlist.pfm\_description = "Setting the policy specifies which native messaging hosts aren't subject to the deny list. A deny list value of \* means all native messaging hosts are denied, unless they're explicitly allowed.\n\nAll native messaging hosts are allowed by default. But, if all native messaging hosts are denied by policy, the admin can use the allow list to change that policy.";

NativeMessagingBlocklist.pfm\_title = "Names of the forbidden native messaging hosts (or \* for all)";

NativeMessagingBlocklist.pfm\_description = "Setting the policy specifies which native messaging hosts shouldn't be loaded. A deny list value of \* means all native messaging hosts are denied, unless they're explicitly allowed.\n\nLeaving the policy unset means Google Chrome loads all installed native messaging hosts.";

NativeMessagingUserLevelHosts.pfm\_title = "Allow user-level Native Messaging hosts (installed without admin permissions)";

NativeMessagingUserLevelHosts.pfm\_description = "Setting the policy to Enabled or leaving it unset means Google Chrome can use native messaging hosts installed at the user level.\n\nSetting the policy to Disabled means Google Chrome can only use these hosts if installed at the system level.";

NetworkPredictionOptions.pfm\_title = "Enable network prediction";

NetworkPredictionOptions.pfm\_description = "0 - Predict network actions on any network connection\n1 - Predict network actions on any network that is not cellular.\n(Deprecated in 50, removed in 52. After 52, if value 1 is set, it will be treated as 0 - predict network actions on any network connection.)\n2 - Do not predict network actions on any network connection\nThis policy controls network prediction in Google Chrome. It controls DNS prefetching, TCP, and SSL preconnection and prerendering of webpages.\n\nIf you set the policy, users can't change it. Leaving it unset turns on network prediction, but the user can change it.";

NewTabPageLocation.pfm\_title = "New Tab page URL";

NewTabPageLocation.pfm\_description = "Setting the policy configures the default New Tab page URL and prevents users from changing it.\n\nThe New Tab page opens with new tabs and windows.\n\nThis policy doesn't decide which pages open on start up. Those are controlled by the RestoreOnStartup policies. This policy does affect the homepage, if that's set to open the New Tab page, as well as the startup page if it's set to open the New Tab page.\n\nIt is a best practice to provide fully canonicalized URL, if the URL is not fully canonicalized Google Chrome will default to https://.\n\nLeaving the policy unset or empty puts the default New Tab page in use.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

NotificationsAllowedForUrls.pfm\_title = "Allow notifications on these sites";

NotificationsAllowedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify the sites that can display notifications.\n\nLeaving the policy unset means DefaultNotificationsSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

NotificationsBlockedForUrls.pfm\_title = "Block notifications on these sites";

NotificationsBlockedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify the sites that can't display notifications.\n\nLeaving the policy unset means DefaultNotificationsSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

NtlmV2Enabled.pfm\_title = "Enable NTLMv2 authentication.";

NtlmV2Enabled.pfm\_description = "Setting the policy to Enabled or leaving it unset turns NTLMv2 on.\n\nSetting the policy to Disabled turns NTLMv2 off.\n\nAll recent versions of Samba and Microsoft® Windows® servers support NTLMv2. This should only be turned off for backward compatibility as it reduces the security of authentication.";

OriginAgentClusterDefaultEnabled.pfm\_title = "Allows origin-keyed agent clustering by default.";

OriginAgentClusterDefaultEnabled.pfm\_description = "This policy allows origin-keyed agent clustering by default.\n\nThe Origin-Agent-Cluster: HTTP header controls whether a document is\nisolated in an origin-keyed agent cluster, or in a site-keyed agent\ncluster. This has security implications since an origin-keyed agent\ncluster allows isolating documents by origin. The developer-visible\nconsequence of this is that the document.domain accessor can no longer\nbe set.\n\nThe default behaviour - when no Origin-Agent-Cluster: header has been set\n- changes in M106 from site-keyed to origin-keyed. If this policy\nis enabled or not set, the browser will follow this new default from that\nversion on. If this policy is disabled this change is reversed and\ndocuments without Origin-Agent-Cluster: headers will be assigned to\nsite-keyed agent clusters. As a consequence, the document.domain accessor\nremains settable by default. This matches the legacy behaviour.\n\nSee https://developer.chrome.com/blog/immutable-document-domain/\nfor additional details.\n";

OverrideSecurityRestrictionsOnInsecureOrigin.pfm\_title = "Origins or hostname patterns for which restrictions on\ninsecure origins should not apply";

OverrideSecurityRestrictionsOnInsecureOrigin.pfm\_description = "Setting the policy specifies a list of origins (URLs) or hostname patterns (such as \*.example.com) for which security restrictions on insecure origins won't apply. Organizations can specify origins for legacy applications that can't deploy TLS or set up a staging server for internal web development, so developers can test out features requiring secure contexts without having to deploy TLS on the staging server. This policy also prevents the origin from being labeled \"Not Secure\" in the address bar.\n\nSetting a list of URLs in this policy amounts to setting the command-line flag --unsafely-treat-insecure-origin-as-secure to a comma-separated list of the same URLs. The policy overrides the command-line flag and UnsafelyTreatInsecureOriginAsSecure, if present.\n\nFor more information on secure contexts, see Secure Contexts ( https://www.w3.org/TR/secure-contexts ).";

PasswordDismissCompromisedAlertEnabled.pfm\_title = "Enable dismissing compromised password alerts for entered credentials";

PasswordDismissCompromisedAlertEnabled.pfm\_description = "Setting the policy to Enabled or leaving it unset gives the user the option to dismiss/restore compromised password alerts.\n\nIf you disable this setting, users will not be able to dismiss alerts about compromised passwords. If enabled, users will be able to dismiss alerts about compromised passwords.";

PasswordLeakDetectionEnabled.pfm\_title = "Enable leak detection for entered credentials";

PasswordLeakDetectionEnabled.pfm\_description = "Setting the policy to Enabled lets users have Google Chrome check whether usernames and passwords entered were part of a leak.\n\nIf the policy is set, users can't change it in Google Chrome. If not set, credential leak checking is allowed, but the user can turn it off.\n\nThis behavior will not trigger if Safe Browsing is disabled (either by policy or by the user). In order to force Safe Browsing on, use the SafeBrowsingEnabled policy or the SafeBrowsingProtectionLevel policy.";

PasswordManagerEnabled.pfm\_title = "Enable saving passwords to the password manager";

PasswordManagerEnabled.pfm\_description = "Setting the policy to Enabled means users have Google Chrome remember passwords and provide them the next time they sign in to a site.\n\nSetting the policy to Disabled means users can't save new passwords, but previously saved passwords will still work.\n\nIf the policy is set, users can't change it in Google Chrome. If not set, the user can turn off password saving.";

PasswordProtectionChangePasswordURL.pfm\_title = "Configure the change password URL.";

PasswordProtectionChangePasswordURL.pfm\_description = "Setting the policy sets the URL for users to change their password after seeing a warning in the browser. The password protection service sends users to the URL (HTTP and HTTPS protocols only) you designate through this policy. For Google Chrome to correctly capture the salted hash of the new password on this change password page, make sure your change password page follows these guidelines ( https://www.chromium.org/developers/design-documents/create-amazing-password-forms ).\n\nTurning the policy off or leaving it unset means the service sends users to https://myaccount.google.com to change their password.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

PasswordProtectionLoginURLs.pfm\_title = "Configure the list of enterprise login URLs where password protection service should capture salted hashes of passwords.";

PasswordProtectionLoginURLs.pfm\_description = "Setting the policy sets the list of enterprise login URLs (HTTP and HTTPS protocols only). Password protection service will capture salted hashes of passwords on these URLs and use them for password reuse detection. For Google Chrome to correctly capture password salted hashes, ensure your sign-in pages follow these guidelines ( https://www.chromium.org/developers/design-documents/create-amazing-password-forms ).\n\nTurning this setting off or leaving it unset means the password protection service only captures the password salted hashes on https://accounts.google.com.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

PasswordProtectionWarningTrigger.pfm\_title = "Password protection warning trigger";

PasswordProtectionWarningTrigger.pfm\_description = "0 - Password protection warning is off\n1 - Password protection warning is triggered by password reuse\n2 - Password protection warning is triggered by password reuse on phishing page\nSetting the policy lets you control the triggering of password protection warning. Password protection alerts users when they reuse their protected password on potentially suspicious sites.\n\nUse PasswordProtectionLoginURLs and PasswordProtectionChangePasswordURL to set which password to protect.\n\nIf this policy is set to:\n\n\* PasswordProtectionWarningOff, no password protection warning will be shown.\n\n\* PasswordProtectionWarningOnPasswordReuse, password protection warning will be shown when the user reuses their protected password on a non-allowed site.\n\n\* PasswordProtectionWarningOnPhishingReuse, password protection warning will be shown when the user reuses their protected password on a phishing site.\n\nLeaving the policy unset has the password protection service only protect Google passwords, but users can change this setting.";

PaymentMethodQueryEnabled.pfm\_title = "Allow websites to query for available payment methods.";

PaymentMethodQueryEnabled.pfm\_description = "Allows you to set whether websites are allowed to check if the user has payment methods saved.\n\nIf this policy is set to disabled, websites that use PaymentRequest.canMakePayment or PaymentRequest.hasEnrolledInstrument API will be informed that no payment methods are available.\n\nIf the setting is enabled or not set then websites are allowed to check if the user has payment methods saved.";

PolicyAtomicGroupsEnabled.pfm\_title = "Enables the concept of policy atomic groups";

PolicyAtomicGroupsEnabled.pfm\_description = "Setting the policy to Enabled means policies coming from an atomic group that don't share the source with the highest priority from that group get ignored.\n\nSetting the policy to Disabled means no policy is ignored because of its source. Policies are ignored only if there's a conflict, and the policy doesn't have the highest priority.\n\nIf this policy is set from a cloud source, it can't target a specific user.";

PolicyDictionaryMultipleSourceMergeList.pfm\_title = "Allow merging dictionary policies from different sources";

PolicyDictionaryMultipleSourceMergeList.pfm\_description = "ContentPackManualBehaviorURLs - Managed user manual exception URLs\nDeviceLoginScreenPowerManagement - Power management on the login screen\nExtensionSettings - Extension management settings\nKeyPermissions - Key Permissions\nPowerManagementIdleSettings - Power management settings when the user becomes idle\nScreenBrightnessPercent - Screen brightness percent\nScreenLockDelays - Screen lock delays\nSetting the policy allows merging of selected policies when they come from different sources, with the same scopes and level. This merging is in the first level keys of the dictionary from each source. The key coming from the highest priority source takes precedence.\n\nUse the wildcard character '\*' to allow merging of all supported dictionary policies.\n\nIf a policy is in the list and there's conflict between sources with:\n\n\* The same scopes and level: The values merge into a new policy dictionary.\n\n\* Different scopes or level: The policy with the highest priority applies.\n\nIf a policy isn't in the list and there's conflict between sources, scopes, or level, the policy with the highest priority applies.";

PolicyListMultipleSourceMergeList.pfm\_title = "Allow merging list policies from different sources";

PolicyListMultipleSourceMergeList.pfm\_description = "Setting the policy allows merging of selected policies when they come from different sources, with the same scopes and level.\n\nUse the wildcard character '\*' to allow merging of all list policies.\n\nIf a policy is in the list and there's conflict between sources with:\n\n\* The same scopes and level: The values merge into a new policy list.\n\n\* Different scopes or level: The policy with the highest priority applies.\n\nIf a policy isn't in the list and there's conflict between sources, scopes, or level, the policy with the highest priority applies.";

PolicyRefreshRate.pfm\_title = "Refresh rate for user policy";

PolicyRefreshRate.pfm\_description = "Setting the policy specifies the period in milliseconds at which the device management service is queried for user policy information. Valid values range from 1,800,000 (30 minutes) to 86,400,000 (1 day). Values outside this range will be clamped to the respective boundary.\n\nLeaving the policy unset uses the default value of 3 hours.\n\nNote: Policy notifications force a refresh when the policy changes, making frequent refreshes unnecessary. So, if the platform supports these notifications, the refresh delay is 24 hours (ignoring defaults and the value of this policy).";

PopupsAllowedForUrls.pfm\_title = "Allow pop-ups on these sites";

PopupsAllowedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify the sites that can open pop-ups.\n\nLeaving the policy unset means DefaultPopupsSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

PopupsBlockedForUrls.pfm\_title = "Block pop-ups on these sites";

PopupsBlockedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify the sites that can't open pop-ups.\n\nLeaving the policy unset means DefaultPopupsSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

PrintHeaderFooter.pfm\_title = "Print Headers and Footers";

PrintHeaderFooter.pfm\_description = "Setting the policy to Enabled turns headers and footers on in print preview. Setting the policy to Disabled turns them off in print preview.\n\nIf you set the policy, users can't change it. If unset, users decides whether headers and footers appear.";

PrintPdfAsImageAvailability.pfm\_title = "Print PDF as Image Available";

PrintPdfAsImageAvailability.pfm\_description = "Controls how Google Chrome makes the Print as image option available on Microsoft® Windows® and macOS when printing PDFs.\n\nWhen printing a PDF on Microsoft® Windows® or macOS, sometimes print jobs need to be rasterized to an image for certain printers to get correct looking output.\n\nWhen this policy is set to Enabled, Google Chrome will make the Print as image option available in the Print Preview when printing a PDF.\n\nWhen this policy is set to Disabled or not set Google Chrome the Print as image option will not be available to users in Print Preview and PDFs will be printed as usual without being rasterized to an image before being sent to the destination.";

PrintPdfAsImageDefault.pfm\_title = "Print PDF as Image Default";

PrintPdfAsImageDefault.pfm\_description = "Controls if Google Chrome makes the Print as image option default to set when printing PDFs.\n\nWhen this policy is set to Enabled, Google Chrome will default to setting the Print as image option in the Print Preview when printing a PDF.\n\nWhen this policy is set to Disabled or not set Google Chrome then the user selection for Print as image option will be initially unset. The user will be allowed to select it for each individual PDFs print job, if the option is available.\n\nFor Microsoft® Windows® or macOS this policy only has an effect if PrintPdfAsImageAvailability is also enabled.";

PrintPreviewUseSystemDefaultPrinter.pfm\_title = "Use System Default Printer as Default";

PrintPreviewUseSystemDefaultPrinter.pfm\_description = "Setting the policy to Enabled means Google Chrome uses the OS default printer as the default destination for print preview.\n\nSetting the policy to Disabled or leaving it unset means Google Chrome uses the most recently used printer as the default destination for print preview.";

PrintRasterizePdfDpi.pfm\_title = "Print Rasterize PDF DPI";

PrintRasterizePdfDpi.pfm\_description = "Controls print image resolution when Google Chrome prints PDFs with rasterization.\n\nWhen printing a PDF using the Print to image option, it can be beneficial to specify a print resolution other than a device's printer setting or the PDF default. A high resolution will significantly increase the processing and printing time while a low resolution can lead to poor imaging quality.\n\nThis policy allows a particular resolution to be specified for use when rasterizing PDFs for printing.\n\nIf this policy is set to zero or not set at all then the system default resolution will be used during rasterization of page images.";

PrinterTypeDenyList.pfm\_title = "Disable printer types on the deny list";

PrinterTypeDenyList.pfm\_description = "privet - Zeroconf-based (mDNS + DNS-SD) protocol destinations (Deprecated)\nextension - Extension-based destinations\npdf - The 'Save as PDF' destination\nlocal - Local printer destinations\ncloud - Google Cloud Print and 'Save to Google Drive' destinations (Deprecated)\nThe printers of types placed on the deny list will be disabled from being discovered or having their capabilities fetched.\n\nPlacing all printer types on the deny list effectively disables printing, as there would be no available destinations to send a document for printing.\n\nIn versions before 102, including cloud on the deny list has the same effect as setting the CloudPrintSubmitEnabled policy to false. In order to keep Google Cloud Print destinations discoverable, the CloudPrintSubmitEnabled policy must be set to true and cloud must not be on the deny list. Beginning in version 102, Google Cloud Print destinations are not supported and will not appear regardless of policy values.\n\nIf the policy is not set, or is set to an empty list, all printer types will be available for discovery.\n\nExtension printers are also known as print provider destinations, and include any destination that belongs to a Google Chrome extension.\n\nLocal printers are also known as native printing destinations, and include destinations available to the local machine and shared network printers.";

PrintingAllowedBackgroundGraphicsModes.pfm\_title = "Restrict background graphics printing mode";

PrintingAllowedBackgroundGraphicsModes.pfm\_description = "any - Allow printing both with and without background graphics\nenabled - Allow printing only with background graphics\ndisabled - Allow printing only without background graphics\nRestricts background graphics printing mode. Unset policy is treated as no restriction.";

PrintingBackgroundGraphicsDefault.pfm\_title = "Default background graphics printing mode";

PrintingBackgroundGraphicsDefault.pfm\_description = "enabled - Enable background graphics printing mode by default\ndisabled - Disable background graphics printing mode by default\nOverrides default background graphics printing mode.";

PrintingEnabled.pfm\_title = "Enable printing";

PrintingEnabled.pfm\_description = "Setting the policy to Enabled or leaving it unset lets users print in Google Chrome, and users can't change this setting.\n\nSetting the policy to Disabled means users can't print from Google Chrome. Printing is off in the three dots menu, extensions, and JavaScript applications.";

PrintingPaperSizeDefault.pfm\_title = "Default printing page size";

PrintingPaperSizeDefault.pfm\_description = "Overrides default printing page size.\n\nname should contain one of the listed formats or 'custom' if required paper size is not in the list. If 'custom' value is provided custom\_size property should be specified. It describes the desired height and width in micrometers. Otherwise custom\_size property shouldn't be specified. Policy that violates these rules is ignored.\n\nIf the page size is unavailable on the printer chosen by the user this policy is ignored.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=PrintingPaperSizeDefault for more information about schema and formatting.";

ProfilePickerOnStartupAvailability.pfm\_title = "Profile picker availability on startup";

ProfilePickerOnStartupAvailability.pfm\_description = "0 - Profile picker available at startup\n1 - Profile picker disabled at startup\n2 - Profile picker forced at startup\nSpecifies whether the profile picker is enabled, disabled or forced at the browser startup.\n\nBy default the profile picker is not shown if the browser starts in guest or incognito mode, a profile directory and/or urls are specified by command line, an app is explicitly requested to open, the browser was launched by a native notification, there is only one profile available or the policy ForceBrowserSignin is set to true.\n\nIf 'Enabled' (0) is selected or the policy is left unset, the profile picker will be shown at startup by default, but users will be able to enable/disable it.\n\nIf 'Disabled' (1) is selected, the profile picker will never be shown, and users will not be able to change the setting.\n\nIf 'Forced' (2) is selected, the profile picker cannot be suppressed by the user. The profile picker will be shown even if there is only one profile available.";

PromotionalTabsEnabled.pfm\_title = "Enable showing full-tab promotional content";

PromotionalTabsEnabled.pfm\_description = "Setting the policy to True or leaving it unset lets Google Chrome show users product information as full-tab content.\n\nSetting the policy to False prevents Google Chrome from showing product information as full-tab content.\n\nSetting the policy controls the presentation of the welcome pages that help users sign in to Google Chrome, set Google Chrome as users' default browser, or otherwise inform them of product features.";

PromptForDownloadLocation.pfm\_title = "Ask where to save each file before downloading";

PromptForDownloadLocation.pfm\_description = "Setting the policy to Enabled means users are asked where to save each file before downloading. Setting the policy to Disabled has downloads start immediately, and users aren't asked where to save the file.\n\nLeaving the policy unset lets users change this setting.";

PromptOnMultipleMatchingCertificates.pfm\_title = "Prompt when multiple certificates match";

PromptOnMultipleMatchingCertificates.pfm\_description = "This policy controls whether the user is prompted to select a client certificate when more than one certificate matches AutoSelectCertificateForUrls.\nIf this policy is set to Enabled, the user is prompted to select a client certificate whenever the auto-selection policy matches multiple certificates.\nIf this policy is set to Disabled or not set, the user may only be prompted when no certificate matches the auto-selection.";

ProxySettings.pfm\_title = "Proxy settings";

ProxySettings.pfm\_description = "Setting the policy configures the proxy settings for Chrome and ARC-apps, which ignore all proxy-related options specified from the command line.\n\n Leaving the policy unset lets users choose their proxy settings.\n\n Setting the ProxySettings policy accepts the following fields:\n \* ProxyMode, which lets you specify the proxy server Chrome uses and prevents users from changing proxy settings\n \* ProxyPacUrl, a URL to a proxy .pac file\n \* ProxyPacMandatory, which prevents the network stack from falling back to direct connections with invalid or unavailable PAC script\n \* ProxyServer, a URL of the proxy server\n \* ProxyBypassList, a list of hosts for which the proxy will be bypassed\n\n The ProxyServerMode field is deprecated in favor of the ProxyMode field.\n\n For ProxyMode, if you choose the value:\n \* direct, a proxy is never used and all other fields are ignored.\n \* system, the systems's proxy is used and all other fields are ignored.\n \* auto\_detect, all other fields are ignored.\n \* fixed\_servers, the ProxyServer and ProxyBypassList fields are used.\n \* pac\_script, the ProxyPacUrl, ProxyPacMandatory and ProxyBypassList fields are used.\n\nNote: For more detailed examples, visit The Chromium Projects ( https://www.chromium.org/developers/design-documents/network-settings#TOC-Command-line-options-for-proxy-sett ).\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=ProxySettings for more information about schema and formatting.";

QuicAllowed.pfm\_title = "Allow QUIC protocol";

QuicAllowed.pfm\_description = "Setting the policy to Enabled or leaving it unset allows the use of QUIC protocol in Google Chrome.\n\nSetting the policy to Disabled disallows the use of QUIC protocol.";

RegisteredProtocolHandlers.pfm\_title = "Register protocol handlers";

RegisteredProtocolHandlers.pfm\_description = "Setting the policy (as recommended only) lets you register a list of protocol handlers, which merge with the ones that the user registers, putting both sets in use. Set the property \"protocol\" to the scheme, such as \"mailto\", and set the property \"URL\" to the URL pattern of the application that handles the scheme specified in the \"protocol\" field. The pattern can include a \"%s\" placeholder, which the handled URL replaces.\n\nUsers can't remove a protocol handler registered by policy. However, by installing a new default handler, they can change the protocol handlers installed by policy.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=RegisteredProtocolHandlers for more information about schema and formatting.";

RelaunchNotification.pfm\_title = "Notify a user that a browser relaunch or device restart is recommended or required";

RelaunchNotification.pfm\_description = "1 - Show a recurring prompt to the user indicating that a relaunch is recommended\n2 - Show a recurring prompt to the user indicating that a relaunch is required\nNotify users that Google Chrome must be relaunched or Google Chrome OS must be restarted to apply a pending update.\n\nThis policy setting enables notifications to inform the user that a browser relaunch or device restart is recommended or required. If not set, Google Chrome indicates to the user that a relaunch is needed via subtle changes to its menu, while Google Chrome OS indicates such via a notification in the system tray. If set to 'Recommended', a recurring warning will be shown to the user that a relaunch is recommended. The user can dismiss this warning to defer the relaunch. If set to 'Required', a recurring warning will be shown to the user indicating that a browser relaunch will be forced once the notification period passes. The default period is seven days for Google Chrome and four days for Google Chrome OS, and may be configured via the RelaunchNotificationPeriod policy setting.\n\nThe user's session is restored following the relaunch/restart.";

RelaunchNotificationPeriod.pfm\_title = "Time period (milliseconds)";

RelaunchNotificationPeriod.pfm\_description = "Allows you to set the time period, in milliseconds, over which users are notified that Google Chrome must be relaunched or that a Google Chrome OS device must be restarted to apply a pending update.\n\nOver this time period, the user will be repeatedly informed of the need for an update. For Google Chrome OS devices, a restart notification appears in the system tray according to the RelaunchHeadsUpPeriod policy. For Google Chrome browsers, the app menu changes to indicate that a relaunch is needed once one third of the notification period passes. This notification changes color once two thirds of the notification period passes, and again once the full notification period has passed. The additional notifications enabled by the RelaunchNotification policy follow this same schedule.\n\nIf not set, the default period of 604800000 milliseconds (one week) is used.";

RelaunchWindow.pfm\_title = "Relaunch time window";

RelaunchWindow.pfm\_description = "Specify a target time window for the end of the relaunch notification period.\n\nUsers are notified of the need for a browser relaunch or device restart based on the RelaunchNotification and RelaunchNotificationPeriod policy settings. Browsers and devices are forcibly restarted at the end of the notification period when the RelaunchNotification policy is set to 'Required'. This RelaunchWindow policy can be used to defer the end of the notification period so that it falls within a specific time window.\n\nIf this policy is not set, the default target time window for Google Chrome OS is between 2 AM and 4 AM. The default target time window for Google Chrome is the whole day (i.e., the end of the notification period is never deferred).\n\nNote: Though the policy can accept multiple items in entries, all but the first item are ignored.\nWarning: Setting this policy may delay application of software updates.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=RelaunchWindow for more information about schema and formatting.";

RemoteAccessHostAllowClientPairing.pfm\_title = "Enable or disable PIN-less authentication for remote access hosts";

RemoteAccessHostAllowClientPairing.pfm\_description = "Setting the policy to Enabled or leaving it unset lets users pair clients and hosts at connection time, eliminating the need to enter a PIN every time.\n\nSetting the policy to Disabled makes this feature unavailable.";

RemoteAccessHostAllowFileTransfer.pfm\_title = "Allow remote access users to transfer files to/from the host";

RemoteAccessHostAllowFileTransfer.pfm\_description = "Setting the policy to Enabled or leaving it unset allows users connected to a remote access host to transfer files between the client and the host. This doesn't apply to remote assistance connections, which don't support file transfer.\n\nSetting the policy to Disabled disallows file transfer.";

RemoteAccessHostAllowRelayedConnection.pfm\_title = "Enable the use of relay servers by the remote access host";

RemoteAccessHostAllowRelayedConnection.pfm\_description = "If RemoteAccessHostFirewallTraversal is set to Enabled, setting RemoteAccessHostAllowRelayedConnection to Enabled or leaving it unset allows the use of remote clients to use relay servers to connect to this machine when a direct connection is not available, for example, because of firewall restrictions.\n\nSetting the policy to Disabled doesn't turn remote access off, but only allows connections from the same network (not NAT traversal or relay).";

RemoteAccessHostAllowRemoteAccessConnections.pfm\_title = "Allow remote access connections to this machine";

RemoteAccessHostAllowRemoteAccessConnections.pfm\_description = "If this policy is Disabled, the remote access host service cannot be started or configured to accept incoming connections. This policy does not affect remote support scenarios.\n\nThis policy has no effect if it is set to Enabled, left empty, or is not set.";

RemoteAccessHostAllowRemoteSupportConnections.pfm\_title = "Allow remote support connections to this machine";

RemoteAccessHostAllowRemoteSupportConnections.pfm\_description = "If this policy is disabled, the remote support host cannot be started or configured to accept incoming connections.\n\nThis policy does not affect remote access scenarios.\n\nThis policy does not prevent enterprise admins from connecting to managed Google Chrome OS devices.\n\nThis policy has no effect if enabled, left empty, or is not set.";

RemoteAccessHostClientDomainList.pfm\_title = "Configure the required domain names for remote access clients";

RemoteAccessHostClientDomainList.pfm\_description = "Setting the policy specifies the client domain names that are imposed on remote access clients, and users can't change them. Only clients from one of the specified domains can connect to the host.\n\nSetting the policy to an empty list or leaving it unset applies the default policy for the connection type. For remote assistance, this allows clients from any domain to connect to the host. For anytime remote access, only the host owner can connect.\n\nSee also RemoteAccessHostDomainList.\n\nNote: This setting overrides RemoteAccessHostClientDomain, if present.";

RemoteAccessHostClipboardSizeBytes.pfm\_title = "The maximum size, in bytes, that can be transferred between client and host via clipboard synchronization";

RemoteAccessHostClipboardSizeBytes.pfm\_description = "If this policy is set, clipboard data sent to and from the host will be truncated to the limit set by this policy.\n\nIf a value of 0 is set, then clipboard sync is disabled.\n\nThis policy affects both remote access and remote support scenarios.\n\nThis policy has no effect if it is not set.\n\nSetting the policy to a value that is not within the min/max range may prevent the host from starting.\n\nPlease note that the actual upper bound for the clipboard size is based on the maximum WebRTC data channel message size which this policy does not control.";

RemoteAccessHostDomainList.pfm\_title = "Configure the required domain names for remote access hosts";

RemoteAccessHostDomainList.pfm\_description = "Setting the policy specifies the host domain names that are imposed on remote access hosts, and users can't change them. Hosts can be shared only using accounts registered on one of the specified domain names.\n\nSetting the policy to an empty list or leaving it unset means hosts can be shared using any account.\n\nSee also RemoteAccessHostClientDomainList.\n\nNote: This setting will override RemoteAccessHostDomain, if present.";

RemoteAccessHostFirewallTraversal.pfm\_title = "Enable firewall traversal from remote access host";

RemoteAccessHostFirewallTraversal.pfm\_description = "Setting the policy to Enabled or leaving it unset allows the usage of STUN servers, letting remote clients discover and connect to this machine, even if separated by a firewall.\n\nSetting the policy to Disabled when outgoing UDP connections are filtered by the firewall means the machine only allows connections from client machines within the local network.";

RemoteAccessHostMatchUsername.pfm\_title = "Require that the name of the local user and the remote access host owner match";

RemoteAccessHostMatchUsername.pfm\_description = "Setting the policy to Enabled has the remote access host compare the name of the local user the host is associated with and the name of the Google Account registered as the host owner (\"johndoe,\" if the host is owned by \"johndoe@example.com\"). This host won't start if the host owner's name differs from the name of the local user that the host is associated with. To enforce that the owner's Google Account is associated with a specific domain, use the policy with RemoteAccessHostDomain.\n\nSetting the policy to Disabled or leaving it unset means the remote access host can be associated with any local user.";

RemoteAccessHostMaximumSessionDurationMinutes.pfm\_title = "Maximum session duration allowed for remote access connections";

RemoteAccessHostMaximumSessionDurationMinutes.pfm\_description = "If this policy is set, remote access connections will automatically disconnect after the number of minutes defined in the policy have elapsed. This does not prevent the client from reconnecting after the maximum session duration has been reached. Setting the policy to a value that is not within the min/max range may prevent the host from starting. This policy does not affect remote support scenarios.\n\nThis policy has no effect if it is not set. In this case, remote access connections will have no maximum duration on this machine.";

RemoteAccessHostRequireCurtain.pfm\_title = "Enable curtaining of remote access hosts";

RemoteAccessHostRequireCurtain.pfm\_description = "Setting the policy to Enabled turns off remote access hosts' physical input and output devices during a remote connection.\n\nSetting the policy to Disabled or leaving it unset lets both local and remote users interact with the host while it's shared.";

RemoteAccessHostUdpPortRange.pfm\_title = "Restrict the UDP port range used by the remote access host";

RemoteAccessHostUdpPortRange.pfm\_description = "Setting the policy restricts the UDP port range used by the remote access host in this machine.\n\nLeaving the policy unset or set to an empty string means the remote access host can use any available port.\n\nNote: If RemoteAccessHostFirewallTraversal is Disabled, the remote access host will use UDP ports in the 12400-12409 range.";

RemoteDebuggingAllowed.pfm\_title = "Allow remote debugging";

RemoteDebuggingAllowed.pfm\_description = "Controls whether users may use remote debugging.\n\nIf this policy is set to Enabled or not set, users may use remote debugging by specifying --remote-debugging-port and --remote-debugging-pipe command line switches.\n\nIf this policy is set to Disabled, users are not allowed to use remote debugging.";

RestoreOnStartup.pfm\_title = "Action on startup";

RestoreOnStartup.pfm\_description = "5 - Open New Tab Page\n1 - Restore the last session\n4 - Open a list of URLs\n6 - Open a list of URLs and restore the last session\nSetting the policy lets you specify system behavior on startup. Turning this setting off amounts to leaving it unset as Google Chrome must have specified start up behavior.\n\nIf you set the policy, users can't change it in Google Chrome. If not set, users can change it.\n\nSetting this policy to RestoreOnStartupIsLastSession or RestoreOnStartupIsLastSessionAndURLs turns off some settings that rely on sessions or that perform actions on exit, such as clearing browsing data on exit or session-only cookies.\n\nIf this policy is set to RestoreOnStartupIsLastSessionAndURLs, browser will restore previous session and open a separate window to show URLs that are set from RestoreOnStartupURLs. Note that users can choose to keep those URLs open and they will also be restored in the future session.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

RestoreOnStartupURLs.pfm\_title = "URLs to open on startup";

RestoreOnStartupURLs.pfm\_description = "If RestoreOnStartup is set to RestoreOnStartupIsURLs, then setting RestoreOnStartupURLs to a list of URLs specify which URLs open.\n\nIf not set, the New Tab page opens on start up.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

RestrictSigninToPattern.pfm\_title = "Restrict which Google accounts are allowed to be set as browser primary accounts in Google Chrome";

RestrictSigninToPattern.pfm\_description = "Contains a regular expression which is used to determine which Google accounts can be set as browser primary accounts in Google Chrome (i.e. the account that is chosen during the Sync opt-in flow).\n\nAn appropriate error is displayed if a user tries to set a browser primary account with a username that does not match this pattern.\n\nIf this policy is left not set or blank, then the user can set any Google account as a browser primary account in Google Chrome.";

RoamingProfileLocation.pfm\_title = "Set the roaming profile directory";

RoamingProfileLocation.pfm\_description = "Configures the directory that Google Chrome will use for storing the roaming copy of the profiles.\n\nIf you set this policy, Google Chrome will use the provided directory to store the roaming copy of the profiles if the RoamingProfileSupportEnabled policy has been enabled. If the RoamingProfileSupportEnabled policy is disabled or left unset the value stored in this policy is not used.\n\nSee https://www.chromium.org/administrators/policy-list-3/user-data-directory-variables for a list of variables that can be used.\n\nOn non-Windows platforms, this policy must be set for roaming profiles to work.\n\nOn Windows, if this policy is left unset, the default roaming profile path will be used.";

RoamingProfileSupportEnabled.pfm\_title = "Enable the creation of roaming copies for Google Chrome profile data.";

RoamingProfileSupportEnabled.pfm\_description = "If you enable this setting, the settings stored in Google Chrome profiles like bookmarks, autofill data, passwords, etc. will also be written to a file stored in the Roaming user profile folder or a location specified by the Administrator through the RoamingProfileLocation policy. Enabling this policy disables cloud sync.\n\nIf this policy is disabled or left not set only the regular local profiles will be used.";

SSLErrorOverrideAllowed.pfm\_title = "Allow proceeding from the SSL warning page";

SSLErrorOverrideAllowed.pfm\_description = "Setting the policy to Enabled or leaving it unset lets users click through warning pages Google Chrome shows when users navigate to sites that have SSL errors.\n\nSetting the policy to Disabled prevent users from clicking through any warning pages.";

SSLErrorOverrideAllowedForOrigins.pfm\_title = "Allow proceeding from the SSL warning page on specific origins";

SSLErrorOverrideAllowedForOrigins.pfm\_description = "If SSLErrorOverrideAllowed is Disabled, setting the policy lets you set a list of origin patterns that specify the sites where a user can click through warning pages Google Chrome shows when users navigate to sites that have SSL errors. Users will not be able to click through SSL warning pages on origins that are not on this list.\n\nIf SSLErrorOverrideAllowed is Enabled or unset, this policy does nothing.\n\nLeaving the policy unset means SSLErrorOverrideAllowed applies for all sites.\n\nFor detailed information on valid input patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy. This policy only matches based on origin, so any path in the URL pattern is ignored.";

SSLVersionMin.pfm\_title = "Minimum SSL version enabled";

SSLVersionMin.pfm\_description = "tls1 - TLS 1.0\ntls1.1 - TLS 1.1\ntls1.2 - TLS 1.2\nSetting the policy to a valid value means Google Chrome won't use SSL/TLS versions less than the specified version. Unrecognized values are ignored.\n\nIf this policy is not set, then Google Chrome will show an error for TLS 1.0 and TLS 1.1, but the user will be able to bypass it.\n\nIf this policy is set to \"tls1.2\", the user will not be able to bypass this error.\n\nSupport for setting this policy to \"tls1\" or \"tls1.1\" was removed in version 91. Suppressing the TLS 1.0/1.1 warning is no longer supported.";

SafeBrowsingAllowlistDomains.pfm\_title = "Configure the list of domains on which Safe Browsing will not trigger warnings.";

SafeBrowsingAllowlistDomains.pfm\_description = "Setting the policy to Enabled means Safe Browsing will trust the domains you designate. It won't check them for dangerous resources such as phishing, malware, or unwanted software. Safe Browsing's download protection service won't check downloads hosted on these domains. Its password protection service won't check for password reuse.\n\nLeaving the policy unset means default Safe Browsing protection applies to all resources.\n\nThis policy must be set as a list of fully qualified domain names. It does not support regular expressions, and will not allowlist subdomains of domains listed in the policy.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

SafeBrowsingExtendedReportingEnabled.pfm\_title = "Enable Safe Browsing Extended Reporting";

SafeBrowsingExtendedReportingEnabled.pfm\_description = "Setting the policy to Enabled turns on Google Chrome's Safe Browsing Extended Reporting, which sends some system information and page content to Google servers to help detect dangerous apps and sites.\n\nSetting the policy to Disabled means reports are never sent.\n\nIf you set this policy, users can't change it. If not set, users can decide whether to send reports or not.\n\nSee more about Safe Browsing ( https://developers.google.com/safe-browsing ).";

SafeBrowsingProtectionLevel.pfm\_title = "Safe Browsing Protection Level";

SafeBrowsingProtectionLevel.pfm\_description = "0 - Safe Browsing is never active.\n1 - Safe Browsing is active in the standard mode.\n2 - Safe Browsing is active in the enhanced mode. This mode provides better security, but requires sharing more browsing information with Google.\nAllows you to control whether Google Chrome's Safe Browsing feature is enabled and the mode it operates in.\n\nIf this policy is set to 'NoProtection' (value 0), Safe Browsing is never active.\n\nIf this policy is set to 'StandardProtection' (value 1, which is the default), Safe Browsing is always active in the standard mode.\n\nIf this policy is set to 'EnhancedProtection' (value 2), Safe Browsing is always active in the enhanced mode, which provides better security, but requires sharing more browsing information with Google.\n\nIf you set this policy as mandatory, users cannot change or override the Safe Browsing setting in Google Chrome.\n\nIf this policy is left not set, Safe Browsing will operate in Standard Protection mode but users can change this setting.\n\nSee https://developers.google.com/safe-browsing for more info on Safe Browsing.";

SafeSitesFilterBehavior.pfm\_title = "Control SafeSites adult content filtering.";

SafeSitesFilterBehavior.pfm\_description = "0 - Do not filter sites for adult content\n1 - Filter top level sites (but not embedded iframes) for adult content\nSetting the policy controls the SafeSites URL filter, which uses the Google Safe Search API to classify URLs as pornographic or not.\n\nWhen this policy is set to:\n\n\* Do not filter sites for adult content, or not set, sites aren't filtered\n\n\* Filter top level sites for adult content, pornographic sites are filtered";

SameOriginTabCaptureAllowedByOrigins.pfm\_title = "Allow Same Origin Tab capture by these origins";

SameOriginTabCaptureAllowedByOrigins.pfm\_description = "Setting the policy lets you set a list of URL patterns that can capture tabs with their same Origin.\n\nLeaving the policy unset means that sites will not be considered for an override at this level of capture.\n\nNote that windowed Chrome Apps with the same origin as this site will still be allowed to be captured.\n\nIf a site matches a URL pattern in this policy, the following policies will not be considered: TabCaptureAllowedByOrigins, WindowCaptureAllowedByOrigins, ScreenCaptureAllowedByOrigins, ScreenCaptureAllowed.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. This policy only matches based on origin, so any path in the URL pattern is ignored.";

SandboxExternalProtocolBlocked.pfm\_title = "Allow Chrome to block navigations toward external protocols in sandboxed iframes";

SandboxExternalProtocolBlocked.pfm\_description = "Chrome will block navigations toward external protocols inside\nsandboxed iframe. See https://chromestatus.com/features/5680742077038592.\n\nWhen True, this lets Chrome blocks those navigations.\n\nWhen False, this prevents Chrome from blocking those navigations.\n\nThis defaults to True: security feature enabled.\n\nThis can be used by administrators who need more time to update their internal website affected by this new restriction. This Enterprise policy is temporary; it's intended to be removed after Google Chrome version 117.\n";

SavingBrowserHistoryDisabled.pfm\_title = "Disable saving browser history";

SavingBrowserHistoryDisabled.pfm\_description = "Setting the policy to Enabled means browsing history is not saved, tab syncing is off and users can't change this setting.\n\nSetting the policy to Disabled or leaving it unset saves browsing history.";

ScreenCaptureAllowed.pfm\_title = "Allow or deny screen capture";

ScreenCaptureAllowed.pfm\_description = "If enabled or not configured (default), a Web page can use\nscreen-share APIs (e.g., getDisplayMedia() or the Desktop Capture extension API)\nto prompt the user to select a tab, window or desktop to capture.\n\nWhen this policy is disabled, any calls to screen-share APIs will fail\nwith an error; however this policy is not considered (and a site will be\nallowed to use screen-share APIs) if the site matches an origin pattern in\nany of the following policies:\nScreenCaptureAllowedByOrigins,\nWindowCaptureAllowedByOrigins,\nTabCaptureAllowedByOrigins,\nSameOriginTabCaptureAllowedByOrigins.\n";

ScreenCaptureAllowedByOrigins.pfm\_title = "Allow Desktop, Window, and Tab capture by these origins";

ScreenCaptureAllowedByOrigins.pfm\_description = "Setting the policy lets you set a list of URL patterns that can use Desktop, Window, and Tab Capture.\n\nLeaving the policy unset means that sites will not be considered for an override at this level of Capture.\n\nThis policy is not considered if a site matches a URL pattern in any of the following policies: WindowCaptureAllowedByOrigins, TabCaptureAllowedByOrigins, SameOriginTabCaptureAllowedByOrigins.\n\nIf a site matches a URL pattern in this policy, the ScreenCaptureAllowed will not be considered.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. This policy only matches based on origin, so any path in the URL pattern is ignored.";

ScrollToTextFragmentEnabled.pfm\_title = "Enable scrolling to text specified in URL fragments";

ScrollToTextFragmentEnabled.pfm\_description = "This feature allows for hyperlinks and address bar URL navigations to target specific text within a web page, which will be scrolled to once the loading of the web page is complete.\n\nIf you enable or don't configure this policy, web page scrolling to specific text fragments via URL will be enabled.\n\nIf you disable this policy, web page scrolling to specific text fragments via URL will be disabled.";

SearchSuggestEnabled.pfm\_title = "Enable search suggestions";

SearchSuggestEnabled.pfm\_description = "Setting the policy to True turns on search suggestions in Google Chrome's address bar. Setting the policy to False turns off these search suggestions.\n\nSuggestions based on bookmarks or history are unaffected by the policy.\n\nIf you set the policy, users can't change it. If not set, search suggestions are on at first, but users can turn them off any time.";

SecurityKeyPermitAttestation.pfm\_title = "URLs/domains automatically permitted direct Security Key attestation";

SecurityKeyPermitAttestation.pfm\_description = "Setting the policy specifies URLs and domains for which no prompt appears when attestation certificates from Security Keys are requested. A signal is also sent to the Security Key indicating that individual attestation may be used. Without this, when sites request attestation of Security Keys, users are prompted in Google Chrome version 65 and later.\n\nURLs will only match as U2F appIDs. Domains only match as webauthn RP IDs. So to cover both U2F and webauthn APIs, list the appID URL and domain for a given site.";

SensorsAllowedForUrls.pfm\_title = "Allow access to sensors on these sites";

SensorsAllowedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify the sites that can access sensors like motion and light sensors.\n\nLeaving the policy unset means DefaultSensorsSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nIf the same URL pattern exists in both this policy and the SensorsBlockedForUrls policy, the latter is prioritized and access to motion or light sensors will be blocked.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

SensorsBlockedForUrls.pfm\_title = "Block access to sensors on these sites";

SensorsBlockedForUrls.pfm\_description = "Setting the policy lets you set a list of URL patterns that specify the sites that can't access sensors like motion and light sensors.\n\nLeaving the policy unset means DefaultSensorsSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nIf the same URL pattern exists in both this policy and the SensorsAllowedForUrls policy, this policy is prioritized and access to motion or light sensors will be blocked.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

SerialAllowAllPortsForUrls.pfm\_title = "Automatically grant permission to sites to connect all serial ports.";

SerialAllowAllPortsForUrls.pfm\_description = "Setting the policy allows you to list sites which are automatically granted permission to access all available serial ports.\n\nThe URLs must be valid, otherwise the policy is ignored. Only the origin (scheme, host and port) of the URL is considered.\n\nOn Google Chrome OS, this policy only applies to affiliated users.\n\nThis policy overrides DefaultSerialGuardSetting, SerialAskForUrls, SerialBlockedForUrls and the user's preferences.";

SerialAllowUsbDevicesForUrls.pfm\_title = "Automatically grant permission to sites to connect to USB serial devices.";

SerialAllowUsbDevicesForUrls.pfm\_description = "Setting the policy allows you to list sites which are automatically granted permission to access USB serial devices with vendor and product IDs matching the vendor\_id and product\_id fields. Omitting the product\_id field allows the given sites permission to access devices with a vendor ID matching the vendor\_id field and any product ID.\n\nThe URLs must be valid, otherwise the policy is ignored. Only the origin (scheme, host and port) of the URL is considered.\n\nOn ChromeOS, this policy only applies to affiliated users.\n\nThis policy overrides DefaultSerialGuardSetting, SerialAskForUrls, SerialBlockedForUrls and the user's preferences.\n\nThis policy only affects access to USB devices through the Web Serial API. To grant access to USB devices through the WebUSB API see the WebUsbAllowDevicesForUrls policy.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=SerialAllowUsbDevicesForUrls for more information about schema and formatting.";

SerialAskForUrls.pfm\_title = "Allow the Serial API on these sites";

SerialAskForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can ask users to grant them access to a serial port.\n\nLeaving the policy unset means DefaultSerialGuardSetting applies for all sites, if it's set. If not, users' personal settings apply.\n\nFor URL patterns which do not match the policy SerialBlockedForUrls (if there is a match), DefaultSerialGuardSetting (if set), or the users' personal settings take precedence, in that order.\n\nURL patterns must not conflict with SerialBlockedForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

SerialBlockedForUrls.pfm\_title = "Block the Serial API on these sites";

SerialBlockedForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can't ask users to grant them access to a serial port.\n\nLeaving the policy unset means DefaultSerialGuardSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nFor URL patterns which do not match the policy SerialAskForUrls (if there is a match), DefaultSerialGuardSetting (if set), or the users' personal settings take precedence, in that order.\n\nURL patterns can't conflict with SerialAskForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

SetTimeoutWithout1MsClampEnabled.pfm\_title = "Control Javascript setTimeout() function minimum timeout.";

SetTimeoutWithout1MsClampEnabled.pfm\_description = "When the policy is set to Enabled, the Javascript setTimeout() with a timeout of 0ms will not clamp to 1ms.\nWhen the policy is set to Disabled, the Javascript setTimeout() with a timeout of 0ms will clamp to 1ms.\nWhen the policy is unset, use the browser's default behavior for setTimeout() function clamp.\n\nThis is a web standards compliant feature, but it may change task ordering\non a web page, leading to unexpected behavior on sites that are dependent on\na certain ordering in some way. It also may affect sites with a lot of setTimeout()\nwith a timeout of 0ms usage, e.g. increasing CPU load.\n\nFor users where this policy is unset, Chrome will roll out the change gradually on the stable channel.\n\nThis is a temporary policy that is planned be removed in Chrome 105. This deadline may be extended if there is a need for it among enterprises.\n";

SharedArrayBufferUnrestrictedAccessAllowed.pfm\_title = "Specifies whether SharedArrayBuffers can be used in a non cross-origin-isolated context";

SharedArrayBufferUnrestrictedAccessAllowed.pfm\_description = "\nSpecifies whether SharedArrayBuffers can be used in a non cross-origin-isolated context. Google Chrome will require cross-origin isolation when using SharedArrayBuffers from Google Chrome 91 onward (2021-05-25) for Web Compatibility reasons. Additional details can be found on: https://developer.chrome.com/blog/enabling-shared-array-buffer/.\n\nWhen set to Enabled, sites can use SharedArrayBuffer with no restrictions.\n\nWhen set to Disabled or not set, sites can only use SharedArrayBuffers when cross-origin isolated.";

SharedClipboardEnabled.pfm\_title = "Enable the Shared Clipboard Feature";

SharedClipboardEnabled.pfm\_description = "Enable the Shared Clipboard feature which allows users to send text between Chrome Desktops and an Android device when Sync is enabled and the user is Signed-in.\n\nIf this policy is set to true, the capability of sending text, cross device, for chrome user is enabled.\n\nIf this policy is set to false, the capability of sending text, cross device, for chrome user is disabled.\n\nIf you set this policy, users cannot change or override it.\n\nIf this policy is left unset, the shared clipboard feature is enabled by default.\n\nIt is up to the admins to set policies in all platforms they care about. It's recommended to set this policy to one value in all platforms.";

ShowAppsShortcutInBookmarkBar.pfm\_title = "Show the apps shortcut in the bookmark bar";

ShowAppsShortcutInBookmarkBar.pfm\_description = "Setting the policy to True displays the apps shortcut. Setting the policy to False means this shortcut never appears.\n\nIf you set the policy, users can't change it. If not set, users decide to show or hide the apps shortcut from the bookmark bar context menu.";

ShowCastIconInToolbar.pfm\_title = "Show the Google Cast toolbar icon";

ShowCastIconInToolbar.pfm\_description = "Setting the policy to Enabled displays the Cast toolbar icon on the toolbar or the overflow menu, and users can't remove it.\n\nSetting the policy to Disabled or leaving it unset lets users pin or remove the icon through its contextual menu.\n\nIf the policy EnableMediaRouter is set to Disabled, then this policy's value has no effect, and the toolbar icon doesn't appear.";

ShowFullUrlsInAddressBar.pfm\_title = "Show Full URLs";

ShowFullUrlsInAddressBar.pfm\_description = "This feature enables display of the full URL in the address bar.\nIf this policy is set to True, then the full URL will be shown in the address bar, including schemes and subdomains.\nIf this policy is set to False, then the default URL display will apply.\nIf this policy is left unset, then the default URL display will apply and the user will be able to toggle between default and full URL display with a context menu option.\n";

ShowHomeButton.pfm\_title = "Show Home button on toolbar";

ShowHomeButton.pfm\_description = "Setting the policy to Enabled shows the Home button on Google Chrome's toolbar. Setting the policy to Disabled keeps the Home button from appearing.\n\nIf you set the policy, users can't change it in Google Chrome. If not set, users chooses whether to show the Home button.";

SideSearchEnabled.pfm\_title = "Allow showing the most recent default search engine results page in a Browser side panel";

SideSearchEnabled.pfm\_description = "Setting the policy to Enabled or leaving the policy unset means that users can bring up their most recent default search engine results page in a side panel via toggling an icon in the toolbar.\n\nSetting the policy to Disabled removes the icon from the toolbar that opens the side panel with the default search engine results page.";

SignedHTTPExchangeEnabled.pfm\_title = "Enable Signed HTTP Exchange (SXG) support";

SignedHTTPExchangeEnabled.pfm\_description = "Setting the policy to True or leaving it unset means Google Chrome will accept web contents served as Signed HTTP Exchanges.\n\nSetting the policy to False prevents Signed HTTP Exchanges from loading.";

SigninInterceptionEnabled.pfm\_title = "Enable signin interception";

SigninInterceptionEnabled.pfm\_description = "This settings enables or disables signin interception.\n\nWhen this policy not set or is enabled, the signin interception dialog triggers when a Google account is added on the web, and the user may benefit from moving this account to another (new or existing) profile.\n\nWhen this is disabled, the signin interception dialog does not trigger.\nWhen this is disabled, a dialog will still be shown if managed account profile separation is enforced by ManagedAccountsSigninRestriction.";

SitePerProcess.pfm\_title = "Require Site Isolation for every site";

SitePerProcess.pfm\_description = "Since Google Chrome 67, site isolation has been enabled by default on all Desktop platforms, causing every site to run in its own process. A site is a scheme plus eTLD+1 (e.g., https://example.com). Setting this policy to Enabled does not change that behavior; it only prevents users from opting out (for example, using Disable site isolation in chrome://flags). Since Google Chrome 76, setting the policy to Disabled or leaving it unset doesn't turn off site isolation, but instead allows users to opt out.\n\nIsolateOrigins might also be useful for isolating specific origins at a finer granularity than site (e.g., https://a.example.com).\n\nOn Google Chrome OS version 76 and earlier, set the DeviceLoginScreenSitePerProcess device policy to the same value. (If the values don't match, a delay can occur when entering a user session.)\n\nNote: For Android, use the SitePerProcessAndroid policy instead.";

SpellCheckServiceEnabled.pfm\_title = "Enable or disable spell checking web service";

SpellCheckServiceEnabled.pfm\_description = "Setting the policy to Enabled puts a Google web service in use to help resolve spelling errors. This policy only controls the use of the online service. Setting the policy to Disabled means this service is never used.\n\nLeaving the policy unset lets users choose whether to use the spellcheck service.\n\nThe spell check can always use a downloaded dictionary locally unless the feature is disabled by SpellcheckEnabled in which case this policy will have no effect.";

SpellcheckEnabled.pfm\_title = "Enable spellcheck";

SpellcheckEnabled.pfm\_description = "Setting the policy to Enabled turns spellcheck on, and users can't turn it off. On Microsoft® Windows®, Google Chrome OS and Linux®, spellcheck languages can be switched on or off individually, so users can still turn spellcheck off by switching off every spellcheck language. To avoid that, use the SpellcheckLanguage to force-enable specific spellcheck languages.\n\nSetting the policy to Disabled turns off spellcheck from all sources, and users can't turn it on. The SpellCheckServiceEnabled, SpellcheckLanguage and SpellcheckLanguageBlocklist policies have no effect when this policy is set to False.\n\nLeaving the policy unset lets users turn spellcheck on or off in the language settings.";

SuppressDifferentOriginSubframeDialogs.pfm\_title = "Suppress JavaScript Dialogs triggered from different origin subframes";

SuppressDifferentOriginSubframeDialogs.pfm\_description = "As described in https://www.chromestatus.com/feature/5148698084376576 , JavaScript modal dialogs, triggered by window.alert, window.confirm, and window.prompt, will be blocked in Google Chrome if triggered from a subframe whose origin is different from the main frame origin.\nThis policy allows overriding that change.\nIf the policy is set to enabled or unset, JavaScript dialogs triggered from a different origin subframe will be blocked.\nIf the policy is set to disabled, JavaScript dialogs triggered from a different origin subframe will not be blocked.\n\nThis policy will be removed in Google Chrome version 95.";

SuppressUnsupportedOSWarning.pfm\_title = "Suppress the unsupported OS warning";

SuppressUnsupportedOSWarning.pfm\_description = "Setting the policy to Enabled suppresses the warning that appears when Google Chrome is running on an unsupported computer or operating system.\n\nSetting the policy to Disabled or leaving it unset means the warnings appear on unsupported systems.";

SyncDisabled.pfm\_title = "Disable synchronization of data with Google";

SyncDisabled.pfm\_description = "Setting the policy to Enabled turns off data synchronization in Google Chrome using Google-hosted synchronization services.\nTo fully turn off Chrome Sync services, we recommend that you turn off the service in the Google Admin console.\n\nIf the policy is set to Disabled or not set, users are allowed to choose whether to use Chrome Sync.\n\nNote: Do not turn on this policy when RoamingProfileSupportEnabled is Enabled, because that feature shares the same client-side functionality. The Google-hosted synchronization is off completely in this case.";

SyncTypesListDisabled.pfm\_title = "List of types that should be excluded from synchronization";

SyncTypesListDisabled.pfm\_description = "If this policy is set all specified data types will be excluded from synchronization both for Chrome Sync as well as for roaming profile synchronization. This can be beneficial to reduce the size of the roaming profile or limit the type of data uploaded to the Chrome Sync Servers.\n\nThe current data types for this policy are: \"bookmarks\", \"readingList\", \"preferences\", \"passwords\", \"autofill\", \"themes\", \"typedUrls\", \"extensions\", \"apps\", \"tabs\", \"wifiConfigurations\". Those names are case sensitive!";

TabCaptureAllowedByOrigins.pfm\_title = "Allow Tab capture by these origins";

TabCaptureAllowedByOrigins.pfm\_description = "Setting the policy lets you set a list of URL patterns that can use Tab Capture.\n\nLeaving the policy unset means that sites will not be considered for an override at this level of capture.\n\nNote that windowed Chrome Apps will still be allowed to be captured.\n\nThis policy is not considered if a site matches a URL pattern in the SameOriginTabCaptureAllowedByOrigins policy.\n\nIf a site matches a URL pattern in this policy, the following policies will not be considered: WindowCaptureAllowedByOrigins, ScreenCaptureAllowedByOrigins, ScreenCaptureAllowed.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. This policy only matches based on origin, so any path in the URL pattern is ignored.";

TaskManagerEndProcessEnabled.pfm\_title = "Enable ending processes in Task Manager";

TaskManagerEndProcessEnabled.pfm\_description = "Setting the policy to Disabled prevents users from ending processes in the Task Manager.\n\nSetting the policy to Enabled or leaving it unset lets users end processes in the Task Manager.";

TotalMemoryLimitMb.pfm\_title = "Set memory limit for Chrome instances";

TotalMemoryLimitMb.pfm\_description = "Configures the amount of memory that a single Google Chrome instance can use before tabs start being discarded (I.E. the memory used by the tab will be freed and the tab will have to be reloaded when switched to) to save memory.\n\nIf the policy is set, browser will begin to discard tabs to save memory once the limitation is exceeded. However, there is no guarantee that the browser is always running under the limit. Any value under 1024 will be rounded up to 1024.\n\nIf this policy is not set, the browser will only begin attempts to save memory once it has detected that the amount of physical memory on its machine is low.";

TranslateEnabled.pfm\_title = "Enable Translate";

TranslateEnabled.pfm\_description = "Setting the policy to True provides translation functionality when it's appropriate for users by showing an integrated translate toolbar in Google Chrome and a translate option on the right-click context menu. Setting the policy to False shuts off all built-in translate features.\n\nIf you set the policy, users can't change this function. Leaving it unset lets them change the setting.";

U2fSecurityKeyApiEnabled.pfm\_title = "Allow using the deprecated U2F Security Key API";

U2fSecurityKeyApiEnabled.pfm\_description = "If set to Enabled, the deprecated U2F Security Key API can be used and the deprecation reminder prompt shown for U2F API requests is suppressed.\n\nIf the policy is set to Disabled or left unset, the default behavior will apply.\n\nThe U2F Security Key API is deprecated and it will be disabled by default in Chrome 98.\n\nThis is a temporary opt-out mechanism. The U2F API will be removed from Chrome in Chrome 104, at which point this policy will cease to be supported.\n\nFor more information about the deprecation of the U2F Security Key API, please refer to https://groups.google.com/a/chromium.org/g/blink-dev/c/xHC3AtU\_65A.";

URLAllowlist.pfm\_title = "Allow access to a list of URLs";

URLAllowlist.pfm\_description = "Setting the policy provides access to the listed URLs, as exceptions to URLBlocklist. See that policy's description for the format of entries of this list. For example, setting URLBlocklist to \* will block all requests, and you can use this policy to allow access to a limited list of URLs. Use it to open exceptions to certain schemes, subdomains of other domains, ports, or specific paths, using the format specified at ( https://www.chromium.org/administrators/url-blocklist-filter-format ). The most specific filter determines if a URL is blocked or allowed. The URLAllowlist policy takes precedence over URLBlocklist. This policy is limited to 1,000 entries.\n\nThis policy also allows enabling the automatic invocation by the browser of external application registered as protocol handlers for the listed protocols like \"tel:\" or \"ssh:\".\n\nLeaving the policy unset allows no exceptions to URLBlocklist.\n\nFrom Google Chrome version 92, this policy is also supported in the headless mode.\n\nOn Microsoft® Windows®, this functionality is only available on instances that are joined to a Microsoft® Active Directory® domain, running on Windows 10 Pro, or enrolled in Chrome Browser Cloud Management. On macOS, this functionality is only available on instances that are managed via MDM, or joined to a domain via MCX.";

URLBlocklist.pfm\_title = "Block access to a list of URLs";

URLBlocklist.pfm\_description = "Setting the policy prevents webpages with prohibited URLs from loading. It provides a list of URL patterns that specify forbidden URLs. Leaving the policy unset means no URLs are prohibited in the browser. Format the URL pattern according to this format ( https://www.chromium.org/administrators/url-blocklist-filter-format ). Up to 1,000 exceptions can be defined in URLAllowlist.\n\nFrom Google Chrome version 73, you can block javascript://\* URLs. However, it affects only JavaScript entered in the address bar (or, for example, bookmarklets). In-page JavaScript URLs with dynamically loaded data aren't subject to this policy. For example, if you block example.com/abc, then example.com can still load example.com/abc using XMLHTTPRequest.\n\nFrom Google Chrome version 92, this policy is also supported in the headless mode.\n\nNote: Blocking internal chrome://\* URLs can lead to unexpected errors.";

UrlKeyedAnonymizedDataCollectionEnabled.pfm\_title = "Enable URL-keyed anonymized data collection";

UrlKeyedAnonymizedDataCollectionEnabled.pfm\_description = "Setting the policy to Enabled means URL-keyed anonymized data collection, which sends URLs of pages the user visits to Google to make searches and browsing better, is always active.\n\nSetting the policy to Disabled results in no URL-keyed anonymized data collection.\n\nIf you set the policy, users can't change. If not set, then URL-keyed anonymized data collection at first, but users can change it.";

UrlParamFilterEnabled.pfm\_title = "Control the URL parameter filter feature";

UrlParamFilterEnabled.pfm\_description = "When enabled or not set, the URL parameter filter may remove some parameters when a user selects \"Open Link in Incognito Window\" from the context menu.\nWhen disabled, no filtering is performed.\nThis policy is temporary and may be removed in a future release.";

UserAgentClientHintsGREASEUpdateEnabled.pfm\_title = "Control the User-Agent Client Hints GREASE Update feature.";

UserAgentClientHintsGREASEUpdateEnabled.pfm\_description = "When enabled the User-Agent Client Hints GREASE Update feature aligns the User-Agent GREASE algorithm with the latest spec.\nThe updated spec may break some websites that restrict the characters that requests may contain. See the spec for more information: https://wicg.github.io/ua-client-hints/#grease\nIf this policy is enabled or not set, the browser will decide which User-Agent GREASE algorithm to use. If the policy is disabled the prior User-Agent GREASE algorithm is used.\nThis policy is a temporary measure and will be removed in a future release.";

UserAgentReduction.pfm\_title = "Enable or disable the User-Agent Reduction.";

UserAgentReduction.pfm\_description = "0 - User Agent reduction will be controllable via Field-Trials and Origin-Trials.\n1 - User Agent reduction disabled, and not enabled by Field-Trials or Origin-Trials.\n2 - User Agent reduction will be enabled for all origins.\nThe User-Agent HTTP request header is scheduled to be reduced. In order to facilitate testing and compatibility, this policy can enable the reduction feature for all websites, or disable the ability for origin trials or field trials to enable the feature.\n\nTo learn more about the User-Agent Reduction and its timeline, read here:\n\nhttps://blog.chromium.org/2021/09/user-agent-reduction-origin-trial-and-dates.html\n";

UserDataDir.pfm\_title = "Set user data directory";

UserDataDir.pfm\_description = "Configures the directory that Google Chrome will use for storing user data.\n\nIf you set this policy, Google Chrome will use the provided directory regardless whether the user has specified the '--user-data-dir' flag or not. To avoid data loss or other unexpected errors this policy should not be set to a directory used for other purposes, because Google Chrome manages its contents.\n\nSee https://support.google.com/chrome/a?p=Supported\_directory\_variables for a list of variables that can be used.\n\nIf this policy is left not set the default profile path will be used and the user will be able to override it with the '--user-data-dir' command line flag.";

UserDataSnapshotRetentionLimit.pfm\_title = "Limits the number of user data snapshots retained for use in case of emergency rollback.";

UserDataSnapshotRetentionLimit.pfm\_description = "Following each major version update, Chrome will create a snapshot of certain portions of the user's browsing data for use in case of a later emergency version rollback. If an emergency rollback is performed to a version for which a user has a corresponding snapshot, the data in the snapshot is restored. This allows users to retain such settings as bookmarks and autofill data.\n\nIf this policy is not set, the default value of 3 is used\n\nIf the policy is set, old snapshots are deleted as needed to respect the limit. If the policy is set to 0, no snapshots will be taken";

UserFeedbackAllowed.pfm\_title = "Allow user feedback";

UserFeedbackAllowed.pfm\_description = "Setting the policy to Enabled or leaving it unset lets users send feedback to Google through Menu > Help > Report an Issue or key combination.\n\nSetting the policy to Disabled means users can't send feedback to Google.";

VideoCaptureAllowed.pfm\_title = "Allow or deny video capture";

VideoCaptureAllowed.pfm\_description = "Setting the policy to Enabled or leaving it unset means that, with the exception of URLs set in the VideoCaptureAllowedUrls list, users get prompted for video capture access.\n\nSetting the policy to Disabled turns off prompts, and video capture is only available to URLs set in the VideoCaptureAllowedUrls list.\n\nNote: The policy affects all video input (not just the built-in camera).";

VideoCaptureAllowedUrls.pfm\_title = "URLs that will be granted access to video capture devices without prompt";

VideoCaptureAllowedUrls.pfm\_description = "Setting the policy means you specify the URL list whose patterns get matched to the security origin of the requesting URL. A match grants access to video capture devices without prompt\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns.";

WPADQuickCheckEnabled.pfm\_title = "Enable WPAD optimization";

WPADQuickCheckEnabled.pfm\_description = "Setting the policy to Enabled or leaving it unset turns on WPAD (Web Proxy Auto-Discovery) optimization in Google Chrome.\n\nSetting the policy to Disabled turns off WPAD optimization, causing Google Chrome to wait longer for DNS-based WPAD servers.\n\nWhether or not this policy is set, users can't change the WPAD optimization setting.";

WarnBeforeQuittingEnabled.pfm\_title = "Show a warning dialog when the user is attempting to quit";

WarnBeforeQuittingEnabled.pfm\_description = "Controls \"Warn Before Quitting (⌘Q)\" dialog when the user is attempting to quit browser.\n\nIf this policy is set to Enabled or not set, a warning dialog is shown when the user is attempting to quit.\n\nIf this policy is set to Disabled, a warning dialog is not shown when the user is attempting to quit.";

WebAppInstallForceList.pfm\_title = "URLs for Web Apps to be silently installed.";

WebAppInstallForceList.pfm\_description = "Setting the policy specifies a list of web apps that install silently, without user interaction, and which users can't uninstall or turn off.\n\nEach list item of the policy is an object with a mandatory member:\nurl (the URL of the web app to install)\n\nand 5 optional members:\n- default\_launch\_container\n(for how the web app opens—a new tab is the default)\n\n- create\_desktop\_shortcut\n(True if you want to create Linux and\nMicrosoft® Windows® desktop shortcuts).\n\n- fallback\_app\_name\n(Starting with Google Chrome version 90,\nallows you to override the app name if it is not a\nProgressive Web App (PWA), or the app name that is temporarily\ninstalled if it is a PWA but authentication is required before the\ninstallation can be completed. If both\ncustom\_name and\nfallback\_app\_name are provided,\nthe latter will be ignored.)\n\n- custom\_name\n(Starting with Google Chrome\nversion 99, allows you to permanently override the app name for all web\napps and PWAs. Currently only supported on\nGoogle Chrome OS.)\n\n- custom\_icon\n(Starting with Google Chrome\nversion 99, allows you to override the app icon of installed apps. The\nicons have to be square, maximal 1 MB in size, and in one of the following\nformats: jpeg, png, gif, webp, ico. The hash value has to be the SHA256\nhash of the icon file. Currently only supported on\nGoogle Chrome OS.)\n\nSee PinnedLauncherApps for pinning apps to the Google Chrome OS shelf.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=WebAppInstallForceList for more information about schema and formatting.";

WebAppSettings.pfm\_title = "Web App management settings";

WebAppSettings.pfm\_description = "This policy allows an admin to specify settings for installed web apps.\n\nThis policy maps a Web App ID to its specific setting. A default configuration can be set using the special ID \"\*\", which applies to all web apps without a custom configuration in this policy.\n\nThe \"manifest\_id\" field is the Manifest ID for the Web App. See https://developer.chrome.com/blog/pwa-manifest-id/ for instructions on how to determine the Manifest ID for an installed web app.\nThe \"run\_on\_os\_login\" field specifies if a web app can be run during OS login. If this field is set to \"blocked\", the web app will not run during OS login and the user will not be able to enable this later. If this field is set to \"run\_windowed\", the web app will run during OS login and the user will not be able to disable this later. If this field is set to \"allowed\", the user will be able to configure the web app to run at OS login. The default configuration only allows the \"allowed\" and \"blocked\" values.\n\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=WebAppSettings for more information about schema and formatting.";

WebHidAllowAllDevicesForUrls.pfm\_title = "Automatically grant permission to sites to connect to any HID device.";

WebHidAllowAllDevicesForUrls.pfm\_description = "Setting the policy allows you to list sites which are automatically granted permission to access all available devices.\n\nThe URLs must be valid, otherwise the policy is ignored. Only the origin (scheme, host and port) of the URL is considered.\n\nOn ChromeOS, this policy only applies to affiliated users.\n\nThis policy overrides DefaultWebHidGuardSetting, WebHidAskForUrls, WebHidBlockedForUrls and the user's preferences.";

WebHidAllowDevicesForUrls.pfm\_title = "Automatically grant permission to these sites to connect to HID devices with the given vendor and product IDs.";

WebHidAllowDevicesForUrls.pfm\_description = "Setting the policy lets you list the URLs that specify which sites are automatically granted permission to access a HID device with the given vendor and product IDs. Each item in the list requires both devices and urls fields for the item to be valid, otherwise the item is ignored. Each item in the devices field must have a vendor\_id and may have a product\_id field. Omitting the product\_id field will create a policy matching any device with the specified vendor ID. An item which has a product\_id field without a vendor\_id field is invalid and is ignored.\n\nLeaving the policy unset means DefaultWebHidGuardSetting applies, if it's set. If not, the user's personal setting applies.\n\nURLs in this policy shouldn't conflict with those configured through WebHidBlockedForUrls. If they do, this policy takes precedence over WebHidBlockedForUrls.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=WebHidAllowDevicesForUrls for more information about schema and formatting.";

WebHidAllowDevicesWithHidUsagesForUrls.pfm\_title = "Automatically grant permission to these sites to connect to HID devices containing top-level collections with the given HID usage.";

WebHidAllowDevicesWithHidUsagesForUrls.pfm\_description = "Setting the policy lets you list the URLs that specify which sites are automatically granted permission to access a HID device containing a top-level collection with the given HID usage. Each item in the list requires both usages and urls fields for the policy to be valid. Each item in the usages field must have a usage\_page and may have a usage field. Omitting the usage field will create a policy matching any device containing a top-level collection with a usage from the specified usage page. An item which has a usage field without a usage\_page field is invalid and is ignored.\n\nLeaving the policy unset means DefaultWebHidGuardSetting applies, if it's set. If not, the user's personal setting applies.\n\nURLs in this policy shouldn't conflict with those configured through WebHidBlockedForUrls. If they do, this policy takes precedence over WebHidBlockedForUrls.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=WebHidAllowDevicesWithHidUsagesForUrls for more information about schema and formatting.";

WebHidAskForUrls.pfm\_title = "Allow the WebHID API on these sites";

WebHidAskForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can ask users to grant them access to a HID device.\n\nLeaving the policy unset means DefaultWebHidGuardSetting applies for all sites, if it's set. If not, users' personal settings apply.\n\nFor URL patterns which do not match the policy, the following take precedence, in this order:\n\n \* WebHidBlockedForUrls (if there is a match),\n\n \* DefaultWebHidGuardSetting (if set), or\n\n \* Users' personal settings.\n\nURL patterns must not conflict with WebHidBlockedForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

WebHidBlockedForUrls.pfm\_title = "Block the WebHID API on these sites";

WebHidBlockedForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can't ask users to grant them access to a HID device.\n\nLeaving the policy unset means DefaultWebHidGuardSetting applies for all sites, if it's set. If not, users' personal settings apply.\n\nFor URL patterns which do not match the policy, the following take precedence, in this order:\n\n \* WebHidAskForUrls (if there is a match),\n\n \* DefaultWebHidGuardSetting (if set), or\n\n \* Users' personal settings.\n\nURL patterns can't conflict with WebHidAskForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

WebRtcAllowLegacyTLSProtocols.pfm\_title = "Allow legacy TLS/DTLS downgrade in WebRTC";

WebRtcAllowLegacyTLSProtocols.pfm\_description = "If enabled, WebRTC peer connections can downgrade to obsolete\nversions of the TLS/DTLS (DTLS 1.0, TLS 1.0 and TLS 1.1) protocols.\nWhen this policy is disabled or not set, these TLS/DTLS versions are\ndisabled.\n\nThis policy is temporary and will be removed in a future version\nof Google Chrome.";

WebRtcEventLogCollectionAllowed.pfm\_title = "Allow collection of WebRTC event logs from Google services";

WebRtcEventLogCollectionAllowed.pfm\_description = "Setting the policy to Enabled means Google Chrome can collect WebRTC event logs from Google services such as Hangouts Meet and upload them to Google. These logs have diagnostic information for debugging issues with audio or video meetings in Google Chrome, such as the time and size of RTP packets, feedback about congestion on the network, and metadata about time and quality of audio and video frames. These logs have no audio or video content from the meeting. To make debugging easier, Google might associate these logs, by means of a session ID, with other logs collected by the Google service itself.\n\nSetting the policy to Disabled results in no collection or uploading of such logs.\n\nLeaving the policy unset on versions up to and including M76 means Google Chrome defaults to not being able to collect and upload these logs. Starting at M77, Google Chrome defaults to being able to collect and upload these logs from most profiles affected by cloud-based, user-level enterprise policies. From M77 up to and including M80, Google Chrome can also collect and upload these logs by default from profiles affected by Google Chrome on-premise management.";

WebRtcIPHandling.pfm\_title = "The IP handling policy of WebRTC";

WebRtcIPHandling.pfm\_description = "default - WebRTC will use all available interfaces when searching for the best path.\ndefault\_public\_and\_private\_interfaces - WebRTC will only use the interface connecting to the public Internet, but may connect using private IP addresses.\ndefault\_public\_interface\_only - WebRTC will only use the interface connecting to the public Internet, and will not connect using private IP addresses.\ndisable\_non\_proxied\_udp - WebRTC will use TCP on the public-facing interface, and will only use UDP if supported by a configured proxy.\nThis policy allows restricting which IP addresses and interfaces WebRTC uses when attempting to find the best available connection. See RFC 8828 section 5.2 (https://tools.ietf.org/html/rfc8828.html#section-5.2). When unset, defaults to using all available interfaces.";

WebRtcLocalIpsAllowedUrls.pfm\_title = "URLs for which local IPs are exposed in WebRTC ICE candidates";

WebRtcLocalIpsAllowedUrls.pfm\_description = "Patterns in this list will be matched against the security origin of the requesting URL.\nIf a match is found or chrome://flags/#enable-webrtc-hide-local-ips-with-mdns is Disabled, the local IP addresses are shown in WebRTC ICE candidates.\nOtherwise, local IP addresses are concealed with mDNS hostnames.\nPlease note that this policy weakens the protection of local IPs if needed by administrators.";

WebRtcUdpPortRange.pfm\_title = "Restrict the range of local UDP ports used by WebRTC";

WebRtcUdpPortRange.pfm\_description = "If the policy is set, the UDP port range used by WebRTC is restricted to the specified port interval (endpoints included).\n\nIf the policy is not set, or if it is set to the empty string or an invalid port range, WebRTC is allowed to use any available local UDP port.";

WebSQLAccess.pfm\_title = "Force WebSQL to be enabled.";

WebSQLAccess.pfm\_description = "WebSQL is on by default as of M101, but can be disabled via Chrome flag.\nIf this policy is set to false or unset, WebSQL can be disabled.\nIf this policy is set to true, WebSQL cannot be disabled.";

WebUsbAllowDevicesForUrls.pfm\_title = "Automatically grant permission to these sites to connect to USB devices with the given vendor and product IDs.";

WebUsbAllowDevicesForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites are automatically granted permission to access a USB device with the given vendor and product IDs. Each item in the list requires both devices and urls fields for the policy to be valid. Each item in the devices field can have a vendor\_id and product\_id field. Omitting the vendor\_id field will create a policy matching any device. Omitting the product\_id field will create a policy matching any device with the given vendor ID. A policy which has a product\_id field without a vendor\_id field is invalid.\n\nThe USB permission model will grant the specified URL permission to access the USB device as a top-level origin. If embedded frames need to access USB devices, the 'usb' feature-policy header should be used to grant access. The URL must be valid, otherwise the policy is ignored.\n\nDeprecated: The USB permission model used to support specifying both the requesting and embedding URLs. This is deprecated and only supported for backwards compatiblity in this manner: if both a requesting and embedding URL is specified, then the embedding URL will be granted the permission as top-level origin and the requsting URL will be ignored entirely.\n\nThis policy overrides DefaultWebUsbGuardSetting, WebUsbAskForUrls, WebUsbBlockedForUrls and the user's preferences.\n\nThis policy only affects access to USB devices through the WebUSB API. To grant access to USB devices through the Web Serial API see the SerialAllowUsbDevicesForUrls policy.\nSee https://cloud.google.com/docs/chrome-enterprise/policies/?policy=WebUsbAllowDevicesForUrls for more information about schema and formatting.";

WebUsbAskForUrls.pfm\_title = "Allow WebUSB on these sites";

WebUsbAskForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can ask users to grant them access to a USB device.\n\nLeaving the policy unset means DefaultWebUsbGuardSetting applies for all sites, if it's set. If not, users' personal settings apply.\n\nURL patterns must not conflict with WebUsbAskForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

WebUsbBlockedForUrls.pfm\_title = "Block WebUSB on these sites";

WebUsbBlockedForUrls.pfm\_description = "Setting the policy lets you list the URL patterns that specify which sites can't ask users to grant them access to a USB device.\n\nLeaving the policy unset means DefaultWebUsbGuardSetting applies for all sites, if it's set. If not, the user's personal setting applies.\n\nURL patterns can't conflict with WebUsbAskForUrls. Neither policy takes precedence if a URL matches with both.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. \* is not an accepted value for this policy.";

WindowCaptureAllowedByOrigins.pfm\_title = "Allow Window and Tab capture by these origins";

WindowCaptureAllowedByOrigins.pfm\_description = "Setting the policy lets you set a list of URL patterns that can use Window and Tab Capture.\n\nLeaving the policy unset means that sites will not be considered for an override at this level of Capture.\n\nThis policy is not considered if a site matches a URL pattern in any of the following policies: TabCaptureAllowedByOrigins, SameOriginTabCaptureAllowedByOrigins.\n\nIf a site matches a URL pattern in this policy, the following policies will not be considered: ScreenCaptureAllowedByOrigins, ScreenCaptureAllowed.\n\nFor detailed information on valid url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. This policy only matches based on origin, so any path in the URL pattern is ignored.";

WindowPlacementAllowedForUrls.pfm\_title = "Allow Window Placement permission on these sites";

WindowPlacementAllowedForUrls.pfm\_description = "Allows you to set a list of site url patterns that specify sites which will automatically grant the window placement permission. This will extend the ability of sites to see information about the device's screens and use that information to open and place windows or request fullscreen on specific screens.\n\nFor detailed information on valid site url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. Wildcards, \*, are allowed. This policy only matches based on origin, so any path in the URL pattern is ignored.\n\nIf this policy is not set for a site then the policy from DefaultWindowPlacementSetting applies to the site, if set, otherwise the permission will follow the browser's defaults and allow users to choose this permission per site.";

WindowPlacementBlockedForUrls.pfm\_title = "Block Window Placement permission on these sites";

WindowPlacementBlockedForUrls.pfm\_description = "Allows you to set a list of site url patterns that specify sites which will automatically deny the window placement permission. This will limit the ability of sites to see information about the device's screens and use that information to open and place windows or request fullscreen on specific screens.\n\nFor detailed information on valid site url patterns, please see https://cloud.google.com/docs/chrome-enterprise/policies/url-patterns. Wildcards, \*, are allowed. This policy only matches based on origin, so any path in the URL pattern is ignored.\n\nIf this policy is not set for a site then the policy from DefaultWindowPlacementSetting applies to the site, if set, otherwise the permission will follow the browser's defaults and allow users to choose this permission per site.";