

Prepare Mozilla Firefox for Enterprise Deployment and Virtualization

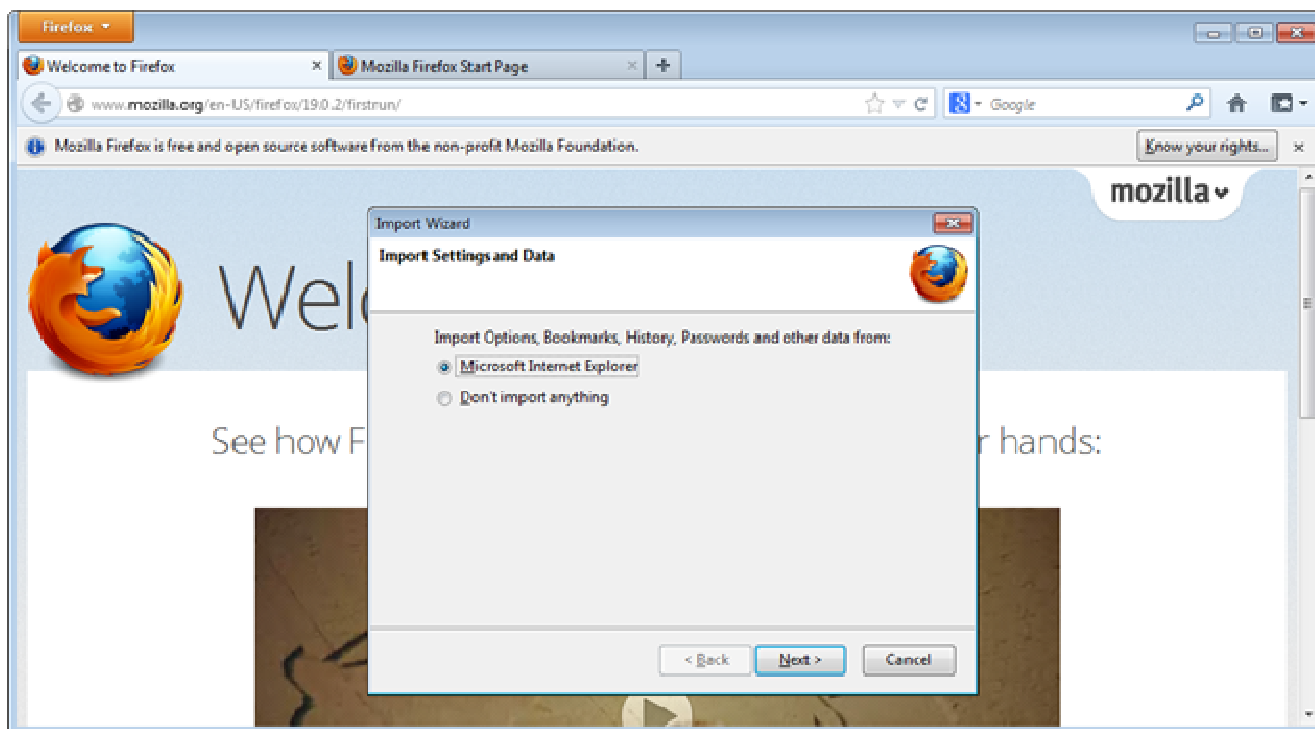
<http://stealthpuppy.com/prepare-mozilla-firefox-for-enterprise-deployment-and-virtualization/>

I've previously written articles on virtualizing Mozilla Firefox, but with Firefox releases more regular these days and the release of App-V 5, it makes sense to split details on configuring Firefox for an enterprise deployment and virtualizing Firefox into separate articles.

Whilst this article will cover some recommendations for configuring a Firefox deployment in an enterprise that can be used when virtualizing Firefox with various solutions, including App-V.

Features to control in an enterprise

There are a number of features that might be considered for disabling or configuring in an enterprise environment where users generally won't have administrative rights and IT may want to control the default user experience.



These might include:

- Import Wizard – Firefox starts a wizard on first run that imports settings from other browsers. You may want to remove this wizard to simplify the startup of Firefox
- Automatic updates for Firefox – *Options / Advanced / Update / Firefox updates*. Firefox updates should be delivered via new App-V packages. Updates for Add-ons and Search Engines should be OK as these are written to the user profile
- *Mozilla Maintenance Service* – Firefox installs an updater service that allows updating Firefox automatically, whilst avoiding UAC prompts. This service should be disabled or not installed in a controlled environment including under application virtualization
- Default browser check – *Options / Advanced / General – Always check to see if Firefox is the default browser on startup*. Under App-V 4,x, once Firefox is isolated from the OS, the user won't be able to make it the default browser (this will work under App-V 5)
- The 'Welcome to Firefox' tab, the 'Know your rights' and 'Improve Firefox' notifications

There is a way to control many of these settings through Group Policy, but if we get these right at install time, there's no need for the overhead of GPOs. I will cover using a few of these customisations to ensure these features are disabled for any new Firefox profile.

Locking down and controlling Firefox options

Firefox can be configured with defaults for any new profile and locked down so that it will contain your required configuration options. Mozilla has made it easy to deploy custom default settings and preferences – by adding some specific files at install time (assuming the default install location):

- %ProgramFiles%\Mozilla Firefox\browser\defaults\preferences\local-settings.js
- %ProgramFiles%\Mozilla Firefox\Mozilla.cfg
- %ProgramFiles%\Mozilla Firefox\browser\override.ini
- %ProgramFiles%\Mozilla Firefox\browser\defaults\profile\chrome\userChrome.css

Firefox itself and any new user profile will be configured with desired preferences and locked down. You can find more detailed documentation on these features in the following articles:

- local-settings.js and Mozilla.cfg (Locking preferences)
- override.ini and Firefox command line options
- UserChrome.css and Chrome element names and IDs
- Enterprise Build Of Firefox For Deployment

To enforce user settings we can leverage the ability to lock Firefox preferences and use UserChrome.css to remove the associated user interface elements.

Local-settings.js

Local-settings.js is used to tell Firefox to read Mozilla.cfg for some configuration items. Add the following lines to local-settings.js:

```
pref("general.config.obscure_value", 0);  
pref("general.config.filename", "mozilla.cfg");
```

Mozilla.cfg

Here's is where we can lock specified Firefox preferences. In the listing below, we've disabled the auto-update feature, the 'Welcome to Firefox' tab, the 'Know your rights' and 'Improve Firefox' notifications. The last (highlighted) line will disable the ability to set the browser as default.

This is useful where you would like to restrict this functionality and is applicable to App-V 4.x environments where attempting to set a virtualized Firefox as default won't work. Under App-V 5, remove this line so that users can set Firefox as the default browser if they wish.

```
lockPref("app.update.auto", false);  
lockPref("app.update.enabled", false);  
lockPref("app.update.service.enabled", false);  
lockPref("toolkit.telemetry.prompted", true);  
lockPref("browser.rights.override", true);  
lockPref("browser.startup.homepage_override.mstone", "ignore");  
lockPref("browser.shell.checkDefaultBrowser", false);
```

Override.ini

To disable the browser Import Wizard on first run, place the following lines into override.ini

```
[XRE]  
EnableProfileMigrator=false
```

userChrome.css

Mozilla has made it a fairly straight forward process to remove browser user interface elements using userChrome.css. Where browser functionality has been disabled, we can remove the corresponding UI to avoid user confusion. Enter the following lines into userChrome.css; however remove the highlighted line if you would like users to be able to set Firefox as the default browser:

```

/* UserChrome.css for Mozilla Firefox */
/* Remove access to user interface elements that aren't suitable for application virtualization */

/* Options - Advanced - General - System Defaults */
#systemDefaultsGroup { display: none !important; }

/* Options / Advanced / Update / Firefox updates group box */
#updateApp { display: none !important; }

/* Help - About - Check for Updates button */
#updateButton { display: none !important; }

```

Installing Firefox

Download the Firefox installer in your target language from the Mozilla site. For most deployments the installer won't require modification and installation can be automated by passing an INI file with setup configuration details to the installer. This enables you to control setup and set options such as preventing the desktop shortcut from being added, or control the target directory that Firefox is installed to (useful when virtualizing).

An installation script for Firefox should perform the following tasks:

- Install Mozilla Firefox with installer command line arguments – use the INI file approach to control for finer control over the installation
- After installing Firefox, copy the configuration files to the correct locations
- Firefox also allows you to add global add-ons by adding them to the Extensions sub-folder of the Firefox installation folder
- Optionally set Mozilla Maintenance service to disabled, if it is installed

For an example script that will automate the install and configuration of Firefox using the recommendations outlined in this article, see the script below. Note the highlighted line, where I can change the target installation directory for Firefox:

```

@ECHO OFF
SET SOURCE=%~dp0
SET SOURCE=%SOURCE:~0,-1%
SET INSTALLPATH=%ProgramFiles%\Mozilla Firefox

REM Create the Firefox answer file
ECHO [Install] > %SOURCE%\Firefox.ini
REM   InstallDirectoryName=Firefox >> "%SOURCE%\Firefox.ini"
ECHO InstallDirectoryPath=%INSTALLPATH% >> "%SOURCE%\Firefox.ini"
ECHO QuickLaunchShortcut=false >> "%SOURCE%\Firefox.ini"
ECHO DesktopShortcut=false >> "%SOURCE%\Firefox.ini"
ECHO StartMenuShortcuts=true >> "%SOURCE%\Firefox.ini"
ECHO MaintenanceService=false >> "%SOURCE%\Firefox.ini"

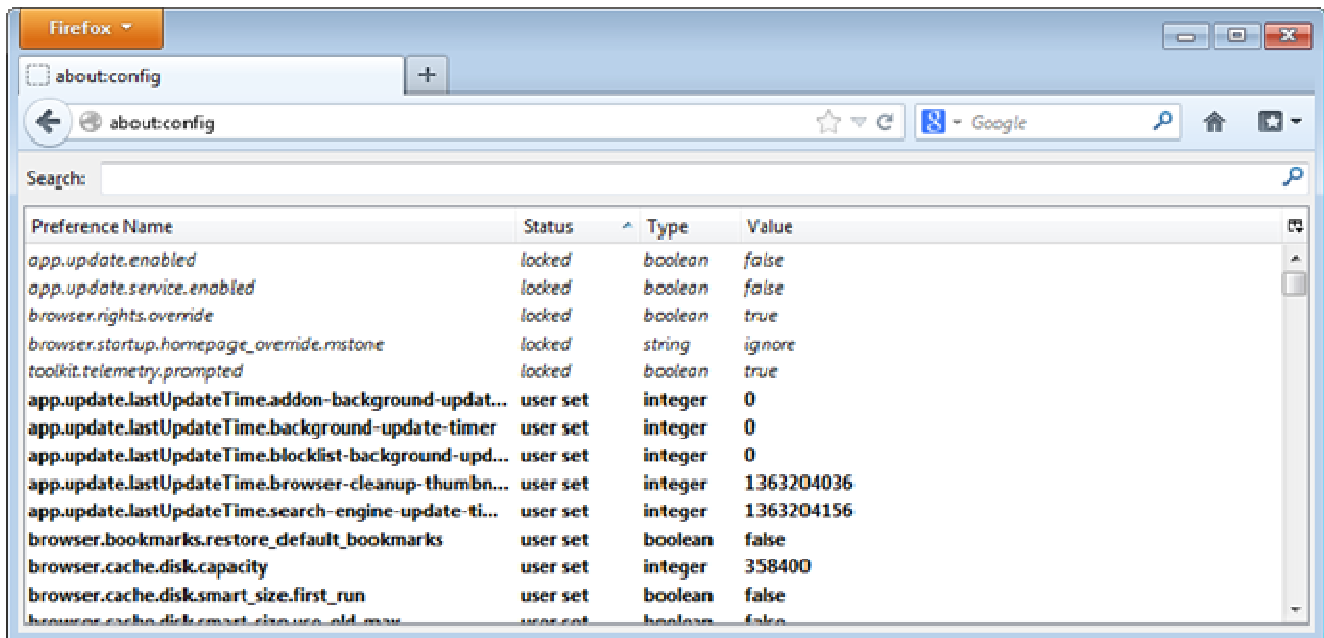
REM Install Firefox
START /WAIT "Firefox" /D "%SOURCE%" "FirefoxSetup21.exe" /INI="%SOURCE%\Firefox.ini"

REM Configure Firefox profile defaults and preferences locking
IF NOT EXIST "%INSTALLPATH%\browser\defaults\profile\chrome" MD
"%INSTALLPATH%\browser\defaults\profile\chrome"
COPY /Y %SOURCE%\userChrome.css "%INSTALLPATH%\browser\defaults\profile\chrome\userChrome.css"
IF NOT EXIST "%INSTALLPATH%\browser\defaults\preferences" MD
"%INSTALLPATH%\browser\defaults\preferences"
COPY /Y %SOURCE%\local-settings.js "%INSTALLPATH%\browser\defaults\preferences\local-settings.js"
COPY /Y %SOURCE%\Mozilla.cfg "%INSTALLPATH%\Mozilla.cfg"
COPY /Y %SOURCE%\override.ini "%INSTALLPATH%\browser\override.ini"

REM Disable the Mozilla Maintenance Service to prevent updates (in the event the service is
installed)
sc config MozillaMaintenance start= disabled

```

If the installation has been configured correctly, Firefox should start and not display the Import Wizard or any of the other prompts and start-up tabs. Additionally, if you navigate to about:config, a number of preferences should be listed as locked:



To make it easier, I've included the configuration files and the installation script listed above in a single ZIP file that you can download here - Mozilla Firefox Enterprise Install and Configure Scripts:

<http://stealthpuppy.com/wp-content/plugins/download-monitor/download.php?id=60>

InstallFirefox.cmd

```
@ECHO OFF
SET SOURCE=%~dp0
SET SOURCE=%SOURCE:~0,-1%
SET INSTALLPATH=%ProgramFiles%\Mozilla Firefox

REM Create the Firefox answer file
ECHO [Install] > %SOURCE%\Firefox.ini
REM InstallDirectoryPath=Firefox >> %SOURCE%\Firefox.ini
ECHO InstallDirectoryPath=%INSTALLPATH% >> %SOURCE%\Firefox.ini
ECHO QuickLaunchShortcut=false >> %SOURCE%\Firefox.ini
ECHO DesktopShortcut=false >> %SOURCE%\Firefox.ini
ECHO StartMenuShortcuts=true >> %SOURCE%\Firefox.ini
ECHO MaintenanceService=false >> %SOURCE%\Firefox.ini

REM Install Firefox - the START command will not work if the Firefox setup filename includes spaces
START /WAIT /D %SOURCE% FirefoxSetup21.exe /INI=%SOURCE%\Firefox.ini

REM Configure Firefox profile defaults and preferences locking
IF NOT EXIST "%INSTALLPATH%\browser\defaults\profile\chrome" MD
"%INSTALLPATH%\browser\defaults\profile\chrome"
COPY /Y %SOURCE%\userChrome.css "%INSTALLPATH%\browser\defaults\profile\chrome\userChrome.css"

IF NOT EXIST "%INSTALLPATH%\browser\defaults\preferences" MD
"%INSTALLPATH%\browser\defaults\preferences"
COPY /Y %SOURCE%\local-settings.js "%INSTALLPATH%\browser\defaults\preferences\local-settings.js"

COPY /Y %SOURCE%\Mozilla.cfg "%INSTALLPATH%\Mozilla.cfg"
COPY /Y %SOURCE%\override.ini "%INSTALLPATH%\browser\override.ini"

REM Disable the Mozilla Maintenance Service to prevent updates (in the event the service is
installed)

sc config MozillaMaintenance start= disabled
```

local-settings.js

```
pref("general.config.obscure_value", 0);
pref("general.config.filename", "mozilla.cfg");
```

Mozilla.cfg

```
lockPref("app.update.auto", false);
lockPref("app.update.enabled", false);
```

```
lockPref("app.update.service.enabled", false);
lockPref("toolkit.telemetry.prompted", true);
lockPref("browser.rights.override", true);
lockPref("browser.startup.homepage_override.mstone", "ignore");
lockPref("browser.shell.checkDefaultBrowser", false);
```

override.ini

UserChrome.css

```
/* UserChrome.css for Mozilla Firefox */
/* Remove access to user interface elements that aren't suitable for application virtualization */

/* Options - Advanced - General - System Defaults */
#systemDefaultsGroup { display: none !important; }

/* Options / Advanced / Update / Firefox updates group box */
#updateApp { display: none !important; }

/* Help - About - Check for Updates button */
#updateButton { display: none !important; }
```

Finally

The approach outlined in this article should provide you with a deployment of Firefox that can be used in an enterprise where control of the browser is required. I've only covered a few of the things that are possible when customising the installation and if you dig a little deeper you can come up with a setup to suit your own environment. This is also a key approach to use when virtualising Firefox. Controlling the browser options and automating the installation will assist in producing better application virtualization packages.

Further Reading

Here's some other articles from around the tubes that cover this topic and are also useful references:

- [Customizing the Firefox Installer on Windows \(2012\)](#)
- [Install and Configure Firefox 18 silently](#)

Реализация от Алекса Корнева

Mozilla рекомендует, для использования в Enterprise Environment, раздавать настройки через с помощью файла **mozilla.cfg**.

<https://support.mozilla.org/en-US/questions/971481>

http://kb.mozillazine.org/Lock_Prefs

<https://support.mozilla.org/en-US/questions/1063418>

Чтобы все заработало, необходим еще один файл - **all-settings.js**, который необходимо поместить в папку **defaults\pref** внутри папки куда установлен Firefox.

all-settings.js

```
pref("general.config.filename", "mozilla.cfg");  
pref("general.config.obscure_value", 0);
```

Далее, средствами GPO/GPP создаем правила удаления/копирования

Минус метода - если рабочая станция x86, то дополнительно будет создана папка *C:\Program File (86)* с вложенной *Mozilla Firefox* (и внутри *defaults\perf*) Можно нагрузить политику (нга самом деле это уже не GPO а GPP) фильтрами по разрядности установленной OS, но ... тут возможен вариант, что установлена x64 а тот же Firefox x86 и можно "промахнуться папкой" для конфигов :-)

И еще раз. Не забыть, что источники файлы с конфигами надо положить туда, где Domain Computers могут читать файлы

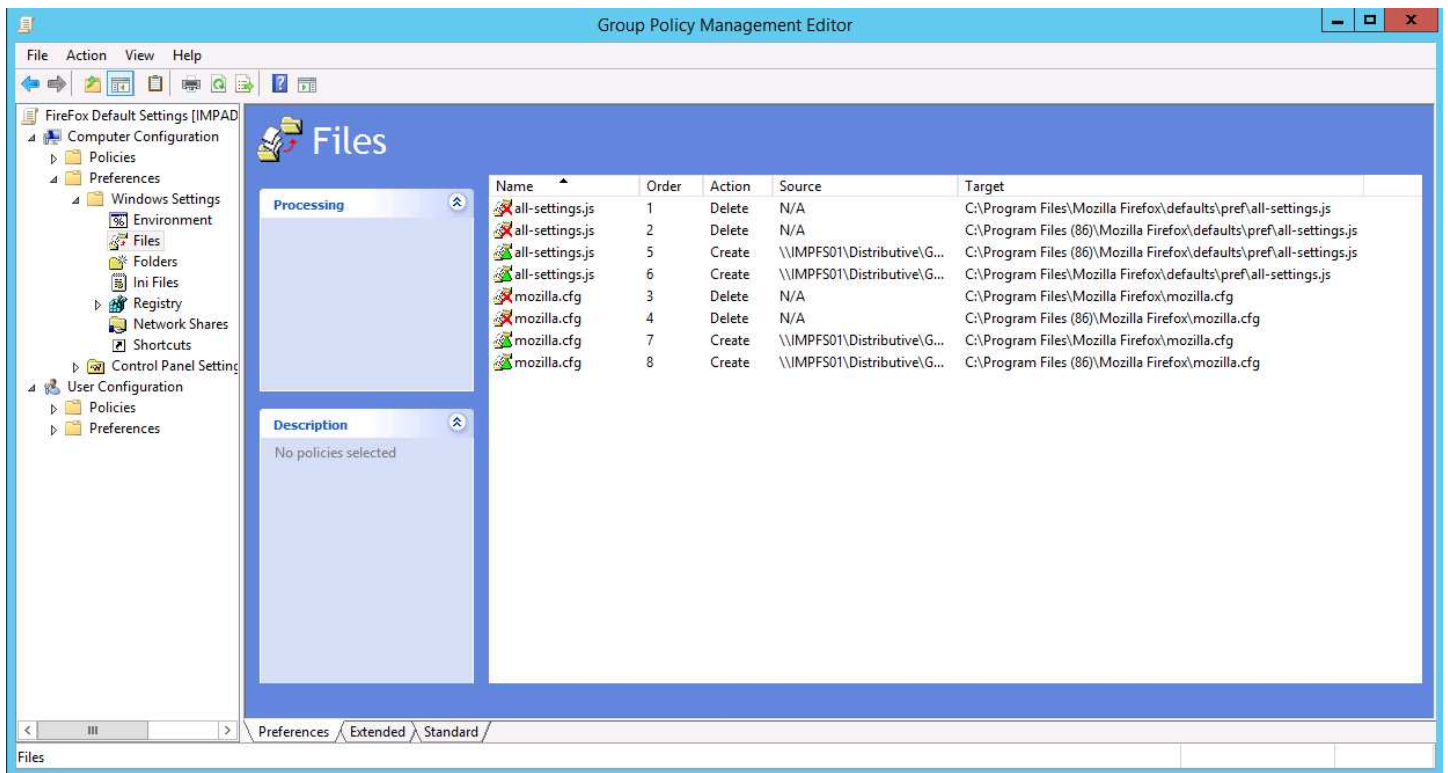
Два варианта:

а) положить файлы в папку NETLOGON - моветон :-)

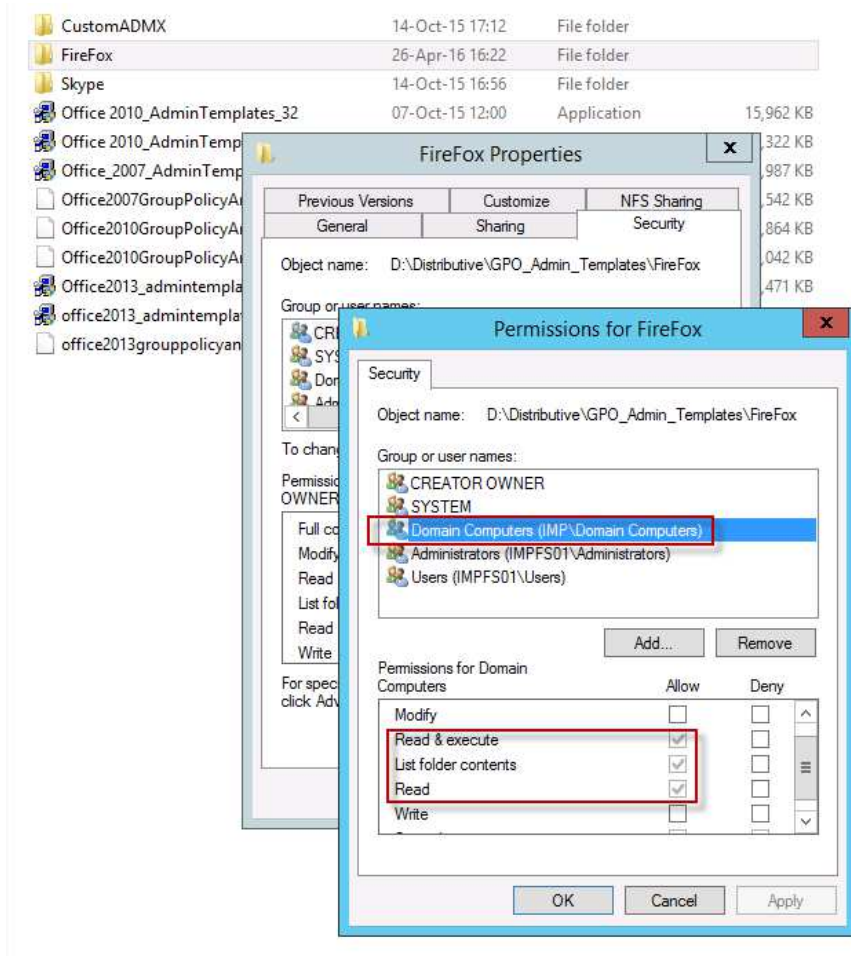
б) положить в шару на общем ресурсе и дать право чтения файлов для Domain Computers. (режим StartUp, запускаемся с правами SYSTEM, пользователя еще нет)

The screenshot shows the 'FireFox Default Settings' window in the Group Policy Editor. It has tabs for 'Scope', 'Details', 'Settings', and 'Delegation'. The 'Settings' tab is active, showing a list of settings under 'Computer Configuration (Enabled)'. The settings are organized into sections: 'Preferences', 'Windows Settings', 'Files', and 'User Configuration (Enabled)'. Under 'Files', there are several entries for 'all-settings.js' and 'mozilla.cfg' files, each with a target path and an order number. The 'all-settings.js' files are located in 'C:\Program Files\Mozilla Firefox\defaults\pref' and 'C:\Program Files (86)\Mozilla Firefox\defaults\pref'. The 'mozilla.cfg' files are located in 'C:\Program Files\Mozilla Firefox' and 'C:\Program Files (86)\Mozilla Firefox'. The 'User Configuration (Enabled)' section is currently empty, showing 'No settings defined.'

GPP "развесистая" слегка получается :-)



Понимаю, что "дую на воду", но, все равно, перед копирование запускаю удаление этих файлов отовсюду



А вот сам файл mozilla.cfg с комментариями.

ВАЖНО:

1. Первая строка этого файла может быть или пустой, или содержать комментарий. FF начинает читать этот файл со второй строки.
2. Файл case-sensitive! Если этого не учесть, Firefox просто не запустится.
3. Параметры можно определить через pref

4. Определить параметр и запретить его изменение пользователю через lockPref

mozilla.cfg

```
// Parameters keywords.
// pref
//   sets the preference as if a user had set it, every time you start the browser.
//   So users can make changes, but they will be erased on restart. If you set a
//   particular preference this way, it shows up in about:config as "user set".
// defaultPref
//   is used to alter the default value, though users can set it normally and their
//   changes will be saved between sessions. If preferences are reset to default
//   through the GUI or some other method, this is what they will go back to.
//   Appears in about:config as "default".
// lockPref
//   is used to lock preferences so they cannot be changed through the GUI or about:config.
//   In many cases the GUI will change to reflect this, graying out or removing options.
//   Appears in about:config as "locked". Some config items require lockPref to be set,
//   such as app.update.enabled. It will not work if it set with just pref.
// clearPref
//   can be used to "blank" certain preferences. This can be useful e.g. to disable functions
//   that rely on comparing version numbers.

// Set browser custom home-page
pref("browser.startup.homepage", "http://www.implanta.ru");
pref("browser.startup.page", 1);

// Proxy settings
pref("network.proxy.ftp", "xxx.xxx.xxx.xxx"); // можно написать доменное имя
pref("network.proxy.ftp_port", 3128);
pref("network.proxy.http", "xxx.xxx.xxx.xxx"); // можно написать доменное имя
pref("network.proxy.http_port", 3128);
pref("network.proxy.share_proxy_settings", true);
pref("network.proxy.socks", "xxx.xxx.xxx.xxx"); // можно написать доменное имя
pref("network.proxy.socks_port", 3128);
pref("network.proxy.ssl", "xxx.xxx.xxx.xxx"); // можно написать доменное имя
pref("network.proxy.ssl_port", 3128);
pref("network.proxy.type", 1);

// Check default browser
pref("browser.shell.checkDefaultBrowser", false);

// Disable updater
pref("app.update.enabled", false);
// make absolutely sure it is really off
pref("app.update.auto", false);
pref("app.update.mode", 0);
pref("app.update.service.enabled", false);

// Disable Add-ons compatibility checking
clearPref("extensions.lastAppVersion");

// Don't show 'know your rights' on first run
pref("browser.rights.3.shown", true);

// Don't show WhatsNew on first run after every update
pref("browser.startup.homepage_override.mstone", "ignore");

// Disable the internal PDF viewer
pref("pdfjs.disabled", true);

// Disable the flash to javascript converter
pref("shumway.disabled", true);

// Don't ask to install the Flash plugin
pref("plugins.notifyMissingFlash", false);

//Disable plugin checking
pref("plugins.hide_infobar_for_outdated_plugin", true);
clearPref("plugins.update.url");

// Disable health reporter
pref("datareporting.healthreport.service.enabled", false);

// Disable all data upload (Telemetry and FHR)
```



```
pref("datareporting.policy.dataSubmissionEnabled", false);  
// Disable telemetry  
pref("toolkit.telemetry.enabled",false);  
pref("toolkit.telemetry.rejected",true);  
pref("toolkit.telemetry.prompted",2);  
  
// Disable Health report  
pref("datareporting.healthreport.uploadEnabled", false);  
  
// Disable warning OnClose multiple tabs  
pref("browser.tabs.warnOnClose", false);
```

Другие параметры, на свой вкус, можно выбрать на странице [about:config](#) :-)