

USB Key Profile Manager:

Part I: Implementation of the Profile Loading Feature

I. Description:

The first major feature of the USB Key Profile Manager is to automatically detect and load profiles from a removable disk such as a USB drive.

The current version of Mozilla allows profiles to be saved to both local or removable disks. However, Mozilla does not recognize these profiles unless they are registered. Registration can only be done when a profile is created, meaning that one cannot easily take a profile created on one computer and use it on another.

My profile loading feature solves this problem and helps to achieve location independence without extra copying and pasting work from the user.

II. The Process:

The USB profile manager is implemented as a XPCOM component for Mozilla. When the browser is started up, the component will register itself with mozilla and start running.

It first checks how many usb storage media are mounted to the system, then loops through each mounted disk searching for profiles. The identification of a profile for now is a simplified process, i.e., all directories of which the name ends with “.slt” are considered valid Mozilla profiles.

Once a profile is found, it is added directly to Mozilla's profile registry.

If no USB profile is found, Mozilla's profile manager will act as usual.

If only one USB profile is located, the component will tell Mozilla to suppress the profile manager dialog at startup and use the USB profile by default.

If multiple are located, the profile manager dialog will be shown with all the profiles available, both on the local drive or the USB drive.

III. Class Definition

```
class nsUSBProfileManager: public nsIObserver,  
                           public nsIUSBProfileManager,
```

```
{
```

```
private:
```

```
    PRBool mLocked;
```

```
public:
```

```
    nsUSBProfileManager ();
```

```
    virtual ~nsUSBProfileManager();
```

```
NS_DECL_ISUPPORTS
```

```
// The Observe function serves as the main function. That's when the component is loaded  
// to Mozilla and started.
```

```
NS_DECL_NSIOBSERVER
```

```
NS_DECL_NSUSBPROFILEMANAGER
```

```

// The following are the helper methods
private:

// @param baseDir – the USB drive to search. For instance: /mnt/usbstick
NS_IMETHODIMP SearchForProfile(nsCOMPtr<nsIFile> baseDir);

/** @param dirEntry – A directory.
    @return PR_TRUE if is a profile, PR_FALSE otherwise.
    More sophisticated validation methods can be implemented later
*/
PRBool IsProfile(nsCOMPtr<nsIFile> dirEntry);

/** @param profileName – The name of the profile, which is actually the name of the
    parent folder of the profile folder xxx.slt
    @param profilePath – the full path name of the profile folder
*/
NS_IMETHODIMP LoadProfile(const PRUnichar *profileName, const PRUnichar
*profilePath);

// record number of profiles
int numOfUSBProfiles;

// Mozilla's special registry file
nsCOMPtr<nsIFile> mNewRegFile;

// The profile to be used, needed when only one USB profile is located
nsString mUSBProfileName;

// the following fields are for resetting to the last state
nsString prevProfileName;

// record the previous startWithLastUsedProfile flag
PRBool prevStartMode;
};

```

IV. Future Tasks:

1. If no profile exists either on the local or USB drive, provide a dialog for the user to create the very first profile, instead of creating one on the local drive by default.
2. Remove USB profiles when the system is shut down.
3. Rewind the profile settings to what they were before the changes made by the USB profile manager.
4. Lock the profile in use