





I tako nekoliko Sekcija sa jednom ili vise **Naziv=Vrednost** komponenti u svakoj od sekcija. Inf fajl editujete uz pomoc **Notepad**-a, a inf fajl kreirate tako sto sacuvate komande u Notepadu i date naziv **imefajla.inf** , znaci bitna je **inf** extenzija. Instalacija inf fajlova je jednostavna: desni klik na inf fajl i install. To je manuelno instaliranje. Postoji drugi nacin instaliranja inf fajlova a to je putem command prompta, primer

**rundll32.exe setupapi , InstallHinfSection DefaultInstall 132 imefajla.inf**

Evo jednog inf fajla koji cemo uzeti kao primer a nazvacemo ga primer.inf

**Kod:**

[Version]

Signature="\$CHICAGO\$"

[DefaultInstall]

AddReg=Add.Settings

DelReg=Del.Settings

[Add.Settings]

HKCR,regfile\shell,,0,"edit"

[Del.Settings]

HKCU,Software\Microsoft\Windows\CurrentVersion\Applets\Regedit

Prva sekcija koju vidimo naziva se **[Version]** i ona je esencijalna sekcija svakog inf fajla. U okviru nje se nalazi Signature komponenta koja oznacava da je u pitanju validan inf fajl. **"\$CHICAGO\$"** je bio m\$-ov "code name" za win 95 sto na kraju znaci da **Signature="\$CHICAGO\$"** identifikuje fajl kao validan Windows 95 INF fajl. U sadasnjosti,





0x00000004 Brise podkljuceve iz redzistrija ili value iz podkljuca.

0x00000008 Pripaja data sa value. Ovaj flag je validan samo u slucaju da je value REG\_MULTI\_SZ. String data se ne pripaja ukoliko vec postoji.

0x00000010 Kreira podkljuc ali ignorise value i data ukoliko je zadata.

0x00000020 Postavlja value jedino ukoliko vec postoji,

0x00001000 Vrsi promene u 64-bit redzistriju. Ukoliko nije zadata, promene se vrse na „normalnom“ redzistriju.

0x00004000 Vrsi promene u 32-bit redzistriju. Ukoliko nije zadata, promene se vrse na „normalnom“ redzistriju.

**data:** Podatak iskljucivo za upisivanje valuesa. Ako value ne postoji, API ce ga kreirati, a ako postoji onda ce API izvorsiti overwrite. Ako je value REG\_MULTI\_SZ a vi stavite 0x00010008 flag, API ce dodati value u postojeću string listu. Ukoliko je data izostavljena, API ce kreirati value bez njenog postavljanja. Primer:

**Kod:**

```
[Version]
```

```
Signature="$CHICAGO$"
```

```
[DefaultInstall]
```

```
AddReg=Add.Settings
```

```
[Reg.Settings]
```

```
; Postavlja defaultni value od HCKU\Software\Sample
```

```
HCKU,Software\Sample,,,"Default"
```

```
; Kreira REG_SZ value pod nazivom Sample
```

```
HCKU,Software\Sample,String,0x00000,"String"
```

```
; Kreira REG_BINARY value pod nazivom Binary
```

```
HCKU,Software\Sample, Binary,0x00001,00,01,30,05
```

```
; Kreira REG_MULTI_SZ value pod nazivom Multisz
```

HKCU,Software\Sample,Multisz,0x10000,"String list"

; Kreira REG\_DWORD value pod nazivom Dword

HKCU,Software\Sample,Dword,0x10001,0x01010102

; Kreira REG\_SZ value pod nazivom Cao

HKCU,Software\Sample,Cao,,"Devojke"

; Kreira REG\_DWORD value i postavlja ga 0x0000 flagu.

HKCU,Software\Sample,Nista,0x10001

## 2b. Brisanje kljuceva i valuesa

**DelReg** naredba u **[DefaultInstall]** sekciji oznacava nazive sekcija koje sadrže podatke koje želite da uklonite iz redzistrija. Te sekcije se jednim imenom zovu **[del-registry-section]** sekcije. Znatno su jednostavnije od [add-registry-section] sekcija, ali imaju slicna pravila: svaka sekcija može sadržati višestruke unose i naziv svake sekcije mora biti unikatan.

### **rootkey i subkey:**

pravila ista kao i kod dodavanja kljuceva

### **value:**

Naziv vrednosti koja se brise. Ova vrednost je opcionalna i ukoliko value nedostaje, vrsi se brisanje podkljuca.

### **flags:**

0x00002000 Brise celokupan podkljuc

0x00004000 Vrsi promene u 32-bit redzistriju. Ukoliko nije zadata, promene se vrse na „normalnom“ redzistriju

0x00018002 Ukoliko je value REG\_MULTI\_SZ, uklanja sve stringove koji obuhvataju string oznacen data-om

**data:**

Upotrebljava se samo kada je value 0x00018002

Primer:

**Kod:**

```
[Version]
```

```
Signature="$CHICAGO$"
```

```
[DefaultInstall]
```

```
DelReg=Reg.Settings
```

```
[Reg.Settings]
```

```
; Uklanja kljuc: HKCU\Software\Sample
```

```
HKCU,Software\Sample
```

```
; Uklanja value Cao iz HKCU\Software\Sample
```

```
HKCU,Software\Sample,Cao
```

```
; Uklanja string "Devojke" iz REG_MULTI_SZ valuesa Cao
```

```
HKCU,Software\Sample,Cao,0x00018002,"Devojke"
```

## 2c. Postavljanje i brisanje bitova (bits)

**BitReg** je ekvivalentno **AddReg** komandi. Znaci, dodaje se **[DefaultInstall]** sekciji i oznacava nazive sekcija koje sadrze bitove koje zelite da postavite ili izbrisete. Te sekcije se jednim imenom zovu **[bit-registry-section]** sekcije. koristite BitReg komandu kada radite sa bit maskama u redzistriju. Razlike izmedju [bit-registry-section] i [add-registry-section] sekcija su sledece:

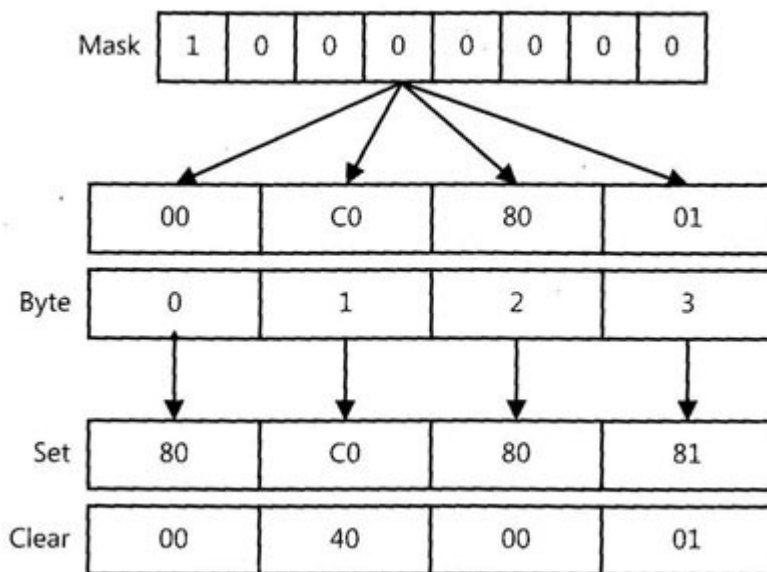
Parametar "value" nije opcionalan. Parametri "mask" i "byte" zamenjuju value "data".

Parametar "mask" je 8 bita dugacak i oznacava bit koji zelite da eneblijete ili diseblujete,

dok parameter "byte" oznacava bajt u binarnom valuesu koji menjate. Oznacava bajtove s leva na desno krecuci od nule.

Sada cemo razmotriti odnos izmedju ove tri komponente koje sam pomenuo a to su: *value*, *mask* i *byte*. Maskiracemo REG\_DWORD value smesten u redzistriju u reverse-byte obliku: 0x0180c000. Kad stavimo masku u bajt 0, rezultat ce biti 0x0180c080. Ako obrisemo masku u bajtu 1, rezultat ce biti 0x0140c080

Na sledecoj slici parameter "byte" oznacava kojem broju bajta zelimo postaviti masku.



### rootkey i subkey:

pravila ista kao i kod dodavanja kljuceva

### value:

Naziv vrednosti koja se edituje. Ova vrednost nije opcionalna i treba da bude REG\_DWORD ili REG\_BINARY value.

### flags:

0x00000000 Brise bitove zadate mask-om

0x00000001 Postavlja bitove zadate mask-om

0x00040000 Vrsi promene u 32-bit redzistriju. Ukoliko nije zadata, promene se vrse na „normalnom“ redzistriju

**mask:**

Maska koja postavlja ili uklanja *bite* u odredjenim bajtima i valuesima. Navodi se u heksadecimalnom zapisu. *Bitovi* koji su 1 ce biti postavljeni ili uklonjeni (zavisno od flagsa) , a *bitovi* koji su 0 ce bice ignorisani.

**byte:**

Oznacava *bajt* u veluesu za koji zelimo da primenimo *mask*. Krajnji levi bajt je 0, a sledeci je 1 i tako redom. Imajte na umu da Windows smesta REG\_DWORD valuese u reverse-byte obliku kada odredjujemo bajt na koji zelimo da primenimo *mask*. Prema tome, u REG\_DWORD valuesu, krajnji levi bajt je smesten na prvom mestu u memoriji.

Primer:

**Kod:**

```
[Version]
```

```
Signature="$CHICAGO$"
```

```
[DefaultInstall]
```

```
BitReg=Bit.Settings
```

```
[Bit.Settings]
```

```
; Menja 50,00,10,00 u 31,00,10,00
```

```
HKCU,Software\Sample,Mask,0x0001,0x01,0
```

```
; Menja 50,00,F0,00 u 30,00,70,00
```

```
HKU,Software\Sample,Mask,0x0000,0x80,2
```

**2e. Upotreba stringova u INF fajlovima**

Ovo se vrši upotrebom **[Strings]** sekcije. Svaka linija u ovoj sekciji je string u formatu **ime="string"**. Takav string mozete koristiti bilo gde u INF fajlu isticuci ga kao **%ime%**

Karakteristike [Strings] sekcije:

- Sakuplja stringove na dno INF fajla tako da ga mozete videti na jednom mestu
- Omogucava vam da unesete string jedanput i da onda koristite taj string na brojnim mestima
- Omogucava lakse prevodjenje INF fajlova jer su stringovi na dnu INF fajla.

**Kod:**

[Version]

Signature="\$CHICAGO\$"

[DefaultInstall]

BitReg=Bit.Set

AddReg=Add.Settings

DelReg=Del.Settings

[Add.Settings]

HKCU,%HK\_Desktop%,ActiveWndTrkTimeout,0x10001x1000

HKLM,%HK\_Setup%,RegisteredOwner..%OWNER%

[Del.Settings]

HKCU,%HK\_EXPLORER%\MenuOrder

HKCU,%HK\_EXPLORER%\RunMRU

HKCU,%HK\_EXPLORER%\RecentDocs

HKCU,%HK\_EXPLORER%\ComDlg32\LastvisitedMRU

HKCU.%HK\_SEARCH%\ACMrU

HKCU,%HK\_INTERNET%\TypedURLs

[Bits.Set]

HKCU,%HK\_Desktop%,UserPreferencesMask,1,0x01,0

HKCU,%HK\_Desktop%,UserPreferencesMask,1,0x40,0

[Strings]

```
HK_Desktop="Control Panel\Desktop"
HK_EXPLORER="Software\Microsoft\Windows\CurrentVersion\Explore"
HK_SEARCH="Software\Microsoft\Search Assistant"
HK_INTERNET="Software\Microsoft\Internet Explorer"
HK_SETUP="Software\Microsoft\Windows NT\CurrentVersion"
OWNER="Doktor Marko"
```

### 3. Kreiranje reg fajlova za popravku kljuceva

Registry fajl je ekvivalentan .inf fajlu i struktura mu je gotovo identicna, s tim da registry fajl ima extenziju **.reg** . REG fajl dodajemo na dva nacina: dvoklikom ( u tom slucaju nas sistem pita za potvrdu) i preko command prompta kucanjem komande **regedit /s nazivfajla.reg** i na taj nacin dodajemo REG fajl bez smaranja od strane sistema. REG fajl editujemo jednostavno, desni klik->edit. Regedit podrzava dva formata: **REG** fajlove i **ANSI**.

**ANSI encoding karaktera** koristi jedan bajt da predstavi svaki karakter. Regedit koristi ANSI za ispisivanje **REG\_EXPAND\_SZ** i **REG\_MULTI\_SZ** stringova REG fajlovima , tako da je svaki karakter ustvari jedan bajt. **Version 5** REG fajlovi su **Unicode**. Unicode encoding karaktera koristi dva bajta za svaki karakter i kada kreirate **Unicode REG fajl**, Regedit koristi dvobajtnu Unicode encoding semu za ispisivanje **REG\_EXPAND\_SZ** i **REG\_MULTI\_SZ** stringova REG fajlu.

#### Kod:

Windows Registry editor Version 5.00

```
[HKEY_CURRENT_USER\Control Panel\Desktop]
"ActiveWndTrkTimeout"=dword:00000000
"ForegroundFlashCount"=dword:00000003
"ForegroundLockTimeout"=dword:00030d40
"MenuShowDelay"=dword:"400"
```

```
"PaintDesktopVersion":00000000
"UserPreferencesMask":hex:9e,3e,07,80
```

```
[HKEY_CURRENT_USER\Control Panel\Desktop\WindowMetrics]
```

```
"Shell Icon BPP"="16"
"Shell Icon Size"="32"
"MinAnimate"="1"
```

```
[HKEY_CURRENT_USER\Control Panel\Mouse]
```

```
@="Mis"
"ActiveWindowTracking"=dword:00000000
"DoubleClickHeight"="4"
"DoubleClickSpeed"="500"
"DoubleClickWidth"="4"
"MouseSensitivity"="10"
"MouseSpeed"="1"
"MouseThreshold1"="6"
"MouseThreshold2"="10"
"SnapToDefaultButton"="0"
"SwapMouseButtons"="0"
```

Sad cu redom da objasnim sta je sta u ovom gronjem primeru.

**Windows Registry editor Version 5.00** oznacava da je u pitanju verzija 5 Unicode REG fajla. Gornji primer predstavlja importovanje podesavanja u tri kljuca: **HKCU\Control Panel\Desktop** , **HKCU\Control Panel\Desktop\WindowMetrics** i **HCU\ Control Panel\Mouse**. Ta tri kljuca predstavljaju posebne sekcije, i u svakoj od njih se nalaze valuesi ciji je format **"ime"=value** . Oznaka **@** predstavlja defaultnu vrednost kljuca. Odredjeni valuesi sadrže **dword** i **hex** , dok su ostali pod navodnicima i takvi valuesi se nazivaju **stringovi** . Valuesi u **hex:values** formatu su REG\_BINARY valuesi, a valuesi u **dword:values** formatu su REG\_DWORD valuesi.

### 3a. Rucno kreiranje REG fajlova

Otvorite Notepad i na vrhu stavite **Windows Registry editor Version 5.00**. Za svaki kljuc u koji zelite da importujete valuse, dodajte sekciju u formatu **[kljuc]** gde je **kljuc** naziv kljuca. Nemojte koristiti skracenice za root kljuceve vec pune nazive tipa:

**HKEY\_CURRENT\_USER**. Za svaki values koji zelite da importujete u redzistri, dodajte values u formatu **"ime"=value** u odgovarajucu sekciju. Koristite **@** za defaultnu vrednost kljuca. Mozete koristiti i obrnutu kosu crtu (backslash) (**\**) da biste nastavili bilo koji unos sa jednog reda na drugi. Kada ste iskuckali reg fajl, idite na File->Save As i sacuvajte fajl npod nazivom **imefajla.reg** (id a naravno, **.reg** extenzija se mora nalaziti na kraju).

Formati Valuesa u REG fajlovima:

Type	Version 4	Version 5
REG_SZ	"String"	"String"
REG_DWORD	dword:00007734	dword:00007734
REG_BINARY	hex:00,00,01,03	hex:00,00,01,03
REG_EXPAND_SZ	hex(2):25,53,59,53, 54,45,4d,52,4f,4f, 54,25,00	hex(2):25,00,53,00,59,00,53,00, 54,00,45,00,4d,00,52,00,4f,00, 4f,00,54,00,25,00,00,00
REG_MULTI_SZ	hex(7):48,65,6c,6c, 6f,20,57,6f,72,6c,64, 00,4a,65,72,72,79,20, 77,61,73,20,68,65,72, 65,00,00	hex(7):48,00,65,00,6c,00,6c,00, 6f,00,20,00,57,00,6f,00,72,00, 6c,00,64,00,00,00,4a,00,65,00, 72,00,72,00,79,00,20,00,77,00, 61,00,73,00,20,00,68,00,65,00, 72,00,65,00,00,00,00,00

### 3.b Brisanje kljuceva upotrebom REG fajla

Ne mozete koristiti REG fajl da bi uklanjali pojedinačne valuse iz redzistrija, vec da bi uklanjali citave kljuceve. Da biste to uradili, jednostavno stavite znak minus (-) ispred naziva kljuca tj: **[-kljuc]**. Evo na pr REG fajl koji ce ukloniti kljuc HKCU\Software\MarkoLegenda

**Kod:**

Windows Registry editor Version 5.00

[-HKCU\Software\MarkoLegenda]

Ipak je dosta praktičnije da vi lepo exportujete ključ u obliku REG fajla, otvorite ga i ručno editujete tj maknete ključeve i valuese iz njega koje želite da ostanu netaknuti (tj koje ne želite ukloniti principom brisanja objasnjenog u gornjem primeru). Nakon toga dodajte znak minus ispred naziva ključa koji želite izbrisati.

#### 4. Editovanje redzistrija putem komandne kozole (batch scriptingom)

Registry ključevi i valuesi se mogu editovati i uz pomoć Command Prompta.

Evo primera batch fajla koji vrsi instaliranje MS Office 2003 (naravno, ukoliko se već ne nalazi na sistemu). Nakon instaliranja Officea

**Kod:**

```
\\Camelot\Office\Setup.exe /settings setup.ini
```

batch fajl dodaje REG\_DWORD value **Flag**, **HKCU\Software\Primer** valuesu

**Kod:**

```
Reg add HKCU\Software\Primer /v Flag /t REG_Dword /d "1"
```

Batch fajl će uvek izvršiti proveru da li su na sistemu već prisutni dotični valuesi i ako jesu onda će preskociti instalaciju, što znači da batch fajlovi instaliraju aplikaciju samo jedanput.

Batch fajl proverava postojanje office 2003 na vašem sistemu komandom:

**Kod:**

```
Reg QUERY HKCU\Software\Microsoft\Office\11.0 >nul
```

Primer:

**Kod:**

```
@Echo off
```

```
Reg QUERY HKCU\Software\Primer /v Flag >nul
```

```
goto %ERRORLEVEL%
```

```
:1
```

```
Echo Instaliranje softvera
```

```
\\Camelot\Office\Setup.exe /settings setup.ini
```

```
Reg add HKCU\Software\Primer /v Flag /t REG_Dword /d "1"
```

```
goto CONTINUE
```

```
:0
```

```
Echo Softver je vec instaliran, ova sekcija se preskace.
```

```
:CONTINUE
```

```
Set HKMS=HKCU\Software\Microsoft
```

```
Set HKCV=HKCU\Software\Microsoft\Windows\CurrentVersion
```

```
REM Obrisi history listu
```

```
Reg DELETE %HKCV%\Explorer\MenuOrder /f
```

```
Reg DELETE %HKCV%\Explorer\RunMRU /f
```

```
Reg DELETE %HKCV%\Explorer\RecentDocs /f
```

```
Reg DELETE %HKCV%\Explorer\ComDlg32\LastVisitedMRU
```

```
Reg DELETE "%HKMS%\Search Assistant\ACMrU" /f
```

```
Reg DELETE "%HKMS%\Internet Explorer\TypedUrls" /f
```

Komande koje ovde srecemo izmedju ostalog su i: Reg, Set, ADD, QUERY i DELETE (a sve moguće komande mozete dobiti kada u cmd unesete komandu **reg /.**)

Opcije su nazivi valuesa, kljuceva i podataka. Kao sto gore vidimo neke komande su i pod navodnicima, a to je zato sto u svom nazivu sadrže razmak.

#### 4.a Dodavanje kljuceva:

Kljuceve dodajemo komandom **ADD**

Sintaksa:

```
REG ADD [\\computer\]key [/v value | /ve] [/t type] [/s separator] [/d data] [/f]
```

<i>\\computer</i>	Ako nedostaje, Redzistri ce se povezati na lokalni racunar, u suprotnom ce se povezati na remote racunar.
<i>key</i>	Naziv kljuca koji pocinje sa root kljucem. Koristite skracenice za root kljuceve: HKCR, HKCU, HKLM i HKU. Jedino su HKLM i HKU moguci ako se vrsi povezivanje sa remote racunatom.
<i>/v value</i>	Dodaje ili menja <i>value</i>
<i>/ve</i>	Menja defaultni value kljuca
<i>/t type</i>	Ovo je tip valuesa: REG_BINARY, REG_DWORD, REG_DWORD_LITTLE_ENDIAN, REG_DWORD_BIG_ENDIAN, REG_EXPAND_SZ, REG_MULTI_SZ ili REG_SZ. Default je REG_SZ
<i>/s separator</i>	Koristi se za razdvajanje stringova pri kreiranju REG_MULTI_SZ valuesa. Default je \0, ili <i>null</i> .
<i>/d data</i>	Podatak koja za povezivanje novom ili postojećem valuesu
<i>/f</i>	Vrsi overwrite postojećeg valuesa uz obavestenje.

primer:

#### **Kod:**

```
REG ADD \\m4rk0\HKLM\Software\MarkoLegenda
REG ADD HKLM\Software\MarkoLegenda /v Data /t REG_BINARY /d CFEF0BC
REG ADD HKLM\Software\MarkoLegenda /v List /t REG_MULTI_SZ /d
Cao\0Devojke
REG ADD HKLM\Software\MarkoLegenda /v Path /t REG_EXPAND_SZ
%%SYSTEMROOT%%
```

Jos da napomenem da se kod reg.exe komandne linije moraju koristiti dva znaka „procenat“ tj. (%%) jer u slucaju da koristimo samo jedan, uslovna varijabla ce se razviti pre pokretanja komande.

#### 4.b Queryng kljuceva:

Komanda **QUERY** radi na tri nacina:

- Moze da prikaze podatke u specificnom valuesu
- Moze da prikaze sve valuese unutar kljuca.
- Moze da izlista sve podkljuceve i valuese unutar kljuca, dodavanjem [/b]/s[/b] switcha.

<i>\computer</i>	Ako nedostaje, Redzistri ce se povezati na lokalni racunar, u suprotnom ce se povezati na remote racunar.
<i>key</i>	Naziv kljuca koji pocinje sa root kljucem. Koristite skracenice za root kljuceve: HKCR, HKCU, HKLM i HKU. Jedino su HKLM i HKU moguci ako se vrsi povezivanje sa remote racunatom.
<i>/v value</i>	Vrsi query vrednosti ( <i>value</i> ) u kljucu ( <i>key</i> ). Ako se izostavi <i>/v</i> automatski ce se vrsiti odabir svih vrednosti u kljucu
<i>/ve</i>	Vrsi query defaultne vrednosti kljuca
<i>/s</i>	Vrsi query svih podkljuceva i valuesa unutar kljuca

Primer

#### **Kod:**

```
REG QUERY HKLM\Software\Microsoft\Windows\CurrentVersion /s  
REG QUERY HKLM\Software\Microsoft\Windows NT\CurrentVersion /v  
CurrentVersion
```

Redzistry postavlja ERRORLEVEL na 0 ako je komanda uspela, a ako nije uspela onda je vrednost 1.

#### 4.c Brisanje kljuceva i valuesa

Za brisanje kljuceva i valuesa koristite komandu **DELETE**.

```
REG DELETE [\\computer\]key [/v value | /ve | /va] /f
```

<i>\\computer</i>	Ako nedostaje, Redzistri ce se povezati na lokalni racunar, u suprotnom ce se povezati na remote racunar.
<i>key</i>	Naziv kljuca koji pocinje sa root kljucem. Koristite skracenice za root kljuceve: HKCR, HKCU, HKLM i HKU. Jedino su HKLM i HKU moguci ako se vrši povezivanje sa remote racunatom.
<i>/v value</i>	Briše <i>value</i> iz kljuca ( <i>key</i> )
<i>/ve</i>	Briše defaultnu vrednost kljuca
<i>/va</i>	Briše sve valuese iz kljuca
<i>/f</i>	Vrši brisanje valuesa uz obavestenje.

primer:

**Kod:**

```
REG DELETE \\m4rk0\HKLM\Software\MarkoLegenda  
REG DELETE HKLM\Software\MarkoLegenda /v Data /f  
REG DELETE HKLM\Software\MarkoLegenda /va
```

#### 4.d Usporedjivanje kljuceva i valuesa

Koristite komandu **COMPARE** za usporedjivanje dva registry kljuca. Ti kljucevi mogu biti na istom ili razlicitim racunarima.

Redzistri postavlja **ERRORLEVEL** prema rezultatima usporedjivanja i mozete upotrebiti taj rezultat u vasem batch fajlu za izvršavanje razlicitih kodova, bilo da su dva kljuca/valuesa ista ili razlicita, bez prikazivanja bilo kakvog rezultata.

Evo **znacenja ERRORLEVEL** komande:

0. Komanda je uspesna, i kljucevi ili valuesi su identicni
1. Komanda je neuspesna
2. Komanda je uspesna, i kljucevi ili valuesi su razliciti.

REG COMPARE [\\computer1\]key1 [\\computer2\]key2 [/v value | /ve] [/oa|/od|/os|/on] [/s]

<i>\\computer1</i>	Ako nedostaje, Redzistri ce se povezati na lokalni racunar, u suprotnom ce se povezati na remote racunar.
<i>\\computer2</i>	Ako nedostaje, Redzistri ce se povezati na lokalni racunar, u suprotnom ce se povezati na remote racunar.
<i>Key1</i>	Naziv kljuca koji pocinje sa root kljucem. Koristite skracenice za root kljuceve: HKCR, HKCU, HKLM i HKU. Jedino su HKLM i HKU moguci ako se vrsi povezivanje sa remote racunatom.
<i>Key2</i>	Naziv kljuca koji pocinje sa root kljucem. Koristite skracenice za root kljuceve: HKCR, HKCU, HKLM i HKU. Jedino su HKLM i HKU moguci ako se vrsi povezivanje sa remote racunatom.
<i>/v value</i>	Uporedjuje <i>value</i>
<i>/ve</i>	Uporedjuje defaultnu vrednost kljuca
<i>/oa</i>	Prikazuje sve razlike i identicnosti (poklapanja)
<i>/od</i>	Prikazuje samo razlike
<i>/os</i>	Prikazuje samo identicnosti
<i>/on</i>	Ne prikazuje nista
<i>/s</i>	Poredi sve podkljuceve i valuese unutar kljuca.

Primer:

**Kod:**

```
REG COMPARE HKCR\txtfile HKCR \txtfile HKR\docfile /ve
REG COMPARE \\Marko1\HKCR \\Marko2\HKCR /od /s
REG COMPARE HKCU\Software \\Marko2\HKCU\Software /s
```

#### 4.e kopiranje kljuceva i valuesa

Koristite komandu **COPY** za kopiranje podkljuca u drugi kljuc. Ova komanda je korisna kod

bekapovanja podkljuceva.

Sintaksa:

```
REG COPY [\\computer1\]key1 [\\computer2\]key2 [/s] [/f]
```

<i>\\computer1</i>	Ako nedostaje, Redzistri ce se povezati na lokalni racunar, u suprotnom ce se povezati na remote racunar.
<i>\\computer2</i>	Ako nedostaje, Redzistri ce se povezati na lokalni racunar, u suprotnom ce se povezati na remote racunar.
<i>Key1</i>	Naziv kljuca koji pocinje sa root kljucem. Koristite skracenice za root kljuceve: HKCR, HKCU, HKLM i HKU. Jedino su HKLM i HKU moguci ako se vrsi povezivanje sa remote racunarem.
<i>Key2</i>	Naziv kljuca koji pocinje sa root kljucem. Koristite skracenice za root kljuceve: HKCR, HKCU, HKLM i HKU. Jedino su HKLM i HKU moguci ako se vrsi povezivanje sa remote racunarem.
<i>/s</i>	Kopira sve podkljuceve i valuese
<i>/f</i>	Vrsi kopiranje uz obavestenje

Primer:

**Kod:**

```
REG COPY HKCU\Software\Microsoft\Office HKCU\Backup\Office /s
```

```
REG COPY HKCU\regfile HKCU\Backup\regfile /s /f
```

#### 4.f Exportovanje kljuceva u vidu REG fajlova

Koristite komandu **EXPORT** da biste exportovali ceo ili deo regdzistrija u obliku REG fajlova. Ovo uspeva samo na lokalnom racunaru, sto znaci da ne mozete da kreirate reg fajl preko redzistrija remote racunara. Takodje, na ovaj nacin ne mozete kreirati ANSI REG fajlove, vec samo version 5, Unicode REG fajlove.

EXPORT komanda je isto sto i **FILE -> Export** u Regeditu.

Sintaksa:

REG EXPORT *key filename*

<i>key</i>	Naziv ključa koji počinje sa root ključem. Koristite skraćene za root ključeve: HKCR, HKCU, HKLM i HKU. Jedino su HKLM i HKU mogući ako se vrši povezivanje sa remote računarom.
<i>filename</i>	Naziv REG fajla koji kreirate

Primer:

**Kod:**

```
REG EXPORT "HKCU\Control Panel" Preferences.reg
```

#### 4.g Importovanje REG fajlova

Koristite komandu **IMPORT** da biste uneli REG fajl u redzistri.

IMPORT komanda je isto sto i **regedit /s filename**. Na taj nacin se REG fajl importuje *tiho* (silently)

Sintaksa:

REG IMPORT *filename*

Primer:

**Kod:**

```
REG IMPORT Settings.reg
```

#### 4.h Sacuvanje kljuceva u vidu Hive fajlova

Ovo se vrši komandom **SAVE**. To je isto kao kad biste u Regeditu isli **File -> Export** i onda promenili tip u **Registry Hive Files (\*.\*)**. Ova komanda radi samo na lokalnom računaru.

Sintaksa

```
REG SAVE key filename
```

Primer:

**Kod:**

```
REG SAVE HKU Backup.dat
```

#### 4.i Restoring Hive fajlova u kljuceve

Ovo se radi komandom **RESTORE** koja obavlja overwrite kljuca i celokupnog njegovog sadrzaja sa sadrzajem hive fajla. To je isto sto i importovanje hive fajla u registry. Razlika izmedju ove komande i učitavanja hive fajla je u tome sto ova komanda vrsi overwrite bilo kog postojećeg kljuca, dok učitavanje hive fajlova kreira novi privremen kljuc koji sadrzi sadrzaj hive fajla. Ova komanda radi samo na lokalnom racunaru.

Sintaksa:

```
REG RESTORE key filename
```

Primer:

**Kod:**

```
REG RESTORE HKCU Backup.dat
```

#### 4.j Loading i unloading Hive fajlova

Komanda **LOADING** vrsi učitavanje hive fajla u privremeni ključ.

Sintaksa:

```
REG LOAD key filename
```

Primer:

**Kod:**

```
REG LOAD HKU\Temporary Settings.dat
```

Komanda **UNLOAD** uklanja hive fajl koji se učitao koristeći komandu LOAD. Bitno je da unloading hiva fajla koji ste učitali, pre pokušaja kopiranja ili bilo čega drugog sa hiva fajlom, jer windows zaključava fajl ukoliko je u upotrebi.

sintaksa:

```
REG UNLOAD key
```

Primer:

**Kod:**

```
REG UNLOAD HKU\Temporary
```