Zbornik radova Fakulteta tehničkih nauka, Novi Sad

**UDK: 621.39**
**DOI: https://doi.org/10.24867/03BE31Spasojevic**

# AUTOMATSKA REINSTALACIJA DMS SCADA SISTEMA POMOĆU DSC-A

# AUTOMATIC REINSTALATION OF DMS SCADA SYSTEM USING DSC

Jelena Spasojević, *Fakultet tehničkih nauka, Novi Sad*

**Oblast - ELEKTROTEHNIKA I RAČUNARSTVO**

**Kratak sadržaj** – *U ovom radu opisani su problemi upravljanja konfiguracijama koji nastaju usled sve veće potražnje za novijim softverom kao rastom kompleksnosti softvera. U radu je dat opis Desired State Configuration sistema za upravljanje konfiguracijama, njegova arhitektura, dat je pregled različitih verzija DSC-a kao i prednosti DSC-a u odnosu na ručno upravljanje konfiguracijama. Na kraju rada dato je poređenje upravljanja konfiguracijama uz pomoć DSC-a i ručno upravljanje kao i rezultati ovog poređenja.*

**Ključne reči:** *Desired State Configuration, upravljanje konfiguracijama, DMS SCADA.*

**Abstract** – *This paper describes issues in configuration management which occur due to high demand in new software and due to rise of its complexity. Desired State Configuration, a system for configuration management was described, its architecture, different versions of DSC were also described and also its advantages over manual configuration management. At the end of this paper there is a comparison between DSC and manual configuration management and the result of comparison is also described.*

**Ključne reči:** *Desired State Configuration, configuration management, DMS SCADA.*

## 1. UVOD

Upravljanje konfiguracijom igra centralnu ulogu u integraciji i plasiranju softvera. Kompanije se danas vode logikom brzog i čestog plasiranja manjih softverskih proizvoda. S tim u vezi, proizvodi se prave sa jednom funkcijom u tom trenutku i to se sa vremenom sve više proširuje i nadograđuje. Da bi to bilo moguće proizvodi se moraju napraviti, testirati a zatim često plasirati na tržište.

Ovde je izazov kako omogućiti stalno plasiranje softvera. Plasiranje softvera mora biti dobro osmišljeno i automatizovano kako bi se eliminisale ljudske greške. Kompleksnost brzog plasiranja raste sa kompleksnošću softvera. Upravo zbog sve veće kompleksnosti softvera kao i težnji kompanija za što bržim plasiranjem softvera cilj je što više eliminisati ljudski faktor prilikom upravljanja konfiguracijama.

Rešenje svih nabrojanih problema jeste automatizacija u upravljanju konfiguracijama odnosno Desired State Configuration (DSC).

---

**NAPOMENA:**
**Ovaj rad proistekao je iz master rada čiji mentor je dr Darko Čapko.**

## 2. DESIRED STATE CONFIGURATION

Desired State Configuration predstavlja opciju upravljanja konfiguracijama koja je ugrađena u Windows operativni sistem. DSC je relativno novi učesnik u svetu upravljanja konfiguracijama. Prvi put se javlja u okviru Windows Powershell 4.0 verzije koja je izašla u oktobru 2013 godine [2]. DSC je veoma značajan jer je proizašao iz samog operativnog sistema i što se alati ugrađeni u sistemu mogu koristiti da se izvršavaju konfiguracione promene. DSC zauzima veoma značajno mesto jer je proizašao iz samog operativnog sistema i što se alati ugrađeni u sistemu mogu koristiti da se izvršavaju konfiguracione promene.

Windows operativni sistem nudi standardizovan način upravljanja konfiguracijama a upravo na tim standardima zasnovan je i DSC. DSC treba shvatiti kao platformu za upravljanje konfiguracijama pre nego kompletno rešenje za upravljanje konfiguracijama. Za razliku od drugih alata ili rešenja, DSC ne nudi alate za upravljanje i nadgledanje konfiguracija po modelu end to end, već nudi API (Application Programming Interface) kao i platformu koju čak drugi alati mogu da koriste [1].

### 2.1 Arhitektura DSC-a

Arhitektura DSC je prilično jednostavna. Na slici 1 prikazana je arhitektura DSC gde su jasno vidljive dve izražene faze.

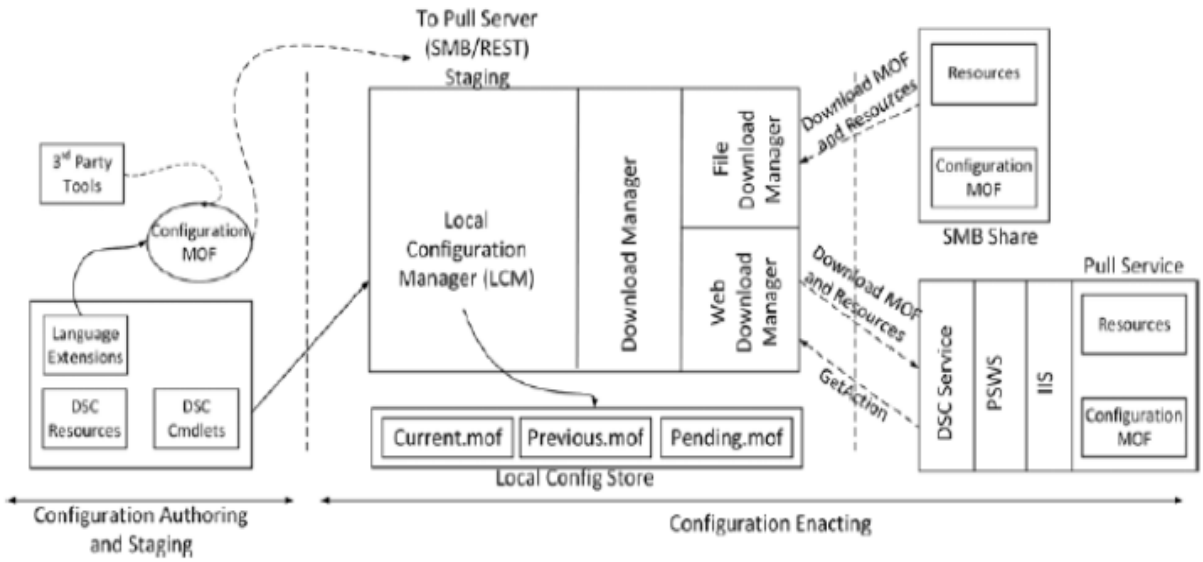
Svaka od ovih faza uključuje komponente DSC koje imaju specifičnu svrhu. Dve glavne faze su:

- faza izrade i testiranja konfiguracije
- faza puštanja u rad

U fazi izrade pišu se konfiguracione skripte koje definišu kako se resursi na sistemu moraju koristiti, kao i koji resursi su neophodni i kako se konfiguracija dalje priprema za sledeću fazu. Nakon faze izrade sledi faza gde se konfiguracija testira i nakon toga sledi faza puštanja u rad. U fazi puštanja u rad konfiguracija se primenjuje na ciljani sistem, a tu se izdvajaju dva različita načina na koje se konfiguracija može primeniti, a to su Push i Pull metode. Zapravo, u fazi izrade dolazi do prijema konfiguracionog fajla na ciljani sistem a zatim primenjivanje promena. Svaka od ovih faza ima specifičnu ulogu u arhitekturi DSC –a [1].

### 2.2 Faza izrade i testiranja

Ova faza je prva faza u DSC arhitekturi. U njoj se pišu konfiguracione skripte i testiraju i pripremaju za fazu puštanja u rad gde se izmenjena konfiguracija primenjuje. Kao što se iz samog imena prve faze vidi, ona se sastoji iz dva dela a to su izrada i testiranje.




*Slika 1 – Arhitektura Desired State Configuration*

U fazi izrade koriste se Windows PowerShell deklarativne skripte. DSC PowerShell modul je prva komponenta koja se posmatra pri pisanju skripti u DSC PowerShell-u. Sve jezičke ekstenzije, koje se koriste u skriptama, implementiraju se unutar DSC PowerShell modula.
Ovde je najvažnije izdvojiti funkciju Configuration. Ona se smatra ključnom komponentom DSC-a, pošto ova funkcija definiše željenu konfiguraciju sistema.
Ono što sledi nakon reči Configuration jeste ime ili identifikator koji je dodeljen konfiguraciji.
Konfiguracija željenog sistema definiše se koristeći PowerShell skript blok koji se identifikuje koristeći blok kod unutar zagrada {}.

```
Configuration IdentifierString {
}
```

Iako je funkcija Configuration ključni aspekt upravljanja konfiguracijama, ova funkcija sama ne definiše konfiguraciju koju želimo da kontrolišemo. Da bi se to uradilo mora se koristiti funkcija Node jer ona određuje na koji sistem se primenjuje konfiguracija. Nakon toga neophodno je navesti šta se menja u konfiguraciji odnosno navode se resursi koje menjamo. Postavlja se pitanje kako da znamo koji resursi su dostupni i koja su njihova imena, jer su nam neophodna da bismo ih koristili.
To se može videti korišćenjem DSC PowerShell modula odnosno naredbe Get-DscResource. Ovom naredbom mogu se videti svi DSC resursi koji su dostupni na sistemu. Unutar Node skript bloka može se navesti bilo koji broj resursa, bilo istih, bilo različitih sve dok se koristi drugačiji identifikator kao i ključne karakteristike poput imena.
DSC skripta bi izgledala ovako

```
Configuration WebsiteConfig {
        WindowsFeture WebServer1{
                Name = "Web-Server"
        }
}
```

Kada je DSC skripta napisana, to nije dovoljno da bi ona napravila promene na sistemu. Neophodno je da se skripta prevede u format koji DSC Local Configuration Manager razume. Taj format je Managed Object Format (MOF). MOF format se generiše jednostavnim učitavanjem konfiguracije u memoriju, a zatim njenim pozivanjem.

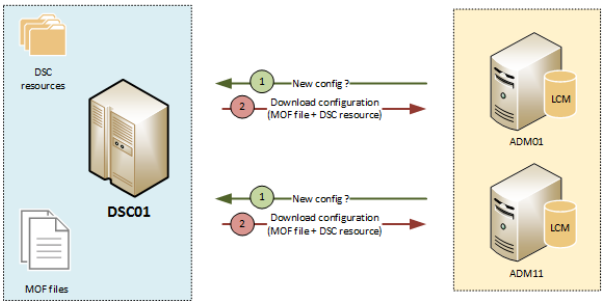
### 2.3 Faza puštanja u rad

Ova faza dolazi nakon faze izrade i testiranja. U fazi izrade generisan je MOF fajl na mrežnoj lokaciji, ali to nije dovoljno da ga sistem koji želimo da promenimo primi i da reaguje na promene. MOF se mora postaviti na željeni sistem. Da bi se to uradilo koriste se Push i Pull funkcije u DSC-u. [1]

### 2.4 Pull model

U Pull modelu nodovi se konfigurišu da dobiju konfiguraciju od strane pull servera. Ovaj način slanja konfiguracije je dobar kada se mašine moraju konfigurisati brzo.

Da bi se koristio Pull model neophodan je server sa instaliranim Windows DSC. Takođe, server mora da ima Windows Management Framework 4 (WMF) instaliran. Jedan jednostavan primer tehničkog okruženja prikazan je na slici ispod.

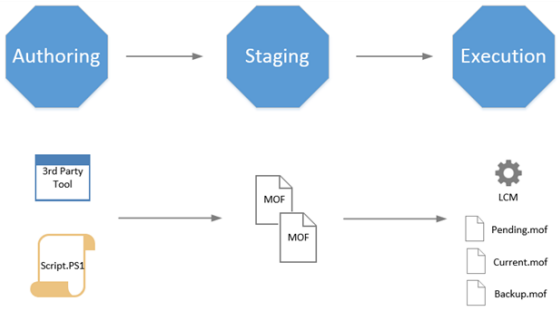



*Slika 2 – Pull model – tehničko okruženje*

Pull mode je najkompleksniji, ali najlakši za održavanje. Jedan primer nedostatka push modela u odnosu na pull model je ako u push modu nod bude offline promena konfiguracije neće uspeti. U tom slučaju Sistem administrator mora stalno da pokušava da pošalje novu konfiguraciju. U pull modu to nije tako, on je mnogo dinamičniji. Kakav god da je status noda konfiguracija će se primeniti čim on bude online.
Pull server je IIS Web server koji sadrži MOF fajlove za nodove. Ovaj server takođe odgovara na zahteve LCM (Local Configuration Manager) zahteve za distribucijom konfiguracija. To je upravo i prednost Pull moda jer se DSC resursi preuzimaju automatski sa servera od strane svakog noda, a zatim u regularnim intervalima nodovi šalju svoj status pull serveru da provere promene u konfiguraciji [4].

### 2.5 Push model

Push model je idealan u uslovima testiranja ili u okruženjima sa serverima gde se konfiguracija primenjuje jednom na duži period. Kod Push modela korisnik odlučuje da ručno primeni konfiguraciju na jedan ili više nodova.

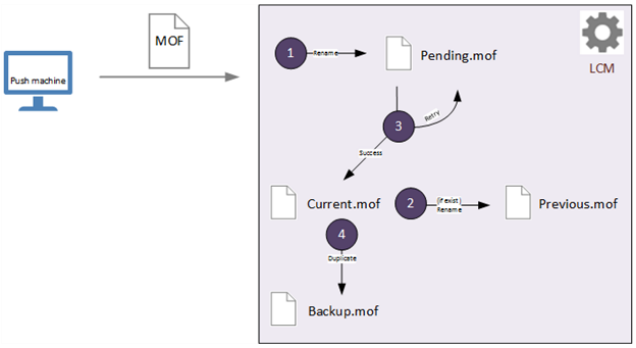



*Slika 3 – Prikaz Push modela*

Kao što je već napomenuto u fazi izrade stvara se DSC konfiguracija. To se može uraditi sa jednostavnim tekstualnim editorom poput Notepad ili PowerShell ISE. Kada se konfiguracija napravi onda se ona mora izvršiti kako bi se napravili jedan ili više MOF fajlova koji sadrže informacije za konfigurisanje nodova. Po nodu može postojati samo jedan MOF fajl. To omogućava da nod ima celokupnu konfiguraciju odjednom i osigurava da je primenio odgovarajuću konfiguraciju.
Nakon sastavljanja MOF fajlova na jednom računaru koristi se push model kako bi se konfiguracija prenela na druge računare uz pomoć Start-DSCConfiguration komandleta. Ovde postoji opcioni parametar -ComputerName koji omogućava korisniku da naznači na koji nod želi da primeni konfiguraciju. Ako nije naznačeno ime noda LCM utvrđuje sve fajlove koji se nalaze u konfiguracionom folderu kako bi identifikovao na koje nodove se fajlovi odnose.
Konačni korak je konfiguracija nodova sa informacijom primljenom preko MOF fajla. LCM odmah izvršava MOF fajlove nakon njihove analize. Fajlovi se smeštaju u svakom nodu na sledećem direktorijumu: **$env:systemRoot/system32/configuration**
Kada se izvrši push jednog MOF fajla na nod pojaviće se nekoliko MOF fajlova. [4]
Ti fajlovi su kreirani prema specifičnoj svrsi. Koraci u njihovom nastajanju su sledeći:

- Prvi korak: LCM prima i izvršava MOF fajl. Preimenuje ga u **pending.mof**
- Drugi korak (opcioni): ako postoji, preimenuje se aktuelni MOF fajl **current.mof** u **previous.mof**
- Treći korak: kada se nova konfiguracija primeni uspešno, preimenuje se **pending.mof** u **current.mof**
- Konačni korak: od **current.mof** se pravi duplikat **backup.mof**

Na sledećoj slici može se videti dijagram ovog procesa.




*Slika 4 - Dijagram procesa MOF fajlova*

## 3. PREDNOSTI DSC-a

Neke od osnovnih prednosti Desired State Configuration su sledeće:

- DSC je veoma lak za čitanje, čuvanje i promenu.
- Opisuje stanje u kom se treba naći ciljani uređaj umesto da sadrži instrukcije kako da uređaj dođe u željeno stanje.
- Manje košta da se nauči i implementira.
- Ponovne implementacije specifičnog seta mašina su mnogo manje podložne greškama.
- Mnogo brže se implementiraju promene konfiguracija i mnogo su pouzdanije implementacije tih promena.
- Konfiguracije se mogu deliti uz pomoć PS galerije, što znači da mogu postojati uobičajeni scenariji.
- Konfiguracije se može dodati u source kontrolu u verziji zajedno sa aplikacijom koja se razvija. [5]

## 4. IMPLEMENTACIJA DSC

Configuration funkcija je najvažnija komponenta DSC-a jer kao što je već napomenuto ona definiše željenu konfiguraciju sistema. Svaki Configuration fajl počinje na isti način.

```
Configuration Name
{
}
```

Nakon reči Configuration sledi ime ili identifikator, a potom sledi kod koji se nalazi unutar zagrada {}.
Konfiguracija se, u ovom radu, sastoji od sledećih koraka opisanih skriptama:

- LogOffUsers – odjavljivanje korisnika
- StopServices – zaustavljanje servisa
- Uninstall – uklanjanje instalacije
- Install – postavljanje nove instalacije
- ConfigSystem – kofigurisanje sistema
- DistribuSys – dobijanje instance CIM klasa
- Database – podešavanje baze podataka

## 5. REZULTATI TESTIRANJA

Bilo kakva vrsta automatizacije, kao osnovni rezultat ima uštedu potrošenog vremena na realizaciji nekog poslovnog procesa.
Tabela 1. prikazuje ukupno utrošeno vreme za automatizaciju. Prikaz podataka je dat hronološkim redosledom realizacije, kao što je prethodnom poglavlju objašnjeno.
***Napomena:** Za potrebe testiranja u ovom radu, razmatraće se vreme koje je potrebno pri reinstalaciji softvera na jednom mikro sistemu.

Tabela 1: Prikaz utrošenog vremena kod kod automatske reinstalacije pomoću DSC-a

| Automatizovani koraci | Vreme trajanja |
|---|---|
| **Odjaviljvanje svih korisnika sa sistema** | 7 sec |
| **Stopiranje servisa** | 25 sec |
| **Uklanjanje postojeće instalacije softvera** | 50 sec |
| **Postavka nove instalacije softvera** | 40 sec |
| **Distribusys** | 12 sec |
| **Konfiguracija** | 5 min |
| **Podešavanje baza** | 15 min |
| **Ukupno utrošeno vreme:** | **22 min i 21 sec** |

Tabela 2: Prikaz utrošenog vremena kod manuelnih koraka za reinstalaciju softvera

| Manuelni koraci | Vreme trajanja |
|---|---|
| **Odjavljivanje svih korisnika sa sistema** | 5 min |
| **Stopiranje servisa** | 10 min |
| **Uklanjanje postojeće instalacije softvera** | 15 min |
| **Postavka nove instalacije softvera** | 10 min |
| **Distribusys** | 15 min |
| **Konfiguracija** | 30 min |
| **Podešavanje baza** | 45 min |
| **Ukupno utrošeno vreme:** | **2h i 5 min** |

Kada se ovakav rezultat utrošenog vremena kod ručne reinstalacije sistema uporedi sa utrošenim vremenom automatskereinstalacije, zaključuje se da je automatizacija ovakvog poslovnog procesa od velike važnosti.

## 6. ZAKLJUČAK

Cilj ovog rada bila je automatizacija reinstalacije softvera na distribuiranom sistemu. Predstavljena je automatizacija u upravljanju konfiguracijama, odnosno Desired State Configuration, njena arhitektura, prednosti kao i razlozi za uvođenje.
Dat je i pregled razvoja DSC-a od prve verzije pa do verzije koja se danas koristi.
Na samom kraju rada iznet je i praktični test sa rezultatima koji jasno govore prednost automatizacije u odnosu na klasične pristupe upravljanja konfiguracijama.
Sam proces automatizacije konfiguracionih koraka za reinstalaciju softvera, dosta je ubrzao rad na distribuiranom sistemu. Prilikom manuelnog izvršavanja ovih konfiguracionih koraka, potrebno je u proseku oko 3 do 4 sata, u zavisnosti od veličine sistema, dok je za ceo proces sa automatizacijom pomenutih koraka potrebno oko 25 minuta, što je znatno ubrzalo sam proces.
Ostale bitne karakteristike i prednosti ovakve automatizacije ogledaju se i u:

- Ponovljivost koraka.
- Minimalizovan faktor ljudske greške.
- Poboljšane performanse koje daju bolje rezultate.

Jedan od daljih koraka razvoja jeste modifikacija i primena navedenih koraka automatskog konfigurisanja na većim distribuiranim sistemima, jer je u ovom radu ovaj proces automatskog konfigurisanja primenjen na mikro (malom) distribuiranom sistemu.
Jasno je da je automatizacija neophodna i da će se u budućnosti sigurno tražiti načini za njeno unapređenje jer se softveri razvijaju sve većom brzinom i postaju sve više i više kompleksniji.

## 7. LITERATURA


[1] Windows PowerShell Desired State Configuration Revealed - Ravikanth Chaganti, izdata 23.09.2014
[2] https://www.itprotoday.com/management-mobility/windows-powershell-40-released-windows-7-windows-2008-r2-and-windows-server-2012, datum pristupa 23.09.2018.
[3] Learning PowerShell DSC Second edition – James Pogran, izdata 19.10.2015
[4] https://www.red-gate.com/simple-talk/sysadmin/powershell/powershell-desired-state-configuration-the-basics/,datum pristupa 22.09.2018.
[5] https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/?view=powershell-6,datum pristupa 22.09.2018.
[6] Pro PowerShell Desired State Configuration – Ravikanth Chaganti, izdata 25.04.2018.


**Kratka biografija:**

**Jelena Spasojević** rođena je 1990. godine u Kragujevcu. Srednju školu je završila u Kragujevcu 2009. godine. Fakultet Tehničkih Nauka u Novom Sadu je upisala 2009. godine. Ispunila je sve obaveze i položila sve ispite predviđene studijskim programom na smeru Energetika, elektronika i telekomunikacije, usmerenju Telekomunikacioni sistemi. Odmah nakon toga je upisala master akademske studije, takodje na istom na smeru i ispunila sve svoje obaveze i položila sve ispite predviđene studijskim smerom.