



KRIGSSKOLEN

KS ING

BACHELOROPPGAVE I INGENIØRFAG
MED FORDYPNING I MILITÆRGEOGRAFI

Automatisering av Etterretningsvurdering Trinn 2 for MilGeo i Hæren

Aqeel Ahmed, Henrik Østmoe Strand og Kjetil Waal

Kull Bing Broch 11-14

**KRIGSSKOLEN**

Norwegian Military Academy

P O Box 42

NO-0517 Oslo NORWAY

UGRADERT

REPORT DOCUMENTATION PAGE

TITTEL: AUTOMATISERING AV ETTERRETNINGSVURDERING TRINN 2 FOR MILGEO I HÆREN	
UTFØRT VED: KRIGSSKOLEN	RAPPORTNR:
OPPDRAGSGIVER: HÆRENS VÅPENSKOLE	
PROSJEKTDeltakere: AQEEL AHMED HENRIK ØSTMØE STRAND KJETIL WAAL	PROSJEKTPERIODE: FRA: 06.01.2014 TIL: 09.05.2014
VEILEDERE: INTERN VEILEDER: HALVARD BJERKE (KRIGSSKOLEN) EKSTERN VEILEDER: STEIN ARNE FEDREHEIM (HÆRENS VÅPENSKOLE)	ANTALL SIDER: HOVEDDOKUMENT: 34 VEDLEGG: 35 TOTALT: 69
EKSTRAKT: BAKGRUNNEN FOR OPPGAVEN ER HVS SIN INTERESSE FOR AUTOMATISERING AV ETTERRETNINGSVURDERING TRINN 2. I OPPGAVEN SES DET PÅ HVORDAN TEKNISKE ARBEIDSPROSESSER I ANALYSEVERKTØYET ARCMAP KAN TILRETTELEGGES MED DATA OG AUTOMATISERES FOR Å SPARE ANALYTIKEREN FOR TID. OPPGAVENS HOVEDTYNGDE LIGGER I PRODUKSJONEN AV ET TEKNISK FORSLAG TIL EN LØSNING SOM BLANT ANNET BYGGER PÅ INTERVJUER OG DOKUMENTSTUDIER. DET KONKLUDERES MED AT EN E-VURDERING TRINN 2 KAN AUTOMATISERES MED ET PRODUKT SOM ER EN KOMBINASJON AV TILRETTELAGT DATA OG TEKNISK AUTOMASJON.	
GRADERING: UGRADERT	
OMSLAGSBILDE: VÅPENSKJOLD KRIGSSKOLEN	
GODKJENT:	
LINDERUD: _____	_____
DATO	VEILEDER



Forord

Hovedoppgaven er skrevet som et ledd av den treårige bachelorutdanningen i ingeniørfag med fordypning i Militær Geografi (MilGeo) ved Krigsskolen (KS) i Oslo. Oppgaven utgjør 15 studiepoeng i det avsluttende semesteret.

Oppgaven omhandler temaet automatisering innenfor MilGeo i Hæren og ble introdusert for oss gjennom Hærens Våpenskole (HVS). Det er et spennende tema med muligheter for utvikling, og arbeid innenfor temaet vil bidra til å gjøre arbeidsprosessene til MilGeo enklere.

Vi vil benytte muligheten til å takke alle som har bidratt gjennom denne læringsprosessen. Vi ønsker å takke Major Anders Calder Haavik Nilsen for god veiledning på oppgavens oppbygning. Spesielt vil vi rette en takk til veiledere Høyskole lektor Halvard Bjerke og Major Stein Arne Fedreheim for god faglig veiledning.

Oslo, 9.mai 2014

Aqeel Ahmed

Henrik Østmoe Strand

Kjetil Waal



Sammendrag

Utviklingen innen fagfeltene Informasjonsteknologi (IT) og Geografiske Informasjonssystemer (GIS) er stor og med dette åpnes mange muligheter for MilGeo i Hæren. HVS ser muligheten til å automatisere en del av arbeidsprosessene til de som jobber i fagfeltet. Spesifikt er det ønskelig med en fullstendig automatisering av Etterretningsvurdering trinn 2 (E-vurdering trinn 2) og dette vil bidra til å frigjøre tid slik at analytikerne kan støtte med vurderinger i enda større grad. GIS består av en teknisk og en menneskelig del. I denne oppgaven har vi valgt å se på den tekniske delen og muligheter for å automatisere den.

Vi har kommet frem til følgende problemstilling:

«Hvordan kan Etterretningsvurdering trinn 2 automatiseres for MilGeo i Hæren?»

Det empiriske grunnlaget kommer primært fra intervjuer med fagpersonell i Ingeniørbataljonen (INGBN), i tillegg har vi hatt samtaler med HVS for å kartlegge behov. En kombinasjon av faglig standpunkt og tilegning av relevant kunnskap har hjulpet oss til å produsere et teknisk forslag som danner grunnlaget for automatisering av E-vurdering trinn 2.

Det konkluderes med at en E-vurdering trinn 2 kan automatiseres med et produkt som er en kombinasjon av tilrettelagt data og ren teknisk automasjon. Produktet er vedlagt oppgaven med en forklaring til oppbygning, bruk og hvordan skriptverktøyet kan endres.



Innholdsfortegnelse

Forord	ii
Sammendrag	iii
Innholdsfortegnelse	iv
Figurliste	vi
Oppgavens disposisjon	vii
1. Innledning.....	1
1.1 Bakgrunn	1
1.2 Problemstilling.....	3
1.3 Mål.....	3
1.3.1 Effektmål	3
1.3.2 Resultatmål.....	3
1.3.3 Læringsmål	3
1.4 Problemanalyse	4
1.5 Faglig standpunkt	6
1.6 Avgrensninger.....	6
2. Metode	8
2.1 Empirisk grunnlag	8
2.2 Intervju	8
2.3 Dokumentundersøkelse	9
2.4 Annen data	9
2.5 Metodekritikk.....	9
2.6 Kildekritikk.....	10
3. Teori.....	11
3.1 Etterretningsvurdering	11
3.2 Kartdokument.....	13
3.3 Layerfiler.....	14
3.4 Geodatabaser	15
3.5 Verktøybokser	15



3.6 Model Builder	16
3.7 Python	16
4. Sammendrag av intervjuer	17
5. Diskusjon	18
5.1 Empiri	18
5.2 Produkt	19
6. Konklusjon	22
7. Veien videre.....	23
8. Referanseliste og Kilder	24
8.1 Litteratur.....	24
8.2 Nettsider	25
8.3 Intervjuobjekter.....	25
9. Ordforklaring	26
Vedlegg 1	



Figurliste

Figur 1: Konseptskisse	5
Figur 2: Arcmap grafisk brukergrensesnitt.....	13
Figur 3: Eksempel layerfiler	14
Figur 4: Geodatabaser	15
Figur 5: Verktøybokser	15
Figur 6: Model Builder	16
Figur 7: Python eksempel.....	16



Oppgavens disposisjon

For at leseren skal få de rette inngangsverdiene har vi inkludert dette avsnittet. Dette er en teknisk oppgave som er lagt på et nivå hvor grunnleggende kjennskap til ArcGIS er nødvendig. I tillegg er det fordelaktig hvis man har kjennskap til Python.

Innledningen danner et bakteppe for oppgaven og forklarer hva vi legger i problemområdet. Dette former en problemstilling som det blir gjort en problemanalyse på. Etter at kjernen i problemet er identifisert, forklares og begrunnes metodevalget. Videre vil teorien og datainnsamlingen bygge grunnlaget for diskusjonen som leder til en konklusjon. Avslutningsvis er det produsert og vedlagt et teknisk forslag til en løsning på problemet.



1. Innledning

1.1 Bakgrunn

Gruppens ambisjon med denne oppgaven er å levere et produkt som er relevant og som i fremtiden kan tjene som et grunnlag for videre arbeid innenfor fagfeltet MilGeo.

Gjennom samtaler med HVS har det blitt presentert følgende:

«Utviklingen innen fagfeltene IT og GIS er veldig stor. Prosessorkraften og datamengden øker, noe som gir mange muligheter for oss innen MilGeo. Spesielt innen automatisering mener vi ved HVS at det er muligheter. En del prosesser kan automatiseres, noe som gir oss tid til å rive oss løs fra tastaturet og bidra med terrengrelaterte vurderinger og anbefalinger.»¹

Videre:

«Fagfeltet har tidligere jobbet mye med automatisering. Det har blitt utviklet en stor mengde modeller internt i avdelingen og fagøvelser arrangert ved HVS. Men disse brukes i varierende grad i avdelingene.»²

Som det kommer frem av sitatene er utviklingen innenfor fagfeltene IT og GIS stor. Dette skaper forutsetningene som trengs for MilGeo sin utvikling innenfor blant annet automatisering. Alt som bidrar til å forenkle hverdagen til MilGeo, vil kunne frigjøre tid for oppgaver som man mener fortjener større bidrag. Per i dag brukes mye tid på gjentakende tekniske prosesser i analyseverktøyet ArcMap. Eksempler på dette er tilrettelegging av data før analyser, organisering av dem og det praktiske analysearbeidet.

En forklaring av hva GIS er skildrer omfanget det er snakk om.

¹ Major Stein Arne Fedreheim, Hovedinstruktør MilGeo/ Ingeniørskolen HVS i E-mail oktober 2013

² Major Stein Arne Fedreheim, Hovedinstruktør MilGeo/ Ingeniørskolen HVS i E-mail oktober 2013



En definisjon av GIS:

«Et geografisk informasjonssystem er samlingen av kartsystemer, geografiske data, rutiner og menneskelig kunnskap og erfaring som gjør det mulig å fremstille, analysere og presentere geografien rundt oss ved hjelp av digital teknologi»³

Som man ser av definisjonen består GIS av to aspekter, det tekniske aspektet og det menneskelige aspektet. Sistnevnte er ikke like enkelt å automatisere, men er essensielt i en analyse. Hvis en menneskelig faktor ikke er involvert, er det vanskelig å si at det har blitt gjort en vurdering. Ved å automatisere de gjentakende tekniske prosessene i en analyse, vil man frigjøre tid som den menneskelige delen i enda større grad kan bruke til vurdering. I denne oppgaven har vi tatt for oss den tekniske delen og ser på muligheter for å automatisere den. Dette er et spennende tema som vil utfordre gruppen på det tekniske planet og kreve at vi utvider horisonten vår innenfor GIS.

Fra HVS har det kommet følgende:

«Vi kunne tenkt oss en fullstendig automatisert terrenganalyse (Trinn 2 IPB). Dette er kanskje en ganske heftig jobb å få til, men definitivt relevant for Hæren.»⁴

ArcMap er hærens primære analyseverktøy og inngår i pakken *ArcGIS for Desktop*. Verktøyet brukes av MilGeo i Hæren og produktene som leveres er varierte. Analysene brukes i stor grad som beslutningsstøtte til operasjoner i både inn- og utland. MilGeo leverer også spesifikke produkter som bidrar til oppdragsstøtte. Selv om produktene er varierte, finnes det eksempler på analyser og produkter som det ofte dukker opp behov for. Slike analyser og produkter kan for eksempel være E-vurderinger, Helikopterlandingsplasser, Objektpakker og Målpakker. Det

³ Grindrud, Knut, Haakon Rasmussen, Steinar Nilsen, Arvid Lillethun, Atle Holten, Øystein Sanderud. GIS: Geografiens språk i vår tidsalder. Trondheim: Tapir Akademisk Forlag, 2008

⁴ Major Stein Arne Fedreheim, Hovedinstruktør MilGeo/ Ingeniørskolen HVS i E-mail oktober 2013



finnes i dag muligheter i programvaren som kan forenkle noen av disse arbeidsprosessene. Som et bidrag til automatisering har det tidligere vært skrevet om og oppfordret til bruk av de mulighetene som finnes i programvaren. Model builder (MB) er eksempel på et verktøy som effektiviserer og skaper automasjon i prosessene, men verktøyet brukes i varierende grad av MilGeo. Python er et annet eksempel på et slikt verktøy, men brukes i liten grad.

Det overordnede målet med oppgaven er å gi et grunnlag for automatisering og forenkling av ofte brukte tekniske prosesser og data som er forbundet med E-vurdering trinn 2.

1.2 Problemstilling

Det finnes et utvalg av prosesser som kan automatiseres og flere av dem henger sammen. I samråd med ønsker fra HVS har gruppen kommet frem til følgende problemstilling:

«Hvordan kan Etterretningsvurdering trinn 2 automatiseres for MilGeo i Hæren?»

1.3 Mål

1.3.1 Effektmål

Oppgaven skal gi et forslag til HVS for hvordan en automatisert E-vurdering trinn 2 kan struktureres. Dette innebærer et felles verktøygrunnlag som i enda større grad kan sikre likhet i sluttproduktets oppbygning og bidra til å øke effektiviteten i analysene.

1.3.2 Resultatmål

Oppgaven har som mål å levere et grunnlag for automatisering av E-vurdering trinn 2. Målet er nådd når et forslag til en løsning foreligger og oppgaven er overlevert HVS.

1.3.3 Læringsmål

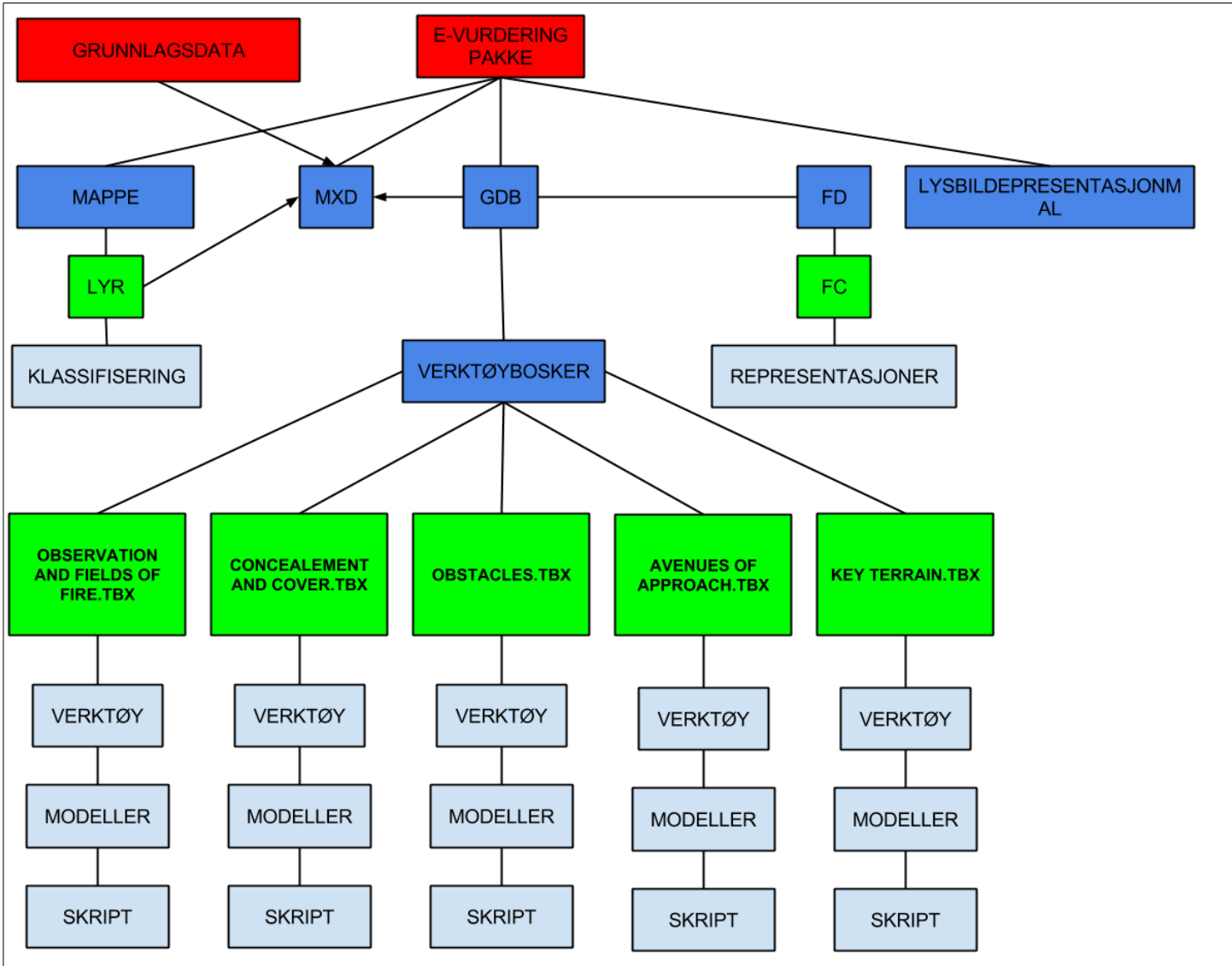
Oppgaven vil gi oss en bedre forståelse for hvordan MilGeo jobber opp mot en E-vurdering



trinn 2 og hva slags produkter som leveres. Videre vil den gi oss kunnskap om hvilke parametere de forskjellige avdelingene jobber med, samt en bedre forståelse for mulighetene programvaren gir innenfor automatisering.

1.4 Problemanalyse

Innledningsvis var tanken at automatisering av E-vurdering trinn 2 utelukkende ville løses gjennom bruk av MB. Da vi begynte på problemanalysen kom det fort frem at E-vurdering trinn 2 ikke vil kunne automatiseres hundre prosent. Det er naturlig å tro at den menneskelige faktoren alltid vil og bør være tilstede. Tross alt spiller vurderingene til analytikerens en betydelig rolle i analysen. Videre ble vi mer bevisst på kjernen i problemet. Man ønsker å oppnå tidsbesparelser for å kunne bidra med støtte og vurderinger i eksempelvis gjennomføringsfasen av en operasjon. Automatisering gjennom MB er bare et bidrag til dette. Det finnes andre faktorer som nødvendigvis ikke er like tekniske, men som også kan bidra til tidsbesparelser. Ett eksempel på en slik faktor er tilrettelegging av ofte brukte data. Et annet eksempel er eksport av bilder fra ArcMap og inn i PowerPoint. Problemanalysen gjorde gruppen bevisst på at vi må tenke på faktorer som sparer tid og ikke isolere oss til en ren teknisk tilnærming som MB. Det var viktig for oss å kartlegge alle de faktorene som spiller inn i E-vurdering trinn 2 for å se hvor omfattende problemområdet er. Basert på tiden vi hadde til oppgaven og det vi så på som relevant for utformingen av grunnlaget, fokuserte vi på å identifisere de viktigste faktorene. Under er det en konseptuell skisse av det vi kaller en «E-vurderingspakke» som inneholder disse faktorene. Ambisjonen er at denne pakken skal inneholde det som trengs for en sømløs gjennomføring av E-vurdering Trinn 2.



Figur 1: Konseptskisse



1.5 Faglig standpunkt

Gruppens geofaglige standpunkt består av fagene Databaser som gikk over høsten 2012, Geomatikk 1 over våren 2013, Advanced GIS over høsten 2013 og GEOINT over våren 2014. Gruppen har i tillegg hatt et tre dagers kurs i Geodatabaser (GDB'er), en kadettøvelse i samarbeid med HVS på Rena og Cold Response 2014 (CR14) hvor enkelte satt som MilGeo. ArcMap har mange bruksområder og vi er godt kjent med mange av verktøyene som kan benyttes til militære formål, men det er fortsatt utallige muligheter å utforske i programvaren.

1.6 Avgrensninger

Vi har foretatt noen avgrensninger av temaer som er høyst relevante, men blir for omfangsrike for denne oppgaven. Tatt i betraktning tidsperspektivet blir det urealistisk å lage en løsning som skal ta høyde for alle avdelinger i Hæren, men det tilstrebes å få et grunnlag man kan jobbe ut i fra og senere skreddersy. Dette innebærer at pakken etter hvert kan inneholde verktøy som tar høyde for alle avdelingers kapasiteter og ressurser. Med dette kommer spesifikke verktøy og enda mer tilrettelegging av data. UTM soner kan være en utfordring hvis man skal kjøre modeller i områder hvor det er et soneskille. Over mindre områder gir dette marginale utslag, men over store områder som for eksempel hele Norge kan dette få større konsekvenser. Vi ser ikke på denne utfordringen i vår oppgave.

Oppgaven er ugradert primært på grunn av lettere tilgjengelighet i fremtiden. Det meste av data og informasjon finnes ugradert og fordi vi produserer en teknisk løsning, blir arbeidsprosessen forenklet ved å jobbe på ugraderte maskiner.

I og med at prosjekt 8009 ikke er ferdig, vet vi ikke hva oppbygningen av servere og maskinpark vil innebære for MilGeo i Hæren. På grunn av dette velger vi å se bort i fra hvordan deling av informasjon skal skje. Dette innebærer deling av modeller, verktøy, data og Web Map Service tjenester (WMS). Dynamiske data har en sentral rolle i lende vurderingen, men på grunn av temaets omfang har vi valgt å se bort i fra dette.



Videre finnes det ikke et overordnet dokument for MilGeo i Hæren, noe som gjør det vanskelig å definere symbologi og dette er derfor et tema vi ikke ser på. I og med at tiden på lende vurderingen varierer, blir det vanskelig å si nøyaktig hvor mye tid man sparer ved automatisering. Dette krever tid og omfattende undersøkelser av den løsningen vi kommer frem til. Løsningen vil derfor ikke testes for tidsbesparelser i løpet av bachelorperioden. En lysbildemal er noe som bør være inkludert, men alle avdelinger har sine faste oppsett. Vi har valgt å ikke lage en fordi dette allerede er godt innarbeidet i avdelingene. INGBN støtter flere avdelinger og gjennomfører mange lende vurderinger. De har et godt bilde av utfordringene rundt automatisering og arbeidsprosessene i fagfeltet. Vi har derfor avgrenset oss fra andre avdelinger, og intervjuene har kun blitt gjennomført med fagpersonell i INGBN.



2. Metode

Metodekapittelet er basert på boken «*Hvordan gjennomføre undersøkelser?*»⁵ av Dag Ingvar Jacobsen. Metode er en strukturert måte å gå frem på for å samle informasjon om virkeligheten. Vi har i denne oppgaven valgt kvalitativ metode. Dette er en metode hvor kunnskapen opparbeides ved å undersøke hva hendelser og erfaringer har å si for personene som opplever dem. I motsetning til kvantitativ metode søker kvalitativ metode å gå i dybden på et mindre antall personer.

En kvalitativ metode var naturlig for problemstillingen vår og dette kan begrunnes i at det finnes lite empiri innenfor fagfeltet MilGeo. Videre finnes det lite dokumentasjon i form av håndbøker eller andre offisielle dokumenter bortsett fra «*Stabshåndboken for Hæren – Plan- og beslutningsprosess*»⁶. Kunnskapsnivået vårt var også begrenset, noe som gjorde at vi med en kvalitativ metode fikk den informasjonen vi trengte for å skrive oppgaven.

Fagpersonenes syn på problemområdet var essensielt for å kartlegge de mulighetene og tiltakene som kunne gjøres for å oppnå et best mulig resultat.

2.1 Empirisk grunnlag

Det empiriske grunnlaget er basert på intervjuer og dokumentstudier.

2.2 Intervju

Vi har hatt åpne intervjuer med både gruppe og enkeltindivid.

Grunnen til dette er at åpne intervjuer egner seg godt for å utvikle ny kunnskap om et ukjent tema. Intervjuene var åpne i form at vi brukte intervjuguide med tema, fast rekkefølge og kun åpne svar. Ved å intervjuer en gruppe fikk vi ikke bare avdekket felles synspunkter på

⁵ Dag Ingvar Jacobsen. *Hvordan gjennomføre undersøkelser?*, 2. utgave. Høyskoleforlaget, 2005

⁶ Hærens skole for taktikk og operasjoner. (HSTO). *Stabshåndbok for Hæren - Plan- og beslutningsprosessen*. Rena: HVS, 2010



problemområdet, men også hvor det var uenigheter. Individuelt intervju ga oss muligheten til å se på hvordan korrelasjonen var mellom gruppe og individ. For å kontrollere notater og sikre at informasjon ikke skulle gå tapt, ble det gjort båndopptak av intervjuene i sin helhet. Personene som har blitt gjort opptak av har ikke blitt sitert i oppgaven.

2.3 Dokumentundersøkelse

For å bygge opp om oppgaven har vi hentet empiri fra e-brev, lysbildepresentasjoner av E-vurderinger, stabshåndboken⁷ og en amerikansk håndbok⁸. Siden det i dag ikke foreligger noen håndbøker på hvordan E-vurdering trinn 2 skal leveres som et produkt, er det variasjoner som baserer seg på erfaringer og skjønn hos den enkelte. Derfor har det vært viktig for oss å se på hvordan andre har gjort det.

2.4 Annen data

E-vurdering har mange faktorer som spiller inn på det endelige resultatet. For å kunne kartlegge hvordan man kan automatisere en E-vurdering trinn 2 er det samlet inn annen type data. Dette er data som type filstruktur, lysbildefremvisninger og elektroniske verktøy. Oppgaven bygger også på erfaringer fra NATO øvelsen Cold Respons 2014 (CR14), hvor enkelte i gruppen satt i MilGeo stillinger.

2.5 Metodekritikk

Generelt er det kritikk som kan rettes mot metoden i seg selv. Metoden bygger mye på personers erfaringer og deres synspunkter, noe som gjør at man ikke vil komme frem til noen bastante svar. En annen begrensning ved metoden er at man samler inn data fra få personer, noe som gjør at man kanskje ikke får generalisert funnene. Bruk av båndopptaker under intervjuene kan være en begrensning i form av at det kan virke hemmende for

⁷ Hærens skole for taktikk og operasjoner. (HSTO). *Stabshåndbok for Hæren - Plan- og beslutningsprosessen*. Rena: HVS, 2010.

⁸ <http://www.fas.org/irp/doddir/army/fm34-130.pdf>



intervjuobjektene. Ved åpne intervjuer vil det komme frem en god del ikke relevant informasjon, noe som kan gjøre det tidkrevende å sile ut den informasjonen som er relevant for oppgaven. En kvalitativ undersøkelse er i utgangspunktet en prosess som krever mye tid. Hvis man ser dette i lys av prosjektperiodens varighet, så er dette en begrensning i form av at man ikke får samlet inn all den informasjonen man trenger.

2.6 Kildekritikk

Alle kildene har ikke det samme erfaringsgrunnlaget, dette kan påvirke oppgavens validitet i form av de funnene som er gjort. Det vil alltid være en viss grad av subjektivitet involvert fra intervjuobjektene side og fra oss som gruppe. Vi har prøvd å gjøre funnene så objektive som mulig ved å sparre opp mot HVS som styrende avdeling.



3. Teori

3.1 Etterretningsvurdering

E-vurdering av stridsfeltet er beskrevet i «*stabshåndboken for Hæren-plan og beslutningsprosess*». Håndboken er skrevet med utgangspunkt i «*NATO's Comprehensive Operations planning Directive (COPD)*» og skal tjene som et utgangspunkt for all planlegging i Hæren⁹. Slik får man forsikret en tilstrekkelig vurdering av alle forhold som er av betydning for en operasjon. Videre i håndboken beskrives det at etterretningsvurdering og metodisk målbekjempning er to delprosesser i plan og beslutningsprosessen (PBP) som danner grunnlaget for et godt produkt.

E-vurdering, tidligere Intelligence preparation of the Battlefield/Battlespace -IPB, er en analyse av omgivelsene og aktørene i et geografisk område. Selv om den utføres på alle nivåer i hierarkiet blir den formelle analytiske prosessen hovedsakelig utført på bataljonsnivå og oppover. Ansvar tillegges S-2 på bataljonsnivå og G-2 på brigadenivå i Hæren. E-vurdering er en firetrinnsprosess for analyse av stridsfeltet. Det er en kontinuerlig prosess som skal utføres både under planleggingen og gjennomføringen av en operasjon.

E-vurderingens fire trinn: Definere operasjonsområdet, beskrive operasjonsområdets innvirkning, beskrive aktørene og utvikle aktørenes handlemåter. Trinn to, beskrive operasjonsområdets innvirkning, kan deles inn i tre hovedområder som skal analyseres: Vær føre og lys, lende-vurdering og andre forhold. MilGeo bidrar med støtte inn mot alle tre områder for å kunne gi en best mulig vurdering av stridsfeltets innvirkning på operasjonen og lende-vurderingen er den mest omfattende av de tre.

⁹ Hærens skole for taktikk og operasjoner. (HSTO). Stabshåndbok for Hæren - Plan- og beslutningsprosessen. Rena: HVS, 2010, s.3



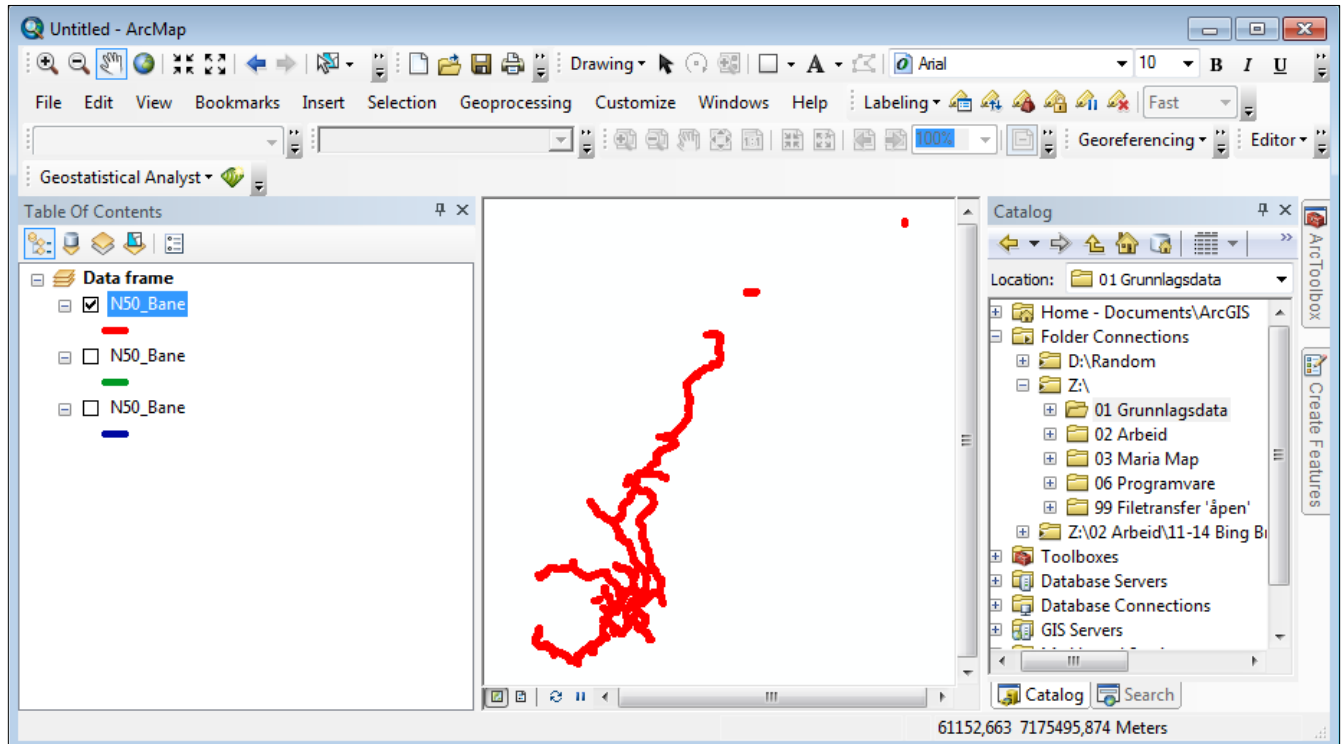
Lendevurderingen gjennomføres og presenteres etter huskeordet VODKA for offensive operasjoner og AKVOD for defensive operasjoner. Vi har tatt utgangspunkt i den amerikanske OCOKA, fordi den blant annet brukes på brigade og bataljonsnivå i Hæren. Den eneste forskjellen er rekkefølgen i stegene. Under følger en oversikt over VODKA og OCOKA.

Terrengvurdering VODKA¹⁰	Terrengvurdering OCOKA
Viktig lende (V)	Observation and fields of fire (O)
Observasjon og skuddfelt (O)	Cover and concealment (C)
Dekning og skjul (D)	Obstacles (O)
Kunstig og naturlig hinder (K)	Key terrain (K)
Akser som kan brukes til fremrykning (A)	Avenues of approach (A)

¹⁰ Hærens skole for taktikk og operasjoner. (HSTO). Stabshåndbok for Hæren - Plan- og beslutningsprosessen. Rena: HVS, 2010, s.107

3.2 Kartdokument

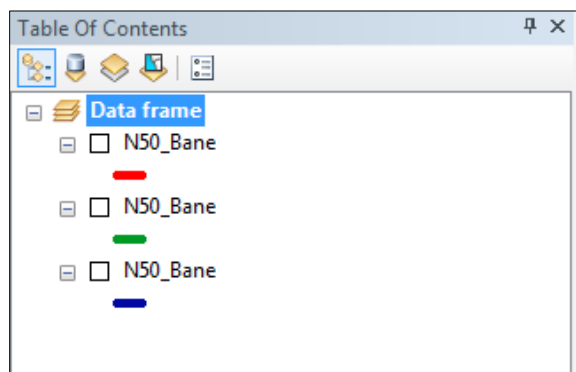
ArcMap sin grafiske brukergrensesnitt er der man gjør de fleste tekniske analysene. Det er her dataene hentes inn i et tomt dokument og bearbeides. Eksempelet nedenfor og alle kartdokumenter i Arcmap lagres i filformatet MXD. Arbeidet som gjøres og alle data som ligger inne i dokumentet blir lagret. I bildet under ser vi fra venstre: «*Table Of Content (TOC)*», «*Data View*» og «*Catalog*».



Figur 2: Arcmap grafisk brukergrensesnitt.

3.3 Layerfiler

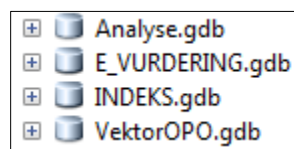
Filformatet i ArcGIS gir muligheter til å lagre symboliseringen for ønskelige filer. Alle data har en symbolisering i ArcMap, men den er ulik hver gang de hentes inn og legges i *TOC'en*. Ved å sette ønskelig symbolisering på dataene og lagre dem som en lyrfile vil man slippe å endre på symboliseringen hver gang de hentes inn. Man henter bare inn lyrfile som peker til dataene og beholder symboliseringen. Bildet under viser *N50_bane* hentet inn tre ganger. Den får ulik symbolisering hver gang. Dette kan unngås ved å bruke lyrfiler og man sparer tid når det begynner å bli mye data. Spesielt nyttig er det hvis man er usikker på hvilken symbolisering dataene skal ha, da er det praktisk med lyrfiler hvor symboliseringen er bestemt.



Figur 3: Eksempel layerfiler

3.4 Geodatabaser

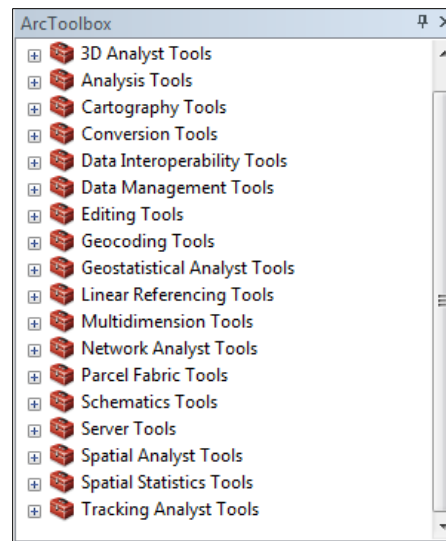
GDB'er er på sitt grunnleggende nivå et felles lagringssted for geografiske data av ulik type, og er ArcGIS sin primære dataformat. Alle dataelementer som støttes av ArcGIS kan legges i en GDB. Man har også muligheten til å dele GDB'er, og en database kan ha flere brukere. En GDB har en omfattende informasjonsmodell for å representere og behandle geografisk informasjon. Denne består av en serie tabeller som inneholder klasser, datasett og attributter. For GDB'er det også mulig å sette rettigheter for hvordan og hvem som kan behandle dataene.



Figur 4: Geodatabaser

3.5 Verktøybokser

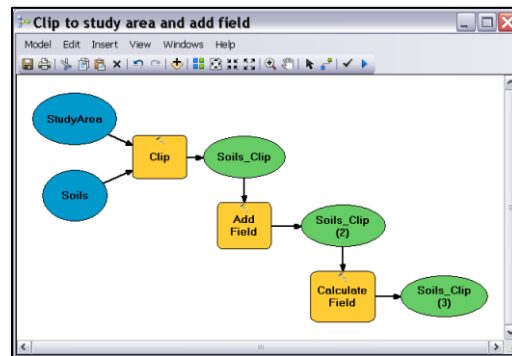
Man kan opprette egne verktøybokser for å lage en samling av ønskelige verktøy. De gir rask tilgang til ofte brukte verktøy og kan eksempelvis legges i en GDB.



Figur 5: Verktøybokser

3.6 Model Builder

MB er en applikasjon som er tilgjengelig i ArcMap og ArcCatalog. Den gir brukeren muligheten til å kjøre data gjennom x-antall verktøy uten opphold. MB er et viktig bidrag til automatisering i den forstand at man slipper å bearbeide data gjennom verktøy enkeltvis. Modellene som lages kan lagres, deles med andre, eksporteres til skript eller tas opp igjen og brukes ved behov. MB er en fleksibel applikasjon og man kan sette opp en arbeidsprosess med nesten alle typer verktøy som ArcMap tilbyr.



Figur 6: Model Builder

3.7 Python

Python er objektorientert programmeringsspråk som er utviklet av Guido Van Rossum, og ble lansert i 1991.¹¹ Det er en utbredt programvare som kan kjøre på flere plattformer. Den er også integrert i ArcGIS, og kan åpnes direkte i ArcMap eller ArcCatalog som et eget dialogvindu med hjelpetekst. Med Python får man tilgang til alle funksjonene i ArcMap og kan skrive alt fra enkle til omfattende skript for en hvilken som helst arbeidsprosess i programvaren.

```
# Importing arcpy
#
import arcpy

# Set the workspace environment and run Clip_analysis
arcpy.env.workspace = "C:/Data/Tongass"
arcpy.Clip_analysis("standb4", "clipcov", "standby_clip", "1.25")
```

Figur 7: Python eksempel

¹¹ <https://docs.python.org/2/license.html>



4. Sammendrag av intervjuer

Vi har primært gjennomført intervjuer med Ingeniørbataljonen og hatt noen samtaler med HVS. Det som kommer frem er at god tid på en E-vurdering trinn 2 regnes til å være tre dager. Generelt gjøres den totale vurderingen på den tiden man har og da blir naturligvis kvaliteten deretter. MB nyttes i noen tilfeller av den enkelte analytiker for å automatisere arbeidsprosessene, men det finnes ingen organisering for deling av disse modellene. De deles gjerne innad i avdelingen, men ellers gjøres det uformelt og på forespørsel. Det kommer frem at man har for lite tid til å fokusere på automatisering, og at dette er noe man burde hatt. «*Slope*» og «*Viewshed*» er blant de analysene som gjøres ofte i lende-vurderingen. I tillegg er «*hillshade*» et virkemiddel som brukes ofte for å kunne fremheve konturer i terrenget. Videre kommer det frem at tilrettelagt lende-vurderingsdata både i form av en MXD-mal og i en struktur ville bidratt til at lende-vurderingen går raskere. Det ble også nevnt at en slik organisering vil fasilitere for deling av eksempelvis modeller. Hva gjelder symbolisering, finnes det lyrfiler som er levert av Forsvarets Militærgeografiske Tjeneste (FMGT) og man prøver å symbolisere på skjønn de gangene M711 symboliseringen ikke kommer frem tydelig nok. Et annet interessant moment som kom frem uformelt etter intervjuene, var eksport av de forskjellige analysene. Isteden for at man skulle eksportere de enkelte lagene fra *TOC'en* selv, ville arbeidsprosessen blitt effektivisert om dette kunne gjøres automatisk.



5. Diskusjon

5.1 Empiri

Lendevurderingen gjøres generelt på den tiden man har til disposisjon og i den tekniske delen er det mange muligheter for automasjon. Man må samtidig være bevisst på at ved å definere og automatisere analysene alt for mye, kan det være fare for å sette en demper på kreativiteten. Ikke minst kan man stå i fare for å ta friheten vekk fra analytikeren. Tar man derimot høyde for dette og har en balansegang, er det gunstig med automatisering.

MB og Python er svært like, begge er programmeringsspråk. Forskjellen er at MB er et visuelt programmeringsspråk mens Python er tekstbasert. For å gjøre det enda tydeligere er alle verktøy i MB visuelle representasjoner av tekstbaserte kode. Med Python har man i enda større grad muligheten til å skreddersy arbeidsprosessene sine, mens man med MB er begrenset til de ferdigskrevne verktøyene fra ESRI. Videre takler Python mer avansert programmeringslogikk og er fleksibel i den forstand at man har tilgang til flere funksjoner som MB ikke har. I og med at språket er så omfattende og versatilt som det er, har man muligheten til å interagere med alle programmer som er støttet i Python biblioteket. Eksempler på dette er Excel eller PowerPoint. En annen funksjonalitet Python har er at det kan kjøres som et eget program utenfor ArcGIS, det betyr at man har muligheten til å sette opp tidsbestemte oppgaver og kjøre analyser selv om ArcMap ikke er åpnet.

På den ene siden er MB et stort bidrag til automatisering og forenkling av arbeidsprosessene i ArcMap. Den gir en god visuell representasjon av arbeidsflyten, og modellene kan deles med andre. På den andre siden skjer ikke deling i stor nok grad. Grunnen kan være at det ikke finnes noen organisering eller struktur som kan fasilitere for det. Dette kan føre til lange opphold mellom hver gang man bruker MB og da kan det tenkes at implementeringen av den i ens daglige rutine ikke blir enkel. I og med at det ikke er et stort fokus rundt det, kan det tenkes at den enkelte analytikeren heller velger å gjøre analysene på den kjente måten isteden for å bruke tiden man har på å lage modeller.



Fordelen med Python er at det er et mer fleksibelt språk enn MB og takler avansert programmeringslogikk. Her har man i enda større grad muligheten til å skreddersy arbeidsprosessene sine og lage egne verktøy. I tillegg er det et relativt enkelt språk å lære seg. Ulempen er at språket krever tid å sette seg inn i og er per i dag mindre brukt enn MB. Dog skal det sies at når man først er på et tilstrekkelig nivå med Python, åpnes det mange muligheter for å interagere med ArcMap.

Tilrettelegging av relevant data for lendeavurderingen i kombinasjon med en struktur, er fordelaktig med tanke på tid, enkelhet og organisering. Hvis man slipper å hente inn aktuelle data hver gang det skal gjøres en lendeavurdering, vil arbeidsprosessen effektiviseres. Videre vil en definert struktur i katalogen for lendeavurderingen bidra til en organisering og alle verktøy, modeller, skript og analyser kan ligge ett sted. Dette har potensialet for deling og gjenbruk i fremtiden. I og med at det ikke finnes et overordnet styrende dokument for MilGeo i Hæren er det noen ulemper med dette. Først og fremst vil en slik tilrettelegging måtte gjøres på skjønn og det kan være positivt, men da kan det også være fare for at den ikke blir hundre prosent komplett. Det kan tenkes at den ikke ivaretar alle kravene som de enkelte analytikerne stiller til data og oppsett. Det kan altså hende at det er noe som mangler i malen eller strukturen.

5.2 Produkt

Utformingen av produktets teoretiske aspekt er basert på intervjuer med INGBN og samtaler med HVS. Dette hjalp oss med å kartlegge hvilke faktorer vi måtte ta høyde for i løsningen og produktets ønskelige virkemåte. Videre forsto vi hvordan grunnlaget burde være. Utformingen av produktets tekniske aspekt derimot, er gjort med utgangspunkt i kompetansen vi allerede besatt eller tilegnet oss i løpet av bachelorperioden. Her er det derfor viktig å presisere at selv om løsningen tar høyde for det som er etterspurt og tilfredsstillende virkemåte, kan det være mulig med en alternativ teknisk tilnærming.



På den ene siden kan styrkene til den preliminære løsningen være at den kommer med et godt grunnlag for både automatisering og tilrettelegging. Dette vil kunne bidra til å spare tid for analytikerne. Den er ikke låst for redigering og man står fritt til å skreddersy den etter ønske. Slik løsningen er nå, skiller den ikke mellom type avdeling. Dette gjør at den kan brukes av alle avdelinger. Både MXD-malen og katalogstrukturen er mottagelige for respektivt data og verktøy, samtidig som katalogstrukturen blant annet tar høyde for gjenbruk av analyser. Videre illustrerer løsningen hvordan programmering med Python kan brukes til å automatisere oppgaver.

På den andre siden kan en del av tilretteleggingsmomentet (malen) også ses på som en svakhet fordi eksport av lagene er knyttet opp mot skriptverktøyet. Alle nye analyser kan legges til i malen, men de må hardkodes i skriptet for å få optimal utbytte av malen ved eksport av lagene. Malen kan altså skreddersys, men det vil få følger for skriptverktøyet. Løsningen er ikke testet og det er vanskelig å si nøyaktig hvor mye tid man vil spare. En annen svakhet med produktet er at det ikke kommer med et forslag til hvordan man skal dele det og skape felles tilgang. Videre har malen en svakhet i at den nødvendigvis ikke er komplett. Det er som nevnt enkelt for brukerne å legge til nye data, men de kunne vært hardkodet i skriptet om vi hadde lagt dem inn da løsningen ble produsert. På denne måten hadde analytikerne sluppet å bruke tid på å gjøre det selv. Python er som tidligere nevnt i oppgaven mindre brukt enn Model Builder, og da kan det tenkes at det, for brukeren, er en høyere terskel for å gjøre endringer i skriptet.

Det positive med skriptverktøyet er at det er enkelt å redigere om man har tilstrekkelig kunnskap innenfor Python. Hvis dette ikke er en utfordring, vil man spare tid ved eksport av mange lag og kunne hardkode nye lag i skriptet. Det skal derimot sies at man bør ha god kjennskap til Python hvis man skal drive modifisering utover dette. Skriptet er statisk per nå og fungerer kun opp mot malen det kommer med.



Fordelen med indeksen er at man fort kan finne igjen tidligere lende vurderinger fordi den inneholder metadata. Slik produktet er nå, legger det opp til at man lager en ny pakke for hver E-vurdering og bruker tidligere E-vurderinger som et supplement. En styrke ved dette er at man underbygger sine egne vurderinger.

En ulempe med slik vi har definert at produktet skal brukes, kan være at indeksen etterhvert blir overfylt med innrammede operasjonsområder som overlapper, og det vil gjøre den uoversiktlig.



6. Konklusjon

Vi har følgende problemstilling:

«Hvordan kan Etterretningsvurdering trinn 2 automatiseres for MilGeo i Hæren?»

En E-vurdering trinn 2 kan automatiseres med et produkt som er en kombinasjon av tilrettelagt data og teknisk automasjon. Vi har lagd en MXD-mal som inneholder ofte brukte data til lende-vurdering. Det er opprettet en struktur i katalogen som gir hurtig tilgang til relevante verktøy og tjener som et lagringssted for fremtidige analyser, skript og modeller. Til slutt har vi med Python lagd et verktøy som eksporterer resultatene fra ArcMap automatisk. Produktet er vedlagt oppgaven i sin helhet.

Den ideelle automatiseringen ville vært at man får ut resultatene ved å trykke på én knapp i ArcMap og der er vi ikke per i dag, men vi kan jobbe opp mot den visjonen. Vi begrenset oss til å se på det tekniske aspektet ved automatisering av lende-vurderingen og har lagd en konseptskisse (figur 1). Denne skissen ble brukt under produksjonen som ble utført i ArcMap. MilGeo-utdanningen på Krigsskolen kombinert med grunnleggende kurs i MB og selvstudie av Python, har gjort oss i stand til å produsere en løsning som automatiserer arbeidsprosessene i ArcMap. Løsningen skrapper overflaten av mulighetene som finnes for automatisering.



7. Veien videre

Det er flere faktorer som kan implementeres i produktet og kan bidra til tidsbesparelser, men som i oppgaven har blitt prioritert vekk på grunn av omfanget. Disse kan sørge for at lende vurderingen i enda større grad kan bli tilrettelagt og automatisert.

Modeller fører til automatisering av arbeidsprosesser og pakken kan brukes som en felles lagringsplass for disse. Slik kan pakken fasilitere for deling, men dette fordrer at den brukes og at det blir lagt til modeller. Skriptet kan gjøres mer dynamisk slik at det gir muligheten til å eksportere ut lag man selv har definert. Per nå er de lagene man kan eksportere ut hardkodet i skriptet, med andre ord kan kun lagene som vises i skriptverktøyet eksporteres. Videre må det skrives koder for å automatisk generere «*legend*» til eksporten.

Det er muligheter for å tilrettelegge mer data. FMGT kan eksempelvis levere «*mosaic dataset*» av «*slope*» og «*hillshade*» over hele Norge. Dette er data som endrer seg lite over tid, men som man alltid har behov for. Det er også muligheter for at FMGT kan inkludere verktøy i pakken hvis den blir en del av grunnlagspakken deres. Videre ser vi også potensialet for forhåndsdefinert symbolisering levert fra FMGT i form av lyrfiler for klassifisering og «*representations*» for oppdragsgrafikk.

Pakken må ut til brukerne for testing og vi ser for oss at dette kan skje gjennom en fagøvelse hos HVS. Videre må pakken forbedres med tilbakemeldinger fra brukerne slik at den kan utvikles til et verktøy som MilGeo analytikeren kan bruke fullt ut i en E-vurdering trinn 2.



8. Referanseliste og Kilder

8.1 Litteratur

Hærens skole for taktikk og operasjoner. (HSTO). *Stabshåndbok for Hæren - Plan- og beslutningsprosessen*. Rena: HVS, 2010.

Dag Ingvar Jacobsen. *Hvordan gjennomføre undersøkelser?*, 2. utgave. Høyskoleforlaget, 2005.

Paul A. Zandbergen, *PYTHON Scripting for ArcGIS*, Esri press, 380 New York street, Redlands, California 92373-8100.

Krigsskolen. Studiehåndbok, *Studieprogram for ingeniørutdanning bachelor i ingeniørfag*. Oslo: Krigsskolen, 2013-2014

Grinderud, Knut, Haakon Rasmussen, Steinar Nilsen, Arvid Lillethun, Atle Holten, Øystein Sanderud.

GIS: Geografiens språk i vår tidsalder. Trondheim: Tapir Akademisk Forlag, 2008



8.2 Nettsider

Navn	Adresse	Kontrollert
Model Builder	http://resources.arcgis.com/en/help/main/10.2/index.html#/What_is_ModelBuilder/002w00000001000000/	02.05.2014
Model Builder kurs	http://training.esri.com/gateway/index.cfm?fa=catalog.webCourseDetail&courseid=2168	02.05.2014
Arcpy	http://resources.arcgis.com/en/help/main/10.2/index.html#/What_is_ArcPy/000v000000v7000000/	02.05.2014
Python	http://resources.arcgis.com/en/help/main/10.2/index.html#/What_is_Python/002z00000001000000/	02.05.2014
Python Range function	https://www.youtube.com/watch?v=tkNsj9UTgPM	03.05.2014
Dokument: Prosjekt 8009	http://rapporter.ffi.no/rapporter/2009/01408.pdf	06.05.2014
Dokument: Intelligence preparation of the battlefield	http://www.fas.org/irp/doddir/army/fm34-130.pdf	08.05.2014

8.3 Intervjuobjekter

Stilling	Avdeling
Fagoffiser MilGeo	INGBN
Fagoffiser MilGeo	INGBN
Fagoffiser MilGeo	INGBN



9. Ordforklaring

ArcGIS = Er et samlebegrep for Programvare levert av ESRI

ArcCatalog = En applikasjon som er en del av ArcGIS-plattformen.

ArcMap = Inngår i ArcGIS-plattformen og er Hærens primære analyseverktøy.

ESRI = Amerikansk programvareleverandør av ArcGIS.

FMGT = Forsvarets Militærgeografiske Tjeneste

GDB = Geodatabase

GeoInt = Geospatial Intelligence

HVS = Hærens Våpenskole

INGBN = Ingeniørbataljonen

IPB = Intelligence Preparation of the Battlefield

IT = Informasjonsteknologi

KS = Krigsskolen

Lyr = Filformat til symboliseringsfil.

MB = Model Builder

MilGeo = Militærgeografi

MXD = Filformatet til ArcMap

NATO = North Atlantic Treaty Organization

P8009 = Prosjekt 8009 er et prosjekt i regi av FMGT som omhandler sentralisering av forsvarets kjernetjenester.

PBP = Plan og Beslutningsprosess

UTM = Universal Transverse Mercator



Vedlegg 1

Til leser av dette vedlegget

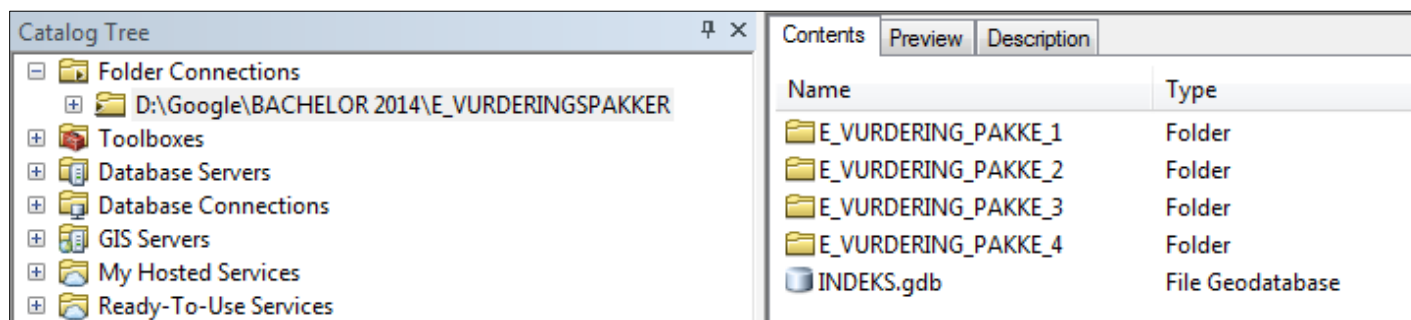
Vedlegget har til hensikt å gi leseren en forståelse for hvordan hvert enkelt element i produktet er bygd opp. Det vil først være en beskrivelse av mappene og strukturen i E-vurderingspakken, etterfulgt av en gjennomgang på hvordan MXD-malen er organisert. Deretter en beskrivelse av hvordan skriptverktøyet er laget. For å få en best mulig forståelse av skriptet er det fordelaktig å ha noe kjennskap til Python. Til slutt vil det være en bruksanvisning for hvordan vi ser for oss at E-vurderingspakken skal brukes i praksis og hvordan man kan editere skriptet.



E-vurderingspakke

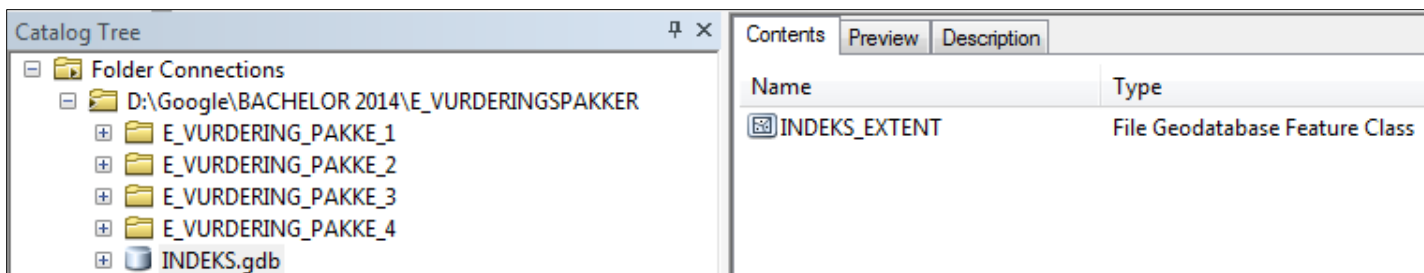
Et av produktene vi har lagd er en E-vurderingspakke. Hensikten med denne pakken er at den skal inneholde alt man trenger til E-vurdering trinn 2. I denne pakken har vi lagd en struktur som skal være til hjelp for analytikeren. Det er selv opp til brukeren av pakken å legge til eventuelle nye verktøy, modeller og skriptverktøy han eller hun bruker. Over tid vil man ha bygd seg opp en pakke som inneholder de hjelpemidlene man trenger, noe som vil føre til at den tekniske delen av arbeidsprosessen vil gå raskere.

E_VURDERINGSPAKKER er en mappe som skal inneholde alle E-vurderingspakkene man lager. Denne har vi valgt å kalle *E_VURDERINGSPAKKER* og ligger på øverste nivå i strukturen. Når man kopierer inn en tom E-vurderingspakke, skal denne nummereres i stigende rekkefølge for å kunne skille mellom dem. Dette nummeret skal brukes i en indeks slik at man i fremtiden kan finne frem til pakken som tilhører det aktuelle området.



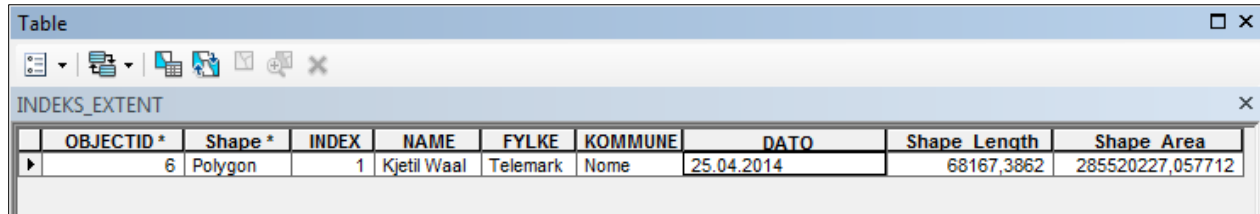
Figur 1: E-vurderingspakke

Under *E_VURDERINGSPAKKER* ligger det også en GDB som heter *INDEKS.gdb*. I denne ligger *INDEKS_EXTENT*, dette er en polygon «*feature class*».



Figur 1.1: E-Vurderingspakke

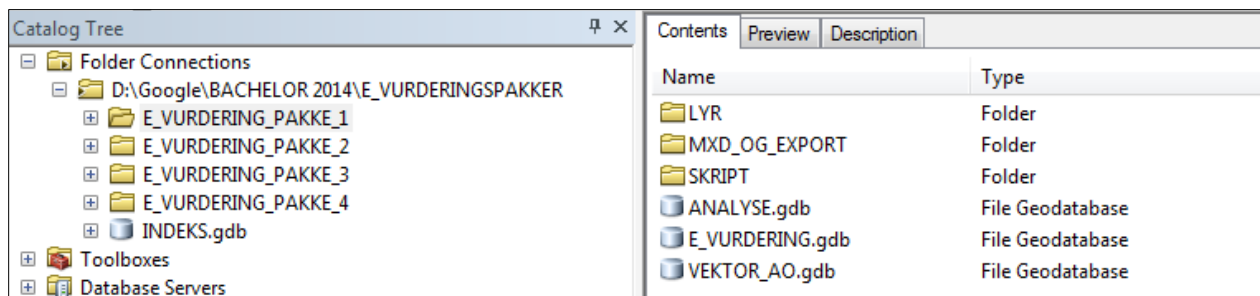
I tabellen til *INDEKS_EXTENT* skal det legges inn informasjon om hvilken pakke som hører til polygonet, navn på hvem som har gjort E-vurderingen, hvilket område den er over og en dato på når vurderingen ble gjort.



OBJECTID*	Shape*	INDEX	NAME	FYLKE	KOMMUNE	DATO	Shape Length	Shape Area
6	Polygon	1	Kjetil Waal	Telemark	Nome	25.04.2014	68167,3862	285520227,057712

Figur 1.2: E-Vurderingspakke

Under *E_VURDERING_PAKKE* er det ulike mapper. *LYR*-mappen er til lagring av lyrfiler for symbolisering. Dette kan være lyrfiler som symboliserer «*slope*», symbolisering knyttet til modeller eller andre spesifikke lag man ønsker å lagre symboliseringen for. *MXD_OG_EXPORT*-mappen skal brukes til å lagre MXD-dokumentet som hører til pakken. Det er i denne mappen all eksporten fra skriptverktøyet kommer til å havne. *SKRIPT*-mappen kan inneholde skriptene som brukes av skriptverktøyene. I *ANALYSE.gdb* lagres alle analyselagene. *E_VURDERING.gdb* skal inneholde modeller, skript og verktøy man bruker i en E-vurdering trinn 2. *VEKTOR_AO.gdb* skal inneholde «feature class'er» som er oppdragsspesifikke. Dette kan være ulike typer administrative grenser som for eksempel etterretningsmessig interesseområde eller teiggrenser.

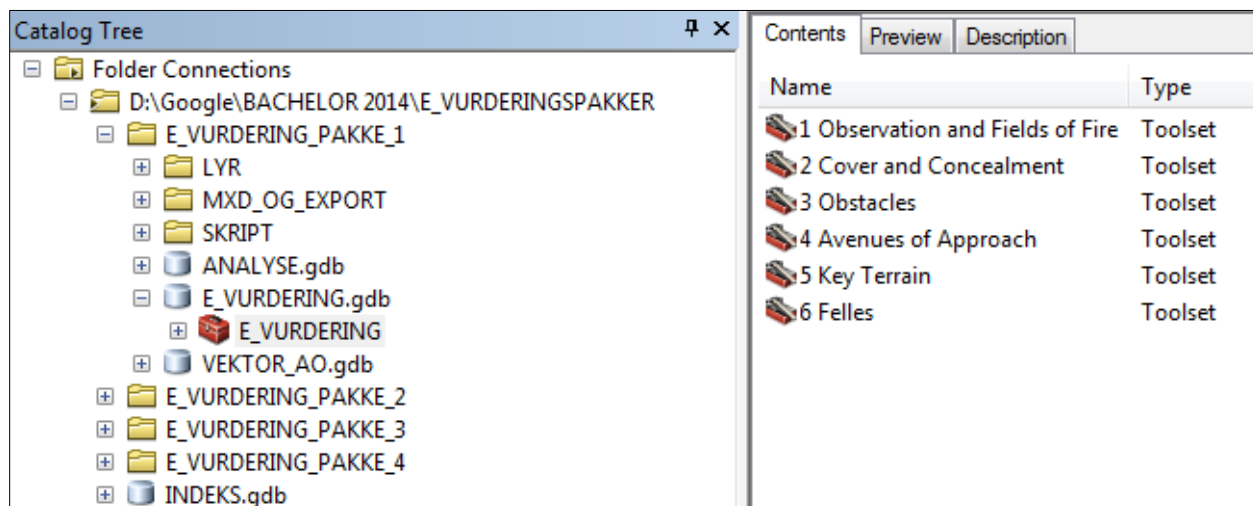


Name	Type
LYR	Folder
MXD_OG_EXPORT	Folder
SKRIPT	Folder
ANALYSE.gdb	File Geodatabase
E_VURDERING.gdb	File Geodatabase
VEKTOR_AO.gdb	File Geodatabase

Figur 1.3: E-Vurderingspakke



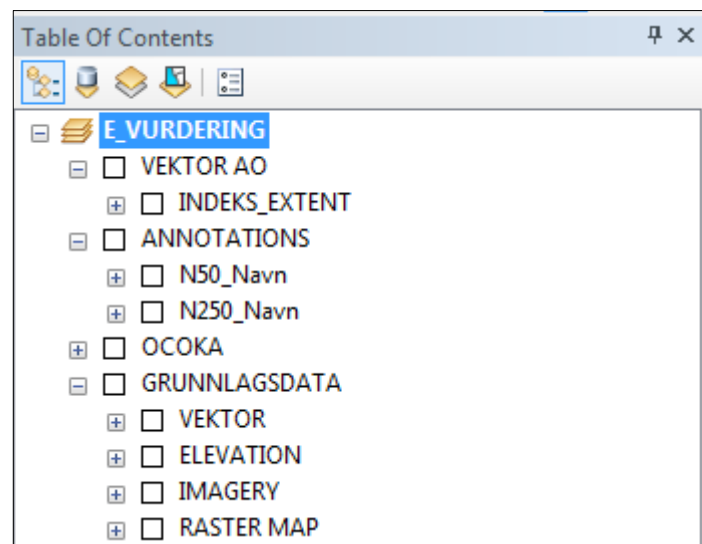
Går vi inn i *E_VURDERING.gdb* inneholder den en verktøyboks, som inneholder verktøysett. Her er hensikten at man skal legge inn de verktøyene, modellene og skriptverktøyene man bruker under de forskjellige stegene i OCOKA'en. Dette gjør at man får samlet alle verktøyene på ett sted og slipper å lete seg frem til dem neste gang. Under verktøysett *6 Felles* legges verktøyene, modellene og skriptverktøyene som ikke faller inn under noen av de andre stegene.



Figur 1.4: E-Vurderingspakke

MXD-mal

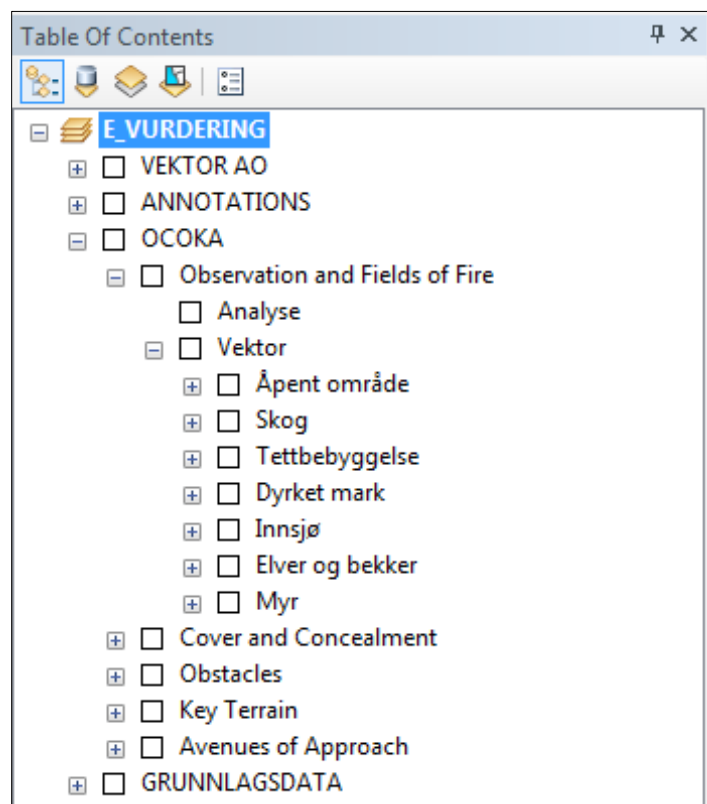
MXD-malen inneholder de fire hovedlagene, *VEKTOR AO*, *ANNOTATIONS*, *OCOKA* og *GRUNNLAGSDATA*. I gruppelaget *VEKTOR AO* skal *INDEX_EXTENT* ligge, det er denne som skal definere operasjonsområdet. Her skal også andre oppdragsspesifikke lag som administrative grenser, etterretningsmessig interesseområde eller teiggrensler ligge. Navn ligger under *ANNOTATIONS*. Hensikten med å plassere navn over de andre lagene er at de skal være synlige når lagene skrur på. Det nederste laget *GRUNNLAGSDATA* inneholder den dataen man skal ha som grunnlag. *VEKTOR* inneholder bane og veier som er kategorisert etter type. Under *ELEVATION* skal det ligge et «*mosaic dataset*» for «hillshade». Denne gir kartet merverdi ved å synliggjøre terrengformasjoner. I *IMAGERY* skal det ligge et «*mosaic dataset*» med flyfoto som kan brukes til å studere områder når man skal gjøre vurderinger. Under *RASTER MAP* ligger det vanlige rasterkart. Primært N5000, N250 og N50. Disse kartene har definerte målestokker som gjør at de slår seg av eller på når man zoomer.



Figur 2: MXD-mal



Det siste laget, *OCOKA*, er delt opp etter trinnene i lende vurderingen. Hvert steg er delt opp i *Analyse* og *Vektor*. Under *Analyse* kan man legge inn relevante analyser som hører til det aktuelle steget. Under *Vektor* ligger det vektordata fra N50_arealdekkeflate som er delt opp etter kategorier med «*Definition Query*». Det som er viktig å poengtere, er sammenhengen mellom skriptverktøyet og MXD-malen. Skriptverktøyet er kun linket opp mot *OCOKA* laget og de forskjellige stegene. Verktøyet vil både skru på *OCOKA* og enkeltvis skru av og på de forskjellige stegene. Dette betyr at man kan legge inn data under *Analyse* og *Vektor* uten at det påvirker skriptverktøyet under eksport av lagene. Lagene *VEKTOR AO*, *ANNOTATIONS* OG *GRUNNLAGSDATA* kan endres navn på eller tillegges lag uten at dette vil ha noen konsekvenser for skriptverktøyet. Det er viktig at man ikke endrer navnene på *OCOKA* og de forskjellige trinnene ettersom de er hardkodet i skriptet.



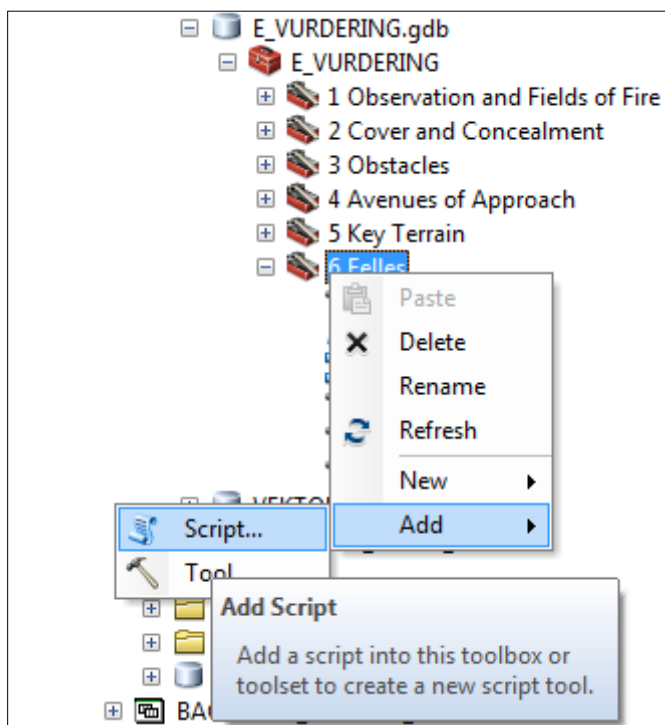
Figur 2.1: MXD-mal

Skriptverktøy

Beskrivelsen av hvordan man oppretter et skriptverktøy er hentet og oversatt fra boken «*Python Scripting for ArcGIS*» av Paul A. Zandbergen.¹²

Opprette skriptverktøy

Når man oppretter et skriptverktøy gjøres det på samme måte som med en modell. Skriptet må enten opprettes i en «*toolbox*» eller i et «*toolset*». Man høyreklikker på en verktøyboks eller et verktøysett og går til *Add > Script*.



Figur 3: Skriptverktøy

¹² Paul A. Zandbergen, *PYTHON Scripting for ArcGIS*, Esri press, 380 New york street, Redlands, California 92373-8100



Når man har opprettet skriptet, vil man få opp en dialog slik som på bildet under. *Name* feltet er navnet på verktøyet og kan brukes hvis man skal kalle på verktøyet i Python. *Label* feltet er det navnet som vil være synlig i verktøysettet. *Description* feltet brukes for å gi en egendefinert beskrivelse av verktøyet. Det er denne teksten som vises i hjelpepanelet når man åpner verktøyet. I *Stylesheet* feltet kan man legge til en egendefinert stil for verktøyet. Når man haker av for *Store relative path names*, vil det bli brukt en relativ stibane for å referere til skriptfilen som kobles til verktøyet. *Always run in foreground* hukes av for at skriptverktøyet ikke skal kjøre i bakgrunnen. I dette tilfellet må den være huket av fordi vi kommer til å bruke nøkkelordet "CURRENT" i skriptet. "CURRENT" vil vi komme tilbake til senere i dette vedlegget under skriptet. Vi fyller inn som på bildet og trykker på *Neste >*.

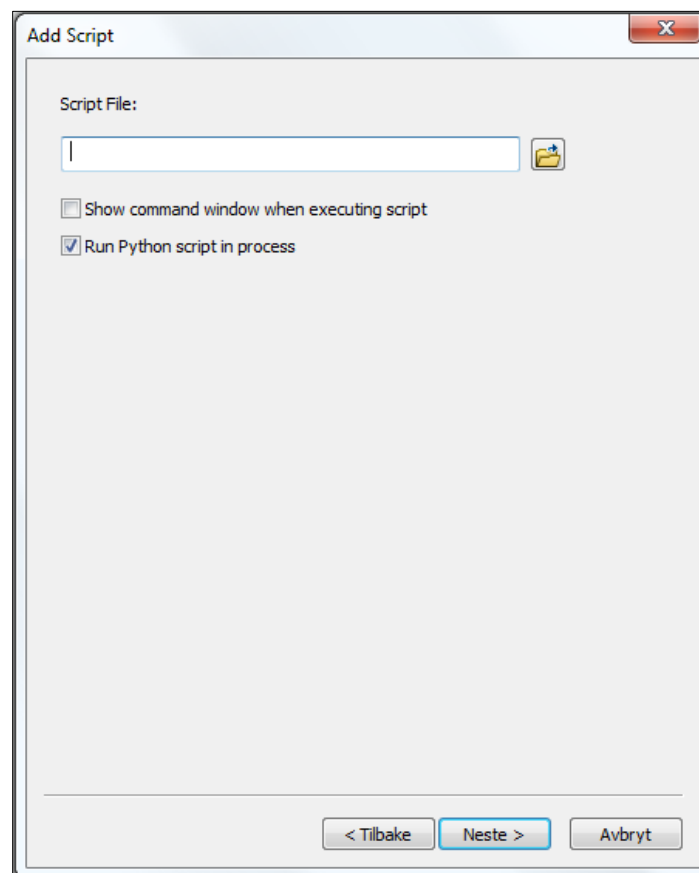
The image shows a dialog box titled "Add Script" with a close button (X) in the top right corner. The dialog contains the following fields and options:

- Name:** A text input field containing "Export".
- Label:** A text input field containing "Export".
- Description:** A text area containing "Dette verktøyet eksporterer ut .png filer fra ArcMap etter OCOKA -trinnene." with a scroll bar on the right.
- Stylesheet:** A text input field that is empty, followed by a folder icon button.
- Store relative path names (instead of absolute paths)
- Always run in foreground

At the bottom of the dialog are three buttons: "< Tilbake", "Neste >" (highlighted in blue), and "Avbryt".

Figur 3.1: Skriptverktøy

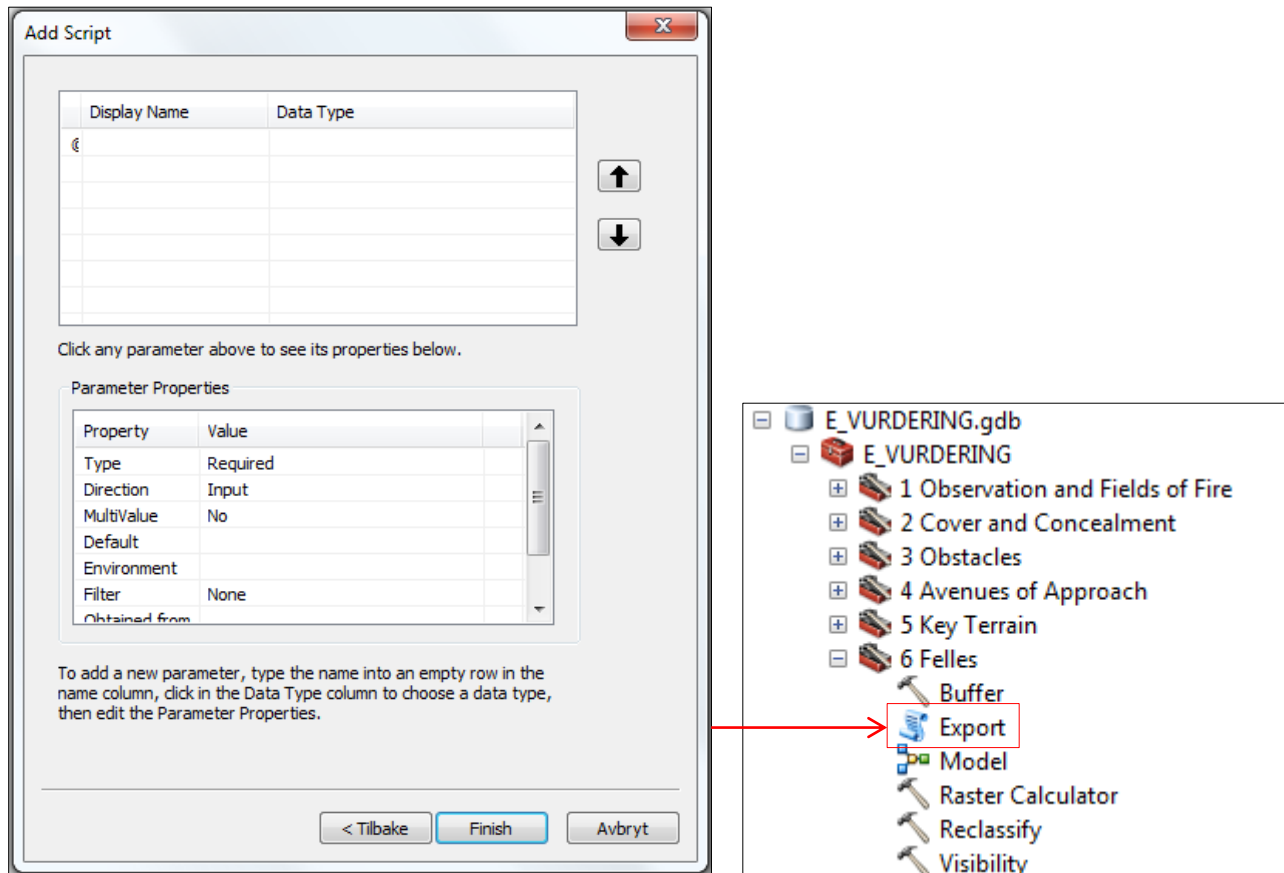
I denne dialogen skal vi legge til en skriptfil under *Script File*. Den kan legges til nå eller på et senere tidspunkt. Vi har ennå ikke lagd skriptet og kommer til å gjøre dette under avsnittet «*Skript*». Hvis man haker av for *Show command window when executing script* vil man få opp et ekstra vindu når man kjører verktøyet. Dette er ikke nødvendig med mindre man har spesielle behov for å se beskjedene skrevet i standard Python format. *Run Python script in process* hukes av slik at skriptet kjører når verktøyet kjører. Her skal *Run Python script in process* være huket av og vi trykker på *Neste >*.



Figur 3.2: Skriptverktøy



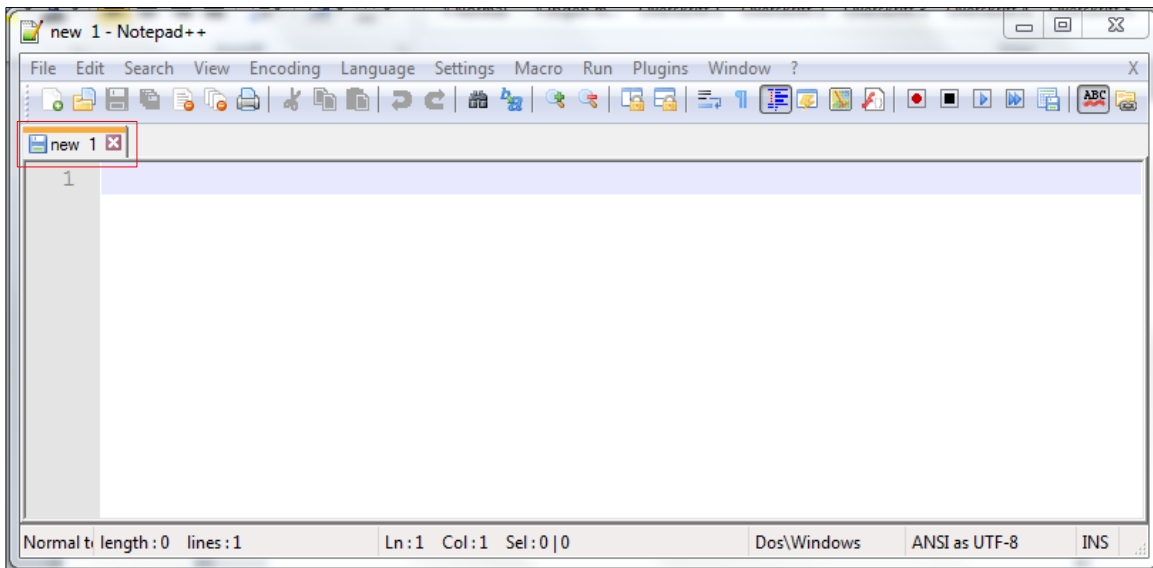
I denne dialogen legger vi inn parameterne som skal brukes i verktøyet. De kan endres på hvis ønskelig. I det øverste vinduet legger man inn de parameterne man ønsker. Vinduet nederst brukes for å definere egenskapene til de forskjellige. Dette skal vi gjøre senere når vi har skrevet skriptet. Her trykker vi bare på *Finish* og det blir opprettet et skriptverktøy i verktøysettet.



Figur 3.3: Skriptverktøy

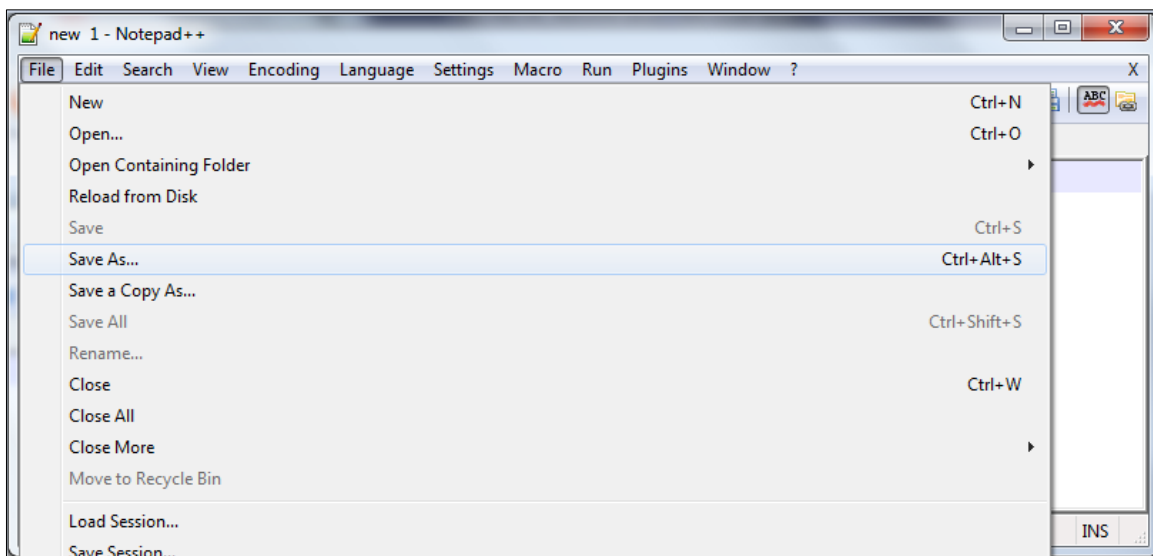
Skript

Til å skrive skriptet har vi brukt tekstediteringsprogrammet *Notepad++*. Dette er et program som støtter flere programmeringsspråk inkludert Python. Det betyr blant annet at den har innebygde stilarter som gjør at den gjenkjenner filtyper og setter formatteringen til den aktuelle filtypen. Når vi åpner Notepad++ for første gang, får vi opp det tomme dokumentet *new 1*.



Figur 4: Skript

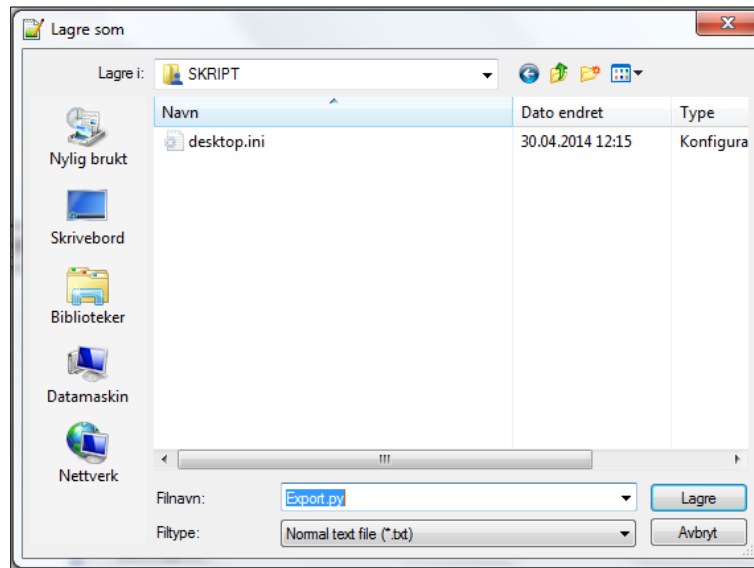
Vi går til *File > Save As...*



Figur 4.1: Skript

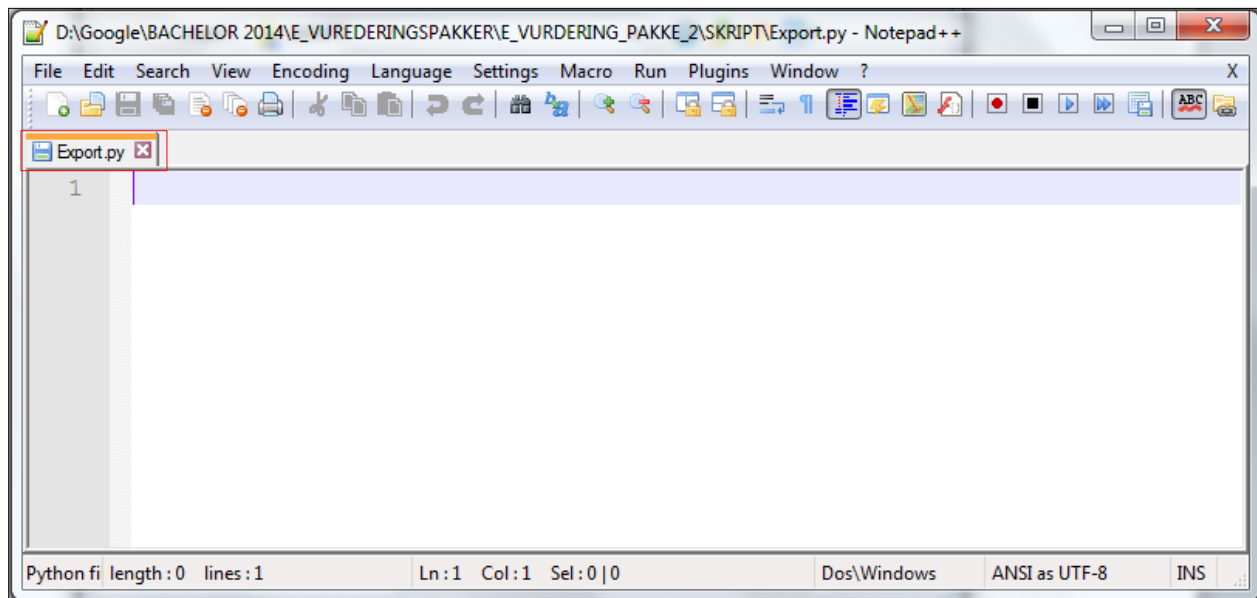


Vi går til en skript-mappe og lagrer filen med navnet *Eksport.py*. Når filen lagres som en Pythonfil, vil Notepad++ gjenkjenne filtypen og sette formateringen til Python syntaks.



Figur 4.2: Skript

I Notepad++ får vi opp en fane som heter *Export.py*, dette er filen skriptet skal skrives i.



Figur 4.3: Skript

Et viktig prinsipp når man skriver skript er å inkludere beskrivelser. Det finnes ingen regler for hva man skal skrive i dokumentasjonen. Hensikten til skriptet, navn på hvem som har skrevet det og når det ble lagret, er informasjon som kan være hensiktsmessig å ta med. En beskrivelse til hva de ulike kodene eller blokkene gjør er viktig å ha med for sin egen del og spesielt andre. Alle har sine egne måter å bygge opp skript på, ved å legge ved en beskrivelse blir det enklere å lese og forstå skriptet for de man deler det med.

I bildet under bruker vi hashtag (#) på starten av linjene. Linjer som starter med hashtag, vil bli ignorert i Python og ikke bli kjørt som en del av skriptet. Hashtag er også en fin måte å feilsøke skript på. Ved å sette hashtag inn foran linjer kan man kjøre deler av skriptet og finne ut hvor det er syntaksfeil eller andre typer feil.

```
1  #-----#
2  # Name:      Export                               #
3  # Purpose:   Export .png files from ArcMap        #
4  # Author:    Aqeel Ahmed, Kjetil Waal and Henrik Strand #
5  # Created:   30/04/2014                          #
6  #-----#
7
```

Figur 4.4: Skript

Vi starter med `import arcpy`. Import er et nøkkelord man bruker for å kunne importere moduler inn i Python. Moduler kan ses på som utvidelsespakker for funksjonaliteten i Python. En modul kan inneholde forskjellige funksjoner eller klasser. Disse funksjonene eller klassene er predefinerte koder som løser ulike oppgaver. Dette gjør at man i Python ikke trenger å skrive alt fra bunnen av. Arcpy er en pakke som bygger på ArcGIS skripting modulen, her importeres denne.

```
7
8  # Import av moduler
9  import arcpy
10
```

Figur 4.5: Skript



`arcpy.env.overwriteOutput = True` er en funksjon som skriver over allerede eksisterende data med samme navn. Funksjonen kan settes til enten `True` eller `False` om man ønsker at skriptet skal kunne skrive over filer eller ikke.

```
10
11 # Overskriv allerede eksisterende export
12 arcpy.env.overwriteOutput = True
13
```

Figur 4.6: Skript

`mxd = arcpy.mapping.MapDocument("CURRENT")` er en funksjon som bestemmer hvilket MXD-dokument man jobber i. Her henter man `mapping` fra `arcpy`, denne brukes for å interagere med MXD-filer. `MapDocument` brukes for å få tilgang til alle funksjonene som ligger i `mapping`. Ved å bruke `MapDocument` får man også tilgang til alle objektene i en MXD-fil som for eksempel «`data frames`», «`layers`» og «`page layout elements`». I dette skriptet bruker vi «`CURRENT`» for å referere til filen vi jobber i. Det er her også mulig å bruke den fullstendige filbanen til MXD-filen, men dette er ikke like fleksibelt. Da må man spesifisere den fullstendige filbanen hver gang man kjører skriptet. Ved å bruke «`CURRENT`» blir filbanen til MXD'en hentet automatisk.

```
13
14 # Definerer hvilket mxd dokument man jobber i
15 mxd = arcpy.mapping.MapDocument("CURRENT")
16
```

Figur 4.7: Skript



`df = arcpy.mapping.ListDataFrames(mxd)[0]` er en variabel som sier hvilken *data frame* man refererer til. `ListDataFrames` ser gjennom *TOC'en* i ArcMap for å se hvor mange *data frames* man har. *MXD* refererer til filen vi jobber i, denne satte vi i forrige linje til "*CURRENT*". Dette vil si at man ser over hvor mange *data frames* det ligger i *TOC'en* på "*CURRENT*" *MXD*-fil. I klammeparentesen `[0]` setter man den *data frame* skriptet skal jobbe opp mot. I dette tilfellet er det den første i listen til *TOC'en*. Hadde *data frame'en* vår ligget som nummer to i *TOC'en* måtte vi ha skrevet `[1]`. Det kan her presiseres at Python teller fra 0 og oppover, det vil si at posisjon nummer 1 er 0.

```
16
17 # Definerer hvilken data frame man referer til
18 df = arcpy.mapping.ListDataFrames(mxd)[0]
19
```

Figur 4.8: Skript

`for lyr in arcpy.mapping.ListLayers(mxd, "OCOKA", df):` er en funksjon som henter alle lagene i *TOC'en*. Den leter i *MXD'en* og finner laget "*OCOKA*" i *df*. Hvis den finner laget, skal dette skrues på fordi `lyr.visible = True` er satt til `True`. Hadde `lyr.visible` vært satt til `False` ville laget blitt skrudd av.

```
19
20 # Definerer hvilke lag som alltid skal være på
21 for lyr in arcpy.mapping.ListLayers(mxd, "OCOKA", df):
22     lyr.visible = True
23 for lyr in arcpy.mapping.ListLayers(mxd, "Obstacles", df):
24     lyr.visible = True
25
```

Figur 4.9: Skript



Første linje definerer en funksjon med linjen `def export(layer)`. Denne funksjonen lager vi for å slippe dupliserende arbeid, dette kommer vi tilbake til. `Def` er et nøkkelord som forteller at vi skal definere en funksjon og denne funksjonen skal ha navnet `export`. For at funksjonen skal kunne motta verdier må man skrive inn en parameter, denne har vi kalt `layer`. Dette betyr at når vi senere kaller på funksjonen `export` så vil verdien bli sendt til parameteren `layer`. Neste linje, `for i in Maplayers`, definerer at vi skal gjøre noe for bokstaven `i`. `if i == layer` sier at hvis `i` er lik `layer` så skal den gjøre det som kommer i neste linje. Denne linjen skrur laget på. I neste linje har vi `else`, dette betyr at hvis `i` ikke er lik `layer` skal resterende lag skrur av.

`arcpy.mapping.ExportToPNG(mxd, layer)` er en funksjon som eksporterer ut laget. `MXD` er det dokumentet man jobber i og `layer` er navnet på filen.

```
25
26 # Funksjon som definerer hvilke lag som skal være av/på
27 def export(layer):
28     for i in Maplayers:
29         if i == layer:
30             arcpy.mapping.ListLayers(mxd, i, df)[0].visible = True
31         else:
32             arcpy.mapping.ListLayers(mxd, i, df)[0].visible = False
33     arcpy.mapping.ExportToPNG(mxd, layer)
34
35 # Liste over lagene som skal eksporteres
36 Maplayers = ["Observation and Fields of Fire", "Cover and Concealment",
37 "Vegetation", "Hydrology", "Slope", "Key Terrain", "Avenues of Approach"]
38
39 # Liste som populeres
40 MapToExport = []
41
42 # Importerer parameteren fra skriptverktøyet med samme rekkefølge som i Maplayers
43 # og legger til listen MapToExport
44 for i in range(7):
45     MapToExport.append(arcpy.GetParameterAsText(i))
46
47 # Sjekker navn fra parametere, hvis navn ikke er "None" så eksporteres laget
48 for q in range(7):
49     if MapToExport[q] != "None":
50         export(Maplayers[q])
```

Figur 4.10: Skript

Det er ønskelig å eksportere ut kun ett lag fra OCOKA om gangen. Det betyr at ett lag må være skrudd på, mens de andre er skrudd av. Hvis man bruker funksjon over, slipper man å skrive en blokk for hvert lag man skal eksportere. Bildet under illustrer en slik blokk.

```
41 if Observation_and_Fields_of_Fire == "None":
42     pass
43 else:
44     for lyr in arcpy.mapping.ListLayers(mxd, "Observation and Fields of Fire", df):
45         lyr.visible = True
46     for lyr in arcpy.mapping.ListLayers(mxd, "Cover and Concealment", df):
47         lyr.visible = False
48     for lyr in arcpy.mapping.ListLayers(mxd, "Obstacles", df):
49         lyr.visible = False
50     for lyr in arcpy.mapping.ListLayers(mxd, "Key Terrain", df):
51         lyr.visible = False
52     for lyr in arcpy.mapping.ListLayers(mxd, "Avenues of Approach", df):
53         lyr.visible = False
54     arcpy.mapping.ExportToPNG(mxd, r"Observation and Fields of Fire")
```

Figur 4.11: Skript

Maplayers = [] er en liste over lagene man finner under "OCOKA" i MXD-malen.

```
34
35 # Liste over lagene som skal eksporteres
36 Maplayers = ["Observation and Fields of Fire", "Cover and Concealment",
37 "Vegetation", "Hydrology", "Slope", "Key Terrain", "Avenues of Approach"]
38
```

Figur 4.12: Skript



`MapToExport = []` er en liste som tillegges verdier fra skriptverktøyet. `MapToExport.append` funksjonen legger disse verdiene inn i listen `MapToExport`. Disse hentes inn med `arcpy.GetParameterAsText(i)`. Inne i parenteser setter man et tall som er ekvivalent med den øverste eller første parameteren i skriptverktøyet. Her legger vi til tallene ved å bruke `range(7)`. Denne funksjonen gir bokstaven `i` et tall fra 0 til 6. Grunnen til at vi har syv ulike tall er fordi det skal legges inn syv parametere i skriptverktøyet senere. Dette henger også sammen med at det er syv elementer i `Maplayers = []` listen (Figur: 4.12).

```
38
39 # Liste som populeres
40 MapToExport = []
41
42 # Importerer parameteren fra skriptverktøyet med samme rekkefølge som i Maplayers
43 # og legger til listen MapToExport
44 for i in range(7):
45     MapToExport.append(arcpy.GetParameterAsText(i))
46
```

Figur 4.13: Skript

Her bruker vi den samme `range()` funksjonen og går over den samme tallrekken. I neste linje sier vi at hvis `MapToExport` ikke er lik `"None"`, skal den sende verdien til `layer` i funksjonen `export` (Figur: 4.10)

```
46
47 # Sjekker navn fra parametere, hvis navn ikke er "None" så eksporteres laget
48 for q in range(7):
49     if MapToExport[q] != "None":
50         export(Maplayers[q])
```

Figur 4.14: Skript



Her er en fullstendig oversikt over skriptet.

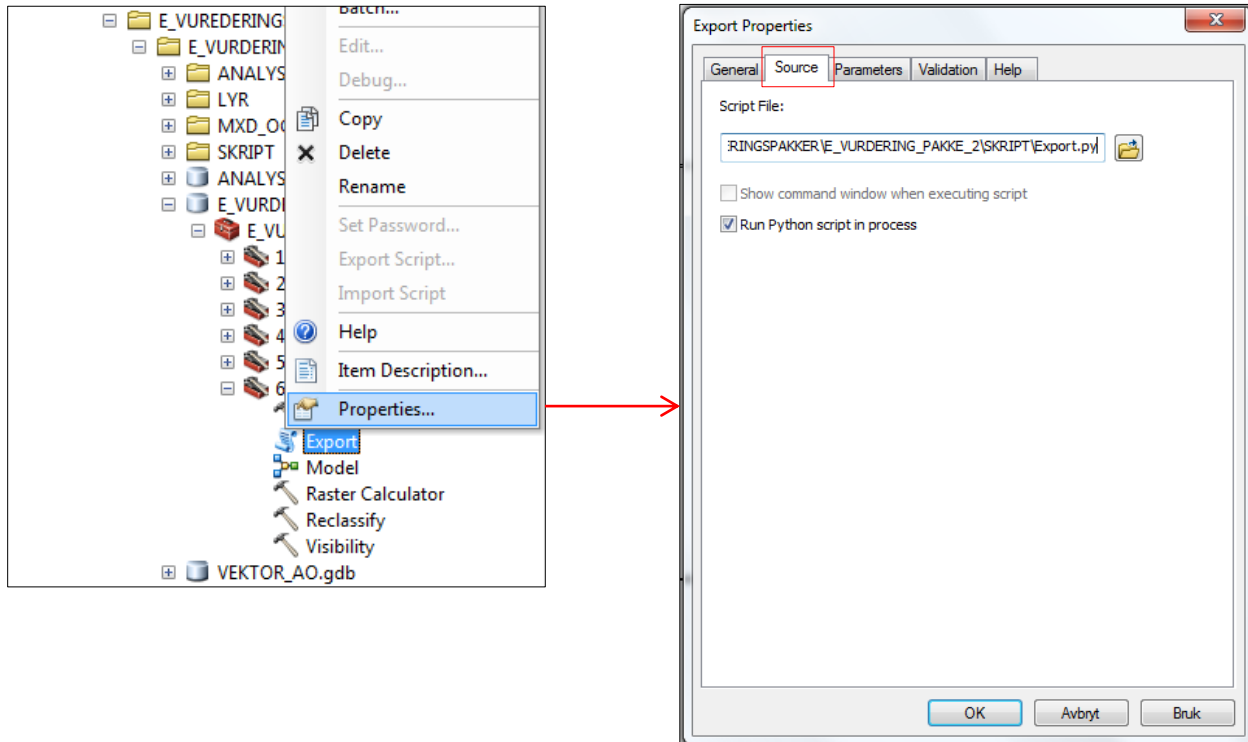
```
8 # Import av moduler
9 import arcpy
10
11 # Overskriv allerede eksisterende export
12 arcpy.env.overwriteOutput = True
13
14 # Definerer hvilket mxd dokument man jobber i
15 mxd = arcpy.mapping.MapDocument("CURRENT")
16
17 # Definerer hvilken data frame man referer til
18 df = arcpy.mapping.ListDataFrames(mxd)[0]
19
20 # Definerer hvilke lag som alltid skal være på
21 for lyr in arcpy.mapping.ListLayers(mxd, "OCOKA", df):
22     lyr.visible = True
23 for lyr in arcpy.mapping.ListLayers(mxd, "Obstacles", df):
24     lyr.visible = True
25
26 # Funksjon som definerer hvilke lag som skal være av/på
27 def export(layer):
28     for i in Maplayers:
29         if i == layer:
30             arcpy.mapping.ListLayers(mxd, i, df)[0].visible = True
31         else:
32             arcpy.mapping.ListLayers(mxd, i, df)[0].visible = False
33     arcpy.mapping.ExportToPNG(mxd, layer)
34
35 # Liste over lagene som skal eksporteres
36 Maplayers = ["Observation and Fields of Fire", "Cover and Concealment",
37 "Vegetation", "Hydrology", "Slope", "Key Terrain", "Avenues of Approach"]
38
39 # Liste som populeres
40 MapToExport = []
41
42 # Importerer parameteren fra skriptverktøyet med samme rekkefølge som i Maplayers
43 # og legger til listen MapToExport
44 for i in range(7):
45     MapToExport.append(arcpy.GetParameterAsText(i))
46
47 # Sjekker navn fra parametere, hvis navn ikke er "None" så eksporteres laget
48 for q in range(7):
49     if MapToExport[q] != "None":
50         export(Maplayers[q])
```

Figur 4.15: Skript fullstendig



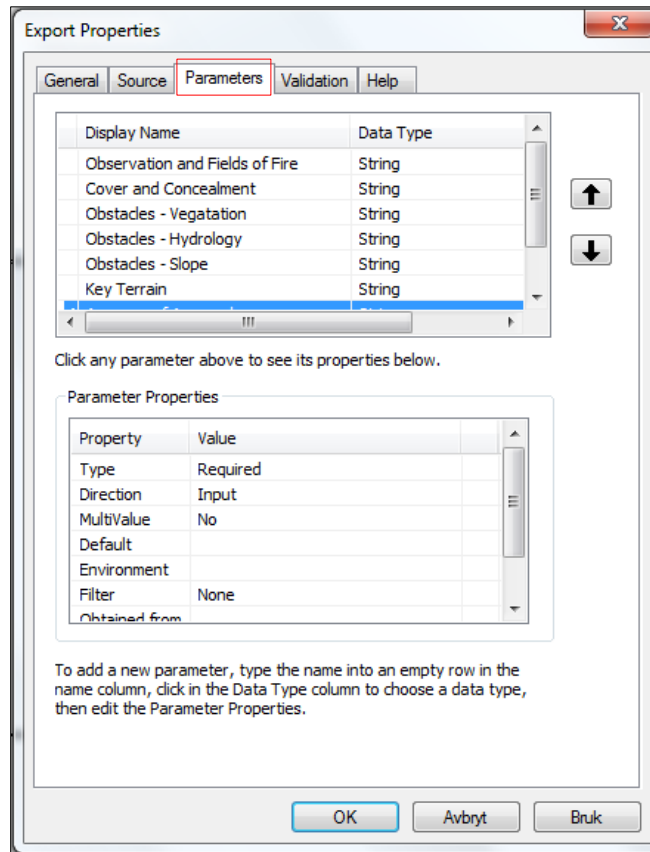
Ferdigstille skriptverktøy

Vi starter med å gå inn i ArcCatalog, høyreklikker på skriptverktøyet og velger *Properties*. Her går vi til fanen *Source* hvor skriptet skal legges til. Nå som skriptet er lagt til, navigeres det til *Parameters*-fanen.



Figur 5: Ferdigstille skriptverktøy

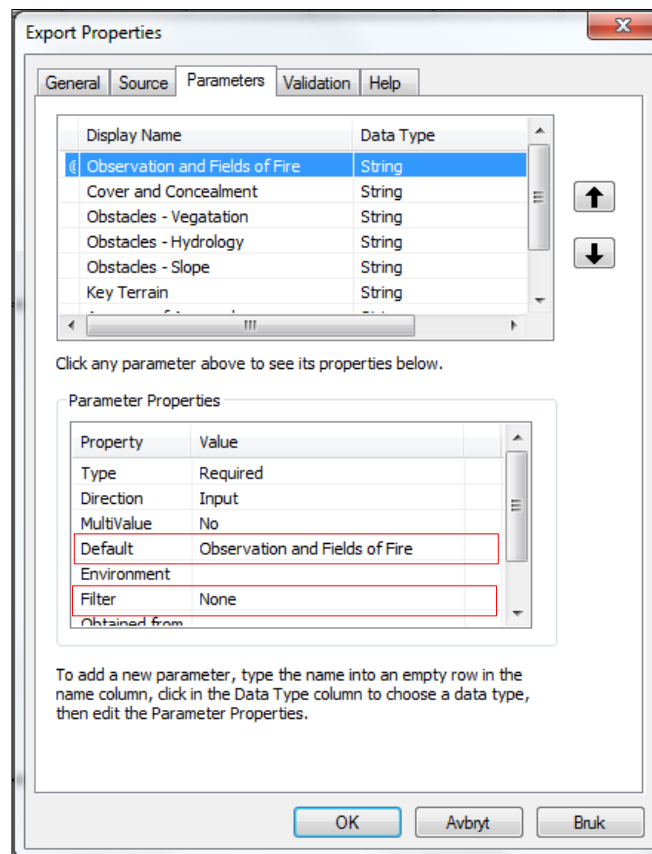
Vi legger inn navn på parameterne under *Display Name*, det er disse som gir navn til de forskjellige parameterne når vi senere åpner skriptverktøyet. Deretter setter vi datatypen til *String* under *Data Type*.



Figur 5.1: Ferdigstille skriptverktøy

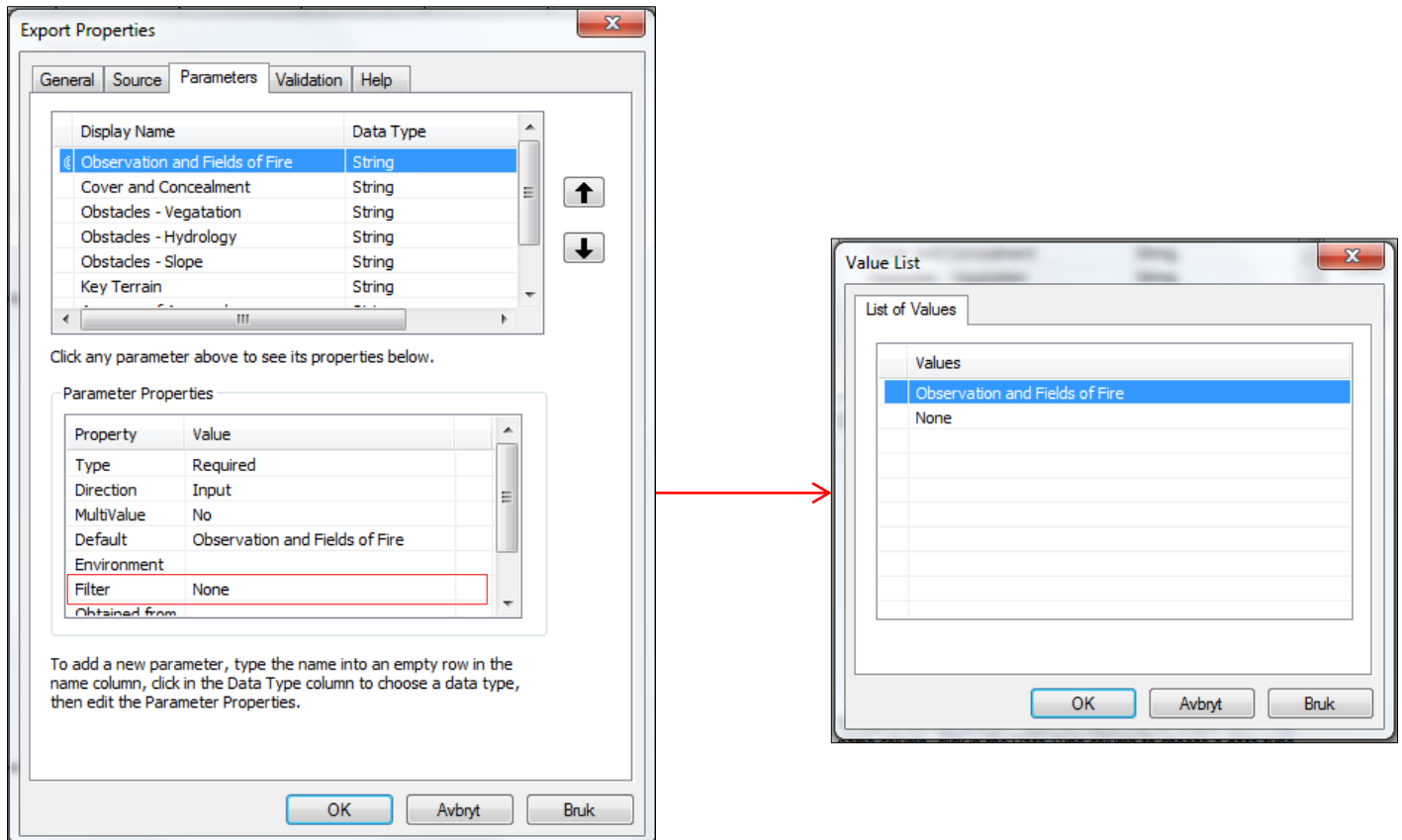


Videre gjøres det to endringer i *Parameter Properties*. Det legges inn en standardverdi i feltet *Default* og en liste i feltet *Filter*. Det er kritisk i denne fasen at de verdiene som legges inn har de samme navnene og rekkefølgen som i *TOC'en* og i listen *Maplayers = ["Observation and Fields of Fire", "Cover and Concealment", "Vegetation", "Hydrology", "Slope", "Key Terrain", "Avenues of Approach"]*. Disse verdiene vil bli sendt til skriptet. Når det legges inn verdier i *Default* feltet, vil verdiene legges automatisk inn når skriptverktøyet åpnes.



Figur 5.2: Ferdigstille skriptverktøy

Nå skal det legges til en liste med verdier, og dette gjør vi ved å trykke på *None* i feltet *Filter*. Da åpnes det en ny dialog, *Value List*. Her legges det inn to verdier: *Laget* som vi setter verdien for og *None*. Når disse legges inn i en liste, får man en nedtrekksmeny i skriptverktøyet. Disse verdiene blir sendt til skriptet og kontrollert opp mot listen *Maplayers[]*. Når dette er gjort for alle lagene trykker, vi på *OK* og fullfører skriptverktøyet.

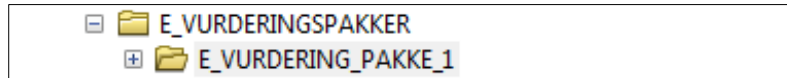


Figur 5.3: Ferdigstille skriptverktøy



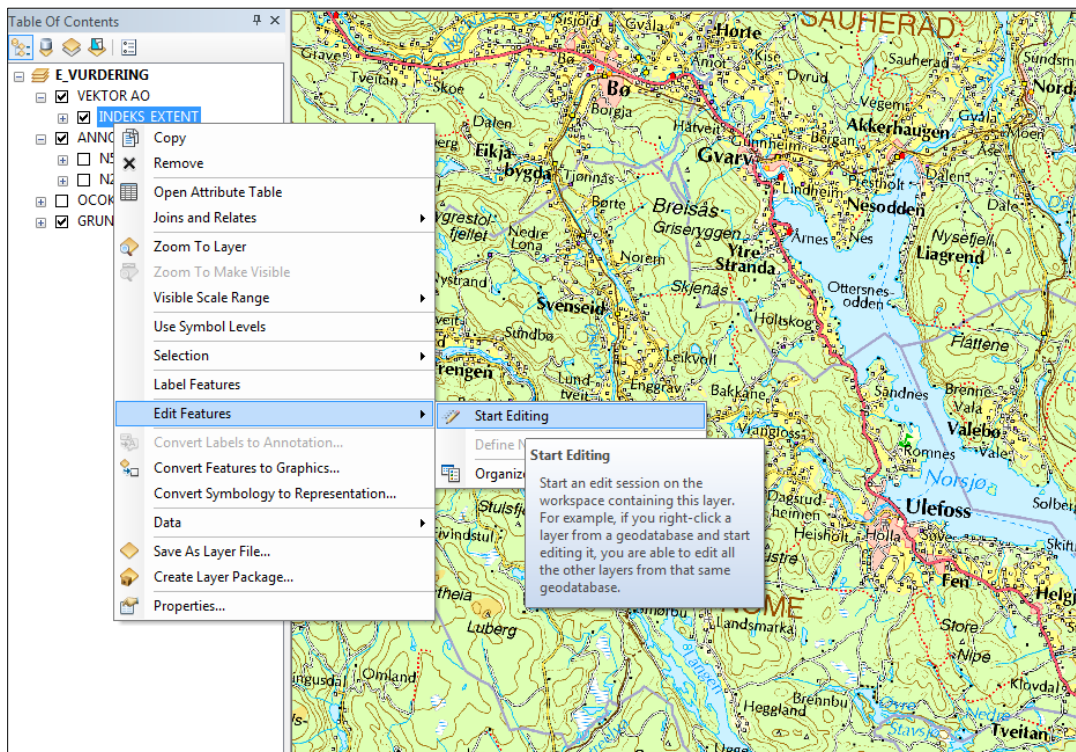
Gjennomgang av bruk

Vi tar for oss følgende scenario: Det skal gjøres en lendvurdering i område Ulefoss. Det første som gjøres er å kopiere inn en tom E-vurderingspakke under *E_VURDERINGSPAKKER* og gi den et nummer. Det er første pakke og får nummeret én.



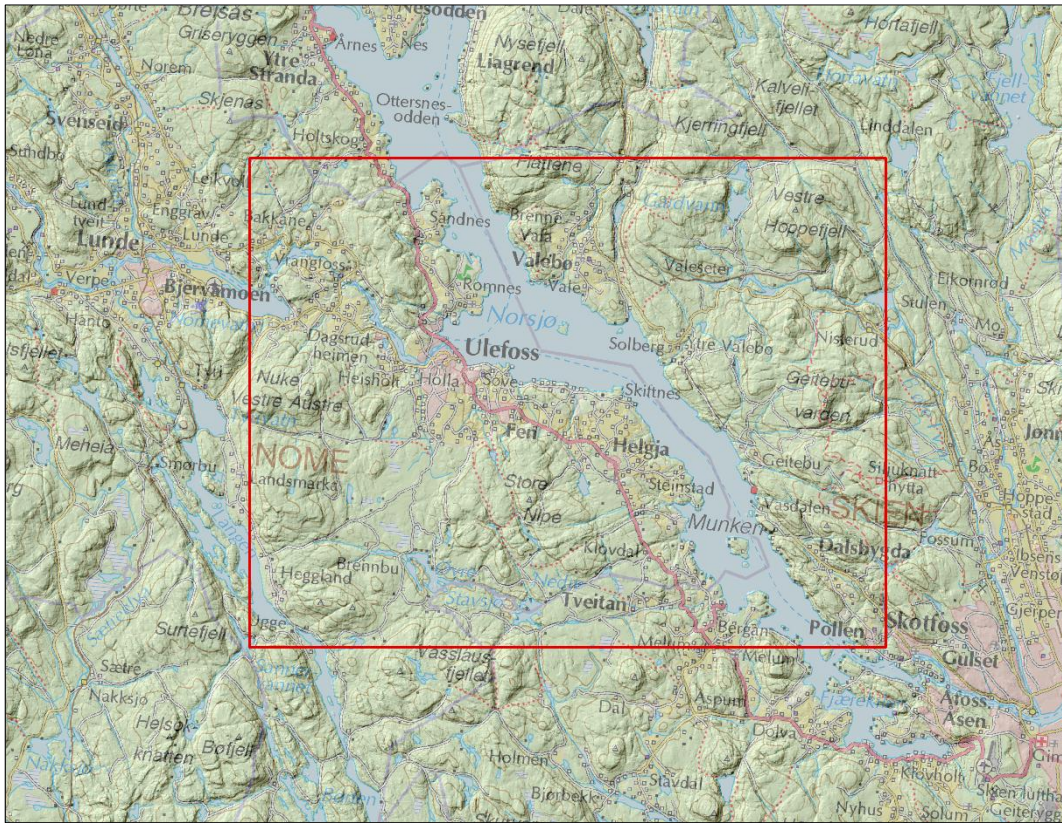
Figur 6: Gjennomgang av bruk

Deretter åpner vi laget *VEKTOR AO* og editerer *INDEKS_EXTENT*.



Figur 6.1: Gjennomgang av bruk

Når operasjonsområdet er innrammet med et polygon, går vi videre inn i tabellen til *INDEX_EXTENT*. Her fylles nummeret på pakken inn under *INDEX*, hvem som har lagd pakken under *NAME*, hvilket fylke, hvilken kommune og til slutt en dato for når pakken har blitt opprettet. Når dette er gjort, lagres disse editeringene.



↓

OBJECTID*	Shape*	INDEX	NAME	FYLKE	KOMMUNE	DATO	Shape Length	Shape Area
6	Polygon	1	Kjetil Waal	Telemark	Nome	25.04.2014	68167,3862	285520227,057712

Figur 6.2: Gjennomgang av bruk

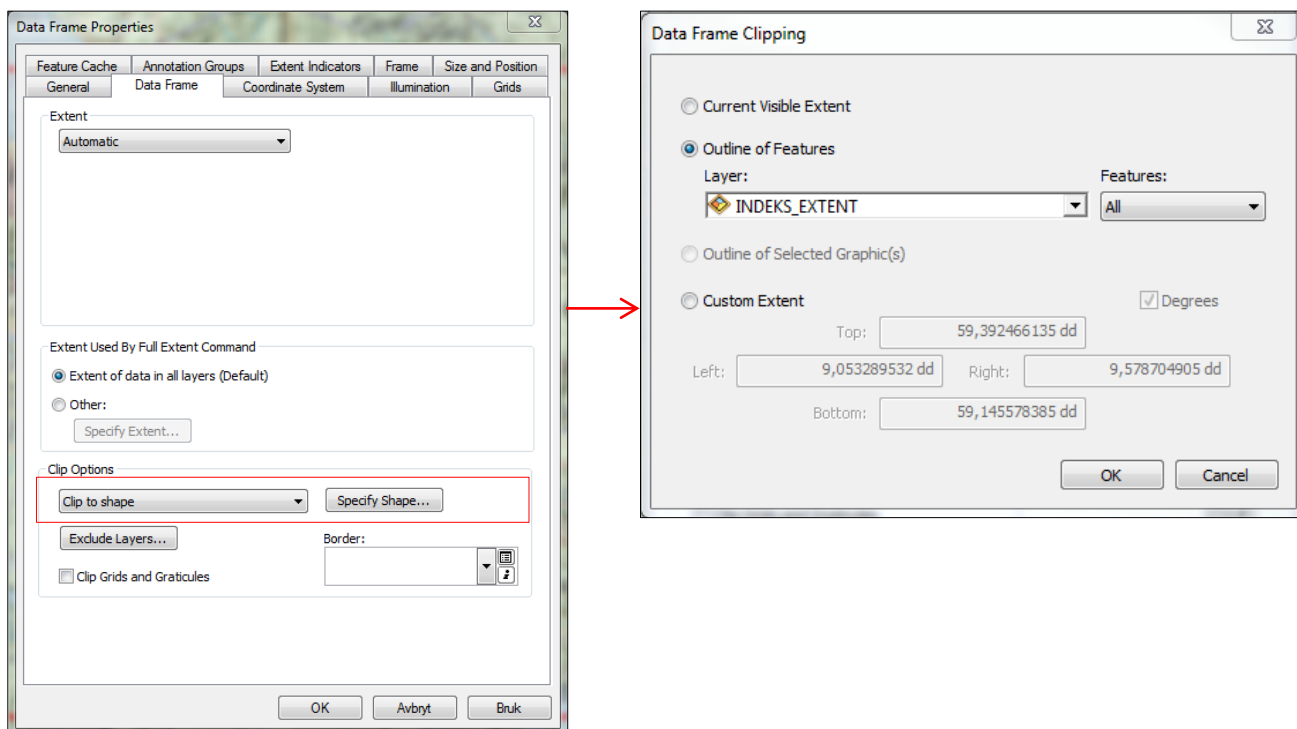


Videre går vi inn i *Properties* i laget *INDEX_EXTENT*. Her går vi til fanen *Defenition Query* og setter *INDEX = 1*. Grunnen til dette er at *data frame* i neste steg skal klippes til operasjonsområdet. Hvis det hadde vært flere polygoner i *INDEX_EXTENT*, ville *data frame* blitt klippet til alle operasjonsområdene for disse polygonene, noe som ikke er ønskelig. Vi vil kun at dataene som ligger i operasjonsområdet skal tegnes.



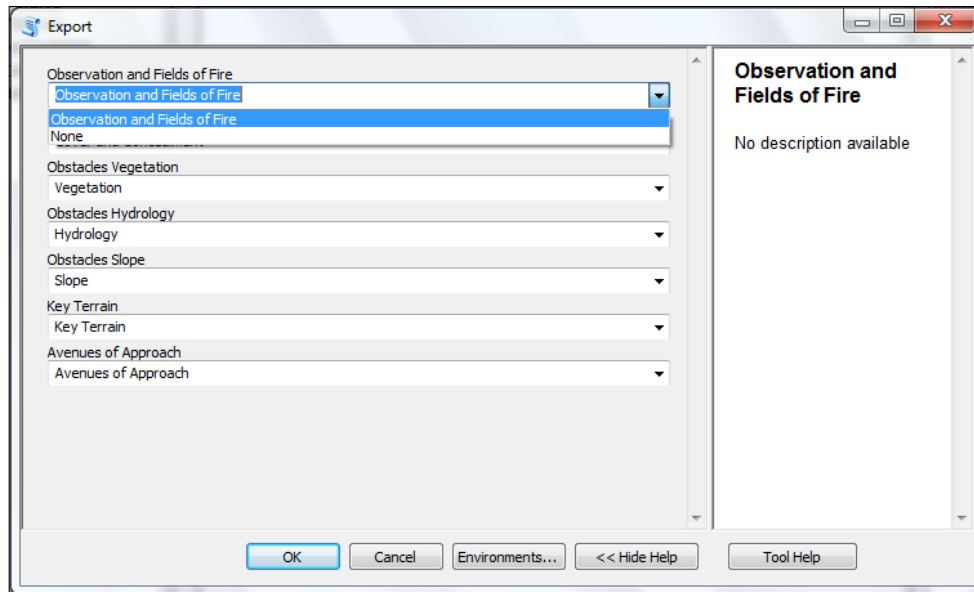
Figur 6.3: Gjennomgang av bruk

Vi høyreklikker på *data frame* vi jobber i og går til *Properties*. Her går vi til fanen *Data Frame* og under *Clip Options* setter vi *Clip to shape*. Deretter trykker vi på *Specify Shape* og huker av på *Outline of Features*. Videre velger man *INDEX_EXTENT* og klikker OK.



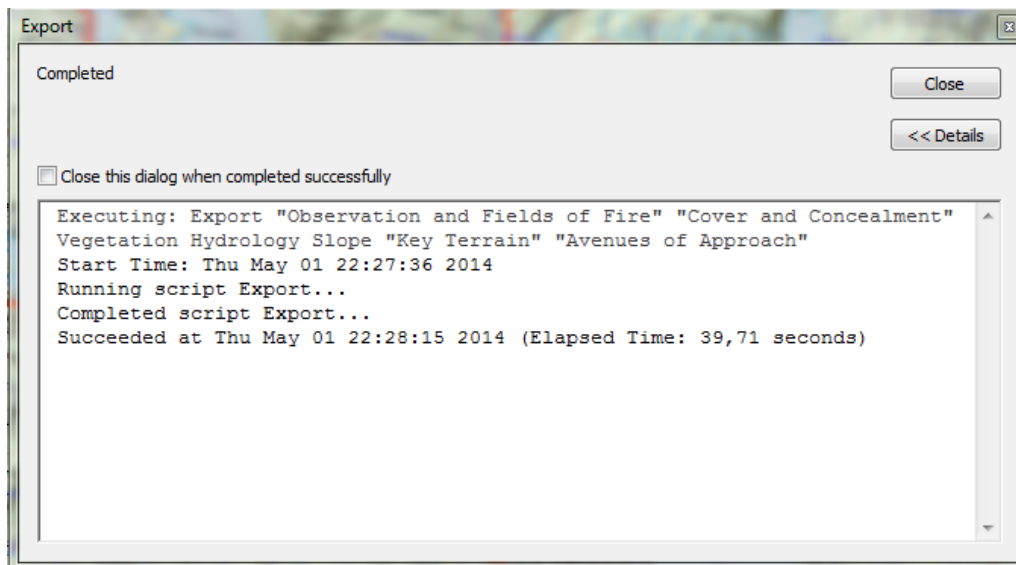
Figur 6.4: Gjennomgang av bruk

Nå som *data frame* er klippet til operasjonsområdet, vises ikke unødvendig informasjon. Det som er synlig i *Layout View* er det som vil bli eksportert. Man setter målestokken og åpner eksportverktøyet. Her kan vi bestemme hvilke lag som skal eksporteres. Ved å sette *None*, vil ikke dette laget eksporteres. Trykk *OK*.



Figur 6.5: Gjennomgang av bruk









Når det står *Completed* øverst til venstre i dialogen er prosessen ferdig.



Figur 6.6: Gjennomgang av bruk



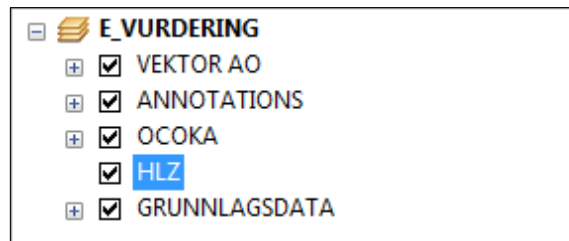
Lagene ender opp i mappen *MXD_OG_EXPORT*.

 Avenues of Approach.png	01.05.2014 22:28	PNG-bilde	806 kB
 Cover and Concealment.png	01.05.2014 22:27	PNG-bilde	778 kB
 E_VURDERING.mxd	01.05.2014 21:45	ArcGIS ArcMap D...	5 710 kB
 Hydrology.png	01.05.2014 22:28	PNG-bilde	810 kB
 Key Terrain.png	01.05.2014 22:28	PNG-bilde	805 kB
 Observation and Fields of Fire.png	01.05.2014 22:27	PNG-bilde	777 kB
 Slope.png	01.05.2014 22:28	PNG-bilde	883 kB
 Vegetation.png	01.05.2014 22:27	PNG-bilde	777 kB

Figur 6.7: Gjennomgang av bruk

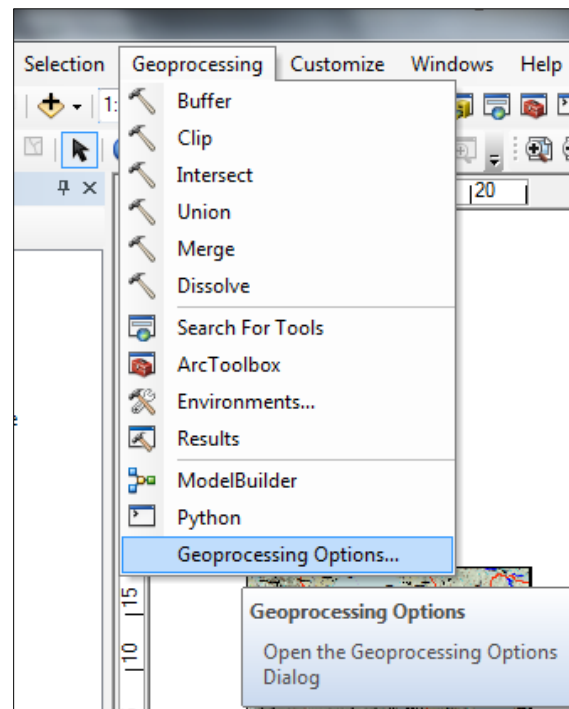
Editering av skriptet

Slik det er i dag, kan skriptet tilpasses for å imøtekomme brukerens behov. Derfor har vi lagt ved en veiledning for hvordan man kan editere skriptet. Hensikten med dette kapittelet er å stegvis belyse hvordan man kan modifisere skriptet. Man kan legge til så mange lag man ønsker ved å gjenta prosessen. Vi åpner MXD'en og legger til et nytt lag med navnet *HLZ* i *TOC'en*.



Figur 7: Editering av skript

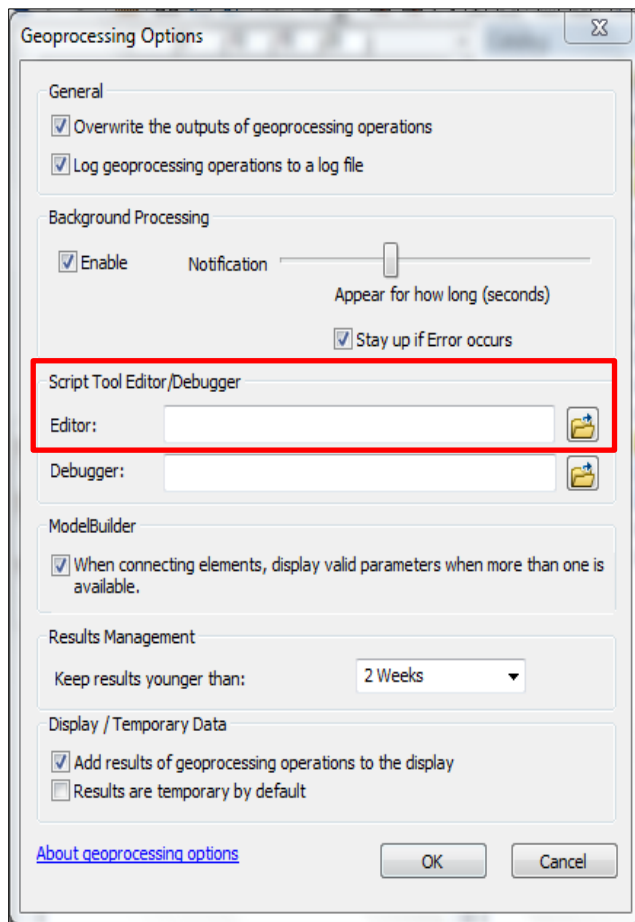
Det er flere måter man kan editere på. For å slippe å navigere til den mappen skriptet ligger i, setter vi «*Notepad++*» som standard teksteditor. Dette gjøres på følgende måte: Trykk på *Geoprocessing* og velg *Geoprocessing Options*.



Figur 7.1: Editering av skript

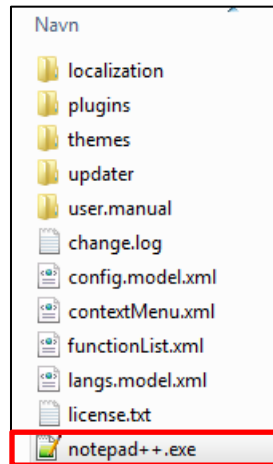


Det dukker opp en ny dialog. Under feltet *Script Tool Editor/Debugger* skal vi legge inn kjør filen (.exe) til «Notepad++».



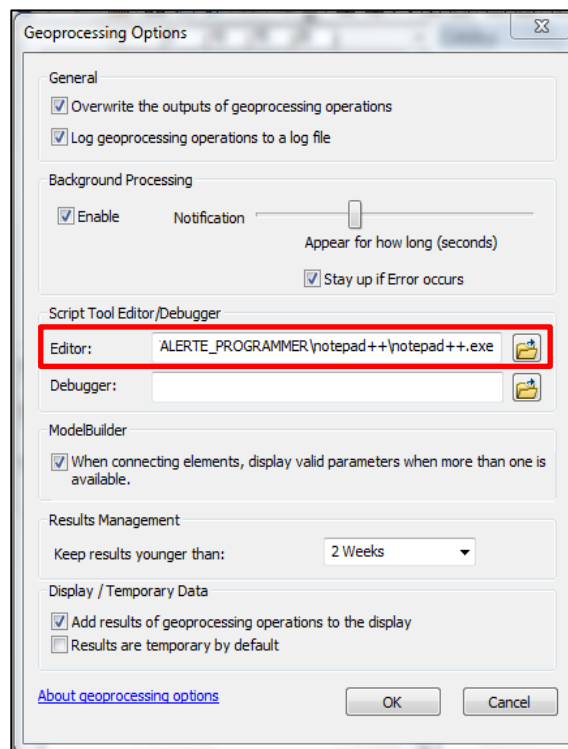
Figur 7.2: Editering av skript

Filen finner vi under installasjonsmappen til «Notepad++». Når den er valgt, trykk *Åpne*.



Figur 7.3: Editering av skript

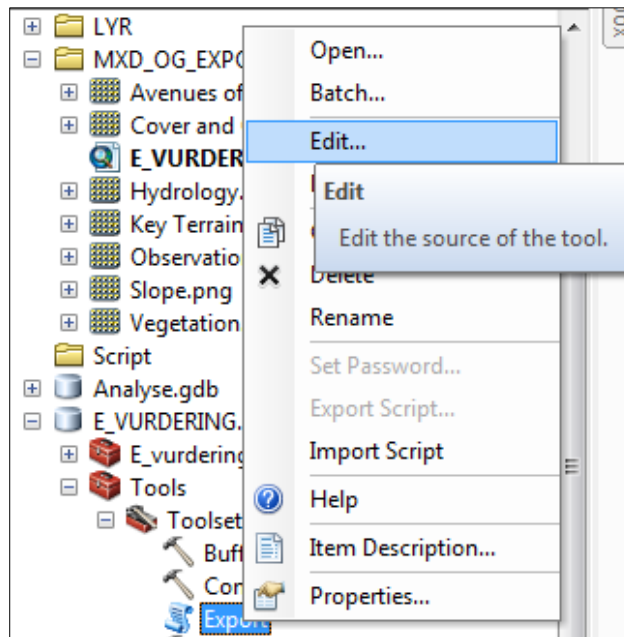
Stibanen til filen blir nå lagt inn. Trykk *OK* og «Notepad++» blir standard tekstediteringsprogram.



Figur 7.4: Editering av skript



Nå kan vi høyreklikke på skriptverktøyet i «Catalogen» og trykke på *Edit*.



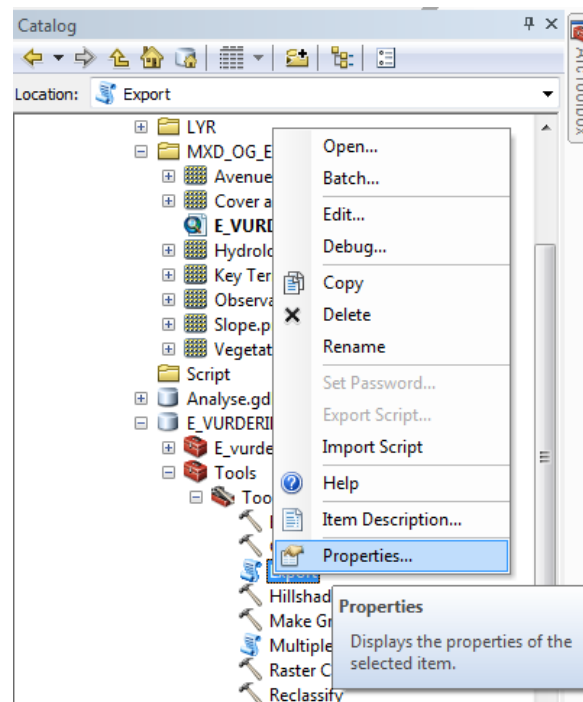
Figur 7.5: Editering av skript

Vi får opp skriptet i «Notepad++». Her skal vi endre på tre ting. Først legger vi til det nye laget i listen *Maplayers*, deretter må vi øke intervallet til *range* funksjonene fra 7 til 8 og til slutt må vi lagre.

```
34
35 # Liste over lagene som skal eksporteres
36 Maplayers = ["Observation and Fields of Fire", "Cover and Concealment",
37 "Vegetation", "Hydrology", "Slope", "Key Terrain", "Avenues of Approach", "HLZ"]
38
39 # Liste som populeres
40 MapToExport = []
41
42 # Importerer parameteren fra skriptverktøyet med samme rekkefølge som i Maplayers
43 # og legger til listen MapToExport
44 for i in range(8):
45     MapToExport.append(arcgpy.GetParameterAsText(i))
46
47 # Sjekker navn fra parametere, hvis navn ikke er "None" så eksporteres laget
48 for q in range(8):
49     if MapToExport[q] != "None":
50         export(Maplayers[q])
```

Figur 7.6: Editering av skript

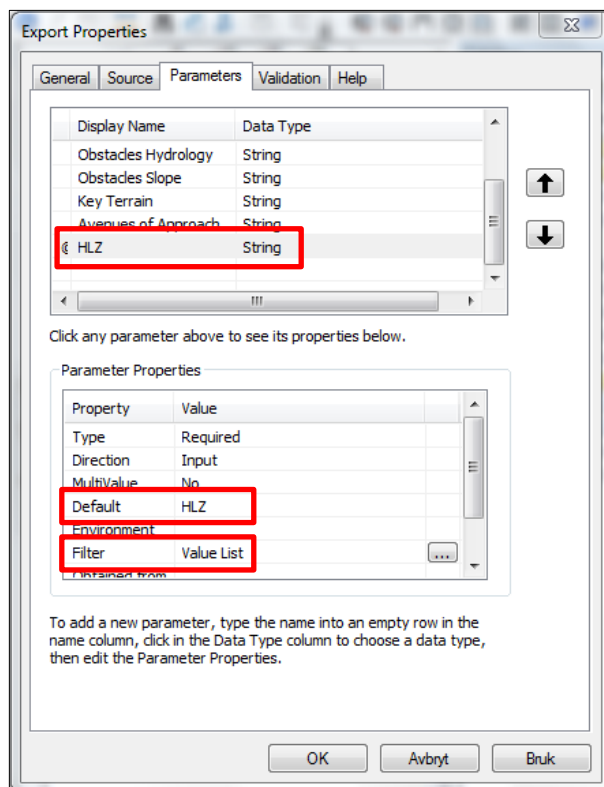
Vi høyreklikker på skriptet og går inn i *Properties*.



Figur 7.7: Editering av skript

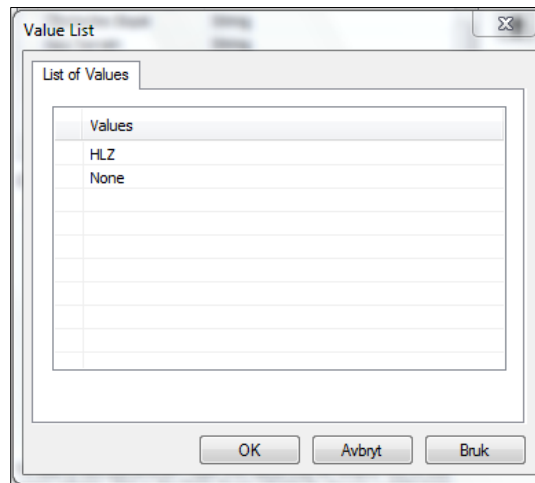


I *Properties* går vi til fanen *Parameters*. Her legger vi inn *HLZ* under *Display Name* og setter datatypen til *String*. Videre legges det inn en standardverdi i feltet *Default* som settes til *HLZ*. Til slutt legger vi til en liste under *Value List* i feltet *Filter*.



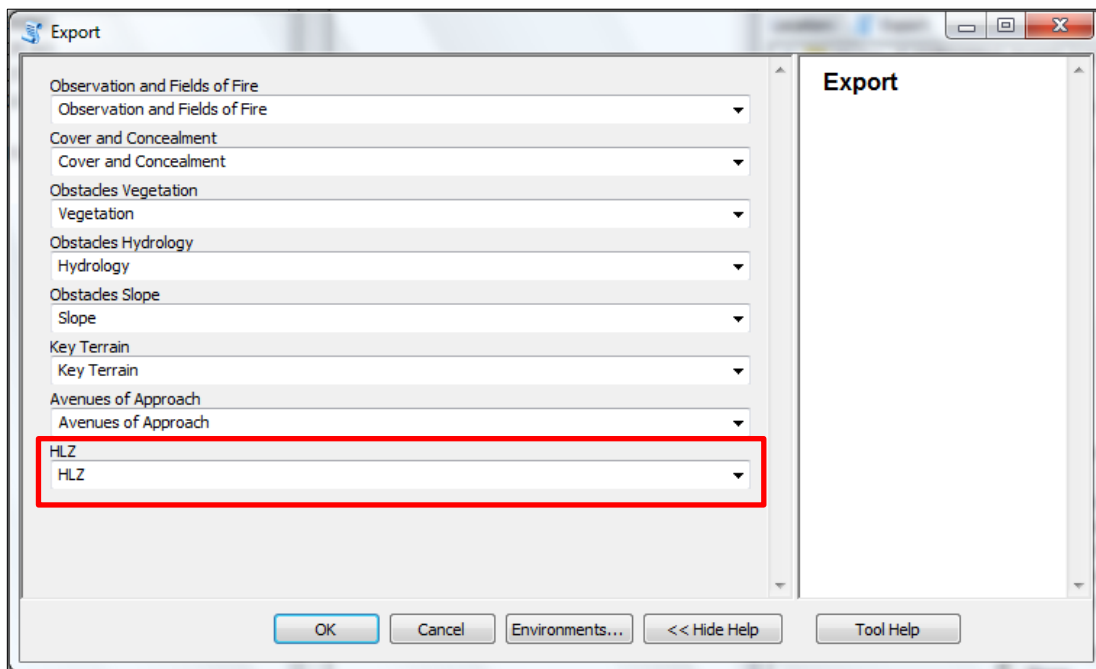
Figur 7.8: Editering av skript

I listen legges verdiene inn som på bildet. Trykk *OK* under her og på *Properties*.



Figur 7.9: Editering av skript

Åpner man verktøyet ser man at laget *HLZ* har blitt lagt til i listen over parametere. Nå kan det legges til nye data under *HLZ* i *TOC'en*.



Figur 7.10: Editering av skript