



# **NILS datafångst och datavärdskap (NIDa)**

## **Projektrapport**

**Saskia Sandring, Pernilla Christensen, Åsa Eriksson, Liselott Nilsson, Anders Pettersson, Johan Svensson**

**Arbetsrapport 430 2014**

---

Sveriges lantbruksuniversitet  
Institutionen för skoglig resurshushållning  
901 83 UMEÅ  
[www.slu.se/sr](http://www.slu.se/sr)  
Tfn: 090-786 81 00



ISSN 1401-1204  
ISRN SLU-SRG-AR-430-SE



# **NILS datafångst och datavårdskap (NIDa)**

## **Projektrapport**

**Saskia Sandring, Pernilla Christensen, Åsa Eriksson, Liselott Nilsson, Anders Pettersson,  
Johan Svensson**

Nyckelord: databas, systemutveckling, kvalitetssäkring, datavårdskap, NILS, datahantering  
database, data quality assurance, data management

Arbetsrapport 430  
Skoglig resurshushållning

---

Sveriges lantbruksuniversitet  
Institutionen för skoglig resurshushållning  
Utgivningsort: Umeå  
Utgivningsår: 2016

ISSN 1401-1204  
ISRN SLU-SRG-AR-430-SE

## Innehåll

1	Inledning.....	3
1.1	Bakgrund .....	3
1.2	Projektets syfte och mål.....	5
1.3	Projektets avgränsning.....	5
1.4	Datavårdskap .....	5
1.5	Studiebesök och kunskapsutbyte .....	6
2	Rutiner och strukturer.....	6
3	Kvalitetssäkring av data .....	6
3.1	NILS data.....	6
3.2	Stöddata .....	8
4	Datasystemet .....	8
4.1	Övergripande .....	8
4.2	Datainsamlingssystem .....	9
4.3	Applikationer för datatvätt och test .....	9
4.4	Applikationer för korrigering av data .....	10
4.5	Lagringsdatabas .....	10
4.6	Analysverktyg och webbgränssnitt.....	10
5	Systemets förvaltning.....	11
5.1	Dataförvaltning.....	11
5.2	SLU:s kvalitetsguide.....	11
6	Utvärdering.....	11
6.1	Delmålen.....	11
6.2	Kostnader i NIDa jämfört med FIND .....	12
6.3	Förutsättningar för datavårdskap .....	12
6.4	Erfarenheter och lärdomar .....	13
7	Framtidsperspektiv .....	14
7.1	Återstående arbete Nilsbas .....	14
7.2	Anpassning till nya fältdatasystemet .....	15
7.3	Anpassning till ny tolkningsmetodik.....	15
7.4	Analysverktyg.....	15
7.5	Webbgränssnitt.....	16
7.6	Tidplan.....	16
8	Sammanfattning .....	3
9	Referenser.....	16
10	Bilagor.....	17

# 1 Sammanfattning

Projektet NIDa (NILS datafångst och datavårdskap) har pågått i 5 år. Syftet har varit att skapa förutsättningar för ett datavårdskap i NILS (Nationell inventering av landskapet i Sverige). Inom projektet har ett system för mottagning av data och grunden för lagringsdatabasen utvecklats. Förutom systemutveckling har projektet genomfört kvalitetssäkring av befintliga data och tagit fram rutiner för korrigerande av data.

I rapporten beskrivs datasystemet för NILS och projektet utvärderas. Komplet systemdokumentation, förvaltningsplan och ekonomisk redovisning av projektet bifogas i bilagor.

## 2 Inledning

### 2.1 Bakgrund

#### 2.1.1 NILS

Miljöövervakningsprogrammet Nationell Inventering av Landskapet i Sverige (NILS) syftar till att följa förändringar i det svenska landskapet och hur dessa påverkar förutsättningar för biologisk mångfald. I ett systematiskt stickprov ingår 631 permanenta landskapsrutor. Inventeringen startades 2003 och har i dagsläget samlat in fältdata i 10 år (två omdrev av hela stickprovet) och fem års data från flygbildstolkningen (ett omdrev).

#### 2.1.2 Förstudie inför NILS datavårdskap (FIND)

NILS behöver ett väl fungerande system för lagring, analys och presentation av data. Projektet FIND (2007) hade som syfte att se över helheten i datahanteringen i NILS för att kunna identifiera särskilt kritiska processer. I rapporten (Esseen mfl. 2008) presenteras en kravspecifikation för de nya system (datahantering, statistik/redovisning och datavårdskap) som behöver byggas upp, de resurser som behövs för detta samt ges ett förslag på hur och i vilken turordning systemen bör byggas upp.

I rapporten föreslås att datahantering i NILS skall delas in i fyra olika huvudprocesser.

Den första processen är Datafångstprocessen, då rådata samlas in, kvalitetssäkras och levereras. I NILS datafångstprocess ingår fältinventering och flygbildstolkning.

Den andra processen är Databasprocessen där kvalitetssäkrade data, metadata och stöddata från datafångstprocessen tas emot och lagras i ett databassystem. I systemet lagras även beräknade data, alternativt beräkningsfunktioner (script) som utför beräkningar direkt beroende på olika användares behov.

Den tredje processen är Statistikprocessen där skattningar, bearbetningar, sammanställningar och resultatrapporter från NILS grunddata produceras i databasen.

Den fjärde processen utgörs av själva Datavårdskapet där det ingår att ta emot och kontrollera data, tillgängliggöra data för externa användare (bl.a. via webb-portal) samt att permanent arkivera kvalitetssäkrade data. I datavårdskapet ingår grunddata från NILS rådatabas samt bearbetade data, resultatrapporter, sammanställningar av data, tabeller, figurer mm. som producerats i Statistikprocessen.

I rapporten gavs följande övergripande förslag till hur huvudprocesserna i NILS datahantering skall genomföras och byggas upp:

- Beställarens uppdrag är klart definierat för utföraren
- Tydligt uppsatta mål finns för arbetet
- Utförlig planering genomförs – baserad på tillräckligt kunskapsunderlag
- Tydligt definierade roller finns för personalen
- Personalen har kompetens, ekonomiska och materiella resurser som behövs för genomförandet
- Hela processen är planerad utifrån delprocesser med tillräcklig upplösning men utan onödig komplexitet
- Delleveranser till avnämare ska ske tidigt och fortlöpande
- Kvalitetssäkring ska finnas i alla processer och delar
- I samtliga huvudprocesser skall det ingå tillräcklig säkerhetskopiering av data, anpassad till varje process särskilda krav
- Tillräcklig dokumentation av alla ingående delprocesser
- Tydlig strategi för att minimera risker
- Fortlöpande dialog och adaptiv utveckling för att säkerställa att målen nås
- Kostnadseffektivitet i alla led – beakta möjligheter till intern och extern samordning
- Samverkan – såväl inom som utom SLU.

Rapporten gav ett förslag till en femårsplan för uppbyggnad av huvudprocesserna. Särskilt kritiska moment i utvecklingsarbetet är bland annat lednings- och organisationsstrukturen, val av programvara för databaserna samt kvalitetssäkring av rådata. Fokus de första åren av arbetet skulle ligga på att skapa förutsättningar för datalagring och kvalitetssäkring i datafångsprocessen. Därpå skulle utvecklingsarbetet inom statistikprocessen genomföras och efter det uppbyggnaden av själva datavärdskapet.

Utifrån femårsplanen har en kostnadsberäkning gjorts för att utveckla och ta i drift samtliga huvudprocesser och komponenter i datahanteringen. Den beräknade kostnaden var per år

4415 kkr	2008
5230 kkr	2009
7147 kkr	2010
7760 kkr	2011
8406 kkr	2012

### **2.1.3 Projektet NILS datafångst och datavärdskap (NIDa)**

Projektet NIDa följde direkt på projektet FIND. Projektet påbörjades 2008 och avslutas 2013 genom leverans av denna slutrapport. NIDa har finansierats av Naturvårdsverket inom

programområdet Landskap, dels via specifika anslag och dels med medel från anslag till NILS basprogram.

## ***2.2 Projektets syfte och mål***

Syftet med projektet NIDa har varit att genomföra utvecklingsarbetet som har föreslagits i FIND.

Under etableringsfasen har fyra konkreta mål formulerats:

- 1) utforma strukturer och rutiner i NILS datafångst del som medger att kvalitetssäkrade data levereras till databasen,
- 2) utveckla och etablera en kvalitetssäkrad databas (Nilsbas) med arkiveringsfunktioner,
- 3) utveckla en analysdel som är direkt kopplad till databaserna och datafångst och som kontinuerligt fungerar som statistikstöd och för kvalitetssäkring av data, samt
- 4) utveckla och etablera publicerings- och exportgränssnitt för kvalitetssäkrade data och information som är tillgängliga för interna för och externa användare, bland annat i form av en webbportal och webb-applikationer för utbyte av data.

Det har tagits beslut att genomföra en del av utvecklingsarbetet i egna projekt (se Projektets avgränsning).

## ***2.3 Projektets avgränsning***

Systemet har utvecklats enbart för hantering av data inom NILS basprogram. Datasystemet har dock byggts på ett flexibelt sätt så att det blir möjligt att även hantera data från andra inventeringsprogram inom ramen för NILS design.

Inom datafångst har projektet fokuserat på kvalitetsäkring, strukturer och rutiner. Utveckling av applikationer som används för datafångst (fältdatorprogram, flygbildstolkingsformulär, datatransfer, flygbildshanteringssystem) har hanterats inom egna utvecklingsprojekt. Frågor kring kvalitetssäkring som gäller personvariation, utbildning i inventeringsmetodik och innehållsfrågor har inte ingått i projektet.

Under 2011 lyftes framtagning av rutiner, ärendehantering, mm. ut till ett eget projekt (Etablering av förvaltningsorganisationen).

Från att ha varit väldigt omfattande i början har NIDa utvecklats till ett mer rent systemutvecklingsprojekt, som också har innefattat praktiskt arbete med att kvalitetsäkra data.

## ***2.4 Datavärdskap***

För att data från miljöövervakningen ska bli tillgängliga finansierar Naturvårdsverket ett system av Nationella datavärddar. Visionen med datavärdskapet är att kvalitetssäkrade data samt metadata ska finnas allmänt tillgängliga via internet. Syftet är att data ska användas och komma till nytta. I syftet med projektet NIDa har det ingått att ta fram förutsättningar för NILS datavärdskap.

## **2.5 Studiebesök och kunskapsutbyte**

NILS principiella design uppvisar stora likheter med Riksskogstaxeringen. Därför sker det ett kontinuerligt utbyte av idéer och erfarenheter kring design, arbetssätt och tekniska lösningar mellan de båda programmen.

För att studera olika lösningar för datalagring och verktyg för att tillgängliggöra data har det genomförts besök på Riksantikvarieämbetet och SMHI. Riksantikvarieämbetet har en databas och hemsida med uppgifter om historiska lämningar (Fornsök). SMHI är datavärdar för marinbiologiska och oceanografiska data som lagras i systemet SHARK. Både riksantikvarieämbetet har valt en ganska flexibel och mer objektbaserad lösning för sin databas. Det gör det möjligt att lagra data som är knuten till väldigt olika typer av objekt. Båda har en extern hemsida där användaren kan söka data antingen med hjälp av ett formulär eller en karta, som i vissa avseenden kan vara intressanta som förebilder för NILS framtida externa webbverktyg.

## **3 Rutiner och strukturer**

Redan i förstudien FIND har organisationsstrukturen utpekats som ett kritiskt moment i utvecklingsarbetet i NILS.

Inom ramen för projektet NIDa har ärendehantering i SharePoint etablerats både för uppföljning av projektet och för dokumentation av korrigeringar i databaserna. Alla ändringar i metodik, databasstruktur och datainnehåll dokumenteras i det nya ärendehanteringssystemet.

Utvecklingsprojektet Etablering av förvaltningsorganisationen har tagit fram en modell för en förvaltningsorganisation vars roll det är att upprätthålla de stödfunktioner och tillhandahålla de hjälpmedel som olika projekt behöver för att uppfylla målsättningen.

Inom ramen för etableringen av förvaltningsområdet Dataförvaltning har ytterligare rutiner och metoder för korrigering av data tagits fram.

## **4 Kvalitetssäkring av data**

### **4.1 NILS data**

#### **4.1.1 Bakgrund**

Sedan starten 2003 har NILS datafångst i fält- och flygbildsinventeringen genomförts med hjälp av datasystem som möjliggör en säker datahantering och flödesstyrning och kontroller vid inmatningen. Framför allt de första åren har det dock funnits begränsningar och brister i systemen som gjorde att det fanns ett kvalitetssäkringsbehov. Dessutom har dokumentationen av redan genomförda tester och korrigeringar varit otillräcklig.

Det genomfördes en ny import av alla originalfiler med NILS fältdata till i en ny temporär sammanslagen databas med hjälp av det nyutvecklade importsystemet. Data från 2004 undantogs, eftersom dessa filer var sparade i ett format som inte gick att läsa med nuvarande



program och datormiljö. Dessa data importerades från en sammanslagen Access-filen som hade skapats efter säsongen 2004.

När det gäller flygbildsinventeringen beror brister i kvaliteten framför allt på fel i tidigare versionen av inmatningsformuläret, då flödesstyrningen ibland inte fungerade.

#### **4.1.2 Korrigering av fel**

Efter dataimporten av fältdata började en arbetsintensiv kvalitetssäkringsprocess, som bestod av:

- (1) Korrigering av fel som fältinventerare rapporterar till kontoret (s.k. blå lappar).
- (2) Rensning av dubletter som uppstår till följd av att olika versioner av fältdatorfiler skickas in vid olika tillfällen av samma lag.
- (3) Korrigering av logiska fel som upptäcktes med det nya importsystemet eller vid analys.
- (4) Kompletterande kartinventering så att alla provytor som ingår i stickprovet finns representerade i databasen.
- (5) Åtgärd av fel som uppstod till följd av buggar i handdatorsystemet vissa år.

#### **4.1.3 Ändringar i inventeringsmetodik**

Ändringar i inventeringsmetodik får ofta till följd att data inte är jämförbara före och efter ändringen. När det är möjligt görs därför anpassningar i databasen. En förutsättning har varit att alla metodändringar som har gjorts över tiden måste dokumenteras. Det finns olika åtgärder för att skapa en kontinuerlig tidserie trots ändringar i inventeringsmetodik. Klassade variabler har i vissa fall klassats om enligt den nya klassningen. I flygbildsinventeringen finns möjligheten att tolka om variabler enligt ny metodik, vilket har gjorts och håller på att göras i vissa fall. I andra fall går det att beräkna ett värde som är jämförbart med den tidigare variabeln. Detta arbete är inte avslutat.

#### **4.1.4 Komplettering av data som saknas**

Det finns flera anledningar till att data saknas i vissa fall. I fält har enstaka provytor saknat data för att dessa har glömts bort vid inventeringen eller för att data har förlorats på vägen och kontrollen vid mottagningen har varit otillräcklig. Den vanligaste orsaken till att det saknas data är dock att provytor inte är tillgängliga för ordinarie inventering på grund av exempelvis beträdnadsförbud eller branter. I flygbildstolkningen saknas vissa data till följd av dålig bildkvalitet. Det kan vara t.ex. skuggor eller moln på bilden.

För att komplettera objekt som saknar data krävs en metodik så att det görs på ett vetenskapligt vedertaget och konsekvent sätt. En kompletterande metodik för flygbildsdata har utvecklats i ett angränsande projekt och korrigeringsprocessen är påbörjad. För fältdata saknas fortfarande en färdigutvecklad metodik för komplettering.

#### **4.1.5 Rimliga värden**

Datansamlingsystemen begränsar vilka värden som är tillåtna för en viss variabel. Trots det kan det ibland matas in värden som inte är rimliga. Orimligheter kan också finnas i

förhållandet mellan variabler. Den sortens fel är svåra att åtgärda i efterhand och kan också vara svåra att upptäcka utan ett komplext regelverk av olika tester. Systematisk felsökning i befintliga data har inte genomförts än. I det nya fältdatansamlingssystemet som utvecklas i projektet NILS Fältdatasystem kommer det att ingå rimlighetskontroller som generar varningsmeddelanden till inventeraren för att åtgärda eventuella felinmatade värden direkt på plats. Planen är att använda regelverket som utvecklas i samband med detta även på befintliga data i framtiden. Även för flygbildstolkningen kan man tänka sig en liknande förbättring av inmatningssystemet framöver.

## 4.2 Stöddata

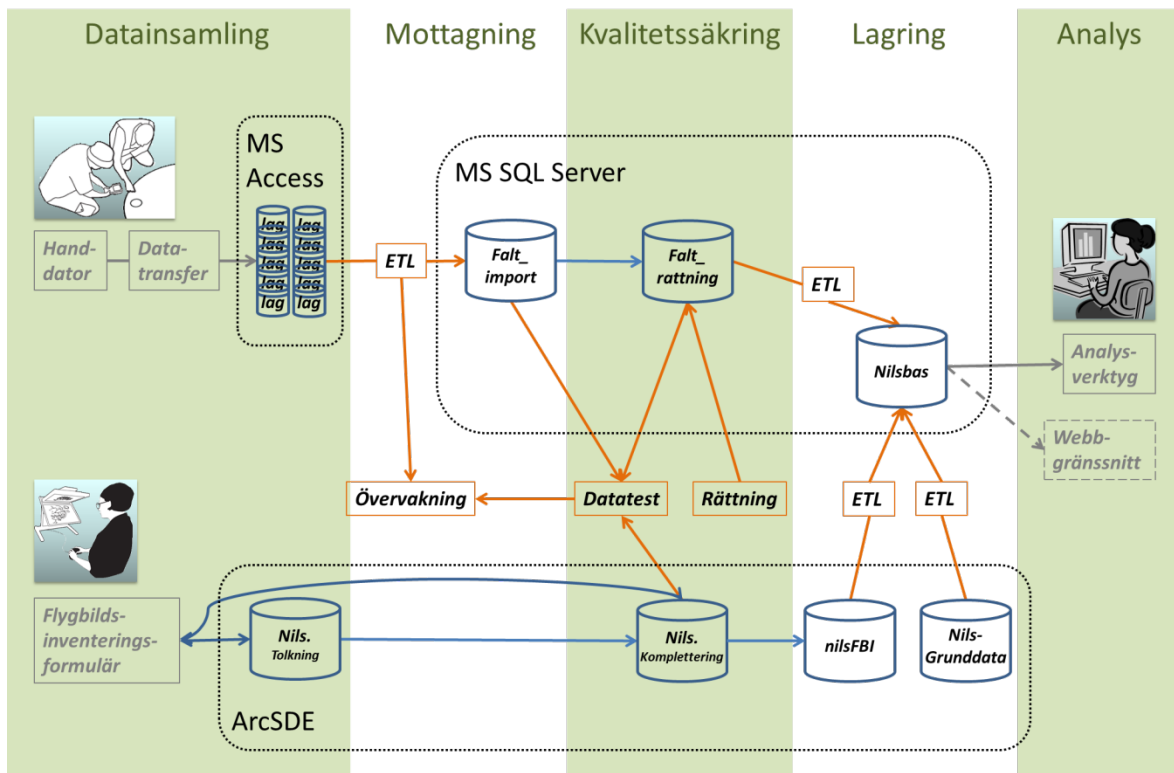
De stöddata som finns i Nilsbas är kvalitetssäkrade, utvärderade och dokumenterade. Exempel på sådana data är NILS grundutlägg, NILS 631 rutor (5 km och 1 km), provytor, inventeringslinjer, rutornas och grundutläggets länstillhörighet samt areal vatten och andel terrester areal som ej ligger inom Sverige för grundutlägg, rutor, provytor och linjer.

Behovet av analysstöd/underlag behöver kontinuerligt utvärderas. Nya stöddataskikt kommer att behöva tas fram, dokumenteras och fastslås. Allt detta skall ske i samråd med övriga avdelningar på Institutionen för skoglig resurshushållning.

## 5 Datasystemet

### 5.1 Övergripande

Syftet med datasystemet är datahantering i NILS från datansamling i fält och flygbildstolkning till analys och leverans av sammanställningar och resultat. Datasystemet består av databaser för mottagning, kvalitetssäkring och lagring av data och egenutvecklade applikationer för överföring och kvalitetssäkring av data.



Översiktlig karta över datasystemet i NILS uppdelat på delprocesser i datahanteringen. Cylindrarna symboliserar databaser och boxarna program/applikationer. Se text för ytterligare förklaring. De utgråade komponenterna har inte ingått i utvecklingsarbetet inom projektet NIDa.

Datainsamling: Flygbildstolkade data matas in direkt i en ArcSDE-databas (**Nils**) med hjälp av ett **formulär**.

I fält matas alla data i en **handdatorenhet**. Innehållet från fältdatasamlarna laddas kontinuerligt ner i **accessfiler**, som inventeringslaget med hjälp av programmet **Datatransfer** skickar till en server på kontoret.

Mottagning och kvalitetssäkring: Varje natt importeras data från accessfilerna till en SQL server-databas (**Falt\_import**). Importen sker via ett program som kallas **ETL** (Extract Transform Load). Programmet **Datatest** används under löpande fältsäsong för att kontrollera att alla data har skickats in komplett. Olika systemdelar rapporterar status om inläsningsjobb och löpande back-uper (**Övervakning**). Efter avslutad fältsäsong sparas en kopia på importdatabasen som back-up och innehållet kopieras till korrigeringsdatabasen (**Falt\_rattning**). Med hjälp av programmet datatest genomförs ytterligare tester.

Efter en komplett årsinventering av flygbilder kopieras innehållet till ett nytt dataset i samma databas för komplettering och korrigerings. Programmet **Datatest** används för validering av logiska regler.

Lagring: I geodatabasen **NilsGrunddata** sammanställs beräkningsunderlag såsom linjelängder och rutornas land- och vattenarealer samt geometrier på inventeringsutlägget, stratumgränser osv. **Nilsbas** är en SQL server-databas som innehåller data från både fält- och flygbildsinventering samt stöddata.

Analys: Framöver kommer det att vara möjligt att komma åt innehållet Nilsbas via **analysverktyg** (interna användare) och ett **webbgränssnitt** (externa användare). Den externa åtkomsten kommer antagligen inte vara direkt utan via ett datalager som mellansteg.

## **5.2 Datainsamlingssystem**

All datainsamling i NILS sker digitalt (se ovan). Systemen har utvecklats och dokumenterats i separata projekt.

## **5.3 Applikationer för datatvätt och test**

Programmet Datatest (Bilaga 1 Systemdokumentation Datatest) validerar data från både fält- och flygbildsinventering.

Fältdata importeras och kontrolleras varje natt under löpande säsong via valideringsregler och facitlistor (Bilaga 2 Systemdokumentation ETL). Lyckad eller misslyckad import rapporteras automatiskt till en SharePoint-lista. Programmet Datatest körs manuellt via ett gränssnitt, där fel kan bearbetas genom att markera dessa som ”behandlade”. Programmet är en .NET-applikation som är programmerad i C#.

Efter avslutad import av årets fältdata sker dubblettrensning i rättningsdatabaserna med hjälp av en funktion i applikationen Datatest. Förutom ytterligare tester i Datatest görs även en uppsättning standardtester i form av SQL-skript manuellt.

Flygbildstolkade data valideras efter genomförd inventering både manuellt med hjälp av FME-skript och programmet Datatest. Först konverteras den tolkade 1100 x 1100 m-rutan från RT 90 till SWEREF 99. Konverteringen ger en icke kvadratisk ruta och rutan klipps därför till 1080 m för att vi även efter konverteringen skall ha data för alla polygoner i rutan och slippa glipor i kanterna. FME-skripten innehåller både topologikontroll av geometrierna och flödeskontroll av attributdata.

#### **5.4 Applikationer för korrigerig av data**

Korrigerig av kända fel genomförs i anslutning till inventeringen varje år i korrigeringsdatabaser (Falt\_rattning för fältdata och NILS komplettering för flygbildstolkade data).

För fältdata sker åtkomst via vyer i MS Access. Eftersom det inte är möjligt att ta bort eller lägga till rader i vyerna görs detta med hjälp av programmet Rättning (Bilaga 3 Systemdokumentation Rättning). Även programmet Rättning är en .NET-applikation i C#.

När Nilsbas är i drift kommer databasen Falt\_rattning att avvecklas och arkiveras. Istället kommer en databas med samma struktur som Nilsbas användas för import och korrigerig av fältdata.

Flygbildstolkade data korrigeras innan avslutad inventering via Flygbildstolkningsformuläret. Data som behöver rättas och/eller kompletteras filtreras fram genom query och calculate field i tabellen för omkodning i ArcGIS alternativt med hjälp av FME queries.

Korrigerig av attributdata görs direkt i tabellen i ArcGIS.

Fel i datainnehållet som upptäcks i olika led av datahanteringen rapporteras till SharePoint-listor. Ärendehanteringslistor fungerar också som dokumentation över vilka ändringar som görs i databaserna.

#### **5.5 Lagringsdatabas**

Data från fält- och flygbildsinventering, samt grunddata för beräkningar lagras i en gemensam databas (Nilsbas) (Bilaga 4 Kravspecifikation Nilsbas, Bilaga 5 Systemdokumentation Nilsbas). Nilsbas är en geodatabas i MS SQL Server med en ny, normaliserad design. Nilsbas är i skrivande stund inte i drift.

#### **5.6 Analysverktyg och webbgränssnitt**

Egna applikationer för datautplock, analys och åtkomst för externa användare har inte utvecklats än. Åtkomst till data för analys sker via vyer. Man kan komma åt vyerna med hjälp av ODBC-uppkoppling från olika program (t.ex. Excel, R, SPSS) beroende på vilka analyser som ska genomföras. Vyerna utformas efter analytikernas behov att genomföra utplock för vidare analyser (Bilaga 4 Kravspecifikation Nilsbas, Bilaga 5 Systemdokumentation Nilsbas). Analysvyerna är specialvyer som byggs upp genom kombination och filtrering av

övergripande vyer. Dessa generella vyer kommer att kunna användas som bas för mer avancerade egenutvecklade analysverktyg.

En testversion av en OLAP-kub har utvecklats till en tidigare version av Nilsbas. Funktionaliteten och användbarheten har dock inte testats än.

## 6 Systemets förvaltning

### 6.1 Dataförvaltning

De delar av systemet som har utvecklats inom projektet förvaltas i förvaltningsområde Dataförvaltning. Förvaltningsplanen (Bilaga 6 Förvaltningsplan Dataförvaltning) synliggör behovet av kompetens och resurser för att underhålla datahanteringssystemet i NILS. I underhållet ingår löpande uppdatering från år till år, användarstöd i verksamheten, uppgradering till nya operativsystem, lagning av fel och mindre förbättringar i funktionen. Vidareutveckling i större skala och nyutveckling av nya systemdelar bedrivs i utvecklingsprojekt. Drift och underhåll av nödvändig hårdvara (dvs. servrar) ingår inte i förvaltningsplanen.

### 6.2 SLU:s kvalitetsguide

SLU har tagit fram en guide för kvaliteten i datahanteringen. NILS deltar i arbetet som innebär att utvärdera datahanteringen utifrån kvalitetskriterier, sätta upp kortfristiga och långfristiga mål för förbättringar och skriva en handlingsplan för det kommande året (Bilaga 7 SLU:s kvalitetsguide).

## 7 Utvärdering

### 7.1 Delmålen

#### 7.1.1 Delmål 1: Strukturer och rutiner i NILS datafångst del som medger att kvalitetssäkrade data levereras till databasen

Dataöverföringen och importen övervakas kontinuerligt och sker på ett säkert sätt. Vid mottagning av data från fält och vid inmatningen i flygbildstolkningen genomförs både automatiska och manuella kontroller. Alla ändringar i databaserna görs enligt rutiner och dokumenteras digitalt. Kvalitetsäkringsprocessen har blivit bättre. Vid fältinventeringen kan datakvaliteten höjas och processen förenklas avsevärt genom att öka kontrollfunktionerna och användarvänligheten redan i inmatningsskedet.

#### 7.1.2 Delmål 2: Kvalitetssäkrad databas (Nilsbas) med arkiveringsfunktioner

Det har genomförts ett stort arbete med att korrigera brister i tidigare års data. Dubletter har tagits bort i fältdatasetet. Felrapporter som har skickats från fält (s.k. blå lappar) har åtgärdats för alla tidigare år. Logiska fel har åtgärdats och databaserna har kompletterats så att alla rutor och provtyper som ingår i NILS stickprov finns representerade. Det som återstår handlar i första hand om att skapa kontinuitet över tiden vid metodändringar, dvs. införa omräkningsmetoder där det är möjligt för fältvariablerna eller genomföra en omtolkning av

variabler i flygbildstolkningen. I andra hand återstår komplettering av data som saknas på grund av brister i datafångstsystemen eller på grund av otillgänglighet i fält eller dålig kvalitet i flygbilderna. Det kräver metodutveckling.

En ny normaliserad databas (Nilsbas) har modellerats. Alla överföring av NILS-data är inte genomförd. Överföringen beräknas vara klar i slutet av 2013.

### **7.1.3 Delmål 3: Analysdel som är direkt kopplad till databaserna och datafångst och som kontinuerligt fungerar som statistikstöd och för kvalitetssäkring av data**

Arbetet med att utveckla analysfunktioner har varit begränsat. När Nilsbas går i drift som analysdatabas kommer analytikerna ha tillgång till vyer, det vill säga sammanställningar av tabeller som underlättar utplock av data. Utvecklingen och testningen av olika analyscript pågår. Färdiga rutiner och verktyg för standardanalyser saknas fortfarande.

### **7.1.4 Delmål 4: Publicerings- och exportgränssnitt för kvalitetssäkrade data och information som är tillgängliga för interna för och externa användare, bland annat i form av en webbportal och webb-applikationer för utbyte av data**

Det har tagits fram ett offertunderlag för utveckling av en webbportal för NILS data. Själva utvecklingen har inte genomförts.

## **7.2 Kostnader i NIDa jämfört med FIND**

I förstudien inför NILS datavårdskap (FIND) (Esseen m. fl. 2008) presenterades ett förslag på tidsplan för att utveckla datahanteringen i NILS under en femårsperiod. Det uppskattade resursbehovet beräknades ligga på totalt ca 33 milj kr för att bygga ett mottagningssystem för fältdata, en ny databas för flygbildstolkning, en databas för datalagring, verktyg för analysverksamhet och datavårdskapet.

Delar av det som togs fram i FIND har genomförts i NIDa som finansierades med totalt ca 9 milj kr (Bilaga 8 Ekonomisk sammanställning NIDa) för mottagnings- och kontrollsystem för datafångsten och grundmodellen för Nilsbas. Enbart dessa delar uppskattades i FIND att uppskattas i förstudien till 16,8 milj kr. För NIDa's del tillkommer då den fortsatta databasutvecklingen 2013-14, vilket innebär att totalkostnaden kommer att bli något högre, men mindre än det uppskattade beloppet i förstudien. Förutom systemutveckling ägnades en stor del av arbetstiden åt kvalitetssäkring. Det påpekas redan i riskanalysen i FIND-rapporten att brister i datakvaliteten samt otillräcklig dokumentation kunde leda till en fördyring av efterföljande led i datahanteringskedjan.

## **7.3 Förutsättningar för datavårdskap**

Vi räknar med att databasen Nilsbas i slutet av 2013 kommer att innehålla NILS data som har samlats in under åren 2003-2012. I och med det kommer kvalitetssäkrade data lagras på ett säkert och effektivt sätt. Den interna åtkomsten kommer att förenklas och leverans av data via NILS analytiker kommer att kunna ske på ett effektivare sätt. Visionen att komma åt NILS data direkt via internet har inte förverkligats än.

Som det påpekades i förstudien FIND skiljer sig NILS från andra miljöövervakningsprogram genom att det är skattningar och bearbetade data som utgör leveransen snarare än grunddata i

sig. Den framtida datavärden måste därför kunna tillhandahålla både grunddata och skattade variabler. Det innebär också att rutiner för standardsammanställningar och analyser måste finnas för att kunna implementeras i ett webbverktyg.

Vid utgången av 2013 kan NILS delvis uppfylla förutsättningar för ett datavärdskap. Ytterligare utveckling behöver genomföras vilket förutsätter att fortsatt finansiering tillförs. Strategier och planering av det fortsatta arbetet bör utgå ifrån en utvärdering av NIDa med FIND i åtanke.

## **7.4 Erfarenheter och lärdomar**

### **7.4.1 Förstudien**

Den gedigna förstudien FIND var av stor nytta för projektet. En stor nackdel var dock att ingen av personerna som var med och genomförde förstudien var med i projektgruppen under det första året. Nyckelpersonerna var delvis inte kvar i NILS och delvis tjänstlediga.

### **7.4.2 Projektgruppens kompetens**

Medlemmarna i projektgruppen var inte bara nya i projektet utan dessutom, med undantag för en person också nya i verksamheten i NILS. Även om FIND-rapporten och det genomförda förarbetet var till stor hjälp för att sätta igång projektet så saknades kontinuitet i tänkandet. Den nya projektgruppen var tvungen att sätta sig in i både verksamhet och lösningar. Det ledde till upprepning av en del arbete som redan var gjort i förstudien.

En annan nackdel var att verksamhetspersonalen och projektledaren inte hade erfarenhet av tidigare systemutvecklingsprojekt.

Under de första åren fanns en anställd systemutvecklare och minst en konsult i projektgruppen. Även den anställde systemutvecklaren var ny i NILS verksamhet och saknade mångårig erfarenhet av systemutveckling. Det saknades även systemarkitektkompetens överlag. Konsulterna hade antingen allmän programmerarkompetens eller specialkompetens (databas).

Eftersom personalen i början hade väldigt liten erfarenhet i rollen som kravställare var det viktigt att systemutvecklarna var lyhörda och hade erfarenhet av samarbete och samverkan i kravfångstprocessen.

### **7.4.3 Projektgruppens kontinuitet**

Projektgruppens sammansättning var till stor del densamma under alla år. Det var en stor fördel för kontinuiteten i arbetet.

De sista två åren har ingen anställd systemutvecklare funnits i projektet p.g.a. finansieringsprioriteringar i NILS och brist på direkt finansiering till NIDa. Under projektets gång har tio olika konsulter varit delaktiga som systemutvecklare. Vissa var planerade för tidsbegränsade insatser med behov av specialistkompetens (databasmodellering, systemarkitektur, kravfångst) medan det i andra fall handlade om att konsulterna flyttades till annan verksamhet utanför SLU. Det optimala hade varit om en erfaren person med bredare

kompetens hade funnits kontinuerligt i rollen som arkitekt och tekniskt projektledare, samt konsulter med specialkompetens i olika perioder av projektet.

#### **7.4.4 Utvecklingsmetodik**

Vi har i utvecklingen arbetat på ett agilt sätt, vilket innebär att arbetet delas upp i regelbundna mindre leveranser och att det sker ett tätt samarbete mellan kravställare och utvecklare. Avstämning sker kontinuerligt och arbetet bedrivs iterativt. Det har fungerat bra med tanke på att kravbilderna och prioriteringar ändras under arbetets gång. Det är också en fördel om kravställaren har en viss insyn i utvecklingsarbetet och utvecklaren en viss insyn i verksamheten, så att båda kan hjälpas åt att hitta bra lösningar ur både det tekniska perspektivet och verksamhetsperspektivet.

Under en period använde vi den agila metoden Scrum (Schwaber & Sutherland 2011). Scrum kräver dock en viss erfarenhet och kompetens för att man ska lyckas fullt ut, vilket vi saknade. Vi märkte också att det i en liten arbetsgrupp med högst två systemutvecklare inte krävs ett så strikt arbetssätt för att få det att fungera. Vi övergav därför Scrums strikta ramar, men behöll arbetssättet med att dela in arbetet i iterationer som slutar med en release av en ny version av applikationen och att ha en aktivitetslista med en prioriteringsordning som ses över i början av varje iteration. Mot slutet av projektet har ärendeflödet förbättrats genom tillämpning av metoden Kanban (Kniberg & Skarin 2010). Rollfördelningen blev tydligare och arbetet mer överskådligt. Metoden är också bra för att synliggöra flaskhalsar. I vårt fall var det testning, som vi behövde ägna mer tid åt för att få arbetet att flyta på.

Det tog lång tid att inse vikten av testverksamhet, dvs. att funktionalitet och användbarhet testas på ett systematiskt sätt innan resultatet godkänns. Det är inte ovanligt att testning glöms bort vid planering av utvecklingsprojekt. Det räcker dock inte att beskriva krav och att implementera kraven i systemet utan det krävs också tid och en metodik för att kvalitetssäkra utvecklingsarbetet och säkerställa att resultatet blev som det var tänkt.

Det är bra om projektledaren känner till olika metoder som används i systemutvecklingen för att hitta ett bra arbetssätt. Framför allt i större utvecklingsprojekt lönar det sig att fundera över vilken metod som ska användas i början av projektet och att se till att alla som är delaktiga i utvecklingen som projektledare, kravställare, arkitekt, programmerare, testare har tillräcklig kompetens i den aktuella metoden.

## **8 Framtidsperspektiv**

### **8.1 Återstående arbete Nilsbas**

Enligt projektplanen 2013 ska Nilsbas gå i drift för analys i slutet av året. Det innebär att alla utplock och analyser av NILS-data görs med hjälp av vyerna i Nilsbas. I början kommer det att finnas behov att åtgärda ytterligare brister som inte har upptäckts vid tidigare tester, tillägg av ytterligare stöddata, rasterdataresultat och beräknade variabler, samt finslipning av vyerna efter behov. Data för vissa tilläggsinventeringar inom NILS basprogram (skogshöns, spillning), variabler som har tagits bort från inventeringen, provytorna 13-16 (inventerades bara 2003) och flaggskeppsrutor har inte importerats.



Kravet att artnamnen i Nilsbas ska uppdateras enligt ArtDatabankens databas Dyntaxa är inte uppfyllt än. TaxonID har tagits fram för de flesta taxa som registreras i NILS, men den tekniska lösningen för uppkoppling och nedladdning av data från Dyntaxa har inte utvecklats än.

En modell för metadata i Nilsbas har inte utvecklats än. Metadata behöver vara tillgängliga både för interna och externa användare vid datautplock och analys och ska uppfylla analytikernas behov och standarder enligt EU:s INSPIRE-direktiv.

Inom MOTH-programmet finns finansiering för att importera MOTH-grunddata, data från provyteinventeringen och, så småningom, även strandinventeringen till Nilsbas våren 2014. Importen av provytedata från tilläggsprogrammen ÄoB och LillNILS och data från fjärilsinventering (ÄoB) och småbiotopsinventering (LillNILS) är inte planerad än.

För att Nilsbas även ska kunna användas för korrigerings av data krävs ett gränssnitt motsvarande applikationen Rättnings, som möjliggör ändringar i databasens innehåll på ett säkert och effektivt sätt.

## ***8.2 Anpassning till nya fältdatasystemet***

NILS håller på att utveckla ett nytt datainsamlingsystem för inventeringen i fält. Omställningen till det nya systemet kommer att innebära vissa variabeländringar som behöver implementeras i Nilsbas. Ambitionen är att båda systemen är anpassade till varandra för att underlätta underhåll och datamottagning. T. ex. ska variabelnamnen vara lika i båda systemen. I huvudsak kommer det nya insamlingsystemet anpassas till det befintliga mottagnings- och lagringssystemet. Men i vissa fall kan det vara mer praktiskt att anpassa databasen.

Programmet Datatest kommer att kunna ersättas av en enklare variant, eftersom felkontroller redan görs vid insamlingen. Samma regelverk ska användas i insamlingsystemet och vid mottagning på kontoret.

## ***8.3 Anpassning till ny tolkningsmetodik***

Metodiken för återinventering av flygbilder håller på att utvecklas. Beroende på hur stora förändringar som görs kommer det att kunna innebära vissa förändringar i databasstrukturen.

## ***8.4 Analysverktyg***

Analysverktyget är tänkt att bestå av två delar, en urvalsfunktion och en beräkningsfunktion (Bilaga 9 Beskrivning av Analysverktyget). Genomgående för byggandet av dessa båda delar är att vi börjat bygga för en liten delmängd i takt med att analysverksamheten växer och fortsätter sedan att utveckla och inkludera fler delar/analysmöjligheter med tiden.

Urvalsfunktionen i Nilsbas består därför av vyer i ett första steg för att först i samband med webbportalen utvecklas vidare till ett urvalsverktyg. Vyerna byggs upp i samband med analyserna och svarar därför väl mot analysbehovet. Utifrån vyerna kan sammanslagningar och urval enkelt genomföras och därefter vidare beräkningar i R baserade på R-script. Utvecklingen av script för olika beräkningstyper sker löpande efter behov och efterfrågan.

## 8.5 Webbgränssnitt

Inom NIDa har ett offertunderlag för en webbportal tagits fram (Bilaga 10). Utvecklingen har dock skjutits på framtiden. En webbportal med extern åtkomst till data skulle ge NILS möjlighet att:

- Publicera analysresultat och rapporter baserade på data insamlade inom ramen för NILS.
- Ge användare möjlighet att ta del av de data som samlats in inom ramen för NILS, både on-line och med möjlighet att exportera data i olika format.
- Öka servicegraden mot avnämare.
- Öka effektiviteten vad gäller möjligheterna att publicera resultat och data på webben.
- Ge användare möjlighet att utnyttja de sök- och analysfunktioner som kommer att finnas på webbportalen.

I offertunderlaget föreslås SharePoint som plattform för webbportalen, men nu förs en diskussion om att andra plattformar kan vara mer funktionella.

## 8.6 Tidplan

När hela datahanteringssystemet i NILS kommer att vara färdigt beror på de finansiella förutsättningarna. Enligt plan avslutas importen av NILS data 2013, så att Nilsbas kan gå i drift. Under 2014 är förslaget att fortsätta utvecklingen av Nilsbas (se ovan) och att fokusera på utveckling av ett internt webbaserat analysverktyg. I så fall skulle det vara realistiskt att vidareutveckla analysverktyget till en webbapplikation för externa användare år 2015.

## 9 Referenser

Esseen, P.-A., P. Christensen, J. Förste, S. Holm, M. Högström, K. Lagerkvist, L. Marklund, A. Ringvall, J. Stensson, S. Sundquist, J. Wikberg & H. Åkesson, 2008. Från datafångst till datavärdskap – översyn av datahanteringen I Nationell Inventering av Landskapet i Sverige (NILS). Arbetsrapport 208. Sveriges Lantbruksuniversitet, Institutionen för skoglig resurshushållning.

Kniberg & Skarin, 2010. Kanban and Scrum - making the most of both.

<http://www.infoq.com/resource/news/2010/01/kanban-scrum-minibook/en/resources/KanbanAndScrum-Swedish.pdf> (svensk översättning från 2013)

Schwaber & Sutherland, 2001. Scrumguiden.

<https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum%20Guide%20-%20SE.pdf>

## **10 Bilagor**

Bilaga 1 Systemdokumentation Datatest

Bilaga 2 Systemdokumentation ETL

Bilaga 3 Systemdokumentation Rättning

Bilaga 4 Kravspecifikation Nilsbas

Bilaga 5 Systemdokumentation Nilsbas

Bilaga 6 Förvaltningsplan Dataförvaltning

Bilaga 7 SLU:s kvalitetsguide

Bilaga 8 Ekonomisk sammanställning NIDa

Bilaga 9 Beskrivning av Analysverktyget

Bilaga 10 Offertunderlag Webbgränssnitt

Bilagorna 7 och 10 publiceras bara på NILS hemsida



## Datatest – systemdokumentation

### Revisionshistorik

Datum	Ändring	Utfört av

<b>1.</b>	<b>Inledning.....</b>	<b>5</b>
1.1	Dokument parametrar .....	5
<b>2.</b>	<b>Bakgrund .....</b>	<b>5</b>
<b>3.</b>	<b>Lösning .....</b>	<b>5</b>
<b>4.</b>	<b>Strukturell modell .....</b>	<b>5</b>
4.1	Komponenter .....	5
4.1.1	Beroenden – Överblick.....	6
4.1.2	Beroenden – External.....	7
4.1.3	Beroenden – Slu.Nils.Base.....	7
4.2	Klassdiagram .....	8
1.1	Baslagret.....	8
1.1.1	ValidationRuleBase.....	9
1.1.2	UndoablePropertyChanged.....	9
1.1.3	SelectStatement .....	9
1.1.4	IBackgroundWorkerOperation .....	10
4.2.1	DBBase.....	10
4.2.2	SerializeManager .....	10
4.2.3	IFieldValueArray .....	10
4.2.4	ICommand .....	10
4.2.5	IWizardSetting .....	10

4.3	Domain – fältinventering & flygbildstolkning .....	10
4.4	Domain – Fjärilsinventering .....	12
4.4.1	FieldDatabaseManagerBase .....	12
4.4.2	FieldValidatorBase .....	12
4.4.3	AerialValidator .....	12
4.4.4	DailyProgressReader .....	12
4.4.5	NilsProject .....	12
4.5	Domain – Flyg .....	13
4.5.1	FeatureRuleTree .....	14
4.5.2	PointRuleTree .....	14
4.5.3	DeponiRuleTree .....	14
4.6	Applications – Datatest .....	14
4.6.1	MainForm .....	15
4.6.2	AerialInterpretationResultViewer .....	15
4.6.3	FieldValidationResultViewer .....	15
4.6.4	NilsAgent .....	15
4.6.5	ValidateCommand .....	15
4.6.6	Settings .....	16
<b>5.</b>	<b>Beteende .....</b>	<b>16</b>
5.1	Kravbild – UCD .....	16
5.1.1	UCD - Validera data för 2010 års data med test "Daglig avprickning" .....	16
5.1.2	UCD – Validera02 - Köra validering .....	18
5.2	Funktionalitet - Activity diagram .....	20
5.2.1	Aktivitetsdiagram 01 - Validera data med test DailyProgress .....	20
<b>6.</b>	<b>Lagring .....</b>	<b>20</b>
6.1	Lagring på fil .....	20

6.1.1	Facitlista.....	21
6.1.2	Behandlade fel.....	21
6.1.3	App.Config .....	21
6.2	Lagring i databas .....	21
<b>7.</b>	<b>Konfigurering .....</b>	<b>22</b>
7.1	Parameterlista .....	23
<b>8.</b>	<b>Deployment .....</b>	<b>24</b>
8.1	Systemkrav .....	24
8.1.1	Förinstallerad programvara.....	24
8.2	Kompilering .....	25
8.3	Installation.....	25
<b>9.</b>	<b>Test.....</b>	<b>25</b>
9.1	Testdata.....	25
9.2	Enhetstester .....	25
9.2.1	Slu.Nils.Domain.Tests .....	25
9.2.2	Slu.Nils.Base.Tests .....	27
9.2.3	Sökväg till test assembly.....	28
<b>10.</b>	<b>Källkod.....</b>	<b>29</b>
<b>11.</b>	<b>FAQ.....</b>	<b>29</b>
11.1	Q: Kan jag testa all flygbildstolkad data samtidigt. Jag får ett minnesfelmeddelande? 29	
11.2	Q: Kan jag testa både punkter, linjer och ytoobjekt samtidigt? .....	29
11.3	Q: Jag får upp fjolårets "behandlade fel" i resultatlistan. Hur tömmer jag detta?..	29

## 1. Inledning

Detta dokument riktar sig till systemutvecklare.

Det syftar att beskriva applikationen i fråga – hur det är uppbyggt strukturellt, hur det beter sig, hur man underhåller det.

Detta dokument tar inte upp hur man installerar och använder programmet, eftersom det finns ett speciellt dokument som behandlar detta, som heter Datatest – Instruktioner.

### 1.1 Dokument parametrar

Namn på servers, databaser, etc , ändras med tiden. I detta dokument refereras därmed dessa namn med parametrar omgivna av hårda parenteser. Dessa parametrar och dess värden är listade nedan:

[server] = terra.srh.slu.se

[instans] = terra.srh.slu.se\nnils

## 2. Bakgrund

Denna applikation föddes pga av behovet att validera och kvalitetssäkra data från fält- och flygbildsinventering. Behovet att fånga upp dessa fel direkt medan inventerare är på plats var essentiellt.

## 3. Lösning

Ett program som varje natt importerar och kontrollerar data via valideringsregler och facitlistor, skickar en rapport till en fältadministratör, som sedan kan ringa ut till inventerare i fält. Programmet kan även köras manuellt via ett GUI, där fel kan bearbetas genom att markera dessa som "behandlade".

## 4. Strukturell modell

Systemet har en 3-skiktad N-tier arkitektur<sup>1</sup>, och nedan beskrivs strukturen på de olika skikten för applikationen, både på en högre nivå där komponenterna i denna applikation beskrivs, men även mer detaljerat på klassnivå.

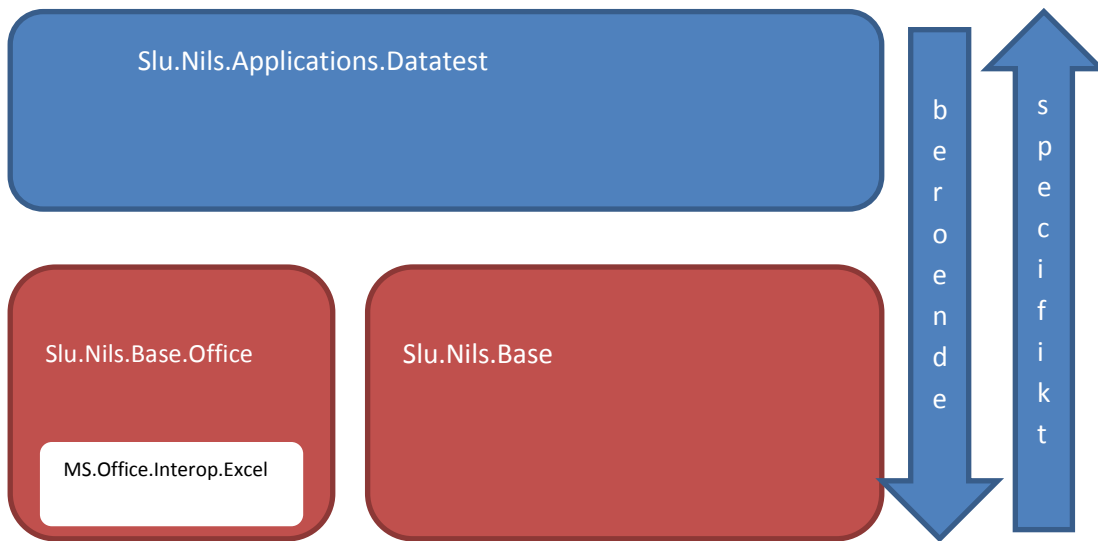
### 4.1 Komponenter

De komponenter som används i denna applikation visas nedan.

---

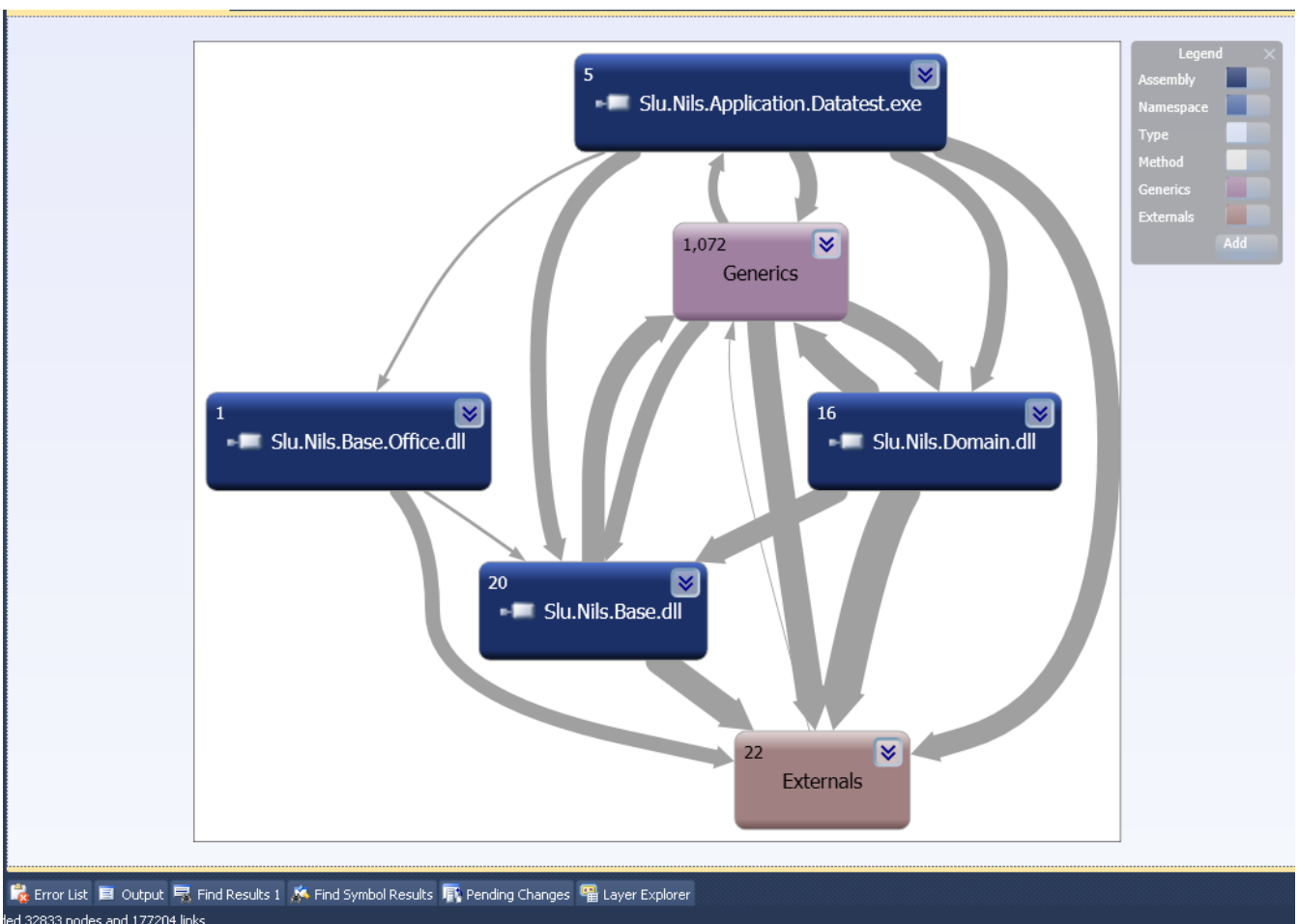
<sup>1</sup> Läs mer om arkitekturen i "LA – Arkitekturbeskrivning".





I Slu.Nils.Base.Office används Microsoft.Office.Interop.Excel, vilket kräver att detta förutom .NET installeras på host-maskinen. [Se avsnittet om Deployment.](#)

#### 4.1.1 Beroenden – Överblick

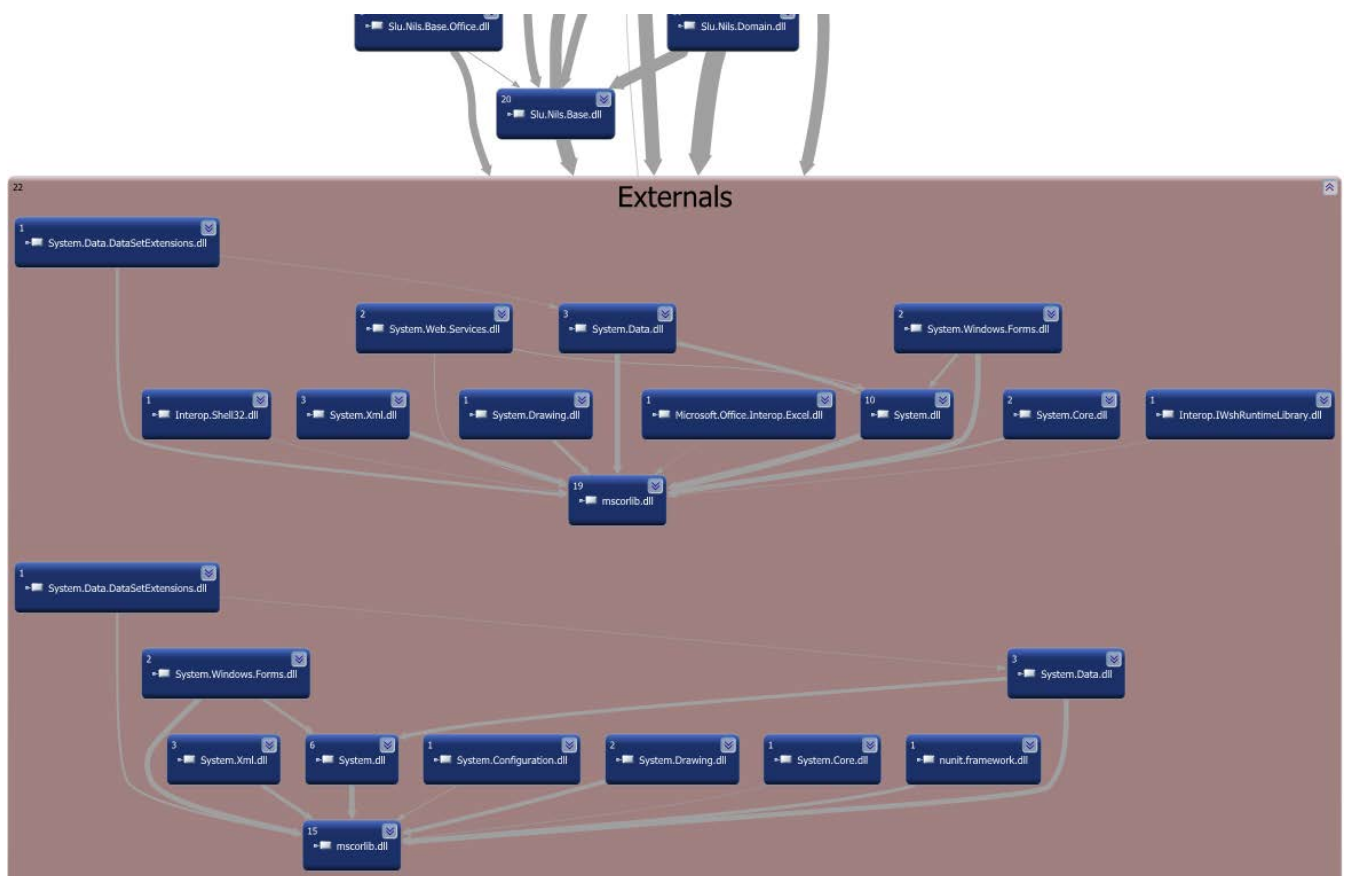


Bilden representerar beroenden mellan olika komponenter mer detaljerat.

- Vi ser att det inte är en strik N-tire arkitektur dvs att applikationslagret endast får prata med domänlagret, eftersom applikationslagret även talar direkt med baslagren.
- Vi har inga beroende uppåt, och inga cirkulära beroenden mellan komponenterna (Assembly)

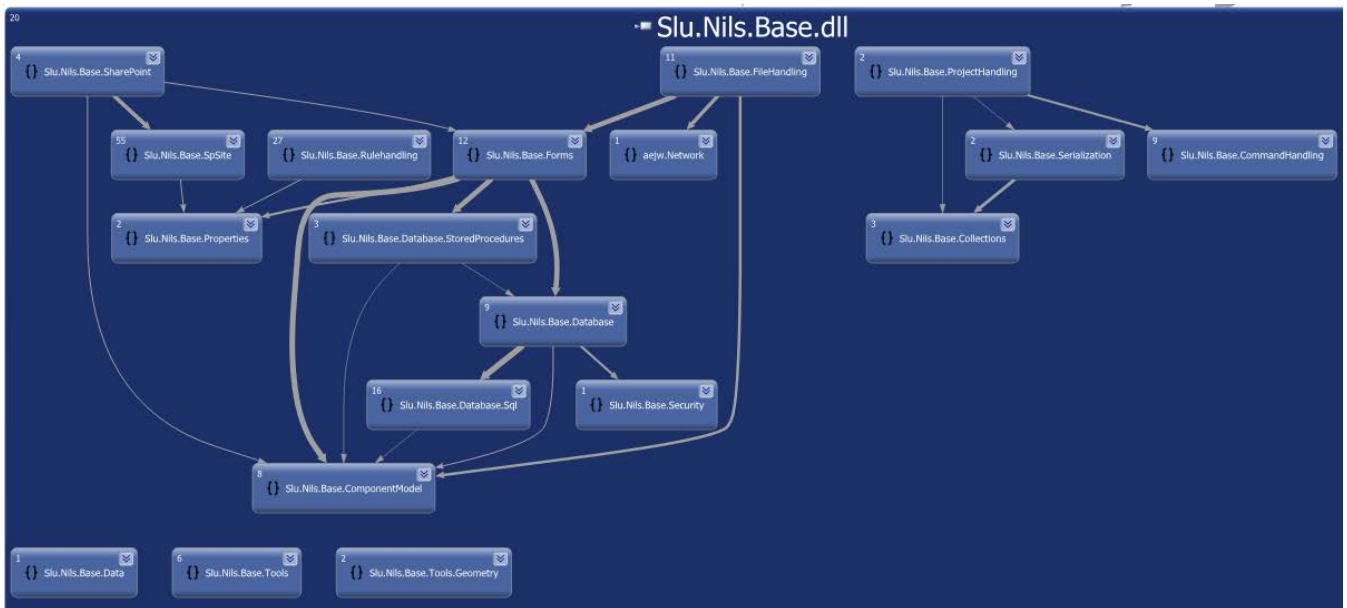
#### 4.1.2 Beroenden – External

Följande .NET komponenter används av Datatest. Se Bilaga och filen: "External.xps" för större och bättre bild.



#### 4.1.3 Beroenden – Slu.Nils.Base

Nedan ses hur olika namespace inom komponenten har beroenden. Se bilaga för större bild "Base.xps".



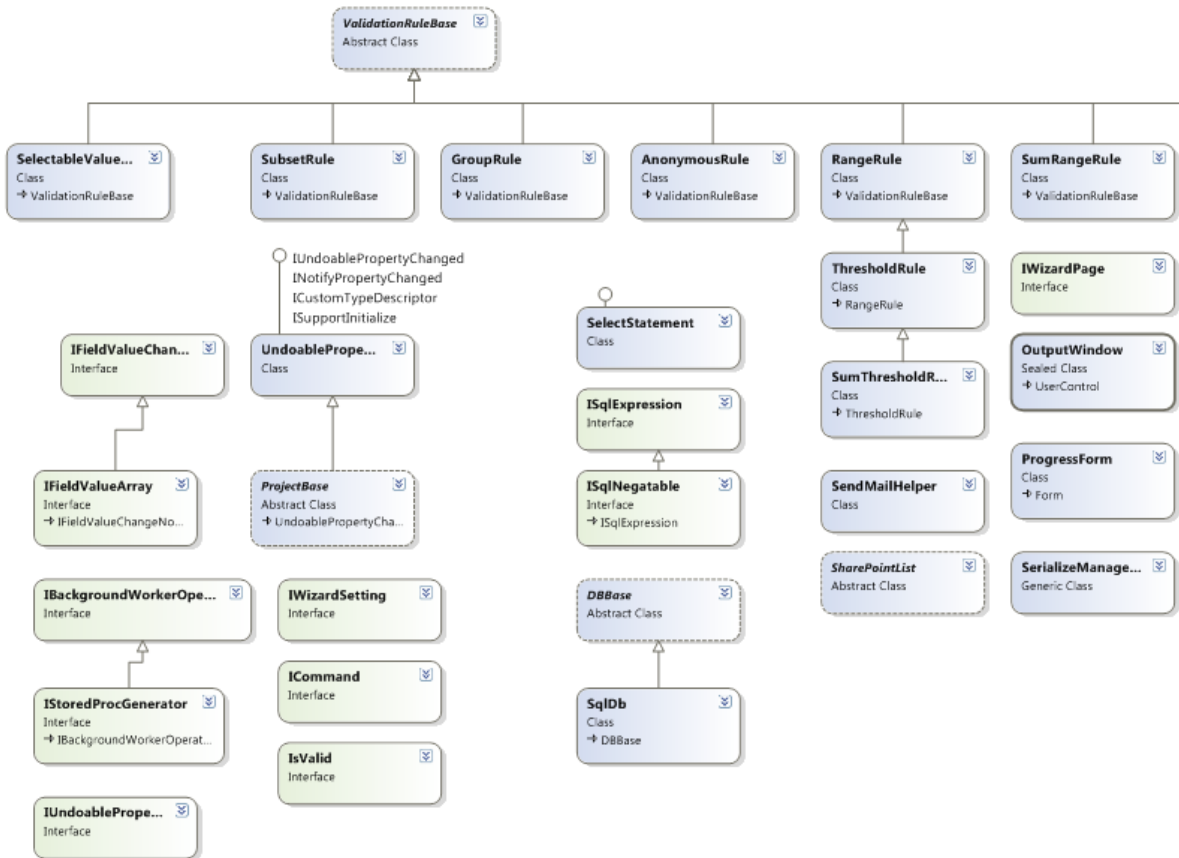
## 4.2 Klassdiagram

Systemet har en 3-skiktad N-tier arkitektur<sup>2</sup>, och nedan beskrivs strukturen på de 3 olika skikten för applikationen Datatest. Att anmärka är att klassdiagrammen endast innehåller en delmängd av alla klasser. Detta för att visa på grundstenarna då inte alla klasser får plats i ett diagram. För att se alla klasser, finns dessa klassdiagram i Visual Studio projekten, och då även i Subversion. I avsnittet 1 läggs fokus på beskrivning av några av de mer betydande klasserna och interfacen i applikationen.

### 1.1 Baslagret

Här nedan beskrivs några klasser i baslagret relevanta för kärnan i systemet och Datatest.

<sup>2</sup> Läs mer om arkitekturen i "LA – Arkitekturbeskrivning".



### 1.1.1 ValidationRuleBase

- **Beskrivning:** Abstrakt basklass där en mängd olika valideringsregler ärver ifrån.
- **Syfte:** Valideringsreglerna används i Datatest samt i Tolkningsformuläret för att kontrollera data mot logiska tester. Både flödesstyrd logik och gränsvärden. Exempel:
  - ”Värde A ska vara mellan 0-100 om värdet B är < 55.

### 1.1.2 UndoablePropertyChanged

- **Beskrivning:** Abstrakt basklass med funktionalitet mot properties och dess eventhantering. Används frekvent i systemet.
- **Syfte:** Att snabbt bygga en klass som ärver av denna för att få funktionalitet att: Undo/redo på en property, innehålla metoder att meddela när en property ändrats via event, Att pausa/starta event rapportering via BeginInit, EndInit.

### 1.1.3 SelectStatement

- **Beskrivning:** Används för att skapa upp SQL utan statiska strängar i koden, och utan att låsa det till TSQL. För närvarande är endast TSQL och SQL-syntax för DataTable.WHERE implementerat. Olika dialekter ges vi ”ToSql, ToAccess”. Denna klass har ett flertal tillhörande klasser som exempelvis WhereCondition, SqlColumn mfl.
- **Syfte:** Att undvika statiska strängar för att skapa dynamisk SQL. Att via ett UI kunna skapa upp klasser som genererar SQL, e.g. QueryBuilder.

#### 1.1.4 IBackgroundWorkerOperation

- **Beskrivning:** Används för att köra en process samtidigt på en annan tråd och automatiskt få en ProgressBar-UI. UI't ProgressBar tar denna som i konstruktorn, och kan då nå metoderna DoWork som den anropar via sin BackgroundWorker. Exempel på en sådan klass är en ValidatorBase-klass eller DatabaseManagerBase-klass, som båda har operationer som tar lång tid: Ladda data och validera data.
- **Syfte:** Att enkelt kunna få upp en progressbar och en process på en annan tråd, för en klass som kör en långtgående process.

#### 4.2.1 DBBase

- **Beskrivning:** Abstrakt basklass att ärva från för att implementera stöd för en ny databasmotor. För närvarande finns endast stöd för Sql server, via klassen SqlDb.
- **Syfte:** Att underlätta stöd för flera databasimplementationer.

#### 4.2.2 SerializeManager

- **Beskrivning:** Klass för att serialisera ner andra klasser till: XML, Binärt eller SOAP.
- **Syfte:** Att spara ner projektinställningar, templates etc.

#### 4.2.3 IFieldValueArray

- **Beskrivning:** Klass att hålla värden som meddelar via event när de ändras via index som mappas mot en databas-tabells kolumnordning.
- **Syfte:** DataTable hade inte rätt funktionalitet, och samtidigt onödig att referera till. Att bygga egna klasser med properties skulle inte fungerat eftersom man ville indexera via kolumnindex på värden och inte fasta properties. Därav byggdes denna för att stödja "IndexValueChanged".

#### 4.2.4 ICommand

- **Beskrivning:** Alla anrop från UI, exempelvis när användaren klickar på knappen Validera, kör ett kommando som i sin tur kör kod, ist för att koden ligger exempelvis ligger i en eventhandler i UI.
- **Syfte:** Dels för att isolera funktionalitet från UI och även för att ge stöd för Undo/Redo på handlingar av användaren.

#### 4.2.5 IWizardSetting

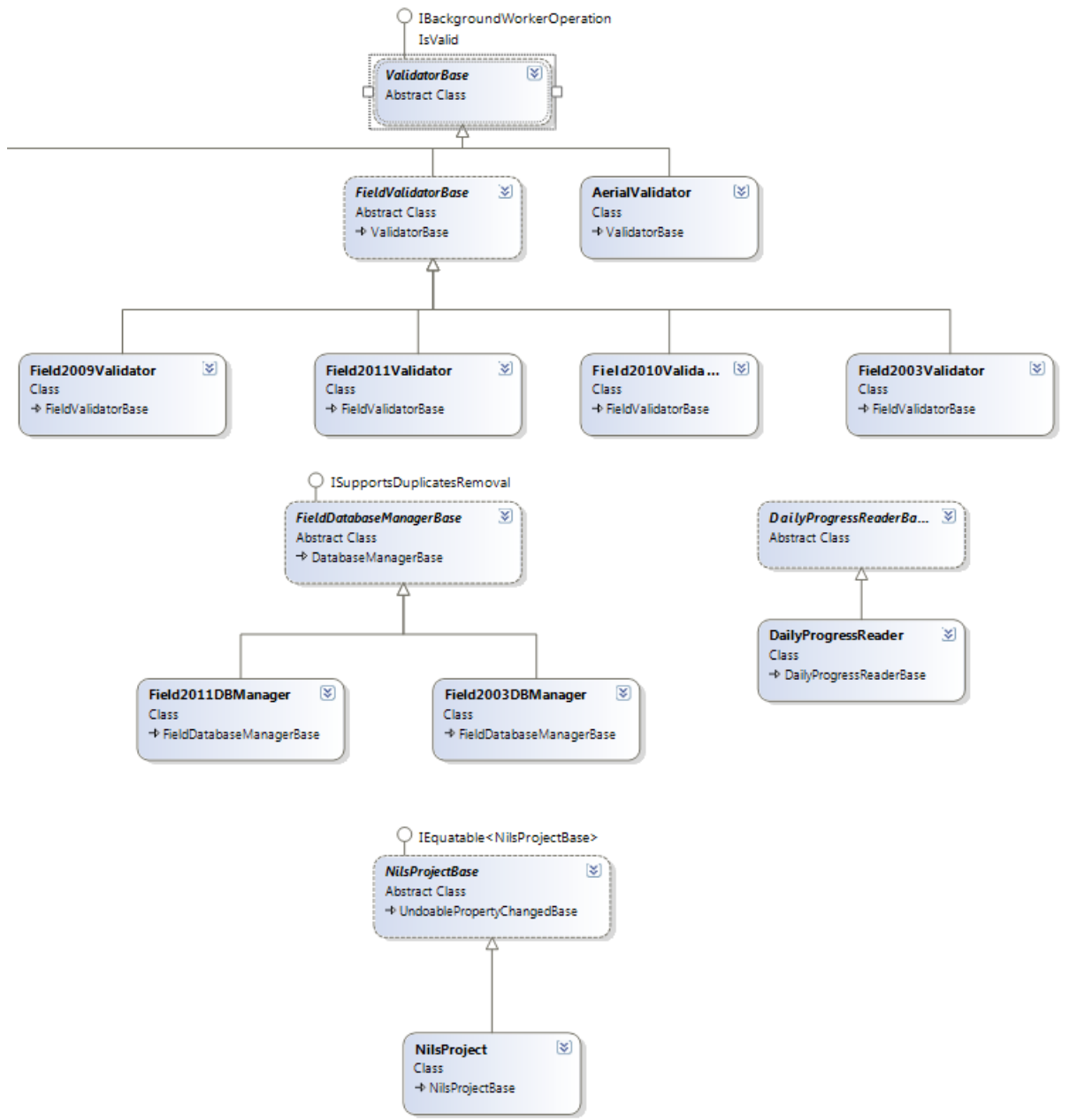
- **Beskrivning:** Om en klass implementerar detta interface, kan klassen konfigureras via en Wizard uppbyggd av IWizardPage-klasser. Man UI gratis.
- **Syfte:** Snabbt skapa en klass med Wizard-funktionalitet som UI för att konfigurera.

### 4.3 Domain – fältinventering & flygbildstolkning

I domän-skiktet finns valideringslogik för både fältdata och flygbildstolkad data. Valideringen sker på olika sätt. Flygbildstolkningen har heltäckande tester, med ca 880 valideringsregler som inkluderar flödeslogik, giltigt intervall för ett värde, saknad av data etc byggt på klassen ValidationRuleBase, medan fältinventeringen har

mer övergripande tester , exempelvis för saknad av provytedata för en Ruta, nullvärden i attribut hörande primärnyckel eller att marktäckte alltid ska anges.

Nedan presenteras en schematisk bild av klasserna som är centrala i domän-skiktet för att validera data. Varje år och inventeringsprojekt har varsin klassuppsättning som bildar en "DataDomain" att validera.



## 4.4 Domain – Fjärilsinventering

### 4.4.1 FieldDatabaseManagerBase

- **Beskrivning:** Hämtar och sparar data till databasen. Innehåller definition på vilken filtrering användaren ska kunna göra via UI när de hämtar data. Innehåller definition på hur dublettrensning ska göras, vilka kolumner som ingår i primärnyckel. Definition på vilken ordning DataTable's ska laddas, sparas ner.
- **Syfte:** Gemensam funktionalitet för kommunikation med databasen för fältinventeringsdata.

### 4.4.2 FieldValidatorBase

- **Beskrivning:** Innehåller valideringstester och funktioner gemensamma för alla inventeringsår för fältdata.
- **Syfte:** Innehålla gemensam funktionalitet för alla år. Impl. IbackgroundWorkerOperation.

### 4.4.3 AerialValidator

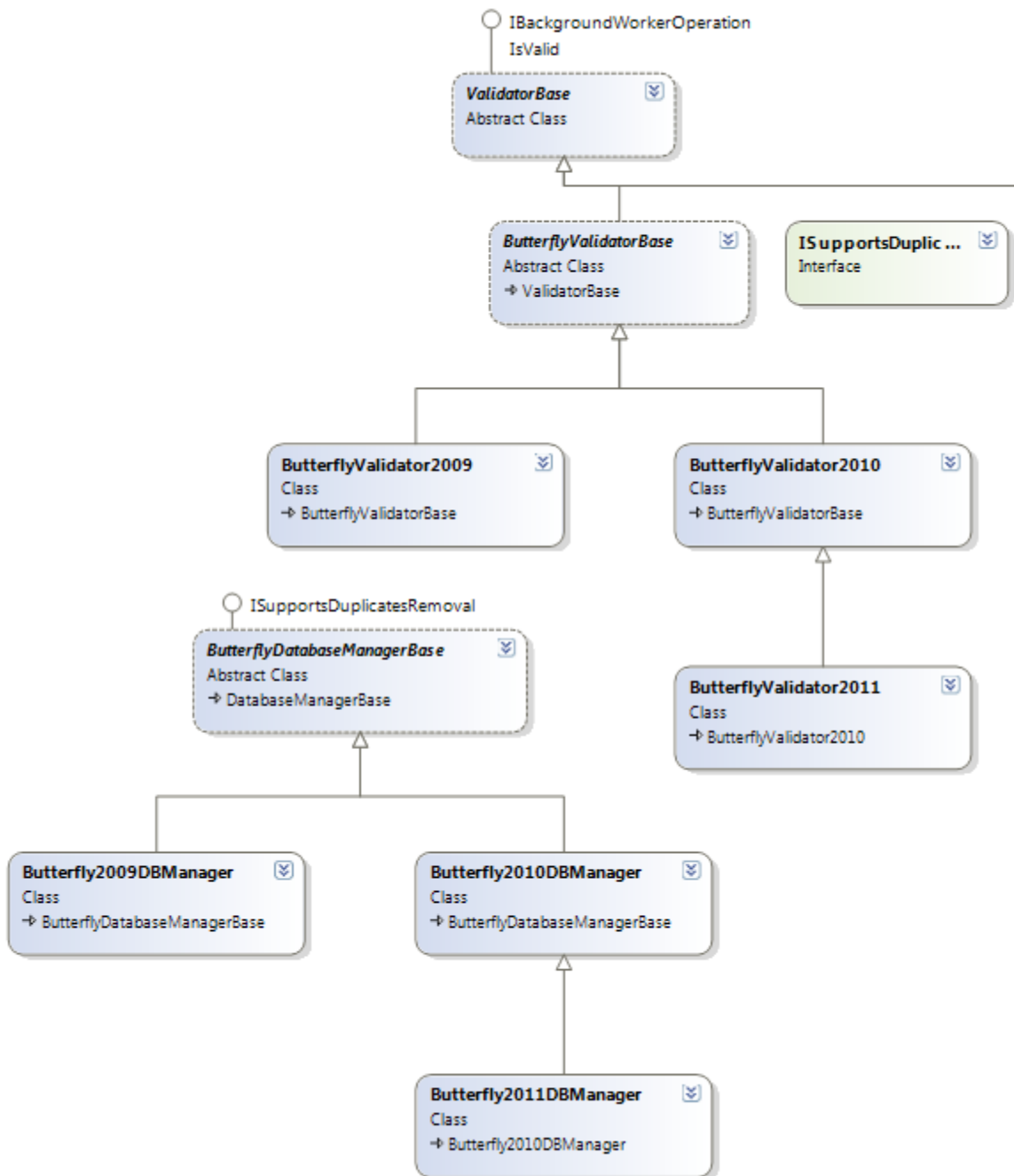
- **Beskrivning:** Innehåller valideringstester och funktioner gemensamma flygbildstolkad data
- **Syfte:** Validera flygbildstolkat data. Impl. IbackgroundWorkerOperation.

### 4.4.4 DailyProgressReader

- **Beskrivning:** Används av valideringsklasser för fjärlil och basnils för att jämföra data i databasen mot dagligen rapporterad data i sharepointlista. Ex: Om inventeraren säger att de har skickat data för provtytor 1-5 för Ruta 123, kollar denna klass om detta stämmer med databasen.
- **Syfte:** Validera att rapporterad data och faktisk data stämmer.

### 4.4.5 NilsProject

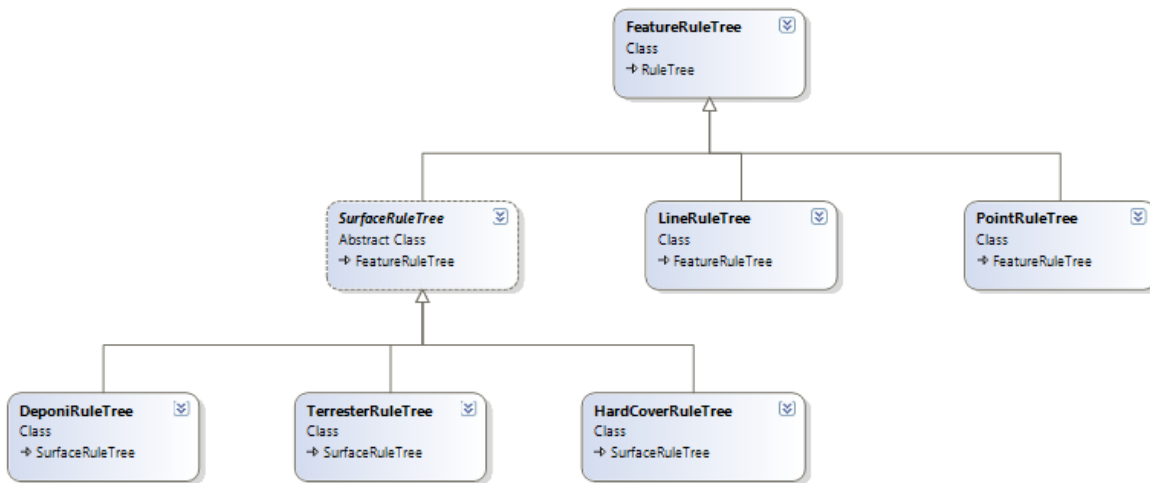
- **Beskrivning:** Projektklass att innehålla definition av en Nils-inventering – dvs att 3 olika data-domäner finns: Fält (som innefattar BasNils, LillNils, MOTH), LillNils provyteinventering samt Flygbildstolkat data. I ett NilsProject finns 3 instanser av klassen DataDomain med respektive ValidatorBase och DatabaseManagerBase klasser.
- **Syfte:** Att definiera vilken data som ska testas, samt spara projektinställningar och resultat för dessa.



## 4.5 Domain – Flyg

Flygbildstolkad data är den data som har mest heltäckande tester av data. Det finns 22 st regelträd (ärver av *FeatureRuleTree*), och i var och ett av dessa träd ca 20-60 regler, totalt ca 880 valideringsregler. Nedan visas endast 5 regelträd: *LineRuleTree*, *PointRuleTree*, *DepoinRuleTree*, *TerresterRuleTree*, *HardCoverRuleTree*, av de 22 träden.





#### 4.5.1 FeatureRuleTree

- **Beskrivning:** Är ett RuleTree som är specialiserat för att fungera även i ESRI's miljö med ArcObjects där klassen Feature är central. Ett RuleTree har en 3 uppsättningar av regler för varje fält/index: Valbara värden, giltigt värde, samt om fältet ska vara enabled eller inte.
- **Syfte:** Att hålla definition av valideringsregler som byggs upp likt grenar i trädet. Ett träd motsvarar ett huvudflöde baserat på ett attribut i tabellen.

#### 4.5.2 PointRuleTree

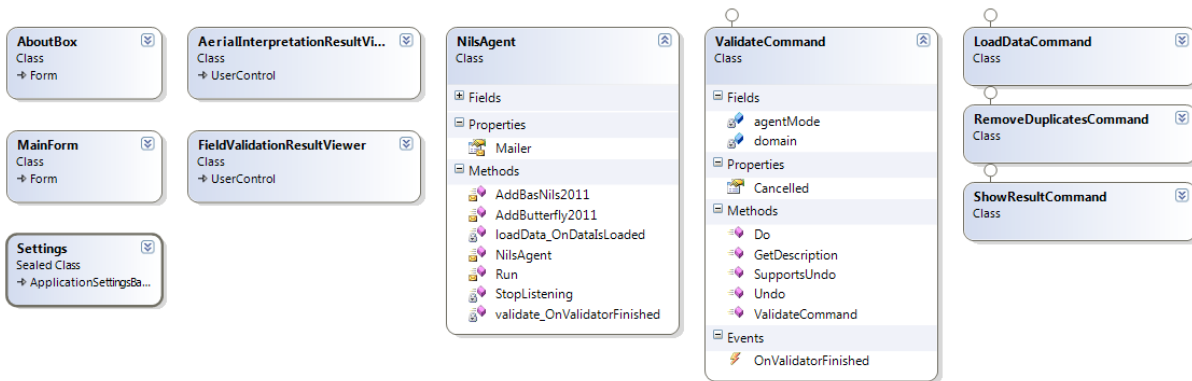
- **Beskrivning:** Är ett RuleTree som är specialiserat mot punktobjekt och de egenskaper flygbildstolkarna kan sätta på dessa.
- **Syfte:** Att styra flöde och validering för punktobjekt när tolkarna tolkar dessa.

#### 4.5.3 DeponiRuleTree

- **Beskrivning:** Är ett RuleTree som ärver av SurfaceRuleTree som representerar ett träd mot ett Ytobjekt, dvs kopplar mot Ytobjekt-tabellen i databasen.
- **Syfte:** Att styra flöde och validering för Ytobjekt som har huvudmarktäcket Deponi.

### 4.6 Applications – Datatest

Nedan presenteras olika klasser i applikationslagret för Datatvätt: Olika UI kontroller och kommandon som används.



#### 4.6.1 MainForm

- **Beskrivning:** Startup form. Formuläret som håller alla andra UI kontroller som: OutputWindow, toolbar, Resultat-UI.
- **Syfte:** Huvudformulär i applikationen.

#### 4.6.2 AerialInterpretationResultViewer

- **Beskrivning:** En resultatvy för resultat från en flygbildstolkad validering i trädform. I Setup-metoden tas en DataDomain som i sin tur håller en ValidatorBase med ett resultat i form av ett RuleTree med RuleError som löv på grenarna. Alla ValidatorBase med ett RuleTree som resultat kan visualiseras.
- **Syfte:** Att visualisera likt ett träd hur flödet blivit fel, och var i flödet felet finns.

#### 4.6.3 FieldValidationResultViewer

- **Beskrivning:** Resultatvy i tabellform med filtrering på Ruta, Lag, Provyta, Feltyp. Fel visas per rad, och man kan markera ett fel som "behandlat" så det färgas grönt och får en kommentar. Informationen om en rad's behandling sparas i klassen NilsError.
- **Syfte:** Att visualisera fel, samt som ärendehantering beta av listan.

#### 4.6.4 NilsAgent

- **Beskrivning:** Klassen som kör Datatest automatiskt vid Autopilot-läge "SuperWizerMode". Innehåller förinställningar av vilken data och tester som ska valideras, samt rapporterar till en sharepointlista via SMTP-mail.
- **Syfte:** Att köra Datatest automatiskt schemalagt.

#### 4.6.5 ValidateCommand

- **Beskrivning:** Ett kommando som kör en validering. Om inte data är laddat så kallas kommandot LoadDataCommand innan. Är ej undoable.
- **Syfte:** Att validera en DataDomain.

#### 4.6.6 Settings

- **Beskrivning:** Innehåller inställningar för applikationen som sparasi app.config. Exempelvis: SMTP-server, SMTP-mailadress, Nuvarande fältår, Default fält databasuppkoppling, Valda datakällor av användaren, Wiki-hjälp-start-url.
- **Syfte:** Att spara inställningar i xml för att slippa kompilera om applikationen för justeringar.

## 5. Beteende

Nedan följer diagram som inte beskriver strukturen av systemet utan snarare beteendet av systemet, hur dess funktionalitet får systemet att agera i sig själv och hur interaktionen med användaren ser ut.

*Aktivitetsdiagram* kan med fördel likt *sekvensdiagram* modellera finare nedbrutna processer, dvs interan processer i systemet för att kommunicera beteende programmerare emellan. *Use case diagram* å andra sidan är bättre lämpad att beskriva interaktion med systemet på en mer övergripande nivå, för att finna aktörer och beskriva funktionalitet för avnämaren. Detta kan iofs även aktivitetsdiagram vilket du ser i detta dokument.

### 5.1 Kravbild – UCD

Use case diagram (UCD) kan användas dels för kravfångst, design och test för att beskriva funktionalitet med aktörer och deras mål/UCD och beroenden mellan olika UCD. Nedan presenteras UCD tillhörande funktionalitet rörande programmet Datatest.

#### 5.1.1 UCD - Validera data för 2010 års data med test "Daglig avprickning"

**Namn:** Validera år 2010 års data med test "Daglig avprickning"

**Ref:** Validera01

**Författare:** jas

**Syfte:** Se att rapporterad data finns inte i databasen samt att inga felskrivningar skett.

**Aktörer:** Lag, Administratör, Sharepoint, Mobiltelefon

**System:** Datatest

**Triggers:** Administratör får en rapport i Outlook att data är inkommer för ruta 123 och fel finns.

**Pre-condition:**

Data för dagen har kommit in i databasen

Administratör har installerat Datatest och har access till databasen att läsa.

Sharepoint-servern är tillgänglig.

**Post-condition:**

Administratör vet felen, har meddelat laget, har markerat felen som behandlande.

Actor (Admin, Lag, sharepoint, Mobiltelefon)

System (Datatest)

<b>Admin</b> startar Datatvätt	Datatest kollar om Admin har access till databasen och skriver sedan uppkopplingstatus i statusfältet. Därefter stänger ner alla knappar som inte användaren ska kunna klicka på: Validate data..., Remove duplicates..., Find duplicates... (samt "Select datasource..." om användaren ej har access till databasen).
<b>Admin</b> klickar på knappen "Database settings"	Datatest öppnar ett formulär med förinställda databaskopplingar.
<b>Admin</b> kollar att datakällan mot "Field data 2010" är inställd mot korrekt server och databas och klickar Ok.	
<b>Admin</b> klickar på knappen "Select datasource"	. Datatest öppnar ett formulär där användaren ser senast valda år och källa i en trädvy med klickbara checkbox-noder.
<b>Admin</b> klickar i noden "Field data 2010"	Datatest klickar i alla underliggande noder under "Field data 2010".
<b>Admin</b> klickar på knappen OK.	Datatest stänger formuläret och gör knappen "Validate data..." valbar för användaren, samt tömmer data i ramminnet om något fanns.
<b>Admin</b> klickar på knappen "Validate data..."	Datatest öppnar ett formulär med en trädvy där olika tester är presenterade. Testerna läses upp via reflection från koden.
<b>Admin</b> letar upp testet "1. DailyProgress..." och klickar på knappen OK.	Datatest visar en progressbar som visar vilka tabeller som läses till ramminnet och laddar in data från det valda året, från alla källor. Progressbaren är modal, vilket betyder att övrig funktionalitet i Datatest är låst för interaktion.
<b>Admin</b> har möjlighet att stoppa processen via en Cancel-knapp på progressbaren, men kan inte interagera med programmet på annat sätt.	Därefter visar Datatest en annan progress bar och kör alla valda test, dvs test "1. DailyProgress..."
	Datatest anropar sharepointlistan för att se vilken data Laget rapporterat att de inventerat.
<b>Sharepoint</b> levererar avprickade provytor för respektive ruta, med status om rutainventeringen är pågående eller klar.	
	Datatest jämför data i databasen mot information från sharepoint för att se om data för marktäcke och provytor för respektive ruta finns. Om det inte stämmer skapas ett fel som sparas i minnet.
	När testet är kört, öppnar datatest en resultatvy för användaren att titta på felen via en tabell-vy som går att filtrera.
<b>Admin</b> skriver in rutanummer 123 i filterfältet "Ruta".	Systemet filtrerar bort alla rader förutom de tillhörande Ruta 123.

<b>Admin</b> ser vilka fel som finns och ringer <b>laget</b> med <b>mobilen</b> .	
<b>Laget</b> svarar och skriver en blå lapp i <b>sharepointlistan</b> , och ger <b>Admin</b> ett Id till blå lappen.	
<b>Admin</b> markerar raden med fel i Datatest och högerklickar.	Datatest visar en pop-up meny där valet "Mark as TREATED" visas.
<b>Admin</b> klickar på knappen "Mark as TREATED"	Datatest markerar raden grön och ger användaren en popup-ruta att skriva kommentar.
<b>Admin</b> skriver blålappsnumret med ett kommentarsmedelande.	Datatest sparar ner: inloggad användare och kommenteraren med blålappsnummer, samt skapar en förhoppningsvis unik nyckel baserat på "felmeddelande"+rutanummer.

### 5.1.2 UCD – Validera02 - Köra validering

Nedan presenteras ett mer generellt use case, som ger en bättre överblick grafiskt, men ej lika specifik som 2.1.1.

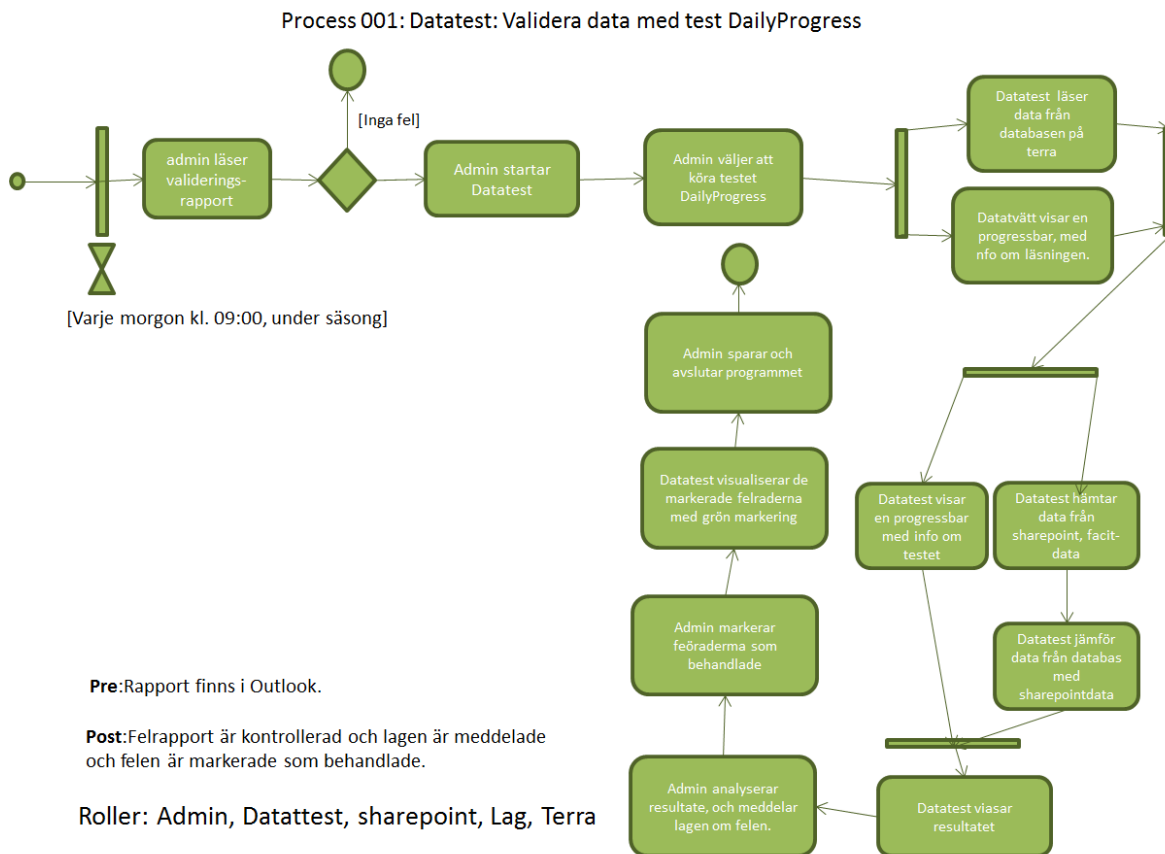
Validera02 –Köra validering

Det man ser tydligt är att systemet består av andra system och är beroende av: mobilnätet, servern terra, sharepoint, Outlook för att fungera. Vi ser att systemet klarar om någon av sharepoint eller Outlook är nere. Förbättringsförslag är att likt att rapporteringen till Admin är redundant från både Outlook och sharepoint, så kan man göra rapporteringen till Laget redundant att även via deras sharepoint-fältportal rapportera felen, som idag endast rings ut till lagen via mobilnätet. På så sätt kan de vara utan mobiltelefon och ändå kunna se dagens felrapport.

## 5.2 Funktionalitet - Activity diagram

Aktivets diagram (AD) används för att arbetsflödet i systemet av stegvisa aktiviteter, med stöd av olika val, iterationer och samtidighet.

### 5.2.1 Aktivitetsdiagram 01 - Validera data med test DailyProgress.



Diagrammet ovan visar de aktiviteter och flöden som sker när administratören kör testet för att validera daglig avprickning mot databasen. Här ser man flödet och samtidighet tydligt, men kanske sämre vilka roller och aktörer som finns med. Därav kompletterar UCD, Aktivitetsdiagram varandra bra.

## 6. Lagring

Nedan beskrivs på vilka sätt denna applikation persisterar data.

### 6.1 Lagring på fil

Vissa saker lagras på fil, exempelvis facitlistor och konfigurering av styrparametrar.

### 6.1.1 Facitlista

Det finns dels en facitlista i Excel som används mot alla provyteinventeringar i testet<sup>3</sup> 1. *Validerar rätt provytor och rutor mot facitlista.*

Filen ligger på terra och är skrivskyddad i editering av struktur, eftersom programmet är beroende av flikens namn, kolumnernas namn och ordning. Denna fil bör inte editeras av annan än en utvecklare, administratör, och detta bör göras på tider på Datatest inte används eftersom man låser filen för programmet när man editerar den. Filen ligger här:

[\\terra.srh.slu.se\2\\_FALT\Fältdata\Metadata\Gemensamt\FACITLISTA\\_ALLA\\_PY\\_2011\\_System.xlsx](\\terra.srh.slu.se\2_FALT\Fältdata\Metadata\Gemensamt\FACITLISTA_ALLA_PY_2011_System.xlsx)

**Endast data i kolumnerna Ruta, Provyta används av datatvätt.** De andra kolumnerna är stödkolumner som kommit med då listan genererats (man skulle i princip kunna ta bort dem). Year-kolumnen skulle man iofs kunna använda om man vill använda samma dokument för alla år, och man skulle kanske vilja välja ut facit för en specifik inventering inventeringar (InvTyp), eller se var provytorna är placerade på en karta (där x och y koordinaterna skulle vara bra).

Denna lista är bra att ha som en Excel-fil (till skillnad från 3.2), eftersom det underlättar att administrera den, via copy-paste.

### 6.1.2 Behandlade fel

De behandlade fel som användaren markerar sparas ner i en Excelfil på Terra. Detta är inte optimalt – bör ändras att denna information sparas i Sql server databasen ist. Anledning varför detta gjordes var av enkelhet.

Filen ligger här:

[\\terra.srh.slu.se\2\\_FALT\Fältdata\Metadata\Gemensamt\BEHANDLADE\\_FEL\\_2011\\_System.xlsx](\\terra.srh.slu.se\2_FALT\Fältdata\Metadata\Gemensamt\BEHANDLADE_FEL_2011_System.xlsx)

### 6.1.3 App.Config

Det finns en hel del inställningar i App.Config, som finns i den katalog programmet körs ifrån, är installerad. Nedan ses vilka parametrar man ställer in i denna:

## 6.2 Lagring i databas

I databas lagras all inventerad data som Datatest ska testa. Denna data uppdateras under säsong varje dag genom att inventerare skickar in data via olika filformat som via ETL-jobb på TERRA läses in i Sql server databasen.

---

<sup>3</sup> <https://arbetsplats.slu.se/sites/srh/Landskapsanalys/NIDa/NIDa%20Wiki/Datatvätt-Tester-Fält.aspx#1>



Det finns en [wikipedia](#) där du kan läsa mer om vilka databaser som finns och vad de gör. Följande databaser är relevanta för denna applikation:

- Falt\_Import
  - Här läser ETL-jobb in data för provyteinventeringarna under säsong. Det finns endast årets data i denna.
- Falt\_Rattning
  - Om man har eftersläntare, dvs access-databaser som kommit efter man redan importerat årets data till Falt\_Rattning, ställer man om ETL-jobbet att läsa in endast eftersläntarna till denna.
- Fjaril\_Import
  - Här läser ETL-jobb in data för provyteinventeringarna under säsong. Det finns endast årets data i denna.
- Fjaril\_Rattning
  - Om man har eftersläntare, dvs filer som kommit efter man redan importerat årets data till Fjaril\_Rattning, ställer man om ETL-jobbet att läsa in endast eftersläntarna till denna.
- Nils6
  - Flyg bildstolkad data.

Konfigurering av vilken data i Falt\_Rattning och Falt\_Import som Datatest ska kontrollera kan styras i databas via tabellen config.ArsvisSokvag, där flaggan "Active" sätts till TRUE för att Datatest ska inkludera den data.

Användaren väljer via UI vilken databas de vill koppla testa. Läs manual om hur detta görs: "Datatest – instruktione.pdf".

## 7. Konfigurering

Följande avsnitt beskriver den konfigurering som kan göras via dess config-fil. Filen finns i samma katalog som exe-filen, och heter *app.config*.

Detta är viktigt att veta när programmet ska sjösättas, flyttas etc. Då måste vissa parametrar ändras. Alla parametrar med typ USER kan användaren sätta om, de andra kan endast systemet ändra på. I detta program är följande parametrar viktiga, och sätta nu.

Name	Type	Scope	Value
DatabaseConnectionString	(Connection...	Application	Data Source=terra.srh.slu.se\nils;Initial Catalog=falt_import;Integrated Security=True
UserConnectionString	string	User	Data Source=terra.srh.slu.se\nils;Initial Catalog=falt_import;Integrated Security=True
SMTPserver	string	Application	193.10.102.40
SMTPmailFrom	string	Application	joakim.astrom@slu.se
SuperVisorFieldYear	string	Application	2011
SMTPmailTo	string	Application	SuSuperwiser@arbetsplats.slu.se
DefaultFieldDbConnection	string	Application	Data Source=terra.srh.slu.se\nils;Initial Catalog=Falt_import;Integrated Security=True
DefaultAerialDbConnection	string	User	Data Source=terra.srh.slu.se\nils;Initial Catalog=Nils6;Integrated Security=True
DefaultButterflyDbConnect...	string	User	Data Source=terra.srh.slu.se\nils;Initial Catalog=Fjaril_import;Integrated Security=True
UserSelecteDatasources	string	User	
UserIgnoreRutor	string	User	999;
WikiHelpStartUrl	string	Application	https://arbetsplats.slu.se/sites/srh/Landskapsanalys/NIDa/NIDa%20Wiki/Dataatest.aspx

## 7.1 Parameterlista

De viktigaste parametrarna beskrivs nedan:

- DatabaseConnectionString
  - Används ej längre. (Om ingen databaskoppling är sparad av användaren används denna. Användes tidigare då endast en databasuppkoppling behövdes)
- UserConnectionString
  - När användaren stänger applikationen sparas databaskopplingen här, och läses nästa gång applikationen öppnas. Används ej längre. (Om ingen databaskoppling är sparad av användaren används denna. Användes tidigare då endast en databasuppkoppling behövdes)
- DefaultFieldDbConnection
  - Används att sätta databasuppkoppling till tester för provyteinventerad data (Moth, Nils, LillNils)
- DefaultAerialDbConnection
  - Används att sätta databasuppkoppling till tester för flygbildstolkad data
- DefaultButterflyDbConnection
  - Används att sätta databasuppkoppling till tester för fjärilsinventerad data.
- SMTPserver
  - Mailserver - Används för att skicka resultat-rapport till en sharepointlista, när programmet körs schemalagt automatiskt.
- SMTPMailFrom
  - Vilken användare som skickar mail. Måste vara en behörig användare att dels använda mailserver, och dels med skrivrättigheter till sharepointlista.
- SMTPMailTo
  - E-post adress till sharepointlista där rapport ska skickas.

- SuperVisorFieldYear
  - Fältår som bestämmer vilka tester som ska köras när programmet körs schemalagt automatiskt.
- UserSelectedDatasources
  - Här sparas de valda datakällor som användaren valt.
- UserSelectedIgnoreRutor
  - Här kan man skriva in rutor som ska ignoreras i tester för provyteinventeringar.
- WikiHelpStartUrl
  - I datatest finns hjälplänkar vid varje test användaren kan välja. Detta är start URL'en till dessa sidor.

## 8. Deployment

### 8.1 Systemkrav

Här nedan ses en tabell över vilka operativsystem som stöds, och vad som krävs för att programmet enligt kan köras enligt användarfallet beskrivet i avsnitt 5. Rekommenderat OS är markerat fetstilt.

OS	X64	X86	Min RAM (GB)
<b>Win 7</b>	X	X	2
Win Vista	?	?	?
Win XP	-	X	2
Win server 2008	X	-	2

#### 8.1.1 Förinstallerad programvara

##### 8.1.1.1 Obligatoriskt

Det krävs även att följande saker finns

- .NET framework 4.0 client profile.
- Microsoft Office Interop 2010
  - DLL'n för denna component heter: "Microsoft Excel 14.0 Object library"

### 8.1.1.2 Valfritt men rekommenderat

Följande programvara bör finnas för att få ut 100% av funktionaliteten:

- Sharepoint-site, krävs för vissa av testerna, som läser från en sharepointlista som "facit-lista".

## 8.2 Kompilering

Här beskrivs hur applikationen måste kompileras, x86 eller x64, vilket .NET framework, om "unmanaged code" används eller inte, debug- VS Release kompilering.

- Eftersom den använder Office komponent, måste den kompileras med "x86" i Visual studio

## 8.3 Installation

I detta avsnitt beskrivs hur applikationen är paketerad för att användaren ska kunna installera den.

Detta är en Click-once applikation och användaren installerar den via en länk som återfinns på sharepoint, i listan "Installationslänkar":

<https://arbetsplats.slu.se/sites/srh/Landskapsanalys/SU/Public/Lists/Installationslnkar/Applikationer.aspx>

## 9. Test

### 9.1 Testdata

Det finns ingen testdata. Testa mot ett tidigare år eller befintligt. Eftersom Datatest enast läser data är detta ofarligt.

### 9.2 Enhetstester

Det en mängd enhetstester för detta program. Se testprojekten i Visual studio och klasserna, eller öppna N-unit och titta på den kompilerade DLL'en.

#### 9.2.1 Slu.Nils.Domain.Tests

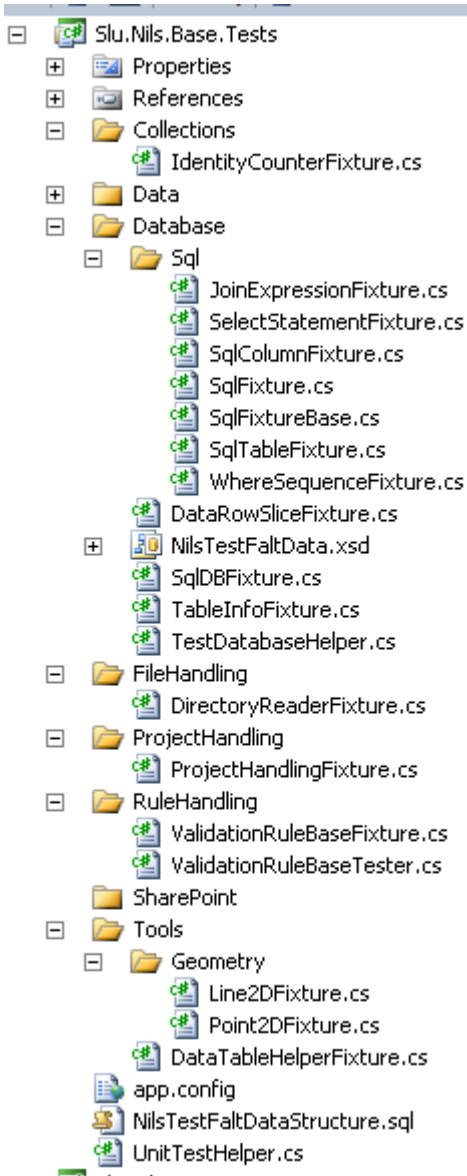
Nedan ses de test-klasser som är relevanta för denna applikation i domänlagret.

- [-] Database
  - [#] DataDuplicateRemoverFixture.cs
  - [#] Field2003DBManagerFixture.cs
  - [#] Field2004DBManagerFixture.cs
  - [#] Field2005DBManagerFixture.cs
  - [#] Field2006DBManagerFixture.cs
  - [#] Field2007DBManagerFixture.cs
  - [#] Field2008DBManagerFixture.cs
  - [#] Field2009DBManagerFixture.cs
  - [#] Field2010DBManagerFixture.cs
  - [#] FieldDBManagerFixtureBase.cs
  - [#] VoidDatabase.cs
  - [#] VoidFieldDBManager.cs
- [-] Sharepoint
  - [#] DailyProgressReaderFixture.cs
  - [#] ErrorLogFixture.cs
- [-] Validation
  - [+] Aerial
    - [#] ButterflyValidator2010Fixture.cs
    - [#] ButterflyValidatorBaseFixture.cs
    - [#] Field2003ValidatorFixture.cs
    - [#] Field2004ValidatorFixture.cs
    - [#] Field2005ValidatorFixture.cs
    - [#] Field2006ValidatorFixture.cs
    - [#] Field2007ValidatorFixture.cs
    - [#] Field2008ValidatorFixture.cs
    - [#] Field2009ValidatorFixture.cs
    - [#] Field2010ValidatorFixture.cs
    - [#] Field2011ValidatorFixture.cs
    - [#] FieldValidatorBaseFixture.cs
    - [#] TestDailyProgressReader.cs
    - [#] TestField2011Validator.cs
    - [#] TestHelper.cs
    - [#] TestSet.cs
    - [#] TestSetFixture.cs



## 9.2.2 Slu.Nils.Base.Tests

Nedan ses de test-klasser som är relevanta för denna applikation i baslaget.



### 9.2.3 Sökväg till test assembly

C:\\$Subversion\Enhetstester\Slu.Nils.Base.Tests

C:\\$Subversion\Enhetstester\Slu.Nils.Domain.Tests

## 10. Källkod

Källkoden, solutionfilen, till detta ligger under följande sökväg i Subversion<sup>4</sup>:

[https://terra.srh.slu.se/svn/Slu\\_Nils/trunk/Applikationer/Slu.Nils.Application.Datatest](https://terra.srh.slu.se/svn/Slu_Nils/trunk/Applikationer/Slu.Nils.Application.Datatest)

**Lokalt:** C:\\$Subversion\Applikationer\Slu.Nils.Application.Datatest

## 11. FAQ

I detta avsnitt tas vanliga frågor och svar upp. Dessa frågor härstammar från användare av applikationen och kan beskriva både möjligheter och begränsningar som applikationen har.

### 11.1 Q: Kan jag testa all flygbildstolkad data samtidigt. Jag får ett minnesfelmeddelande?

Nej, i dagsläget tar ram-minnet slut eftersom applikationen är kompilerad som en 32-bitars applikation<sup>5</sup>, och därav endast kan nyttja 4GB trots att du har mer minne i din dator. Detta går att lösa på 2 sätt:

A. Dela upp testningen. Kör exempelvis 20 rutor i taget.

B. Kompilera om Datatest så den kan nyttja mer ram-minne.

En systemutvecklare kan kompilera om programmet till 64 bitars läge, och då först plockar bort funktionalitet som gör den beroende att vara en 32-bitars applikation. Dvs, att man har en version av Datatest för flyg (64-bit) och en version för fält (32-bit).

Tidsskattning för detta mellan tummen och pekfingret skulle jag säga är 8 h, förutsatt att programmeraren läst systemdokumentationen och kan Visual studio, C# .NET.

### 11.2 Q: Kan jag testa både punkter, linjer och ytoobjekt samtidigt?

Ja, men rekommenderas ej. Det finns inget gränssnitt att därefter filtrera resultatet på att "titta endast på punktobjekt". Det som visas är en lista med objekt-id'n: punkter, linjer, ytor, blandade.

Rekommenderat sätt är att titta på en typ i taget.

### 11.3 Q: Jag får upp fjolårets "behandlade fel" i resultatlistan. Hur tömmer jag detta?

- A. Antingen kopiera det befintliga Excel-dokumentet, döpa om det, tömma det på data, och konfigurera om en parameter i Datatest. På detta sätt har ni spårbarhet kvar från förra årets behandlade fel. Ex på kopia:

<sup>4</sup> Läs Systemarkitekturbeskrivningen, under dokumentationsmappen "1. Enterprise arkitektur", för mer info om Subversion.

<sup>5</sup> Detta pga av att den använder Office-komponenter för facitlista och att komma ihåg behandlade fel för tester för provyteinventeringen.



- [\\terra.srh.slu.se\2\\_FALT\Fältdata\Metadata\Gemensamt\BEHANDLADE\\_FEL\\_2012\System.xlsx](\\terra.srh.slu.se\2_FALT\Fältdata\Metadata\Gemensamt\BEHANDLADE_FEL_2012\System.xlsx)

B. Töm det befintliga dokumentet. Ni förlorar fjolårets "ärendehantering", men å andra sidan är denna kanske inte så intressant att spara om ni redan är klar med rättningarna.

Bästa är att bygga om denna funktionalitet, att istället spara denna info i databasen<sup>6</sup> och göra ett UI i Datatest att "tömma befintligt års logg", "byta år". Om man då sedan även bygger om att läsa alla facitlistor från sharepoint är man inte beroende av Excel eller Office-komponenten längre.

---

<sup>6</sup> Detta för att kapa bort beroenden till Office-komponenten, som gör att programet måste kompileras i x86.



Sveriges lantbruksuniversitet  
Swedish University of Agricultural Sciences

## ETL – systemdokumentation

<b>1. Inledning</b>	<b>4</b>
1.1 Dokument parametrar	4
<b>2. Bakgrund</b>	<b>4</b>
<b>3. Lösning</b>	<b>4</b>
<b>4. Struktur</b>	<b>5</b>
4.1 Komponenter	5
<b>5. Beteende</b>	<b>6</b>
5.1 UCD	7
5.2 Aktivitetsdiagram	7
<b>6. Lagring</b>	<b>7</b>
6.1 Indata – source	7
6.1.1 CSV	8
6.1.2 MDB	8
6.1.3 Shapefiler	8
6.2 Utdata – target	8
6.2.1 Importdatabaser	8
6.2.2 Nilsbas	9
6.3 Konfigurering	9
6.3.1 Databas	9
6.3.2 Enironmentvariabler	9
<b>7. Konfigurering</b>	<b>9</b>
7.1 Fjäril	9
7.1.1 Environment variable	10
7.1.2 Datasources i ETL-jobben.	10

7.2	Provyteinventering: Nils, Moth, LillNils .....	10
7.2.1	Environment variable .....	11
<b>8.</b>	<b>Deployment.....</b>	<b>12</b>
8.1	Systemkrav.....	12
8.1.1	Förinstallerad programvara .....	12
8.2	Kompilering .....	13
8.3	Installation.....	13
<b>9.</b>	<b>FAQ.....</b>	<b>13</b>
9.1	Q: Hur gör jag om strukturen ändrats från förra året? .....	13
9.2	Q: Det blir dubletter av rader pga av i, hur hanteras detta? .....	13

## 1. Inledning

Detta dokument riktar sig till systemutvecklare.

Det syftar att beskriva ETL-jobben rörande NILS data – hur de är uppbyggda, hur det beter sig, hur man underhåller dem.

Detta dokument tar inte upp hur man installerar och använder dem, eftersom det finns ett speciellt dokument som behandlar detta under i dokumentationen här:

*”3. Manualer/Admin – ”Årsvi, databaser, program.docx.*

Läs även detta för att få en bra förståelse att bygga ETL-jobben flexibla, ex: så användaren kan sätta parametrar för att peka om mot en annan databas.

### 1.1 Dokument parametrar

Namn på servers, databaser, etc, ändras med tiden. I detta dokument refereras därmed dessa namn med parametrar omgivna av hårda parenteser. Dessa parametrar och dess värden är listade nedan:

[server] = terra.srh.slu.se

[instans] = terra.srh.slu.se\nnils

## 2. Bakgrund

ETL jobben byggdes för att automatiskt läsa in data från olika dataformat till en gemensam sql server databas. De olika formaten var:

- Tabblade textfiler från fjärilsinventering.
- Pockes access databaser (.cdb) från BasNils, Moth, LillNils provyteinventering

Det är olika format från år till år. Det finns data från år 2003 till 2011 i skrivande stund.

## 3. Lösning

Bygga 2 olika ETL-jobb:

- Ett för Fjärilsinventeringen som importerar textfiler.
- Ett för provyteinventeringen som importerar Access-databaser.

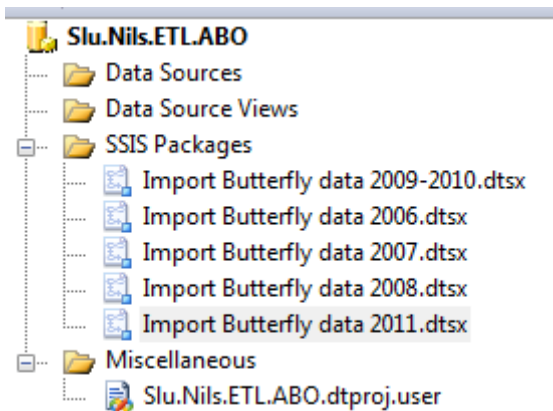
- Ett paket för varje år/struktur.

Dessa 2 jobben körs schemalagt varje natt på en server, terra.srh.slu.se, under säsong. Efter säsongen finns all data i 2 sql server databaser: Falt\_Import samt Fjaril\_Import.

Efter ett ETL-jobb är klart, körs ett sql-script som läser loggen hur det gick, och mailar detta till en lista på en sharepoint-site<sup>1</sup> som en administratör kan prenumerera på, för att se daglig status på import-jobben.

## 4. Struktur

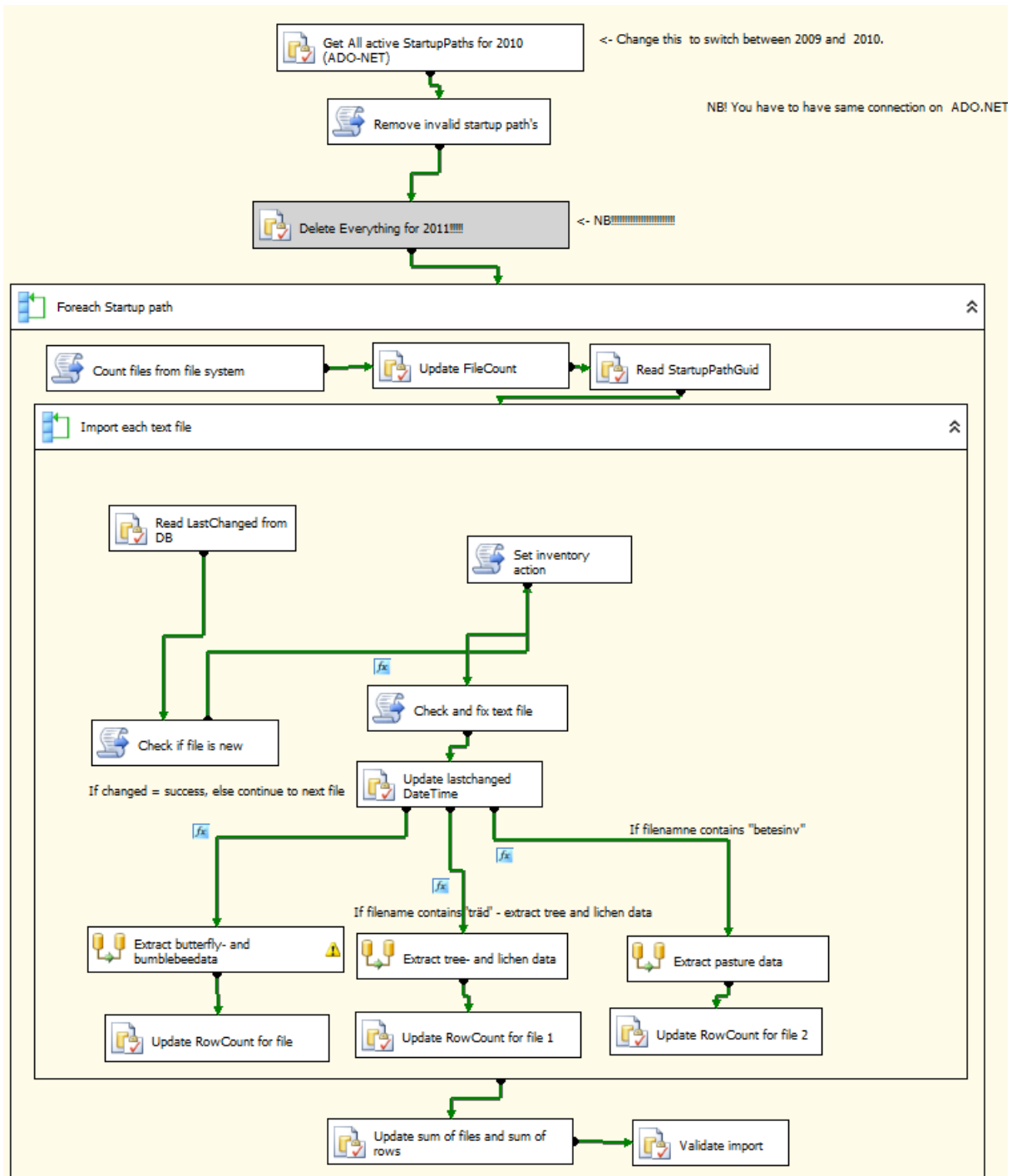
ETL-jobben är byggda i SSIS, med standardkomponenter. Från år till år kan föregående året kopieras och modifieras. Se bild nedan:



### 4.1 Komponenter

De komponenter som används i denna applikation. Nedan ses vilka SSIS komponenter som används. Den nedanstående bilden är från ETL-jobbet för Fjäril 2011:

<sup>1</sup> SU-enhetens sharepointsite:  
<https://arbetsplats.slu.se/sites/srh/Landskapsanalys/SU/Public/Lists/Sql%20server%20Importera%20data/Succeeded.aspx>



## 5. Beteende

Nedan följer diagram som inte beskriver strukturen av systemet utan snarare beteendet av systemet, hur dess funktionalitet får systemet att agera i sig själv och hur interaktionen med användaren ser ut.

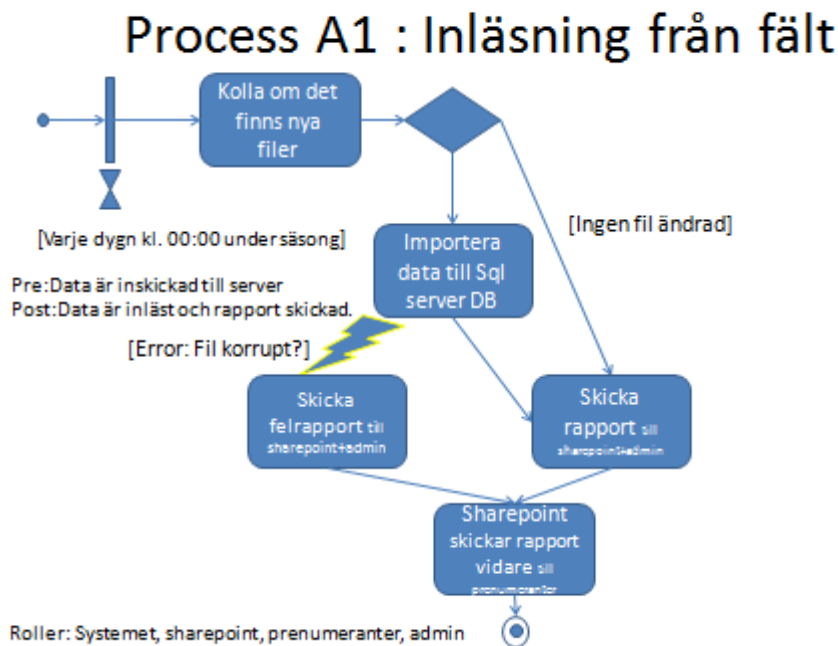
*Aktivitetsdiagram* kan med fördel likt *sekvensdiagram* modellera finare nedbrutna processer, dvs interan processer i systemet för att kommunicera beteende programmerare emellan. *Use case diagram* å andra sidan är bättre lämpad att beskriva interaktion med systemet på en mer övergripande nivå, för att finna aktörer och beskriva funktionalitet för avnämaren. Detta kan iofs även aktivitetsdiagram vilket du ser i detta dokument.

## 5.1 UCD

Use case diagram (UCD) kan användas dels för kravfångst, design och test för att beskriva funktionalitet med aktörer och deras mål/UCD och beroenden mellan olika UCD. Eftersom dessa ETL-jobb körs schemalagts sätts ingen fokus på UCD.

## 5.2 Aktivitetsdiagram

Aktivitetsdiagram (AD) används för att arbetsflödet i systemet av stegvisa aktiviteter, med stöd av olika val, iterationer och samtidighet. Diagrammet nedan beskriver hur ETL-jobben generellt fungerar:



## 6. Lagring

### 6.1 Indata – source

Det finns olika format för indata. Nedan presenteras vilka och från vilken källa.



### 6.1.1 CSV

Fjärilsinventeringen har handdatorer med tabbade textfiler (CSV).

### 6.1.2 MDB

All provyteinventering (NILS, LillNILS, MOTH) har MS Access databaser.

### 6.1.3 Shapefiler

LillNils använder handdatorer som skriver till Shapefiler. Dessa hanteras dock inte via MS SSIS, utan via ett .NET program som sköter inläsningen. Se systemdokumentation "LillNils – SBK".

## 6.2 Utdata – target

Det finns 5 databaser vi föder med data på [server] från ETL-jobben. Alla är av typ MS Sql server 2008 STD R2, och instansen heter [instans]. För att läsa mer om dessa databaser, läs wikipedia. Nedan beskrivs i korthet dessa databaser.

### 6.2.1 Importdatabaser

Alla importdata hamnar i följande 4 databaser:

- Falt\_Import
  - Här läser ETL-jobb in data för provyteinventeringarna under säsong. Det finns endast årets data i denna.
- Falt\_Rattning
  - Om man har eftersläntare, dvs access-databaser som kommit efter man redan importerat årets data till Falt\_Rattning, ställer man om ETL-jobbet att läsa in endast eftersläntarna till denna.
- Fjaril\_Import
  - Här läser ETL-jobb in data för provyteinventeringarna under säsong. Det finns endast årets data i denna.
- Fjaril\_Rattning
  - Om man har eftersläntare, dvs filer som kommit efter man redan importerat årets data till Fjaril\_Rattning, ställer man om ETL-jobbet att läsa in endast eftersläntarna till denna.

## 6.2.2 Nilsbas

Nilsbas är en normaliserad databas där all data ska importeras. Denna databas är under utveckling och har därför flera versioner av sig på [instans]. Databaserna har namnet "Nils" följt av en versionskod.

### 6.2.2.1 Namn och versionsnummer

Ex: "Nils\_i20.0.0", där i20 betyder iteration 20 – dvs i vilken utvecklings-iteration databasen är publiceras. Den första efterföljande nollan räknas upp om ny funktionalitet tillkommit under iterationen, och den senare nollan räknas upp om buggfixar på databasmodellen gjorts under iterationen.

## 6.3 Konfigurering

### 6.3.1 Databas

På [instans] finns en tabell som heter Databaskonfiguration. I denna konfigureras ETL-jobb hörande Nilsbas-utveckling, dvs ETL-jobbet:

C:\\$Subversion\Databasprojekt\Slu.Nils.ETL.Nilsbas

### 6.3.2 Enironmentvariabler

Det finns även jobb som inte använder databaskonfiguration utan istället miljövariabler. Dessa jobb är de hörande import från fält under säsong:

C:\\$Subversion\Databasprojekt\Slu.Nils.ETL.Faltdata

- **NILS\_FALTIMPORT\_SETTINGS\_FILE**

C:\\$Subversion\Databasprojekt\Slu.Nils.ETL.ABO

- **NILS\_FJARILIMPORT\_SETTINGS\_FILE**

## 7. Konfigurering

Följande avsnitt beskriver den konfigurering som kan göras via dess "config-fil", dvs sättas utan att programmet behöver kompileras om.

### 7.1 Fjäril

Connectionstring för databasen är definierade på 2 ställen:

1. Environmentvariabel

- a. ETL-jobbet försöker först läsa från denna. Finn den inte går den på den i ETL-jobbets connections.

## 2. I Connections i ETL jobbet

### 7.1.1 Environment variable

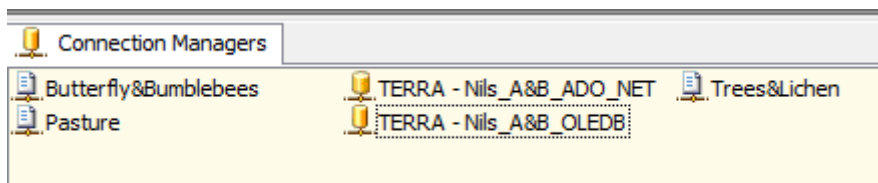
Connectionstring till target Sql server databasen, kan med fördel definieras i environmentvariabeln "NILS\_FJARILIMPORT\_SETTINGS\_FILE".

Då kan man ställa om ET-jobbet på [server] , dvs köra mot olika databaser direkt utan att behöva Visual Studio.

### 7.1.2 Datasources i ETL-jobben.

ETL-jobbet heter Slu.Nils.ETL.Abo och finns i Subversion under mappen 'C:\\$Subversion\Databasprojekt\Slu.Nils.ETL.ABO'

För fjäril finns det en del datakällor att ställa in.



De 3 fil-connections i bilen används för att koppla mot årets struktur, för att kunna sätta rätt datatyper enklare. Själva kopplingen mot filerna sätts om i en loop i ETL-jobbet.

De 2 databaskopplingarna är speglade. OLEDB går snabbare, och ADO\_NET är smidigare då det ger tillbaka ett Dataset. Dessa måste ställas in mot den server och databas man vill skriva data till. I nuläget vill man skriva till databasinstansen 'servern terra.srh.slu.se\nils' och databasen "Fjaril\_Import".

Förutom inställningar i ETL-jobbet, finns inställningar i måldatabasen i tabellen 'dbo.ImportSettings'. Där ställer man in:

- Startsökväg
- År
- Aktivt jobb eller inte

## 7.2 Provyteinventering: Nils, Moth, LillNils

Connectionstring för databasen är definierade på 2 ställen:

### 3. Environmentvariabel

- a. ETL-jobbet försöker först läsa från denna. Finn den inte går den på den i ETL-jobbets connections.

#### 4. I Connections i ETL jobbet

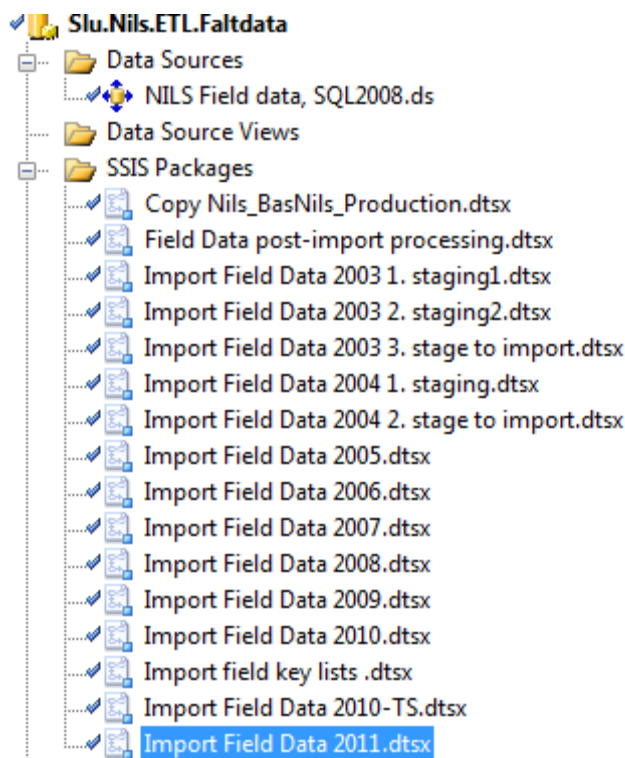
### 7.2.1 Environment variable

Connectionstring till target Sql server databasen, kan med fördel definieras i environmentvariabeln ”NILS\_FALTIMPORT\_SETTINGS\_FILE”.

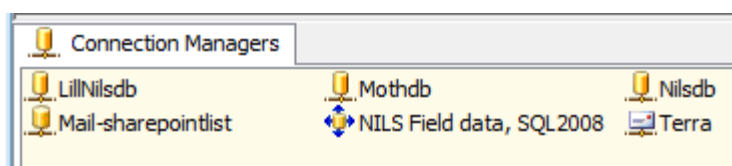
Då kan man ställa om ET-jobbet på [server] , dvs köra mot olika databaser direkt utan att behöva Visual Studio.

ETL-projekt för provyteinventering finns i Subversion under:  
C:\\$Subversion\Databasprojekt\Slu.Nils.ETL.Faltdata

De jobb som finns idag ses i bilden nedan:



Det finns även en gemensam datasource, som används som target. Denna ställs in mot den databas man vill skriva data till. I nuläget terra.srh.slu.se\nils och databasen Falt\_Import.



Förutom target datasourDet finns 3 databasuppkopplingar som används som källa, vilka är accessdatabaser: LillNilsdb, Mothdb, Nilsdb.

## 8. Deployment

### 8.1 Systemkrav

Här nedan ses en tabell över vilka operativsystem som stöds, och vad som krävs för att programmet enligt kan köras enligt användarfallet beskrivet i avsnitt 5. Rekommenderat OS är markerat fetstilt.

OS	X64	X86	Min RAM (GB)
<b>Win 7</b>	-	X	2
Win Vista	-	?	?
Win XP	-	X	2
<b>Win server 2008</b>	-	X	2

#### 8.1.1 Förinstallerad programvara

##### 8.1.1.1 Obligatoriskt

Det krävs även att följande saker finns

- Sql server 2008 R2, med Sql server integration services. (SSIS)
- **Microsoft.Office.Interop** för att kunna läsa Access databaserna.

##### 8.1.1.2 Valfritt men rekommenderat

Följande programvara bör finnas för att få ut 100% av funktionaliteten:

- Åtkomst till en server med SMTP-tjänst, för att kunna maila rapport.

- Sharepoint-site, där rapporten mailas.

## 8.2 Kompilering

Här beskrivs hur applikationen måste kompileras, x86 eller x64, vilket .NET framework, om "unmanaged code" används eller inte, debug- VS Release kompilering.

- Den måste kompileras med "x86" i Visual studio, eftersom Oledb från Office komponenter endast finns i 32-bitars OS.

## 8.3 Installation

I detta avsnitt beskrivs hur applikationen behöver paketeras för att användaren ska kunna installera den.

1. Dessa ETL-jobb publiceras från Visual studio till Sql server instansen 'Nils' på servern terra.srh.slu.se.
  - a. Öppna SSMS och öppna Sql server Agent, för att se de 2 ETL-jobben.
  - b. Användare att köra jobben är: NilsAgent med lösenord: SluNils2010.
2. Om man ändrar namnet på ETL-jobben, annat än de heter på servern, måste man även konfigurera de sql-script som ska rapportera resultat till sharepoint. Dessa finner du genom att titta på de 2 schemalagda jobben på terra.srh.slu.se.

## 9. FAQ

I detta avsnitt tas vanliga frågor och svar upp. Dessa frågor härstammar från användare av applikationen och kan beskriva både möjligheter och begränsningar som applikationen har.

### 9.1 Q: Hur gör jag om strukturen ändrats från förra året?

Kopiera ETL-paketet i SSIS från förra året, ändra efter den nya strukturen.

### 9.2 Q: Det blir dubletter av rader pga av i, hur hanteras detta?

I slutet på säsongen kör man en funktion i Datatest, "Dublettrensning", som sparar den senast raden och tar bort övriga dubletter.

## **Rättning – systemdokumentation**

<b>1.</b>	<b>Inledning</b> .....	<b>4</b>
<b>2.</b>	<b>Bakgrund</b> .....	<b>4</b>
<b>3.</b>	<b>Lösning</b> .....	<b>4</b>
<b>4.</b>	<b>Strukturell modell</b> .....	<b>5</b>
4.1	Komponenter.....	5
4.2	Klassdiagram.....	5
4.2.1	UIValidation .....	6
4.2.2	DeleteProcedure .....	6
4.2.3	AddProcedure .....	6
4.2.4	Field2010Procedures.....	7
4.2.5	IStoredProcGenerator .....	7
<b>5.</b>	<b>Beteende</b> .....	<b>7</b>
5.1	UCD .....	7
5.2	Aktivitetsdiagram.....	9
<b>6.</b>	<b>Lagring</b> .....	<b>10</b>
<b>7.</b>	<b>Konfigurering</b> .....	<b>10</b>
7.1	FieldConnectionString .....	10
7.2	ButterflyConnectionString.....	10
7.3	UserTable, UserYear.....	10
7.4	LogPath .....	10
7.5	LogUserName, LogPassword .....	10
<b>8.</b>	<b>Deployment</b> .....	<b>11</b>
8.1	Systemkrav.....	11
8.1.1	Förinstallerad programvara.....	11



8.2	Kompilering .....	11
8.3	Installation.....	12
<b>9.</b>	<b>FAQ.....</b>	<b>12</b>
9.1	Q: Kan man ändra på data via applikationen? .....	12
9.2	Q: Kan man bara köra mot Falt_Rattning och Falt_Import? .....	12

## 1. Inledning

Detta dokument riktar sig till systemutvecklare.

Det syftar att beskriva applikationen i fråga – hur det är uppbyggt strukturellt, hur det betar sig, hur man underhåller det.

Detta dokument tar inte upp hur man installerar och använder programmet, eftersom det finns ett speciellt dokument som behandlar detta

## 2. Bakgrund

Under säsong finner man fel i data via Datatest och skriver felrapporter om dessa fel ”blå lappar”. Efter säsongen rättar man dessa fel. För att detta ska göras på ett kontrollerat och användarvänligt sätt vill man inte att rättaren ska sitta direkt mot Sql server databasen och skriva TSQL. Dels kanske rättaren inte kan script-språket TSQL, dels inte har Sql server management studio installerat, och dels inte kan tabell-relationerna till fullo.

## 3. Lösning

Den första lösningen blev att göra användarvänliga vyer och använda MS Access länkad mot databasen för ett användarvänligare utseende. Men detta fungerar endast om man ska ändra en rad i en tabell. Man kan inte lägga till eller ta bort rader via vyer.<sup>1</sup>

Därför byggdes ett **rättningsprogram** i .NET som instället tar hand om att generera TSQL och stored procedures<sup>2</sup>, har kunskap om tabellrelationerna, ett mer kontrollerat och användarvänligt sätt att arbeta på. Alla ändringar loggas av programmet vilket ger spårbarhet vill vem, när och vad användaren ändrat i databasen.

Rättningsprogrammets främsta uppgift är att göra det möjligt för användaren att: skapa rader och ta bort rader dock inte ändra på rader, eftersom detta går att göra via MS Access och vyer.

---

<sup>1</sup> Det går inte lägga till eller ta bort rader för Vyer som är baserade på flera olika tabeller.

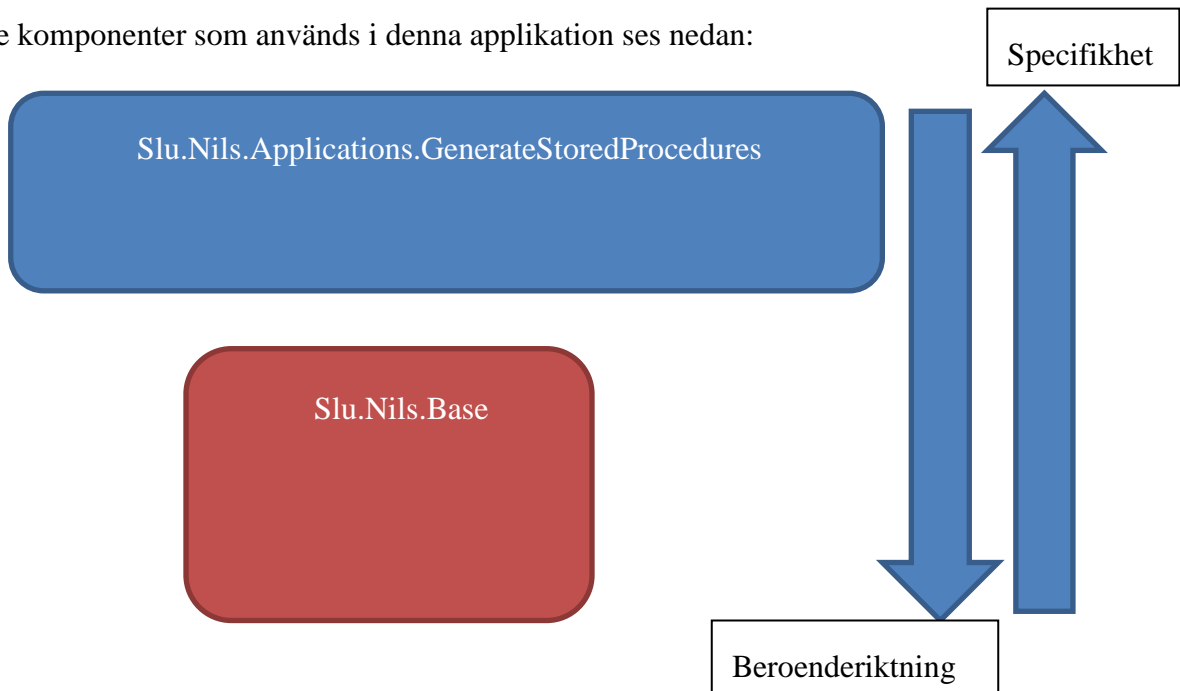
<sup>2</sup> Att skapa dessa stored procedures för hand skulle ta 20 min per procedur: ca 200 procedurer, vilket ger 8-9 arbetsdagar varje år. Detta program tog 3 dagar att bygga.

## 4. Strukturell modell

Systemet har en 2-skiktad N-tier arkitektur<sup>3</sup>, och nedan beskrivs strukturen på de olika skikten för applikationen, både på en högre nivå där komponenterna i denna applikation beskrivs, men även mer detaljerat på klass-nivå.

### 4.1 Komponenter

De komponenter som används i denna applikation ses nedan:

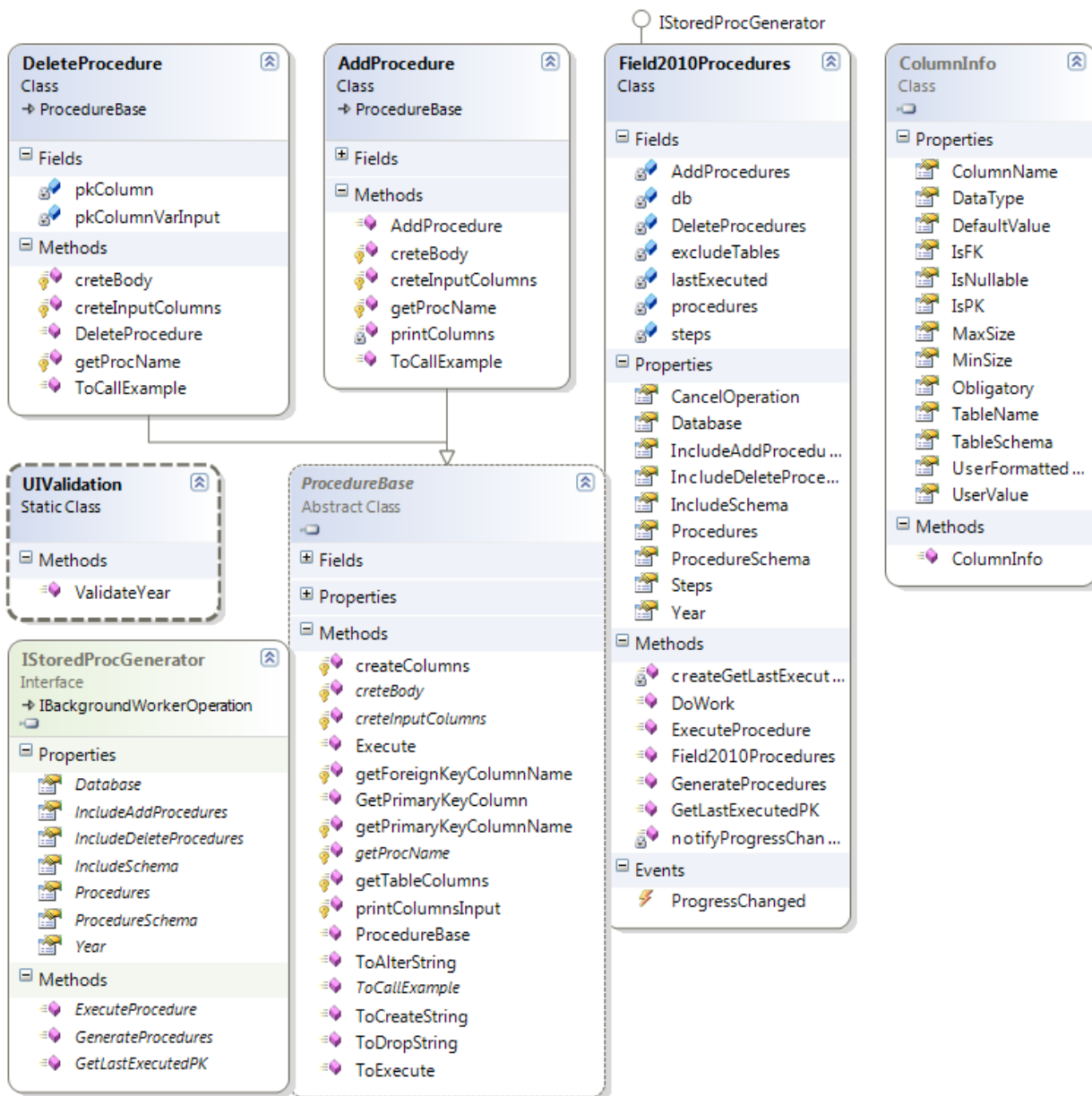


- Applikationslagret: Här finns specifik kod för databasstrukturen.
- Baslagret: I Baslagret finns interface och en abstrakt basklass att generera och exekvera stored procedures. IStoredProcGenerator, SqlDb, ProcedureBase

### 4.2 Klassdiagram

Att anmärka är att klassdiagrammen endast innehåller en delmängd av alla klasser. Detta för att visa på grundstenarna då inte alla klasser får plats i ett diagram. För att se alla klasser, finns dessa klassdiagram i Visual Studio projekten, och då även i Subversion. I detta avsnitt läggs fokus på beskrivning av några av de mer betydande klasserna och interface.

<sup>3</sup> Läs mer om arkitekturen i "LA – Arkitekturbeskrivning".



#### 4.2.1 UIValidation

Validaderar input av år, att det är angivet som en fyrsiffrig integer större än 2000.

#### 4.2.2 DeleteProcedure

Genererar TSQL för en delete-procedure för en viss tabell.

Subklass av ProcedureBase.

#### 4.2.3 AddProcedure

Genererar TSQL för en add-procedure för en viss tabell.

Subklass av ProcedureBase.

#### 4.2.4 Field2010Procedures

Genererar TSQL för en lägga till och ta bort rader i tabeller för databasstrukturen år 2010 för databasen Falt\_Rattning och Falt\_Import .

Denna klass har visat sig även fungera för 2011 års struktur. Om strukturen ändras mycket från år till år, gör man en ny klass som exempelvis kan ärva från föregående år.

#### 4.2.5 IStoredProcGenerator

Interface att generera och exekvera stored procedures.

## 5. Beteende

Nedan följer diagram som inte beskriver strukturen av systemet utan snarare beteendet av systemet, hur dess funktionalitet får systemet att agera i sig själv och hur interaktionen med användaren ser ut.

*Aktivitetsdiagram* kan med fördel likt *sekvensdiagram* modellera finare nedbrutna processer, dvs interan processer i systemet för att kommunicera beteende programmerare emellan. *Use case diagram* å andra sidan är bättre lämpad att beskriva interaktion med systemet på en mer övergripande nivå, för att finna aktörer och beskriva funktionalitet för avnämaren. Detta kan iofs även aktivitetsdiagram vilket du ser i detta dokument.

### 5.1 UCD

Use case diagram (UCD) kan användas dels för kravfångst, design och test för att beskriva funktionalitet med aktörer och deras mål/UCD och beroenden mellan olika UCD.

**Namn:** Generera stored procedures.

**Ref:** GenereraSP

**Författare:** jas

**Syfte:** Generera nya stored procedures för år 2012, samt skapa upp dessa i Sql server databasen Falt\_Rattning på servern Terra och även generera script att plocka bort dessa som sparas till nästa år, om nya ska läggas in.

**Aktörer:** Administratör, Terra, Sql server Manangement Studio

**System:** Rättning

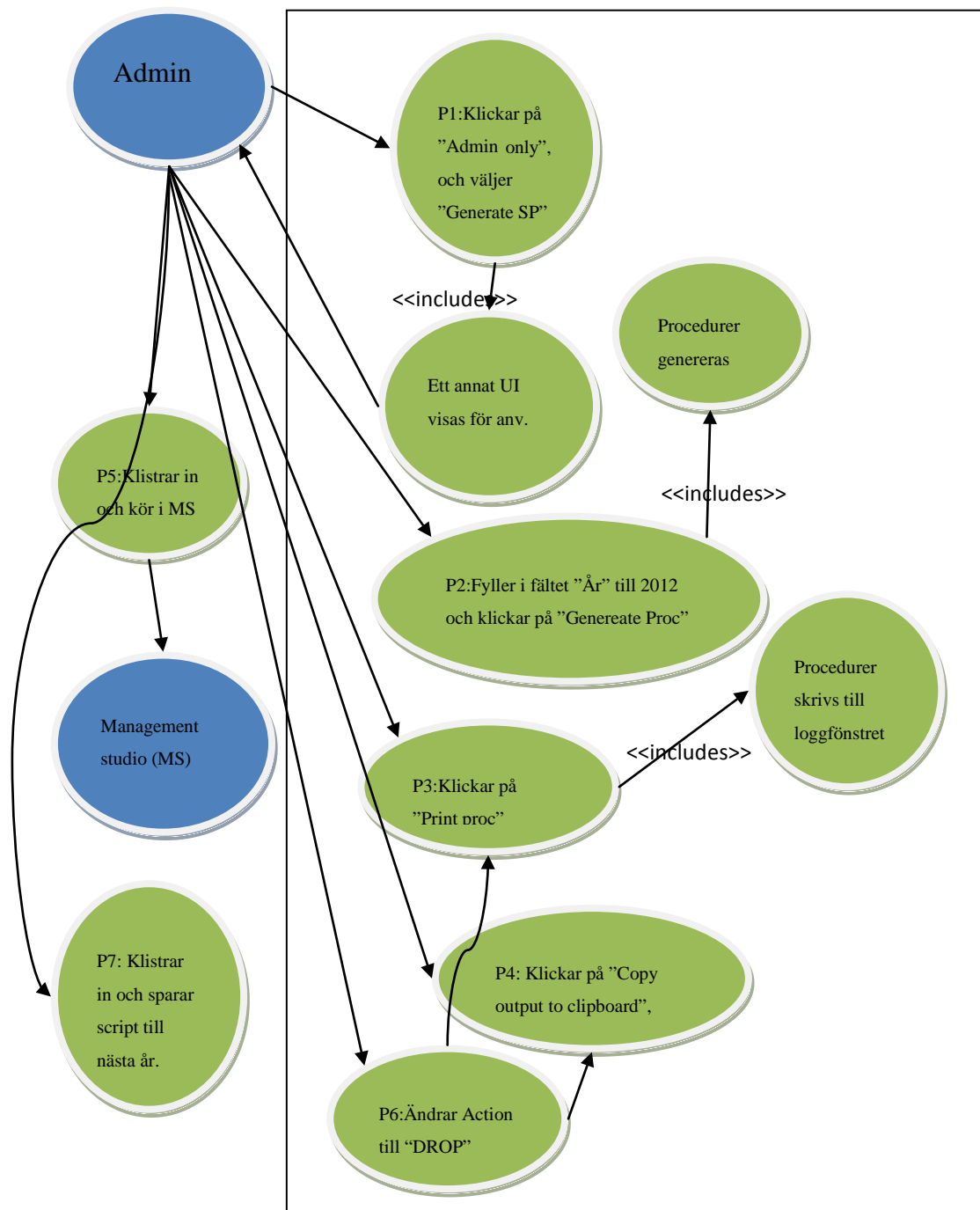
**Triggers:** Administratören vill generera nya stored procedures

**Pre-condition:**

Rättning är installerad och startad. Föregående års sql-script att ”ta bort befintliga stored” procedures är sparad

**Post-condition:**

Nya fräscha stored procedures skapade via reflektion av databasen finns i Sql servern Falt\_Rattning . Script för hur man plockar bort dem är genererade och sparade.



## 5.2 Aktivitetsdiagram

Aktivitetsdiagram (AD) används för att arbetsflödet i systemet av stegvisa aktiviteter, med stöd av olika val, iterationer och samtidighet.

## 6. Lagring

Ingen data behöver lagras.

## 7. Konfigurering

Följande avsnitt beskriver den konfigurering som kan göras via dess app.config fil, dvs sätts utan att programmet behöver konfigureras om.

Detta är viktigt att veta när programmet ska sjösättas, flyttas etc. Då måste vissa parametrar ändras. Alla parametrar med typ USER kan användaren sätta om, de andra kan endast systemet ändra på. I detta program är följande parametrar viktiga, och satta nu.

Name	Type	Scope	Value
FieldConnectionString	(Connectio...	Application	Data Source="terra.srh.slu.se\nils";Initial Catalog=falt_rattning;Integrated Security=True
ButterflyConnectionString	(Connectio...	Application	Data Source="terra.srh.slu.se\nils";Initial Catalog=fjaril_import;Integrated Security=True
UserTable	string	User	
UserYear	string	User	2011
LogPath	string	Application	\\terra.srh.slu.se\2_FALT\Fältdata\1.Rättning\Log
LogUserName	string	Application	NilsAgent
LogUserPassWord	string	User	SluNils2010
*			

### 7.1 FieldConnectionString

Databasuppkoppling mot databasen som läses av och körs mot.

### 7.2 ButterflyConnectionString

Används ej. Det var tänkt att även Fjäril skulle använda samma program, dock gjordes dessa stored procedures för hand eftersom det endast var 8 st.

### 7.3 UserTable, UserYear

När programmet stängs sparas tabell och år, så användaren nästa gång inte behöver fylla i dessa uppgifter.

### 7.4 LogPath

Sökväg till var loggfilen ska sparas.

### 7.5 LogUserName, LogPassword

AD-Användaruppgifter för konto som har rättigheter att skriva loggfilen.



## 8. Deployment

### 8.1 Systemkrav

Här nedan ses en tabell över vilka operativsystem som stöds, och vad som krävs för att programmet enligt kan köras enligt användarfallet beskrivet i avsnitt 5. Rekommenderat OS är markerat fetstilt.

OS	X64	X86	Min RAM (GB)
<b>Win 7</b>	X	X	2
Win Vista	?	?	?
Win XP	-	X	2
Win server 2008	-	-	-

#### 8.1.1 Förinstallerad programvara

##### 8.1.1.1 Obligatoriskt

Det krävs även att följande saker finns

- .NET framework 4 Client profile

##### 8.1.1.2 Valfritt men rekommenderat

Följande programvara bör finnas för att få ut 100% av funktionaliteten:

- Åtkomst till en server där loggfil kan skrivas till.

## 8.2 Kompilering

Här beskrivs hur applikationen måste kompileras, x86 eller x64, vilket .NET framework, om "unmanaged code" används eller inte, debug- VS Release kompilering.

- Den kan kompileras med "Any CPU" i Visual studio, dvs köras på x64 om det finns.

### 8.3 Installation

I detta avsnitt beskrivs hur applikationen behöver paketeras för att användaren ska kunna installera den. Det är en Click-once applikation. Deployas till valfri server, och länken användarna använder är:

<https://arbetsplats.slu.se/sites/srh/Landskapsanalys/SU/Public/Lists/Installationslnkar/Applikationer.aspx>

## 9. FAQ

I detta avsnitt tas vanliga frågor och svar upp. Dessa frågor härstammar från användare av applikationen och kan beskriva både möjligheter och begränsningar som applikationen har.

### 9.1 Q: Kan man ändra på data via applikationen?

Nej. Eftersom detta går att göra via MS Access och vyer, byggdes ej denna funktionalitet. Dock skulle jag rekommendera att bygga detta, så användaren endast behöver ett program, samt att bättre valideringar (affärslogik) kan läggas in att ge varningar om värden ändras till orimligheter.

### 9.2 Q: Kan man bara köra mot Falt\_Rattning och Falt\_Import?

Ja. I dagsläget är det så. För att göra stöd för fler databasstrukturer, behöver du i stort sett bara göra 2 nya klasser likt: AddProcedure och Field2010Procedures, samt bygga UI för detta.



Sveriges lantbruksuniversitet  
Swedish University of Agricultural Sciences

# Kravspecifikation databas

---

## Innehållsförteckning

<b><i>Innehållsförteckning</i></b> .....	<b>2</b>
<b>1 Syfte</b> .....	<b>4</b>
<b>2 Mål</b> .....	<b>4</b>
<b>3 Databasstruktur</b> .....	<b>4</b>
<b>4 Vad ställer skattningarna för krav på databasen?</b> .....	<b>5</b>
<b>4.1 Övergripande krav på databasen</b> .....	<b>5</b>
<b>4.2 Stöddata</b> .....	<b>6</b>
<b>4.3 Beräknade data</b> .....	<b>7</b>
<b>4.4 Rasterdata</b> .....	<b>7</b>
<b>4.5 Dyntaxa</b> .....	<b>7</b>
<b>4.6 Metadata &amp; Standards</b> .....	<b>8</b>
4.6.1 Datasetnivå .....	8
4.6.2 Variabelnivå.....	10
4.6.3 Andra nivåer .....	10
<b>4.7 Referenssystem för geodata</b> .....	<b>11</b>
<b>4.8 Kombinationsskattningar</b> .....	<b>11</b>
<b>4.9 Kopplingen mellan tolkade data och fältdata – tvåfasskattningar och poststratifieringar</b> <b>11</b>	
<b>4.10 Övergångar</b> .....	<b>13</b>
<b>4.11 Tidsintervall</b> .....	<b>14</b>
<b>4.12 Geografiskt urval vid utplock</b> .....	<b>15</b>
<b>4.13 Utseendet på utdata från databasen dvs. vyerna:</b> .....	<b>15</b>
4.13.1 Övergripande.....	15
4.13.2 NILS grundvyer för rutan, linjer, provytor och småprovytor .....	16

4.13.3 Vyer för artdata, markanvändning, marktäckte, markbeskrivning, åtgärder/påverkan och detaljerade träddata 17

<b>5</b>	<b><i>Arkivering och backups</i></b> .....	<b>18</b>
<b>5.1</b>	<b>Vad skall sparas i databasen?</b> .....	<b>18</b>
<b>5.2</b>	<b>Formell arkivering</b> .....	<b>18</b>
<b>5.3</b>	<b>Backuper, versionshantering, rättningar/korrigeringar i databasen</b> .....	<b>18</b>

## 1 Syfte

Att skapa en normaliserad databas som underlättar utplock och korrigering av data.

## 2 Mål

Specifikationen för databasen innefattar att bygga databasen samt att läsa in data.

## 3 Databasstruktur

Vi vet att databasstrukturen ändras från år till år, då befintliga inventeringsprojekt ändrar variabler eller ett nytt inventeringsprojekt läggs till. Vi håller läsbarhet av databasen högt prioriterat, analytiker ska kunna skapa vyer relativt enkelt. Vi kommer alltid ha systemutvecklarresurser tillhanda för att underhålla systemet.

Det är viktigt att databasstrukturen klarar av att hantera exempelvis byte av handdator och därmed struktur på det data som kommer in från fält. Då allt data läses in i SQL och genom programmering översätts till att kunna plockas in i databasen så har ett byte av handdatorprogram ingen betydelse för databasstrukturen.

Våren 2009 besökte vi andra datavårdare för att undersöka olika alternativ för att hantera detta. Med detta och systemutvecklarnas tidigare erfarenheter fann vi två alternativ på design som skulle kunna passa:

1. En metadatastyrd objektorienterad design.
2. En traditionell relationsdatabas.

Fördelen med alternativ 1 är att det blir en liten ansträngning för systemutvecklare när databasstrukturen behöver förändras. Nackdelen är att databasdesignen kräver lång eftertanke, och resultatet blir en komplex databas, svår att läsa för en icke-databasspecialist.

Fördelen med alternativ 2 är att det går fortare att designa då sättet är mer vedertaget och inte lika komplext. Detta medger större läsbarhet för en icke-databasspecialist. Nackdelen är att när strukturen måste ändras, kräver det en större insats av systemutveckling.

Vi väljer en traditionell relationsdatabas som databasstruktur. Eftersom:

- Vi vill få tillstånd en databas så fort som möjligt
- Ändringar kommer inte att ske mer än 1-2 gånger per år
- Den lägre komplexiteten hos denna databas kräver färre resurser
- Databasen är mindre personberoende

## 4 Vad ställer skattningarna för krav på databasen?

För att komma igång med leveranser behöver vi snabbt få till en funktionell databas. Om vi börjar med en bas som tillgodoser våra behov just nu så kan vi därefter utveckla databasen vidare för att klara våra framtida behov.

### 4.1 Övergripande krav på databasen

Databasen skall:

- På sikt kunna förvalta bl.a. ÄoB och LillNILS data
- Förenkla för oss att kunna göra urval på variabelnivå samt utplock årsvis och/eller sammanlästa år
- Förenkla skattningarna genom:
  - Alla data som behövs vid analys skall finnas tillgängliga, såsom stöddata, metadata, beräknade data i tabellform eller beräknade enligt funktioner som ligger i databasen
  - Funktioner såsom volymeräkningar, arealviktningar för enkla summeringar skall ligga i databasen (krav föreligger på att koden skall vara lätt att förstå (T-sql))
- Variabelnamnen i databasen skall vara lätta att förstå och följa en enkel enhetlig struktur dvs enhetliga konventioner
- Svenska variabelnamnen gäller som default i databasen. En översättningstabell, där engelska och vetenskapliga namn kan kopplas samman med de svenska vid behov, ska finnas. Liknande kan ske för att korta de svenska variabelnamnen inför analys, dvs en översättningstabell med kortare variabelnamn kopplas till utpocket.
- Variabler som bytt namn eller inventeringsmetod mellan år skall synkroniseras (namnges eller beräknas om för att stämma överens) i så stor utsträckning som möjligt så att enhetliga variabler över tiden kan plockas fram ur databasen
- När en variabel ändrar klassning finns det tre alternativ: a) den gamla klassningen konverteras till den nya klassningen b) när konverteringen inte är möjlig bibehålls den gamla klassningen men den nya klassningen utgör en ny variabel i en ny kolumn, c) när konverteringen är delvis möjlig sparas den gamla klassningen och konverteringen görs delvis i den nya kolumnen.

År	VariabelNY	VariabelGAM	Variabel
2010	1	NULL	1
2010	2	NULL	2
2010	3	NULL	4
2010	4	NULL	5
2005		1	1
2005		2	2
2005		3	3

- Data som inte används i särskilt hög utsträckning, såsom exempelvis provyta 13-16 från 2003, skall finnas med i databasen men inte i vyerna
- Ha fullständiga utläggslistor (där alla rutor i utlägget finns beskrivna, till exempel om de ligger i Norge eller helt i vatten),
- Fullständiga facitlistor ska finnas, det vill säga Som visar vilka rutor och provytor som har inventerats varje år
- Data ska presenteras i vyer och det ska vara möjligt för oss att göra egna SQL utsökningar från databasen
- Kunna byggas på med ett datalager

## 4.2 Stöddata

Med stöddata menas kompletterande data som används direkt eller indirekt inom NILS för flygbildstolkning eller fältinventering samt vid statistikproduktion och redovisning.

Vad gäller rättigheter till stöddata kan dessa delas in i data som är skyddade av upphovrättslagen och offentliga data. Stöddata från externa datakällor såsom Lantmäteriets produkter är skyddade av upphovsrättslagen vilket gör det möjligt för NILS att använda data för egna ändamål men inte sprida dem till tredje part. Denna kategori ingår därför inte i datavårdskapet. Upphovsrättskyddade kartdata som ingår i analyserna och därigenom omvandlas, det vill säga egenproducerade stöddata såsom gränser, arealuppgifter och annan information som är kopplad till NILS-rutorna (förutom koordinater), utgör offentliga data. Dessa data kan vid behov tillgängliggöras inom ramen för ett datavårdskap (Esseen et al. 2008).

- Stöddata som frekvent kommer att användas för analys skall lagras och uppdateras direkt i Nilsbas. Framförallt gäller detta de stöddata som ingår i standardanalyserna. En del stöddata kommer därför att dubbellagras, dels på [\\gis.slu.se](http://gis.slu.se) och dels i Nilsbas, där data på [\\gis.slu.se](http://gis.slu.se) utgör originalen från Lantmäteriet som uppdateras regelbundet vid varje leverans.
- Lagring och kopplingar mot stöddata som behövs för mindre vanliga analyser eller uppdrag kommer att ske utanför Nilsbas.
- Kopplingar mellan fältdata, flygbildstolkade data och stöddata kan ske genom geografiska kopplingar direkt i databasen eller genom specificerade nycklar såsom RutaID, ProvytaID etc.
- Viss extra stödinformation för till exempel. kartproduktion behöver också hanteras utanför Nilsbas.
- Det är viktigt att ange vilket **år (eller version)** av data som ligger i Nilsbas samt vilken **datakälla** som använts och **var metadata finns**. Ibland kommer vissa stöddata att förekomma flera gånger i olika versioner eftersom det används för olika syften. Vid alla analyser och datautplock kommer det att bli standard att ange datakälla och aktualitetsår (version).



### 4.3 Beräknade data

Vissa beräknade data ska finnas "färdiga" i databasen (se nedan vilka). Hur beräkningarna genomfördes utgör dock metadata och föreslås sparas separat i en wiki som är kopplad till den beräknade variabeln i framtida analysverktyg. Beräknade data som skall sparas i databasen genomgår kvalitetsgranskning och får beskrivande metadata i wikin för att sedan levereras till databasen i ett exceldokument. För de flesta förslagen nedan skall en beräkningsalgoritm lagras i databasen. Beräkningsalgoritmen utför beräkningarna vid behov. På detta sätt används hela tiden de senaste och mest korrekta data vid analyserna.

Beräkningarna görs i första hand i databasen genom funktioner (lagrade procedurer eller i vyer beroende på hur prestanda påverkas).

Beräknade variabler som kräver underlag som inte finns lagrade i databasen importeras från filer. Ett exempel är landskapsindex baserade på rasterdata (se nedan).

Exempel på beräknade data:

- Arealer för delytor
- Summor per ruta
- Bältes-arealer
- Linjekorsningspunkters koordinater
- Trädvolym
- Olika klassificeringar grundade på de insamlade variablerna exv naturtyper och markslag
- Landskapsindex
- Arealviktningstal för standardredovisningsgrupper

I databasen skall det:

- Finnas utrymme att på sikt lägga in beräkningar (se ovan) som funktioner
- Finnas möjlighet att enkelt lägga in beräknade variabler kontinuerligt

### 4.4 Rasterdata

Rasterdata kommer inte att lagras i Nilsbas. Vi kommer att ta fram olika typer av raster beroende på klassificeringssystem och rasterstorlek. Data från valt rasterskikt aggregeras först på rutnivå och därefter på stratumnivå. På stratumnivå beräknas aktuellt index utifrån aggregerat data.

### 4.5 Dyntaxa

Samtliga taxa (familjer, släkten, arter, artgrupper o s v) i Nilsbas ska ha ett taxon-id som överensstämmer med namngivningen i Dyntaxa vid ArtDatabanken. I denna databas sker en kontinuerlig uppdatering av vetenskapliga namn och den innehåller även rekommenderade engelska och svenska namn. När vi levererar "artdata" ska det inte råda något tvivel om vilka arter

som avses. Vid byggandet av databasen är det således nödvändigt att planera för en översättningsfunktion som översätter våra nuvarande artkoder till taxon-id.

Både Riksskogstaxeringen och NILS har artgrupper som inte är taxonomiskt enhetliga (t ex smalbladiga gräs), vilket komplicerar översättningen till taxon-id. I sådana fall kan man eventuellt begära att Artdatabanken skapar särskilda taxon-id för dessa grupper. Detta ligger dock utanför ramarna för denna kravspecifikation.

I databasen:

- Skall koppling mellan taxataxa och taxon-id enligt Dyntaxa vara möjlig via ArtDatabankens webbtjänst.
- Skall uppdatering av taxon-id på grund av förändringar i Dyntaxa göras en gång per år.

## 4.6 Metadata & Standards

Metadata har olika nivåer. Metadata på datasetnivå krävs bland annat av INSPIRE-direktivet och är viktiga för externa användare. Dessa metadata levereras till metadatabaser som nationella Geodataportalen (Geodata.se). Metadata på variabelnivå är viktiga vid analys för både interna och externa användare. Andra metadata berör grupper av variabler (som ingår i samma moment) eller en tabell.

### 4.6.1 Datasetnivå

Varken INSPIRE-direktivet eller ISO-standarden påverkar egentligen själva byggandet av databasen. ISO-standarden (19139) ställer högst krav angående metadata och uppfyller därigenom INSPIRE-kraven. I tabellen nedan listas tre 'SLU-profiler' angående metadatanivåer som pilotprojektet 'Metadata SLU' föreslår i sin rapport. Den högsta nivån motsvarar ISO19139.

Tabellen ger ett förslag på vilka metadataelement som kan ingå i de tre SLU-profilerna; SLU - Nationella, SLU – Inspire, SLU – mini. Siffrorna i profilkolumnerna beskriver hur många gånger ett element får eller ska förekomma. Observera att tabellen inte är en fullständig metadatataprofil som t ex Nationella profilen,

- 1 = Elementet är obligatoriskt och får förekomma max en gång  
 0..1 = Elementet är frivilligt och får förekomma max en gång  
 1.\* = Elementet är obligatoriskt och får förekomma flera gånger  
 0.\* = Elementet är frivilligt och får förekomma flera gånger

Metadataelement	SLU profil Nationella "Högsta"	SLU profil Inspire "Mellan"	SLU profil Mini "Lägsta"
Språk i metadata	1	1	1
Metadatakontakt – Organisation	0..1	0..1	
Metadatakontakt – Person	0..1	0	
Metadatakontakt – E-post	1	1	
Metadatakontakt – Ansvarsområde	1		
Datum för metadata	1	1	1
Metadatastandard	1		
Version av metadatastandard	0..1		

Bilaga 4 NILS Datafångst och Datavårdskap

Metadataelement	SLU profil Nationella "Högsta"	SLU profil Inspire "Mellan"	SLU profil Mini "Lägsta"
Hierarkisk nivå (datamängd, series, services, software)	1	1	
Identifierare för metadatamängd (UUID)	1		1
Resursens namn (Titel)	1	1	1
Alternativ titel	0..1		
Händelsedatum	1..*	1	
Händelse (skapande, publicering, revision)	1..*		
Resurskontakt – Organisation	0..1	1	1
Resurskontakt – Person (anges <b>endast</b> om inte org kan anges)	0..1	0	
Resurskontakt – E-post	1	1	1
Resurskontakt – Ansvarsområde	1	1	
Identifieringskod (UUID)	1	1	
Namnrymd (för identifieraren)	0..1	1	
Språk i resurs	1	1	
Omskrivande rektangel	1	1	
Lägsta och högsta höjd	1		
Vertikalt datum	1		
Tidsperiod	1	1	
Textbeskrivning	0..1		
Ämneskategori (tex environment)	1..*	1..*	1..*
Sammanfattning	1	1	1
Filnamn för exempelbildsfil	1		
Bildtext	0..1		
Bildformat	0..1		
Nyckelord	1..*	1..*	
Nyckelordslexikon	0..*	0..*	
Typ av tjänst	0..1	0..1	
Identitet för kopplad resurs	0..1	0..1	
Villkor för åtkomst	1	1	
Lagliga skäl	1	1	
Skyddsklass	1..*	1..*	1..*
Övriga restriktioner	0..1		
Datakvalitet – Omfattning	1		
Datakvalitet – Tillkomsthistorik – Redogörelse	1	1	
Datakvalitet - Specifikation – Beteckning	1	1	
Datakvalitet - Specifikation – Datum	1	1	
Datakvalitet - Specifikation – Händelse	1..*	1..*	
Datakvalitet – Beskrivning	1	1	
Datakvalitet – Specifikationsuppfyllelse	1	1	
Underhållsfrekvens	1		
Anmärkning	0..1		
Metod för rumslig representation	0..1		
Skala	0..1	0..1	
Avstånd mellan observationer (Upplösning)	0..1	0..1	
Identifieringskod	1		

Metadataelement	SLU profil Nationella "Högsta"	SLU profil Inspire "Mellan"	SLU profil Mini "Lägsta"
Ansvarig för namnrymd	1		
Tillhandahållande – Webbadress	0..*	0..1	
Tillhandahållande – Funktion	0..1		
Tillhandahållande – Beskrivning av webbplats	0..1		
Tillhandahållande – Organisation	0..1	0..1	
Tillhandahållande – E-post	1..*	1..*	
Tillhandahållande – Ansvarsområde	1		
Tillhandahållande – Format	0..*		

För många av dessa rubriker genereras metadata automatiskt av GIS-programvaran. För övriga rubriker krävs dock att det för varje skikt med geografiska data måste finnas en metadataansvarig som fyller i och kontinuerligt uppdaterar dessa uppgifter, förslagsvis den aktuella administratören för respektive geodatabas. De krav som ISO-standarderna ställer på oss är alltså snarare av organisations- än databaskaraktär.

All metadata ska inte bara lagras i GIS-programvaran utan i SQL server. Metadata ska vara åtkomlig via utplock från SQL server.

#### 4.6.2 Variabelnivå

Enskilda variabler vill vi beskriva med:

- Enhet
- Syfte
- Definition i instruktioner för datainsamling
- Hur de är mätta/insamlade, noggrannhet, metodik
- Dokumentation över korrigeringar
- Förklaring till bortfall på olika nivåer: till exempel. Provytan eller småprovytan kan inte besökas, eller data registreras inte på grund av flödet
- Beräkningsalgoritmer för beräknade variabler

Metadata på variabelnivå ska vara enkelt att uppdatera.

Vid ändring eller förtydligande av metodik vill man uppdatera på så få ställen som möjligt. Därför ska definitioner och andra metadata på sikt läsas från samma källa vid analys och vid framställning av instruktioner.

Metadata på variabelnivå ska vara tillgängliga i framtida analysverktyg och webbapplikationer.

#### 4.6.3 Andra nivåer

Vissa metadata berör en grupp av variabler snarare än enskilda variabler. T.ex. om flera variabler ingår i samma moment och som tillsammans uppfyller ett syfte.

## 4.7 Referenssystem för geodata

I Nilsbas hanteras endast SWEREF99-koordinater och -geometrier.

Utanför Nilsbas hanteras både RT90 och SWEREF99 systemen under en övergångsperiod.

Grundleveranser från Nilsbas kommer att ske i SWEREF99 men kan levereras i valfritt koordinatsystem vid behov.

## 4.8 Kombinationsskattningar

Kombinationsskattningar, där analysen för en variabel förbättras genom att kombinera data från flera olika stickprov (NILS, Riksskogstaxeringen, MOTH) skall vara genomförbara.

Följande krav skall uppfyllas i databasen för att göra samkörning/kombinationsskattningar möjligt:

- Det skall vara enkelt att såväl ta ut bara data från ett projekt i taget som att lägga ihop hela eller delar av stickprov från flera program.
- Data skall ha gemensamma nycklar (provyteID eller polygonID) där samlokalisering är nödvändig (inte nödvändig vid de flesta samskattningar dock)
- Beskrivande metadata skall vara kopplade till data/enskilda variabler som gör det möjligt att bedöma om variablerna mäts på samma sätt (är samma sak), alternativt gör det möjligt att omvandla variabler till samma enhet. Detta gäller i synnerhet om data från Riksskogstaxeringen ska användas, eftersom de övriga fyra programmen redan har väl integrerad metodik med samma definitioner och inventeringsrutiner.

## 4.9 Kopplingen mellan tolkade data och fältdata – tvåfasskattningar och poststratifieringar

NILS fält- och flygbildsinventeringar kan kombineras och därigenom förbättra vissa analyser och resultera i noggrannare skattningar. En tvåfasmetodik är när man utnyttjar hjälpinformation som inte är heltäckande, det vill säga som också samlats in med ett stickprov (exempel på denna samplingmetodik är MOTH samplingen).

Poststratifiering använder liksom tvåfasmetodiken två olika dataset men vid poststratifiering så finns redan två dataset och poststratifieringen bygger därför inte på en tvåfassampling som vid tvåfasmetodiken. Vid en poststratifiering används informationen från det större stickprovet i efterhand, t.ex. om urvalet av det mindre stickprovet gjorts slumpmässigt från det större. Då används informationen från det större stickprovet i själva skattningen, t.ex. genom en poststratifiering av det mindre stickprovet baserat på informationen i det större stickprovet.

Precis som för kombinationsskattningar behöver man försäkra sig om att den variabel man är intresserad av är jämförbar för de båda dataseten, i detta fall flygbildstolkningen och provyteinventeringen, då man vill korrigera för möjliga systematiska fel i flygbildstolkningen. Vill man

förbättra skattningen för en variabel mätt i fält behöver inte flyg- och fältmätningar vara helt överensstämmande, i alla fall inte i teorin, men möjligheten till förbättringen beror ju av hur väl den fältmätning man vill förbättra korrelerar med den indelning som gjorts i flygbilderna.

I databasen skall:

- Beskrivande metadata vara kopplade till data/enskilda variabler som gör det möjligt att bedöma om variablerna mäts på samma sätt (är samma sak) alternativt gör det möjligt att omvandla variabler till samma enhet.
- Variabeln för en polygon kunna kopplas till motsvarande variabel i provytor.

I princip kan man också tänka sig att göra skattningar som kombinerar linjekorsningsinventeringspunkter med flygbildstolkade linjeobjekt, men beroende på den geometriska lägesosäkerheten vid inventeringarna (fält och flyg) är det osäkert hur väl kombinationen av data faller ut.

För småprovytor är det troligen inte aktuellt att göra sådan koppling, förutom genom att aggregera informationen för alla småprovytor per provyta.

Även för ängs- och betesmarker ska tvåfassskattningar och poststratifieringar kunna göras, baserat på ÄBO-information från TUVA-databasen eller på särskild flygbildstolkning i ängs- och betesmarksobjekt, om sådan blir aktuell. Ytterligare ett tänkbart alternativ för tvåfassskattningar och poststratifieringar är att använda ett heltäckande kartunderlag i kombination med NILS mer detaljerade flygbildstolkningsdata.

Samkörningar (utplock och skattningar) mellan tolkade data och fältdata skall därför vara möjliga, exempelvis:

- Kombinera variabler för att klassa provytorna och polygonerna till en naturtyp för att beräkna arealer
- Beräkna medelvärden för trädäckning eller annan variabel som är likartad i de två dataseten.

Det finns också många andra typer av skattningar där man kan ha stor nytta av att kombinera flygbildstolkningsdata och fältdata som inte är tvåfassskattningar och poststratifieringar, exempelvis när man vill koppla detaljinformation om mängd, typ eller egenskap hos linjeobjekt efter i vilken naturtyp de ligger.

Tolkade data och fältdata skall kunna kopplas samman via:

- Geografisk läge
- Rutanummer
- DelytaID
- ÄBO\_ID
- GlobalID (polygon)
- Inventeringsår – när inventeringen faktiskt genomfördes i fält

- Fotoår – när flygfoto togs

## 4.10 Övergångar

För att kunna följa förändringar i landskapet räcker det inte att jämföra aggregerade arealförändringar. Summan av arealer kan vara densamma samtidigt som det faktiskt har skett förändringar. Därför är det viktigt att kunna undersöka vad som har övergått till vad när det gäller exempelvis marktäcketyper eller markanvändningsklasser. Detta ska kunna göras både för delytor i fälldata och polygoner i tolkade data. Vanligtvis skapas övergångsmatriser för två tillfällen i taget, som kan ligga nära (fem år) eller längre bort i tiden. I regel vill man jämföra två femårs-perioder med varandra. Den första är valfri (inte nödvändigtvis inventeringsvarvet) och den andra ska ligga i samma fas, så att tidsskillnaden blir den samma för alla delprovtytor/polygoner som jämförs. I fall då jämförbara data saknas för en hel femårs-period, på grund av metodändringar kan det vara aktuellt att jämföra t.ex. två treårs-perioder (när stickprovet tillåter detta).

I Nilsbas skall:

- Provtytor, delytor och polygoner kunna följas över tiden.
- Geometrier för delprovtytor, provtytor, ÄBO och flygbildsinventeringens polygoner lagras i databasen och sammankoppling kan göras genom geografiskt läge med SQL-kod. Övergångsmatriser från en klassning till en annan i provtytor, delytor och tolkade polygoner kommer därmed att kunna beräknas direkt från databasen via ytornas/polygonernas geometrier (fig. 1).
- Småprovtytor har inga fasta koordinater och det finns därmed inget behov av att följa enstaka småprovtytor över tiden utan vi nöjer oss med att aggregera småprovtytodata på delyte-/provtytenivå.
- Linjekorsningspunkter är inte permanent markerade i fält och koordinaterna varierar från varv till varv. Vill man kunna följa dessa borde det vara möjligt att göra en analys där man söker efter en korsningspunkt av samma typ inom ett visst avstånd (buffer) vid den andra tidpunkten. Här handlar det egentligen inte om övergångar i meningen ovan (Linjeobjekt *väg* övergår inte till linjeobjekt *dike*). Snarare handlar det om hur enskilda variabler (t.ex. mittremsans bredd) ändras över tiden.
- För att kunna göra en analys enligt exemplet i fig. 1 måste det vara möjligt att göra två utplock från samma variabel vid olika tidpunkter och ge dessa utplock tillfälliga namn.
- Data måste kunna plockas ut och exporteras i matrisform exv. excel för vidare analyser av program som behärskar matrisalgebra (exv. som indata till modelleringsprogram). Det skall även finnas möjlighet att lägga till punktgifterdata.

The screenshot shows a PostgreSQL SQL Editor window with the following SQL query:

```
SELECT a.name AS nameA,
       b.name AS nameB,
       st_Area(st_Intersection(a.geom,b.geom)) AS area
FROM "ExempelA" a JOIN "ExempelB" b
ON st_Intersects(a.geom,b.geom)
ORDER BY a.name,b.name
```

The Output pane displays the following table:

	namea character vari	nameb character vari	area double precis
1	Skog	Hygge	14728.0302734
2	Skog	Skog	32539.9423828
3	Skog	Strand	227.15625
4	Skog	Tomt	2241.36914062
5	Strand	Skog	243.670898437
6	Strand	Strand	17507.5224609
7	Strand	Tomt	3156.59570312
8	Strand	Vatten	483.071289062
9	Vatten	Strand	8028.25195312
10	Vatten	Vatten	116322.647460

The status bar at the bottom indicates: OK, Unix, Ln 6 Col 24 Ch 186, 10 rows., 63 ms

Figur 1. Exempel på SQL-kod som genererar en transformationsmatris från en marktyp till en annan vid två olika tidpunkter med hjälp av geometrier.

#### 4.11 Tidsintervall

Vid alla slags förändringsanalyser, dvs. då förändringen i en variabel mellan två på varandra följande varv är det man analyserar, skall det finnas valfrihet att redovisa med avseende på indelning vid tidpunkt 1 respektive tidpunkt 2. Detta är relevant för länsgränser (pga. sammanslagningar län eller kommuner) men i än högre grad indelning efter exempelvis typ av markslag.

Databasen skall vara upplagd så att man skall:

- Kunna välja ett inventeringsvarv i taget
- Kunna välja antingen fältår, fotoår, eller inventeringsår.
- Kunna använda både t1 och t2 samt bedömd förändring som urval (framförallt vid förändringsskattningar och övergångsmatriser) – förändring i en viss variabel mellan t1 och t2



för olika redovisningsenheter exv. strata kopplad till övergångsmatriser - ger möjligheten att följa exv. om förändring av lingon beror på att vi huggit ned alla skog.

- Kunna välja en valfri femårsperiod i taget
- Kunna välja överlappande femårsperioder, för glidande 5-årsmedelvärden
- Kunna välja enstaka år, där stickprovet räcker för att se skillnaden mellan sådana.

## 4.12 Geografiskt urval vid utplock

Urval vid utplock och analys sker framförallt genom fördefinierade enheter såsom Nils-strata, län, kommuner, biogeografiska regioner etc. För intern analys finns dock ingen begränsning där det gäller geografiska urval och på sikt är ambitionen att externa beställare skall kunna genomföra egna val för utplock/redovisning i webbportalen. Beställaren skall kunna eller ladda upp ett eget område (Shape-fil) eller ha möjligheten att själv markera (rita in) lämpligt område. Om området är litet och inkluderar lite data bör dock beställaren få en varning om detta.

## 4.13 Utseendet på utdata från databasen dvs. vyerna:

Format på utdata skall vara flexibelt för att kunna anpassas till många olika användares behov och förändringar i behovsbild över tiden. Med tiden kommer fler och andra format att efterfrågas och detta måste vi kunna svara mot.

Genom vyerna skapar vi fasta uttagstabeller men genom SQL-sökningar kommer man alltid att kunna plocka fram egna tabeller.

### 4.13.1 Övergripande

#### Vyerna skall:

- I stort sett följa aktuella fält- och tolkningsmanualerna, det vill säga variabelkolumnerna ska vara i samma ordning som i manualerna.
- Ordinarie vyer innehåller endast **typruta=1 (ordinarie inventering)**. Motsvarande vyer för kontrollinventeringar och flaggskepp görs senare
- Vyer görs **programvisa** och namnges med programnamn som schemanamn. Först görs vyerna för NILS. De andra görs senare.
- Ordinarie vyer skall endast ha med **variabler som finns insamlade kontinuerligt** över alla år och inte de som tagits bort tidigare år. Variabler som tillkommit med åren och fortlöpande inventeras presenteras i vyerna.
- **Provyta 13-16** basNILS 2003-2004 skall **inte** visas i ordinarie vyer
- **År och inventeringsår eller fotoår** skall finnas i varje vy där det finns data
- Kolumnnamn - **korta svenska namn**. Sen ska det finnas en översättningstabell, så att namnen lätt kan bytas mot långa beskrivande namn, engelska eller latinska namn.

- Klasserna skall vara i text och i kod, exempelvis för huvudtyp skall det finnas både kod 1 och texten som tillhör den koden dvs. terrester/semiakvatisk mark och dessa båda skall ligga i olika kolumner.

### **Exempel på vyer som vi behöver:**

För de olika programmen krävs egna utläggsvyer (vyer som visar samtliga ytor eller linjer som inventerats) som därefter kopplas samman med data från programmen och för NILS exv. som har 3 olika inventeringar (flyg, linjer och provytor) behöver man även anpassa utläggsvyerna (se nedan) efter inventeringstypen.

#### *4.13.2 NILS grundvyer för rutan, linjer, provytor och småprovytor*

För de olika analyserna som vi genomför behöver vi ett antal grundvyer som innehåller nycklar för att underlätta kopplingen mellan olika vyer (guids), redovisningsenheter, viktningsfaktorer samt om rutan, linjerna, provytorna och småprovytorna är inventerade eller inte.

Grundvy för rutan: I denna vy läggs all information kring vilken redovisningsenhet (exv. stratum eller län) som rutan tillhör samt arealen vatten och land på rutan och de olika viktningsfaktorerna som krävs för att skattningarna skall kunna genomföras. Detta är övergripande information som gäller oberoende av vilket varv som inventeringen utförs under.

Grundvy för linjer: I denna vy läggs vilken ruta linjen tillhör, hur mycket av linjen som ligger på land eller i vatten (framtaget från kartmaterial), linjens start och slut koordinat samt linjens geometri. Detta är övergripande information som gäller oberoende av vilket varv som inventeringen utförs under.

Grundvy för linjesegmenten: I denna vy finns den fältinventerade linjelängden och den kartinventerade längden och för den kartinventerade längden finns även orsaken till varför den var kartinventerad, det vill säga om den delen av linjen låg i vatten eller i ett område med beträdnadsförbud. I denna vy är linje och ruta kopplat till år då nya data samlas in för linjen under nästa varv.

Grundvy för provytorna: I denna vy läggs vilken ruta provytan tillhör, hur mycket av provytan som ligger på land eller i vatten (framtaget från kartmaterial), centrumkoordinat för provytan samt provytans geometri. Detta är övergripande information som gäller oberoende av vilket varv som inventeringen utförs under.

Grundvy för delytorna: I denna vy finns den inventerade delytearean (grundar sig på delningen som genomförts av fältlagen i fält) och om delytan ligger på land, vatten, glaciär eller snötäckt mark samt om den är fältinventerad eller inte och orsaken till om den inte är inventerad. I denna vy finns även

kopplingen till vilken polygon i tolkade data som delytan tillhör. I denna vy är delytan, provytan och rutan kopplade till år då nya data samlas in för delytan under nästa varv.

Grundvy för småprovytor: I denna vy finns uppgifter om småprovytan är inventerad eller inte och orsaken till om den inte är inventerad. I denna vy är småprovytan, delytan, provytan och rutan kopplade till år då nya data samlas in för delytan under nästa varv. I grundvyn på rutanivå kommer de redovisningsenheter som det finns behov av att läggas till eftersom. Redovisningsenheter av intresse är: Regioner, län, naturgeografiska och växtbiogeografiska regioner, ovan och under HK, avrinningsområde samt administrativ tillhörighet såsom om rutan ingår i Häckfågelstaxeringens standardrutter ochskyddade respektive oskyddade områden.

Grundvy för flygbildstolkade data: I denna vy finns kopplingen mellan vilken ruta som den tolkade polygonen ligger i samt vilket fotoår som flygbilden som data insamlats från har men även inventeringsår.

Viktningsfaktorer: För att kunna genomföra skattningarna krävs viktningsfaktorer. Viktningsfaktorerna utgörs av antal rutor som inventeras per år samt totalt utlägg per inventering vilket utgörs av totalt antal 5 x 5 km rutor som finns i de olika stratummen samt land och vattenareal för stratummet.

#### *4.13.3 Vyer för artdata, markanvändning, marktäcke, markbeskrivning, åtgärder/påverkan och detaljerade träddata*

Grundvyerna läggs samman med de olika tabellerna för tolkade data, linjer i fält respektive provytor i fält till totalvyer för exempelvis marktäcke, småprovytor, dike/vattendrag eller tolkade data.

Principen är att en rad vyer som utgör byggstenar byggs upp och dessa vyer benämns *art.v\_exv delyta*. Dessa vyer sammanlänkas därefter med hjälp av en nyckelvy som benämns *nils.v\_exv delyta* till större analysvyer som benämns *nils.v\_exv delyta\_exv buskar*.

Artdata på delytanivå skall i första hand finnas i ej pivoterat format det vill säga alla data ligger i en och samma kolumn.

I ett senare skede kan det vara aktuellt att låta varje art stå i en egen kolumn och för att underlätta överblicken dela upp data så att det blir en vy per inventeringsgrupp. Arter som inte är registrerade skall vara 0. För täckningsgrader skall både förekomst och täckningsgradskolumner finnas med så att de som är en verklig nolla kan skiljas ut från de som betyder mindre än 0.5%. Från 2008 kan förekomst i kolumn läggas till.

För marktäckedata skapas enskilda analysvyer för träd, bottenskikt, buskar, fältskikt och stora arter.

För småprovytedata skapas enskilda analysvyer för mossor, lavar, örter, fältskikt, ormbunksväxter, ris, graminider, bottenskikt och renlavar.

Övergripande klassningsvariabler, såsom skog med produktiv skogsmark, skogFAO, trädbuskFAO och fjälltyp, utgör en egen vy och likaså gör myr med fastmatta, mjukmatta, lösbotten.

Markbeskrivningsdata, markanvändningsdata, åtgärder/påverkan och detaljerade träddata utgör var och en egna analysvyer.

För linjedata skapas enskilda analysvyer för de olika linjeobjekten: transportled, hägnader, vegetationsremсор, dike/vattendrag, skogskanter och stränder.

För de tolkade ytorna skapas enskilda analysvyer för marktäckte och markbeskrivning, trädskikt, busk/fält/bottenskikt, klasser av myr, markanvändning, åtgärder och påverkan samt en övrigt vy. Utöver dessa vyer skapas även en totalanalysvy som innehåller alla ovanstående delar i en och samma vy.

De tolkade linjerna och punkterna kommer att presenteras i enskilda analysvyer där kopplingen mellan i vilken ruta linjen eller punkten finns beskrivs och de olika beskrivande attributen finns redovisade därefter.

## 5 Arkivering och backups

### 5.1 Vad skall sparas i databasen?

Databasen skall utöver basNILS data spara beräkningar och funktioner som är nödvändiga för fortsatt analys i analysverktyg och webbportal.

Under punkterna stöddata, metadata och beräknade data beskrivs vad som skall finnas i databasen.

### 5.2 Formell arkivering

Vi följer [arkivlagen](#) för myndigheter, SLU:s arkivlagar och det som föreskrivs oss från den gemensamma dataplattformen.

### 5.3 Backupер, versionshantering, rättningar/korrigeringar i databasen

Databasen ligger på en server som administreras av IT-enheten och backupер görs enligt deras rutiner.

Någon form av versionshantering i Nilsbas kommer sannolikt att efterfrågas av exempelvis SLUs dataplattformar. Ett förslag är att en ny version skapas varje år efter rättning och övrig kvalitetssäkring.

Vårt eget behov av versionshantering i Nilsbas är litet. Det är däremot mycket viktigt att versionshantering finns i "rättningsdatabaserna".



Sveriges lantbruksuniversitet  
Swedish University of Agricultural Sciences

# Nilsbas

## Databasdesign – designprinciper

## Innehåll

1. Bakgrund & Syfte .....	4
2. Utvecklingsgång.....	4
Grundförutsättningar och datarelationer .....	4
Modellering .....	4
Från modell till fysisk Databas.....	4
Dokumentation .....	4
Exempel på dokumentation .....	4
<b>Table: geo.Delyta</b> .....	5
3. Designprinciper.....	7
Grundläggande entitetsgrupper.....	7
Schemanamn .....	8
Normalisering (3NF) .....	8
Namnkonventioner .....	8
Sammansatta kolumnnamn .....	8
Suffix och Prefix.....	8
Språkstöd.....	9
Hur hanteras Å,Ä,Ö.....	9
5. Bygga bort generalisering.....	9
6. Använda affärsnyckel eller surrogatnyckel .....	9
Identifierare: Guid eller INT.....	9
7. Sql's inbyggda integritetsskydd .....	10
8. Använda stored procedures .....	10
Fördelar .....	10
Använda SP på rätt sätt .....	10
Vyer .....	10
Namngivning för vyer i Nilsbas.....	11
Grundvyer.....	11
Artlistavyer .....	12
Totalvyer.....	12
9. Tester.....	12
10. Faser .....	13
Utvecklingsfas.....	13
Driftfas.....	13

Versionshantering .....	13
Versionsnummer .....	13
11. Dataimport .....	14

## 1. Bakgrund & Syfte

Syftet med detta dokument är att beskriva de designprinciper som gäller vid framtagningen av Nilsbas. Detta dokument är ämnat till utvecklare som stöd för design av databasen.

## 2. Utvecklingsgång

Här beskrivs kortfattat hur arbetet med framtagandet av databasdesignen går till.

### Grundförutsättningar och datarelationer

Det är framför allt *fältinstruktion för nationell inventering av landskapet i Sverige* som i samråd med analytiker vid institutionen legat till grund för den design som tagits fram. Denna skrift är en manual för fältarbetare som är indelad i menyer och baserad på hur inmatningen ute i fält går till. Denna kan ge en förståelse för vilket data som logiskt hör samman. Hur uppdelningen i tabeller sedan görs är dock beroende av andra faktorer tex kraven på normalisering och ev. förestående metodikförändringar.

### Modellering

Databasen modelleras med hjälp av entity-relationship principer. Det valda verktyget för detta är ER-Studio. Där görs modelleringen i ett grafiskt gränssnitt på en konceptuell nivå och är inte kopplad till en viss databasversion. Här talar man därför om entiteter och attribut och inte tabeller och kolumner. Man anger primärnycklar och relationer och deras kardinalitet.

### Från modell till fysisk Databas

I Er-studio kan man välja vilken databasserver som är aktuell och generera ett script som ska skapa de tabeller och nycklar som kopplar ihop dem för just den plattformen. Detta ger en exakt överensstämmelse mellan modell och databasinstans. ER-diagrammet är alltså inte bara en fristående dokument utan är kopplat till den faktiska, resulterande databasen. Det ger även en flexibilitet beträffande målsystemet vilket kan blir värdefullt vid övergång till en annan databasserver.

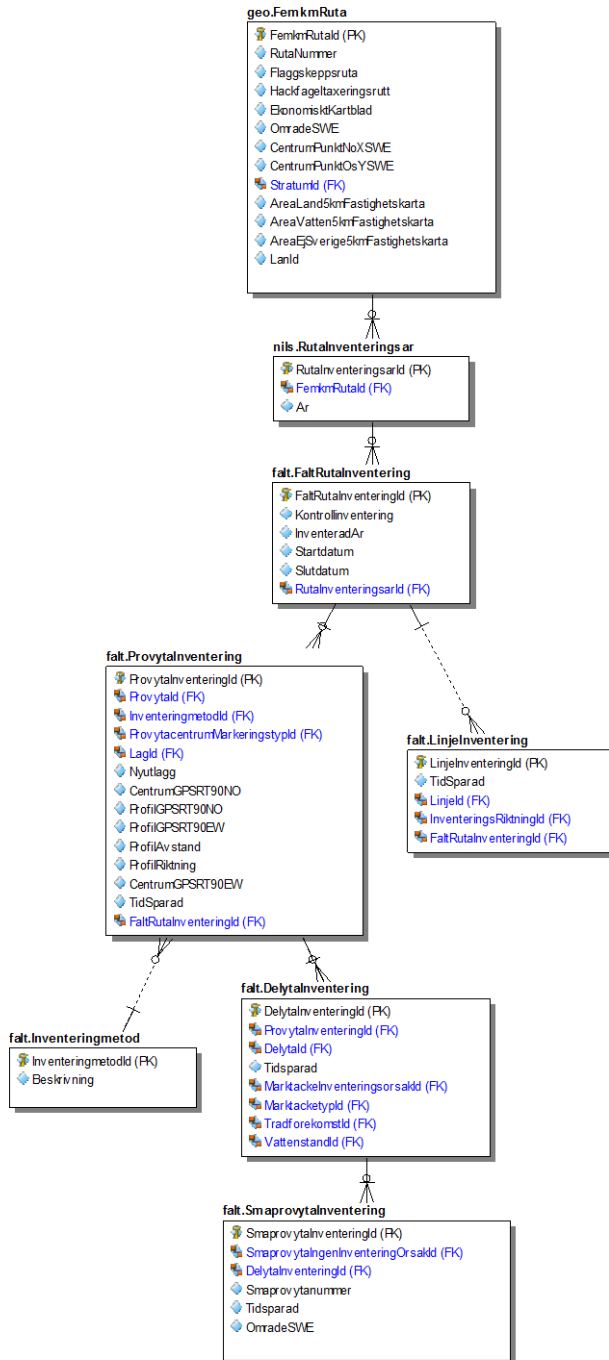
### Dokumentation

I ER-studio kan man generera rapporter i html-format som visar alla entiteter och relationerna mellan dem. Antingen på logisk eller fysisk nivå. Både listor och bilder går att få med i rapporten. I verktyget kan man också ange Extended properties och description fält på kolumner och tabeller och där kommer vi att beskriva så mkt det går.

### Exempel på dokumentation

ER-diagram från ER-studio beskrivande relationer mellan tex femkmruta, Provyta- och Delytainventering.





Exempel på automatgenererad dokumentation av databasen i tabellform

Table: geo.Delyta		Properties
<b>Owner:</b>	geo	
<b>Creation Date:</b>	05-16-2013	

**Located On:** PRIMARY  
**Data Size KB:** 123424  
**Index Size KB:** 1072  
**Rows:** 9280  
**Description:** - Semistatisk tabell - Definierar en delyta

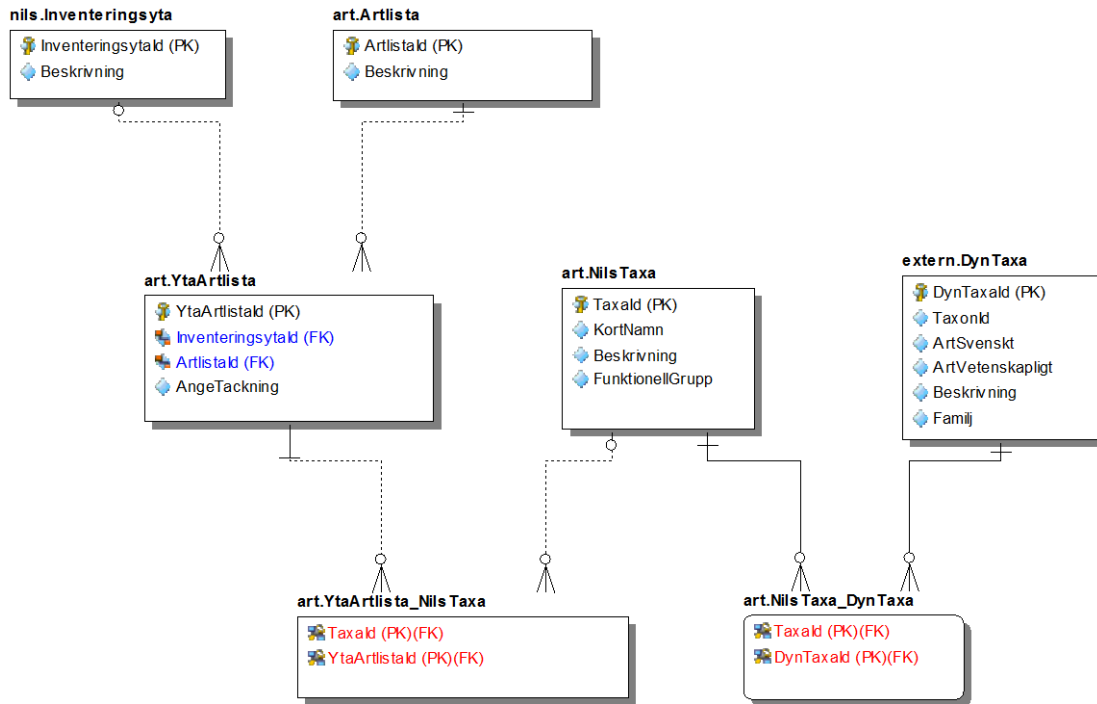
### Columns

Name	Data Type	Length	NULL	Default	IsIdentity	IsGUID	Description
Delytald	uniqueidentifier	16	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	
DelyteArea	decimal	5	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	
Provytald	uniqueidentifier	16	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	
DelytaNummer	int	4	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	
OmradeSWE	geometry	-1	<input checked="" type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	

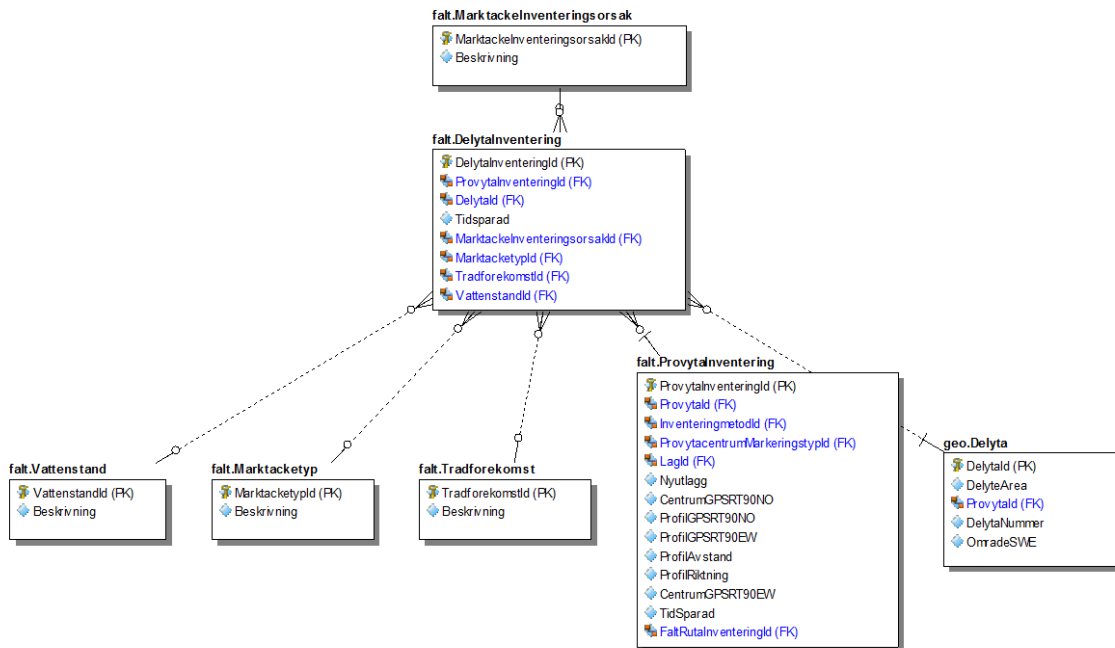
### Indexes

Index	Primary	Unique	Description
PK_Delyta	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
AK_Delyta_Delytanummer	<input type="checkbox"/>	<input type="checkbox"/>	
AK_Delyta_Provytald	<input type="checkbox"/>	<input type="checkbox"/>	

Ytterligare exempel på dokumentation för Arthantering



Delytainventering.



### 3. Designprinciper

#### Grundläggande entitetsgrupper

Datat som ska lagras i Nilsbas kan grovt indelas i tre grupper Statiskt, Semistatiskt och temporalt data. Beroende på om det är data som inte förändras eller sådant som läggs till över tid. Beroende på

vilken kategori en tabell tillhör så råder olika designkriterier. Denna egenskap ska anges vid beskrivningen av entiteten i ER-diagrammet (ER-studio).

## Schemanamn

Datat, och därmed den logiska modellen som representerar det, kan delas upp i submodeller som representerar separata delar av datat. Denna tillhörighet kommer att existera i databasen som ett Schemanamn som sätts på tabeller, lagrade procedurer och vyer. Exempel på submodeller/ schemanamn är nils, fält, flyg och art.

## Normalisering (3NF)

Databasen ska normaliseras till minst 3NF, men det får finnas undantag, men då ska dessa vara väl motiverade.

## Namnkonventioner

Hur vi väljer att döpa kolumner är av stor vikt. Högsta prioritet är att de är informativa för utvecklare och att man är konsistent. Slut användare kommer se vyer av databasen där andra namn på kolumner användas (och slut användarna kan föredra att ha konstiga förkortningar på variabler framför långa deskriptiva namn.)

## Sammanstatta kolumnnamn

Att konkatenera namn kan göras med understreck eller med inledande liten eller stor bokstav etc.

Exempel på olika sätt att skriva ihop flera ord.

- *CustomerName VS Name.*
- *TeamNumber VS Team\_Number*

Vi väljer att inleda med versal både initialt och vid nästa ord.

Exempel: ***KalMossaAndel***

## Suffix och Prefix

Det finns olika förkortningar och konventioner. Exempelvis:

### Suffix:

- **Number VS No**
- **Count VS Counter**

### Prefix:

- **TeamName VS Name**

Vi väljer följande suffix:

- **Count:** för att visa att det är en räknare av något slag
- **Name "utan prefix":** för att tabellnamnet redan berättar vems namn det är.

*Number* skriver vi inte som suffix, eftersom datatypen redan berättar om det en siffra. *Team* använder vi inte som prefix eftersom tabellnamnet redan definierar tillhörighet, och om inte bör designen kollas över.

## Språkstöd

Nilsbas kommer inte att erbjuda flerspråksstöd, eller översätta variabler till engelska, och använder därför tills vidare svenska namn.

## Hur hanteras Å, Ä, Ö

Vi hanterar det som följande:

Å = A

Ä = A

Ö = O

Huvudsaken är att vi är konsekventa och att man förstår variabeln. Om man inte förstår variabeln med dessa översättningar finns alltid förklaring i kolumnbeskrivningen, som alltid ska skrivas.

## 5. Bygga bort generalisering

Undvika generalisering eftersom sql har svårigheter för sådant. Det kan verka smart till en början, men blir slutligen ofta till problem.

Ex: En generisk "Status" tabell med kolumner : ID, RelatedToTable, RelatedToColumn, Code, Description, kan lagra status för 3 olika tabeller som har kolumner: CustomerStatus, ItemStatus, OrderStatus"

Vi vill undvika och bryta ner generiska tabeller. Det blir långa harranger av SQL och ett krångligt sätt att både plocka ut data och sätta in data.

## 6. Använda affärsnyckel eller surrogatnyckel

Det är enkelt och kanske snabbare att använda index och inte tänka på hur affärsnyckel skulle fungera som primärnyckel. Men mkt mer beskrivande att definiera naturliga nycklar. Vi vill använda surrogatnycklar för prestanda och enkelhet, men vi måste kunna finna och definiera affärsnyckeln. Detta kan vi definiera som metadata i description fältet på tabellen.

Fördelar med affärsnyckel:

- Lättare att förstå data. Man ser direkt vilken eller vilka kolumner som gör en rad unik
- Undviker att dubletter finns med olika surrogatnycklar.

**Slutsats:** Vi vill använda surrogatnycklar som, GUID, för att det ger bättre kompatibilitet, med exempelvis ORM, och snabbare prestanda än affärsnyckel, som ofta kan bestå av flera kolumner, men vi ska även dokumentera affärsnyckeln som metadata i databasen.

## Identifierare: Guid eller INT

Än så länge har vi inget prestanda problem, och att använda GUID löser många problem när man ska exportera, slå ihop data mellan databaser. Därför använder vi denna som identitet vid surrogatnyckel i första hand före en 4 byte integer som är snabbare.

## 7. Sql's inbyggda integritetsskydd

Trots att vi har tester på data innan vi importerar det, ska vi sätta skydd mot data-inkonsistens på databasnivå.

Vi ska definiera:

- Varchar/sträng längder
- Nullable eller inte?
- Foreign key constraints: delete/update cascade?

## 8. Använda stored procedures

### Fördelar

Det finns många fördelar med Stored procedures.

- Modifierbarhet
  - o Man kan ändra i SP och tabellstruktur bara man håller SP gränssnitt detsamma
- Prestanda
  - o Man kan casha SP i databasen, vilket gör dem snabbare
- Säkerhet
  - o Man kan inte sätta behörighet på en användares login att endast kunna ändra vissa kolumner på en rad, men man kan sätta behörighet att endast kunna köra en stored procedure som endast ändrar vissa kolumner.

### Använda SP på rätt sätt

1. Vi ska inte försöka göra generella stored procedures där man exempelvis skickar med tabellnamn och kolumnnamn, eftersom då förlorar man många av dess fördelar. Då kan man hellre använda dynamisk sql.
  - o Test och underhåll: Man behöver inte söka efter annat än argument och output.
  - o Säkerhet: Mindre risk för injections då endast sökargument eller output ges.
  - o Prestanda: Kan ha en single-compile-plan, och återanvändas.
2. Vi ska använda dem till ofta återkommande queries, eller sådana som kräver specifik säkerhet.
3. Vi ska inte använda dem för att göra avancerade beräkningar, det är komplicerad kod mkt bättre på, och vi ska inte göra oss beroende av vilken databas när vi gör beräkningar, detta gör vi i primärminnet med kod.

### Vyer

Vyer är nödvändiga för att förenkla åtkomst av data i en normaliserad databasstruktur där datat är spritt över en mängd tabeller.

## Namngivning för vyer i Nilsbas

Det finns tre nivåer i namnsättningen av vyer:

GrundVyns 'adress' består av  
[databas].[schema].[namn]

Art Vyns 'adress' består av  
[databas].[schema].[namn]\_[Artlista]

TotalVy  
[databas].[schema].[namn]\_AllaMenyer?

### Grundvyer

[databas]  
*Nilsbas*

[schema]

Vilket uppdrag hör det till

*nils*

*aob*

*moth*

*lillnils*

*system* (för vyer som enbart görs som stöd för databasprocedurer, andra applikationer eller liknande)

[namn]

v\_[inventeringsobjekt]\_[aggregeringsnivå]\_[innehåll]

[inventeringsobjekt]

*Ruta*

*Delyta*

*Provyta*

*InvLinje*

*Polygon*

*Linje*

*Punkt*

...

[aggregeringsnivå]

faller bort om vyn inte är aggregerad

*P* - Provyta

*R* - Ruta

*ABO* - Ängs-och betesobjekt

...

[innehåll]

Det beror på hur vi väljer att organisera vyerna. Här är några exempel.

*Grunddata*

*Berakningsunderlag*

*Markbeskrivning*

*Skogskant*

*Humlor*  
*Smavatten*

...

#### Exempel

Nilsbas.aob.v\_Delyta\_Markanvändning  
Nilsbas.aob.v\_Transekt\_Fjarilar  
Nilsbas.lillnils.v\_Provyta\_Utlagg  
Nilsbas.lillnils.v\_Linje\_Vegetationsremsa  
Nilsbas.lillnils.v\_Punkt\_SkyddsvartTrad  
Nilsbas.moth.v\_Delyta\_Marktacke  
Nilsbas.nils.v\_Delyta\_Markbeskrivning  
Nilsbas.nils.v\_InvLinje\_R\_Skogskant  
Nilsbas.nils.v\_InvLinje\_Skogskant  
Nilsbas.nils.v\_Linje\_Hagnad  
Nilsbas.nils.v\_Polygon\_Marktacke  
Nilsbas.nils.v\_Ruta\_Grunddata  
Nilsbas.system.v\_Vadsomhelst

#### (1) Flyg och fält

Det syns inte i namnet om inventeringen har skett i flygbild eller i fält. Istället går man direkt på vilken typ av objekt man är ute efter. För att skilja mellan linjekorsningsinventering i fält och flygbildsinventering av linjeobjekt heter linjer i fält *InvLinje* och linjer i flygbilder *Linje*.

Detta gäller våra ordinarie vyer.

#### (2) LillNILS

Provytorna är indelade i myr och gräs. Det ska framgå. Redan i schemanamnet?

#### (3) Kontrollinventeringar

Listor över vyer där man gör sitt val är i bokstavsordning och där vill man att kontrollinventeringsvyer finns som en egen grupp. Det som märker upp kontrollinventeringsvyer ska då kanske ligga direkt efter v (för vy).

#### (4) Flaggskepp

Samma som för kontrollinventeringar

### **Artlistavyer**

Artlistavyerna fungerar som modulära byggstenar till de mer slutanvändaranpassade vyerna. De har schemanamnet art och är baserade på

### **Totalvyer**

Totalvyer är ett slutanvändarvy som kombinerar en eller flera andra underliggande vyer.

## **9. Tester**

Vi vill kunna köra igenom automatiserade tester som ser till att

- data plockas ut på rätt sätt



- data uppdateras på rätt sätt
- data plockas bort på rätt sätt

Detta kan vi göra som stored procedures eller som Nunit-tester via C#, och göra exempelvis 1 gång i veckan som ett sql-jobb eller schemalagt program.

## 10. Faser

Nilsbas kommer att genomgå två faser: en utvecklingsfas och en driftfas. Dessa faser kommer att ha olika arbetssätt vid tillägg och förändringar av struktur och data. Övergången mellan dessa faser kommer att göras när databasen har nått tillräcklig mognadsgrad och när organisationen är redo att styra om sitt arbetssätt till den nya plattformen.

### Utvecklingsfas

Under utvecklingsfasen kommer hela strukturen och all data att laddas om för varje ny version av Nilsbas. Detta innebär att förändringar/rättningar av data i rättningsdatabasen kommer att slå igenom och överförs till NilsBas. Under utvecklingsfasen finns ingen garanti för att data är konsistent eller samma över tid och relativt omfattande strukturförändringar kan göras utan att förvarna eller ta hänsyn till eventuella användare av databasen. Hanteringen av nya versioner görs informellt utan dokumentationskrav och tidigare versioner behöver inte arkiveras.

### Driftfas

Under driftfasen kommer strukturändringar göras genom att exekvera databasscript eller SSIS-paket. De förändringar som är gjorda på data i Nilsbas kommer att följa med till den nya versionen. Däremot kommer inte ändringar av redan inläst data i försystemet att återspeglas i Nilsbas. Hanteringen av nya versioner görs formellt med versionsnummer och utförlig dokumentation av förändringar sedan förra versionen. Tidigare versioner arkiveras. Import av årsvis data kommer att skötas med hjälp av ETL-jobb. Utformningen av dessa ETL-jobb beror på vilket försystem som kommer att vara rådande. I nuläget är Falt\_rattning försystem. Vid tex byte till ny handdator kan försystemet ändras och därmed också villkoren för inläsning av nya fältdata. Tillägg eller borttagning av variabler i inventeringen inom samma försystem hanteras genom att ett nytt ETL-jobb skapas för det året. Det är rimligt att anta att ny version av databasen kommer att sammanfalla med ny årsvis import av data. Eventuell rensning och omläsning av ett inventeringsår i Nilsbas är ett scenario som ännu inte utretts. En annan aspekt av är.

## Versionshantering

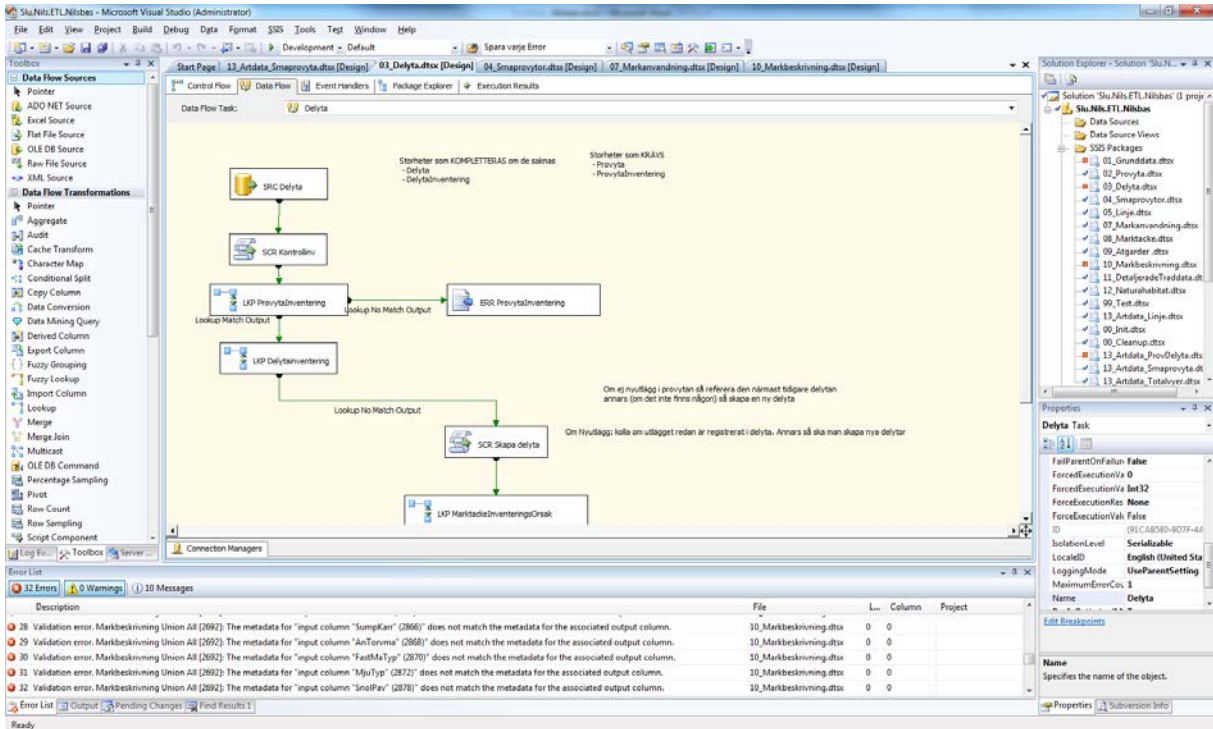
### Versionsnummer

Versionsnummer är treställigt och består av "stort" versionsnummer, modellversionsnummer och dataversionsnummer. Stora versionsnumret är noll under utvecklingsfasen och ett under driftfasen. Det kan även tänkas stiga när större ändringar görs i framtiden. Modellversionsnummer indikerar ändringar i databasstruktur och dataversionsnummer indikerar en ändring/utökning av det data som finns i databasen. Resp. versionsnummer är nollbaserade och en ökning av ett högre modellnummer nollställer alltid underliggande versionsnummer: om tex vi går från 0 till 1 i stort versionsnummer så sätts modell- och dataversionsnummer till 0.

Äldre versioner av Nilsbas ska finnas kvar och vara online på servern tills vidare. Den existerande Nilsbas kommer vid en ny release att byta namn till Nilsbas\_v[Versionsnummer] alltså tex

## 11. Dataimport

En betydande del av utvecklingsarbetet runt Nilsbas går ut på att fylla på tabellerna med historiskt data. Till detta används ett verktyg som heter SSIS och som är en Microsoftprodukt.



Här skapar man i ett visuellt gränssnitt kopplingar mod datakällor och målsystem och har möjlighet att hantera flödet, applicera logik och mappa det historiska indatat mot den nya tabellstrukturen.



## Innehållsförteckning

Innehållsförteckning .....	2
1. Inledning .....	3
1.1. Bakgrund och syfte .....	3
1.2. Omfattning.....	3
1.3. Intressenter.....	4
1.4. Ledning och funktioner i förvaltningsarbetet .....	5
1.5. Förvaltningsplan.....	6
2. Verksamhetsåret.....	7
2.1. Förvaltnings- och utvecklingsbehov.....	7
2.2. Mål för det aktuella året .....	7
2.3. Förvaltningsgruppen .....	8
2.4. Aktiviteter och tidplan .....	8
2.5. Genomförande.....	15
2.6. Förutsättningar .....	15
2.7. Riskanalys.....	15
3. Kommunikation.....	16
3.1. Intern kommunikation .....	16
3.2. Extern kommunikation.....	17
4. Budget.....	18
4.1. Finansiering.....	18
4.2. Kostnader .....	18

## 1. Inledning

### 1.1. Bakgrund och syfte

Det här dokumentet är en generell version av förvaltningsplanen för förvaltningsområdet dataförvaltning. Specifika mål och aktiviteter tillkommer för varje år.

Avdelningen för landskapsanalys förvaltar internt producerade data och metadata samt externt producerade data. Det handlar om stora mängder data som ska vara kvalitetssäkrade, tillgängliga och aktuella, och som kan arkiveras. Även flygbilder och fältfoton med tillhörande metadata ingår.

Syftet med FO Dataförvaltning är att tillhandahålla stöd för lagring av data och metadata, att bistå projekt som genomför datafångst med ändamålsenliga och användarvänliga system för datamottagning, kvalitetskontroll och lagring av data, samt hantering av flygbilder och fältfoton. Detta förutsätter underhåll och vidareutveckling av egenutvecklade systemapplikationer för dataförvaltning, samt ajourhållning av extern producerade systemapplikationer och kommersiella programvaror.

Övergripande mål

- 1 Egenproducerade data lagras, korrigeras och arkiveras på ett effektivt och säkert sätt
- 2 Metadata, samt nödvändig information om datahantering och rutiner är tillgängliga.
- 3 Externt producerade data är aktuella, fullständiga och tillgängliga för interna användare.
- 4 Egenutvecklade applikationer för dataförvaltning är effektiva och lättanvända.
- 5 Tredje parts programvara för dataförvaltning som svarar mot verksamhetens behov.

### 1.2. Omfattning

Förvaltningsområdet omfattar följande förvaltningsobjekt:

**1. Egenproducerade data:** alla data som samlas in inom NILS, ÄoB och LillNILS. Förvaltningen av data innebär ändringshantering av datainnehållet.

- Data som samlas in i olika projekt
- Grunddata för programmen och beräkningsunderlag för analyser
- Rutiner för hur och i vilken databas ändringar görs i datainnehållet
- Ändringsloggar
- Rutiner för versionshantering av databaser

### 2. Datasystem

Förvaltning av datasystemet tar vid efter datainsamlingen, d.v.s. vid dataöverföring från fältdator till kontor för data från fältinventeringar och efter incheckning av en färdigtolkad ruta för data från flygbildsinventeringar.

I objektet ingår:

- Alla databaser som innehåller egenproducerade data, inklusive backup-rutiner.
- Applikationer för överföring av data från en databas till en annan (ETL-jobb)
- Applikationer som sköter nätverksöverföring av data från fält till kontor (Datatransfer, LillNILS SBK)

- Applikationer för test av data. Vissa tester genomförs under pågående fältsäsong och andra under korrigeringsfasen. Kopplas till olika databaser. (Datatest, SQL-skript)
- Applikationer för editering i databaser, behövs för att genomföra korrigeringar i vissa databaser (Rättning, vyer, skript)
- Applikationer för övervakning av back-up, arkivering och import: Meddelar när back-up, arkiveringen eller import-jobb inte fungerar (Övervakning)
- Tredje parts programvara för datalagring, import, databasadministration, korrigering (SS Management Studio, SS Integration services, ArcMap, FME)

**3. Externa data:** Data som inte har producerats inom avdelningen och används som stöddata för datafångst, analys och utveckling.

- Höjdmodell, marktäckedata, fjällkartan, fastighetskartan, vägkartan, topografiska kartan, skyddade områden (GSD - Geografiska Sverigedata) från Lantmäteriet
- Blockdatabasen, Tuva, Naturtyp från Jordbruksverket
- Skyddsvärda områden, naturgeografiska regioner, naturgeografiska regioner, limniska ekoregioner, myrregioner från Naturvårdsverket
- Geodata Wiki: förvaltas av IT-stöd. Förvaltningen innebär kontakt med IT-stöd och bevakning av våra intressen.

**4. Flygbilder:** Förvaltningen av flygbilder börjar efter att UP Flygbildshantering har avslutats. Projektet tar fram en modell för förvaltningen.

- Flygbilder
- Flygbildshanteringssystem (FBHS)
- Ortofoton
- Ortofotodatabas

**5. Fältfoton:**

- Fältfoton
- Skript för hantering och sortering av fältbilder

**6. Fältkartor:**

- ej aktiv än

**7. Administration:** Verktyg för förvaltningen

- Arbetsplats: Information och ärendehantering
- Förvaltningsplan

### 1.3. Intressenter

Intressent	Vari ligger intresset?	Vad (delobjekt)?
------------	------------------------	------------------

<u>Analys- och rapporteringsprojekt</u> - PP NILS Analys och rapportering - PP Ä&B Rapportering - PP MOTH Analys och rapportering	Användarstöd för dataåtkomst Hantering av användarbehörigheter Åtkomst fältfoton och ortofoton	Egenproducerade data Externa data Databaser för lagring av egenproducerade data Fältfoton Ortofoton ArcMap programvara
<u>Datafångstprojekt</u> - PP NILS flygbildstolkning - PP LiLiNILS flygbildstolkning - PP MOTH flygbildstolkning - PP NILS fältinventering - PP LiLiNILS Småbiotopsinventering - PP ÄoB fältinventering - PP MOTH fältinventering	Rutiner för överföring av data Rutiner för korrigerig och eventuell efterbearbetning av data Rutiner för leverans av metadata Hantering av Användarbehörigheter och annan databasstöd Åtkomst stöddata Användarstöd för test- och korrigeringsapplikationer Åtkomst fältfoton Hantering av flygbilder	Flygbilder Flygbildshanteringssystem Externa stöddata Datatransfer Fältfoton
<u>Metodutvecklingsprojekt</u>	Rutiner för dokumentation och arkivering av data som produceras Användarstöd för databaser Åtkomst externa data Samordning av datahantering	Externa data Databaser Flygbilder ArcMap programvara
<u>Systemutvecklingsprojekt</u>	Testdata som behövs under utvecklingens testfas Användarstöd för databaser. Rutiner för uppsättning av utvecklingsdatabaser Samordning av datahantering mellan projekt	Databaser Egeninsamlade data
Leverantörer av flygbilder	Rutiner för beställning och leverans av flygbilder	Flygbildshanteringssystem
Institutionen/IT-stöd	Kontakt	Geodata Wiki
Analysstöd inom NILS Analys och rapportering	Samarbete kring datautplock	Databaser
FO Datafångst	Informationsutbyte kring metodändringar som påverkar databas och ETL	Databaser ETL
Avdelningsadministration	Informationsutbyte om lagringsutrymme för data, servrar Ärendehantering	Databaser Arbetsplats
Andra institutioner & avdelningar på SLU	Samarbete & kontakt Dataplattformen, utvecklingsmöten	
Externa dataanvändare (tidigast 2014)	Hantering av användarbehörigheter	Egenproducerade data och metadata

#### 1.4. Ledning och funktioner i förvaltningsarbetet

Arbetet med dataförvaltning leds av förvaltningsledare och IT-systemansvarig. Deras ansvar är att planera, följa upp och utvärdera verksamheten. Utöver förvaltningsledningen finns ett antal utförare, vilka är specialister på specifika delobjekt.

NILS programchef är objektägare och fastställer årlig budget för dataförvaltningen.

Avdelningens ledningsgrupp fastställer förvaltningsplanen, fattar beslut om aktiviteter och prioriteringar och bemannar av förvaltningsledningen.

Fasta funktioner i förvaltningsgruppen:

Förvaltningsledare

- utarbetar förvaltningsplan tillsammans med IT-systemansvarig
- verkställer förvaltningsplanens mål inom givna ramar
- bemannar roller på operativ nivå (objektspecialister) i samråd med LA-LG

IT-systemansvarig

- utarbetar förvaltningsplan tillsammans med förvaltningsledare
- verkställer förvaltningsplanens mål avseende IT-stöd
- bemannar roller på operativ nivå (applikationsansvariga) i samråd med LA-LG

Objektspecialister

- utför förvaltningsaktiviteter
- fångar och formulerar krav ur ett verksamhetsperspektiv
- säkrar användbarheten ur ett användarperspektiv

Applikationsansvariga

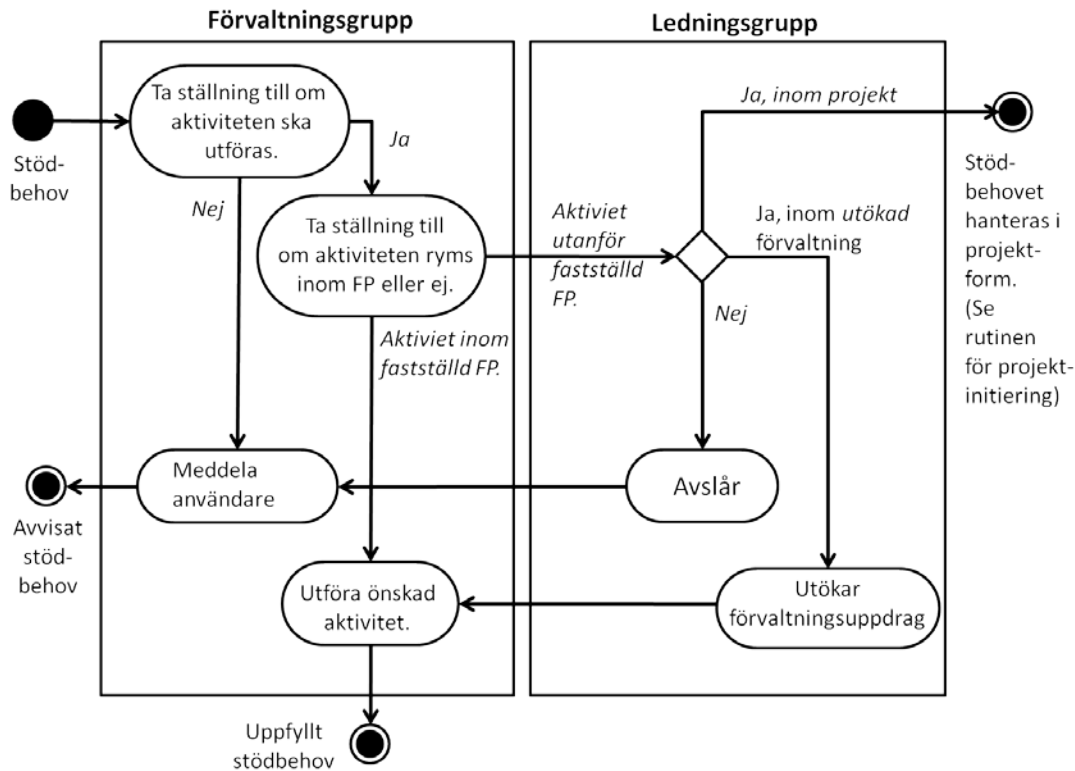
- utför förvaltningsaktiviteter (programmering)
- omvandlar verksamhetskrav till systemkrav
- säkrar användbarheten ur ett IT-perspektiv

För utförligare rollbeskrivning se separata dokument.

## **1.5. Förvaltningsplan**

Syftet med förvaltningsplanen är att klargöra hur förvaltningsarbetet ska styras, med fokus på fullgott stöd till projekt och annan verksamhet inom avdelningen för landskapsanalys. Planen är ett verktyg för förvaltningsgruppen i deras dagliga arbete. Förvaltningsarbetet styrs med hjälp av rutinen för operativ förvaltningsstyrning (figur 1).





Figur 1. Rutin för operativ förvaltningsstyrning.

## 2. Verksamhetsåret

### 2.1. Förvaltnings- och utvecklingsbehov

Behovet beror på verksamheten i övrigt. I slutet av varje förvaltningsår sammanställs hur förvaltningsobjekten har fungerat i verksamheten för att identifiera brister som behöver åtgärdas.

Återkommande aktiviteter handlar om att datasystemet behöver anpassas inför en ny inventeringssäsong.

Nyutvecklade komponenter kan kräva anpassning av befintliga komponenter i systemet. När utvecklingsprojektet avslutas övergår den nya delen i förvaltningen och inkluderas i förvaltningsplanen.

### 2.2. Mål för det aktuella året

Varje år setts specifika mål utifrån förvaltnings- och utvecklingsbehovet.

### 2.3. Förvaltningsgruppen

Verksamhetskompetens	IT-kompetens
<u>Förvaltningsledare och objektansvarig</u>	<u>IT-systemansvarig</u>
<u>Objektspecialister</u>	<u>Ansvariga/programmerare</u>
Egeninsamlade data NILS	-
Egeninsamlade data ÄoB	
Egeninsamlade data lillNILS	
Databaser NILS	Databaser och ETL
Databaser ÄoB	
Databaser lillNILS	
Datatransfer	.NET applikationer
Datatest	
lillNILS SBK	
Rättningsapp	
Externa data	GIT
Flygbilder, flygbilds-hanteringssystem	SharePoint, .NET
Ortofotodatabas, ortofoton	SDE-databas
Fältbilder och fältbildhantering	Python-app
Arbetsplats	SharePoint

### 2.4. Aktiviteter och tidplan

Förvaltningsaktiviteter kan vara engångsaktiviteter eller återkommande aktiviteter som genomförs varje år.

För mer detaljerad information se arbetsplatsens listor Återkommande aktiviteter och Ändringar datasystem

**Aktiviteter som utförs varje år**

1. Egenproducerade data

Delobjekt	Aktivitet	Planerad period	Ansvarig/Utförare	Tidsåtgång (i dagar)	Berörda projekt
NILS data	Uppdatera rutiner för korrigerings	våren	Fältingenjör + flygbilds-ingenjör	2+2	PP NILS Analys, PP NILS Fält
NILS fältdata	Utföra löpande korrigerings	kontinuerligt	Fältingenjör	8+4	PP NILS Analys
ÄoB Fjärilsdata	Utföra löpande korrigerings	hösten	Fältingenjör	2	PP ÄoB Analys
NILS tolkade data	Utföra löpande korrigerings	hösten	Flygbilds-ingenjör	?	PP NILS Analys
NILS fältdata	Anpassa tidigare års data efter årets ändringar	sommaren	Analytiker	6	PP NILS Analys, PP ÄoB Analys
NILS tolkade data	Anpassa tidigare års data efter årets ändringar		Analytiker	9 (ej i år)	PP NILS Analys
ÄoB Fjärilsdata	Anpassa tidigare års data efter årets ändringar		Analytiker	1	PP ÄoB Analys

2. Datasystem

Delobjekt	Aktivitet	Planerad period	Ansvarig/Utförare	Tidsåtgång (i dagar)	Berörda projekt
alla db	Övervakning av arkivering och back-uper	kont	DBA	1	alla
	Support, användaradministration	kont	DBA	5	
	Se över och rensa bland gamla db		DBA	0,5	
db Falt_import	Övervaka dataimport, rensning, dataöverföring till korrigeringsdatabas	maj-sep	DBA	20	PP NILS Fält
db Fjaril_import	Övervaka dataimport, rensning, dataöverföring till korrigeringsdatabas		DBA	7	PP ÄoB Fält

db Falt_import db Falt_rattning, ETL-jobb	Fånga & formulera krav	feb	Analytiker	1	PP NILS Fält
	Uppdatera efter årets behov	mar	Programmerare	5	
	Testa	mar-apr	Analytiker	2	
db Fjaril_import, rattning, ETL-jobb	Fånga & formulera krav	feb	Analytiker	1	PP ÄoB Fält
	Uppdatera efter årets behov	mar	Programmerare	5	
	Testa	mar-apr	Analytiker	1	
db tolkningsdatabas	Administration/Underhåll	kontinuerligt	DBA	5	PP NILS Flyg
db Småbiotoper	Administration/Underhåll	kontinuerligt	DBA	3	PP LiINILS Fält
Datatransfer	Fånga och formulera krav,	dec-jan	Fältingenjör	1+1	PP NILS Fält, PP ÄoB Fält
	Uppdatera inför årets behov	mar	Programmerare	3	
	Fixa buggar, förbättra funktion	mar-apr	Programmerare	5	
	Uppdatera systemdokumentation	maj	Programmerare	1	
	Uppdatera instruktioner, testa funktion	apr	Fältingenjör	4+3	
	Support och utbildning av fältpersonalen	maj-sep	Fältingenjör	2+1	
Datatest	Fånga och formulera krav,	dec-jan	Fältingenjör	1+1	PP NILS Fält, PP ÄoB Fält, PP NILS Flyg
	Uppdatera inför årets behov	mar	Programmerare	4	
	Fixa buggar, förbättra funktion	mar-apr	Programmerare	5	
	Uppdatera systemdokumentation	maj	Programmerare	1	
	Uppdatera instruktioner, testa funktion	apr	Fältingenjör	3+3+0,5	
	kontrollera att rapporter skickas	maj-sep	DBA	1	
	Utbildning och support av fältjouren	maj-sep	Fältingenjörer	1+1+1	
Rättning och vyer (mot NILS falt_rattning)	Fånga och formulera krav,	dec-jan	Fältingenjör	0,5	PP NILS Fält
	Uppdatera inför årets behov	maj	Programmerare	2	
	fixa buggar, förbättra funktion	maj	Programmerare	3	
	Uppdatera systemdokumentation	maj	Programmerare	1	

Uppdatera instruktioner, testa funktion	maj	Fältingenjör	1	
Fånga krav för vyer för kontroll under säsongen	mar	Fältingenjör	0,5	PP NILS Fält
Skapa vyer i falt_import	apr	Programmerare	1	
Utbilda fältjouren i åtkomst under säsongen	maj	Fältingenjör	0,5	

## 3. Externa data

Delobjekt	Aktivitet	Planerad period	Ansvarig/Utförare	Tidsåtgång (i dagar)	Berörda projekt
Alla	Omvärldsbevaka och fånga in krav från användarna	kontinuerligt	GIS-specialist + Analytiker	3+1	PP NILS Analys, PP ÄoB Analys, PP NILS Flyg
	Användarsupport	kontinuerligt	GIS-specialist		
	Uppdatera sammanställningar och länkar	kontinuerligt	GIS-specialist	2	
	Uppdatera lathundar för åtkomst	hösten	GIS-specialist	6+14	

## 4. Flygbilder

Delobjekt	Aktivitet	Planerad period	Ansvarig/Utförare	Tidsåtgång (i dagar)	Berörda projekt
Flygbildshanteringssystem	Fånga och formulera krav	höst	Flygbilds-ingenjör		PP NILS Flyg
	Fixa buggar och förbättra funktion		Programmerare		
	Testa funktion		Flygbilds-ingenjör		
	Uppdatera dokumentation		Programmerare		
	Uppdatera instruktioner		Flygbilds-ingenjör		
Flygbilder	Korrigera och komplettera metadata	kontinuerligt	Flygbilds-ingenjör		PP NILS Analys

	Samordna flygbildsbeställningar på avdelningen	Vid behov	Flygbilds-ingenjör		PP NILS Flyg, utvecklingsprojekt
	Granska beställningsunderlag när beställning görs inom enskilda projekt	Vid behov	Flygbilds-ingenjör		
	Mottagning av beställda bilder. Filöverföring (från ftp/hårddisk/sticka till Terra) och dokumentation i FBHS. Granskning av bildkvalitet och godkännande av leverans.	Vid behov	Flygbilds-ingenjör		
	Stöd vid upphandling av flygbilder eller fotografering	Vid behov	Flygbilds-ingenjör		
	Hantering av analoga bilder		Flygbilds-ingenjör		
	Se över och uppdatera lagringsrutiner för flygbilder		Flygbilds-ingenjör		
	Uppdatera lathundar för orientering av flygbilder mm.	höst	Flygbilds-ingenjör		
	Utbilda flygbildstolkare i fotogrammetri	vinter	Flygbilds-ingenjör		
	Support vid användning av flygbilder	kontinuerligt	Flygbilds-ingenjör		
	Utvärdera och planera	höst	Flygbilds-ingenjör		
Ortofotodatabas	Utvärdera, fånga och formulera krav		Analytiker		PP NILS Analys, PP NILS Fält?, PP ÄoB Analys, PP ÄoB Fält?
	Uppdatera systemet vid behov		Programmerare		
	Testa		Analytiker		
Ortofoton	Generera bilder som saknas		Flygbilds-ingenjör		PP NILS Analys, PP ÄoB Analys
	Korrigera och komplettera metadata		Flygbilds-		

			ingenjör	
	Utvärdera behovet		Analytiker	

## 5. Fältfoton

Delobjekt	Aktivitet	Planerad period	Ansvarig/Utförare	Tidsåtgång (i dagar)	Berörda projekt
Säsongsskript	Beskriva årets behov (sökväg t.ex.)	mar	Fältingenjör		PP NILS Fält
	Testköra programmet för sortering under säsongen och testa resultatet	mar	DBA + Fältingenjör	1+2	
	Fixa buggar och uppdatera efter årets behov	mar	Programmerare		
	Förbättra funktion	klart	Programmerare		
	Utbildning för fältjouren	apr	Fältingenjör	1	
	Övervaka körning av programmet under säsongen	maj-sep	DBA	2	
	Uppdatera dokumentation	mar	Programmerare	0,5	
	Uppdatera rutin, lathund	?	Fältingenjör	0,5	
Arkiveringsskript	Köra om skriptet (några gånger)	mar	KO*		PP NILS Analys, PP ÄoB Analys
	Testa	mar	Analytiker	2	
	Fixa eventuella buggar		Programmerare		
	Uppdatera dokumentation	mar	Programmerare	0,5	
	Uppdatera rutin för körning av skriptet	?	Analytiker	0,5	
NILS Provytebilder	Korrigera fel och leta efter saknade bilder	mar	Fältingenjör	3	PP NILS Fält

## 6. Administration

Delobjekt	Aktivitet	Planerad period	Ansvarig/Utförare	Tidsåtgång (i dagar)	Berörda projekt
-----------	-----------	-----------------	-------------------	----------------------	-----------------

Förvaltningsplan	Skriva förvaltningsplan, göra budget	nov	Förvaltningsledare + IT-systemansvarig	5 + 5	alla
	Utvärdera verksamhetsåret	jan	Förvaltningsledare + IT-systemansvarig	3 + 3	
	Rapportera avvikelser till objektägarna och projektledarna	kont	Förvaltningsledare	2	
Arbetsplatsen	Se över strukturen och fånga önskemål		Förvaltningsledare	2	alla
	Uppdatera allmänna sidor efter behov		Förvaltningsledare	3	
	Stämna av ärendehantering med projekt och andra FO:n	?	Förvaltningsledare	2	
	Administrera att-göra-listor		Förvaltningsledare + IT-systemansvarig	2	
	Uppdatera beskrivning av aktiviteter, tiduppskattning, lägga till länkar till lathundar, mm	efter utförd aktivitet	Alla		
Terra 8_databaser	Uppdatera rutiner		DBA + Förvaltningsledare	2+2	alla
	Rensa och städa		DBA + Förvaltningsledare	2+2	



## 2.5. Genomförande

Tidplanen för genomförandet av aktiviteterna stäms av med berörda program- och utvecklingsprojekt.

Under våren fokuserar förvaltningsgruppens arbete på att uppdatera datasystem och fältfotohantering inför fältsäsongen.

Övervakning och administration av databaser sker kontinuerligt under året.

## 2.6. Förutsättningar

Ansvarsfördelningen (objektägare, objektansvarig, objektspecialist) måste vara förankrat på avdelningen.

Många aktiviteter kräver specialkompetens av utföraren (t.ex. systemutveckling, GIS). Det är därför viktigt att nödvändig kompetens finns på avdelningen.

Löpande kommunikation med programprojekten är en förutsättning för att stämma av tidplan och uppdatering av rutiner.

En effektiv ärendehantering underlättar arbetet och organisation inom FO:t.

Avdelningens gemensamma resurser (ex. servrar och SP) förvaltas på ett fungerande sätt.

## 2.7. Riskanalys

Oklar ansvarsfördelning kan leda till dubbelarbete.

Rollen som IT-systemansvarig är inte bemannad i dagsläget. Det gör det svårt att uppskatta tid och klara ut beroenden av teknisk karaktär mellan olika aktiviteter. Felbedömningar kan göras vid bemanning av applikationsansvariga. Omvärdsbevakning ur ett it-perspektiv görs inte. Ramar för design, arkitektur, it-strategi uppdateras inte och ingen är ansvarig för att de följs av programmerarna. Riktlinjer för dokumentation (se nedan)

Aktiviteter om databaser och egenutvecklade applikationer kräver specialkompetens. Det innebär en risk om kompetensen är personberoende, framför allt så länge dokumentationen av rutiner inte är fullständig. Nyckelkompetens för varje objekt borde finnas hos minst två personer.

Applikationsansvariga bemannas i stor utsträckning med konsulter. När de slutar försvinnar kompetensen. Konsulternas uppdrag pågår inte under hela året. Konsulten finns i princip inte tillgänglig för akuta insatser efter utvecklings-/uppdateringstillfället. Arbetets effektivitet och kvalitet är beroende på dokumentationens kvalitet.

Planeringen förutsätter att arbetet med andra FO:n går hand i hand. Det kan påverka arbetet att förvaltningen av vissa objekt är fortfarande under utveckling eller vilande.

### 3. Kommunikation

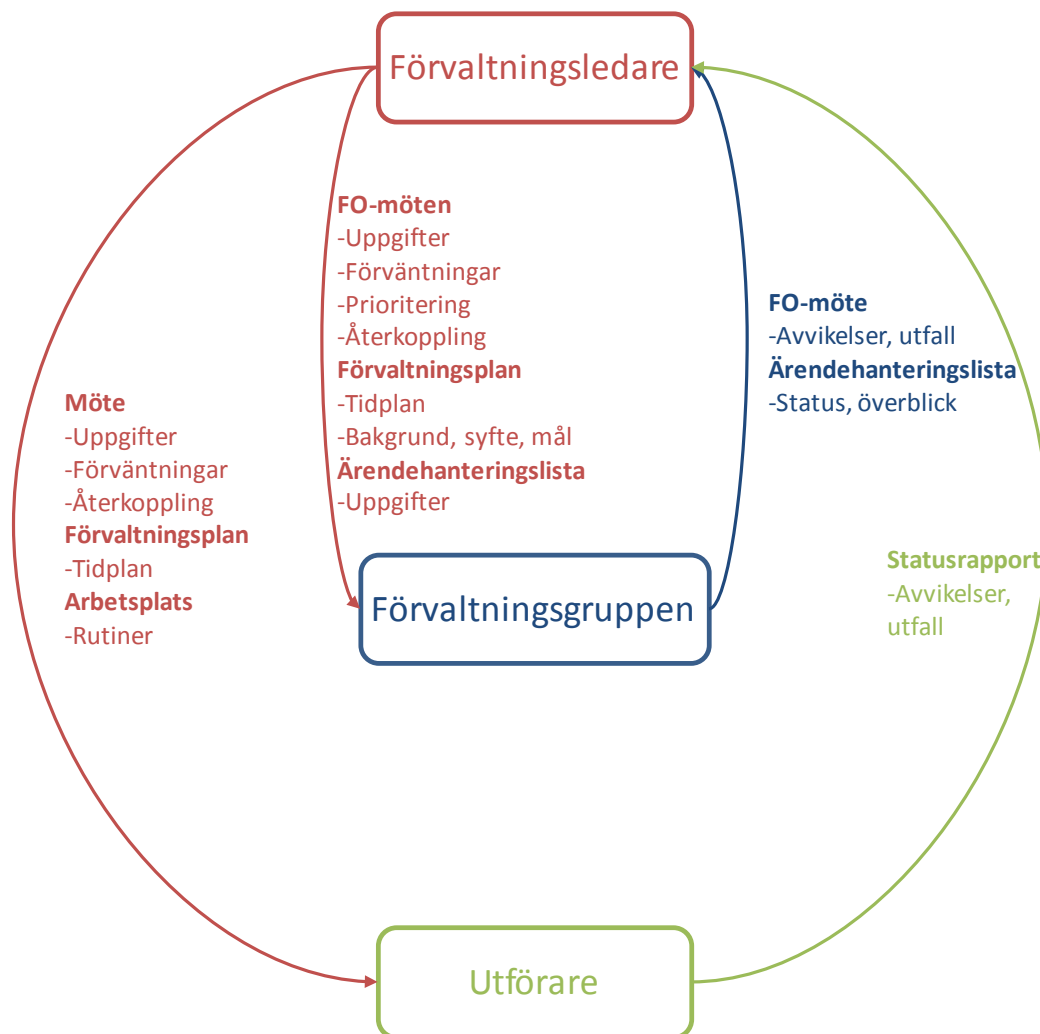
#### 3.1. Intern kommunikation

Möten sker i huvudsak på objektsnivå. Förvaltningsledaren är sammankallande för FO-möten. Utförare kallas till FO-möten vid behov. I övrigt har utförare direkt kontakt med förvaltningsledaren för avstämning av aktiviteter.

Hur återkommande aktiviteter ska genomföras beskrivs i en aktivitetslista på FO:ts arbetsplats med eventuell hänvisning till rutiner.

I förvaltningsplanen samlas information om planering, bakgrund, syfte och mål.

Akuta åtgärder och buggfixar tilldelas via en ärendehanteringslista på SharePoint.



Översikt över kommunikationskanaler inom FO-gruppen.

### 3.2. Extern kommunikation

En stor del av informationsutbytet med andra FO:n, projekt och programchef sker på programinterna verksamhetsmöten. Om det ska fungera behöver mötena vara effektiva och väl förberedda. Förvaltningsledaren ansvarar för att nödvändig information sammanställs inför mötet.

I förvaltningsplanen sammanställs resursbehovet åt programchef och ledningsgrupp.

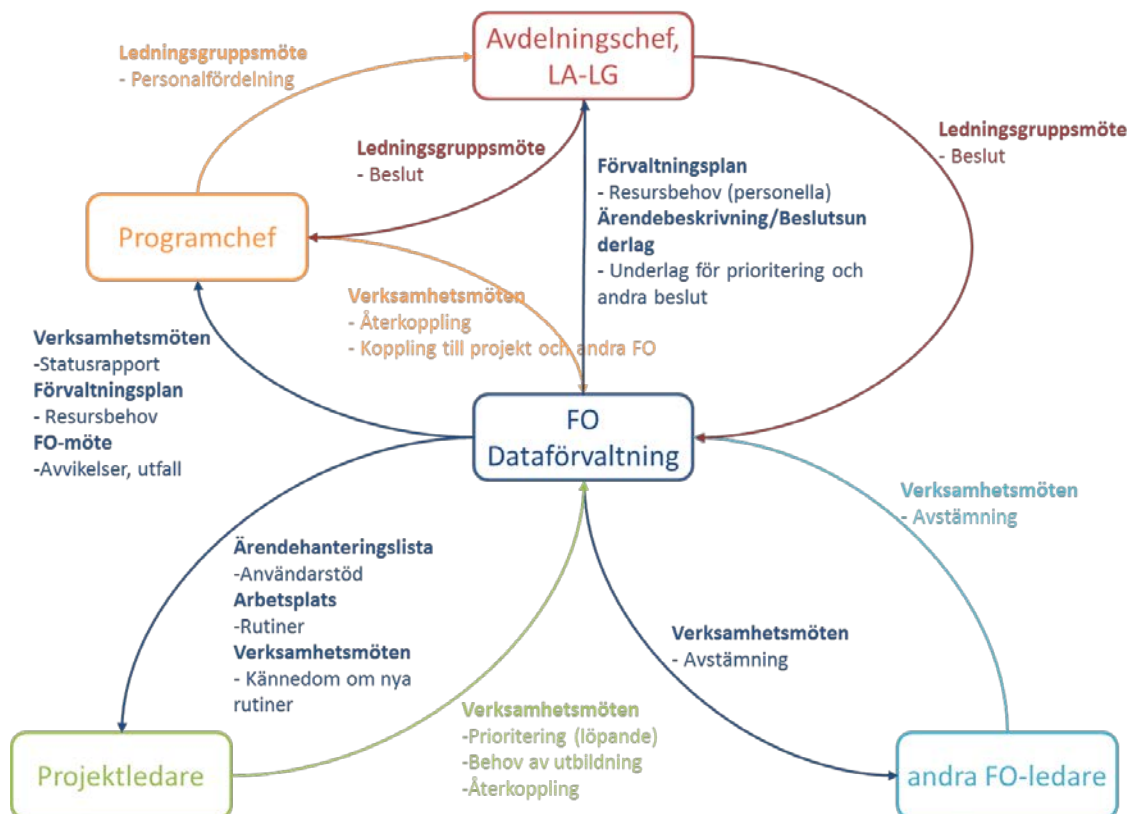
Ärenden till ledningsgruppen beskrivs i skriftlig form med bilagor som underlag för beslut.

Förvaltningsledaren adjungeras på ledningsgruppsmöte vid behov. Annan information från ledningsgruppen förmedlas till förvaltningsledaren via NILS-programchef.

Rutiner tillgängliggörs via avdelningens arbetsplats. Ansvariga för uppdatering av rutiner behöver utses.

Programcheferna kallas till FO-möten och deltar vid behov.

Kommunikationen med övriga intressenter utanför avdelningen sköts av respektive ansvarig för beställning av exempelvis flygbilder, externa geodata eller hantering av användarbehörigheter. Kommunikationskanaler och -ansvar beskrivs i respektive aktivitetsbeskrivning/rutin.



Översikt över kommunikationskanaler inom avdelningen som berör FO Datafångst.

## 4. Budget

### 4.1. Finansiering

-

### 4.2. Kostnader

#### NILS

KOSTNADSPOST	%	KKR
Personal (anställda på Srh)		
Förvaltningsledare	25	
IT-systemansvarig	25	
DBA	10	
Fältingenjörer	15	
Flygbildsingenjörer	15	
GIS-specialist	15	
Analytiker	20	
Utvecklare (.NET, SQL)	30	
Utvecklare (Python)	5	
Summa lönekostnader inkl LKP och semesterlönetillägg		
Indirekta kostnader (påslag) och lokalkostnader		
Övriga lönekostnader (ej anställda på Srh)		
Summa övriga lönekostnader		
Resekostnader, inklusive hyrbilar och traktamenten		
Övriga kostnader (datorkostnader, material m m)		0
Konsult		
Totalt		

#### ÄoB

KOSTNADSPOST	%	KKR
Personal (anställda på Srh)		
Förvaltningsledare		
IT-systemansvarig		
DBA	3,5	
Fältingenjörer	7,5	
Flygbildsingenjörer		
GIS-specialist		
Analytiker	3,5	
Utvecklare (.NET, SQL)	3	
Utvecklare (Python)		
Summa lönekostnader inkl LKP och semesterlönetillägg		

Indirekta kostnader (påslag) och lokalkostnader		
Övriga lönekostnader (ej anställda på Srh)		
Summa övriga lönekostnader		
Resekostnader, inklusive hyrbilar och traktamenten		
Övriga kostnader (datorkostnader, material m m) Magnus Flagge 0%, Daniel Sivertsson 3%		0
Totalt		

## LIIINILS

KOSTNADSPOST	%	KKR
Personal (anställda på Srh)		
Förvaltningsledare		
IT-systemansvarig		
DBA	3	
Fältingenjörer		
Flygbildsingenjörer		
GIS-specialist		
Analytiker		
Utvecklare (.NET, SQL)		
Utvecklare (Python)		
Förvaltningsledare		
Summa lönekostnader inkl LKP och semesterlönstillägg		
Indirekta kostnader (påslag) och lokalkostnader		
Övriga lönekostnader (ej anställda på Srh)		
Summa övriga lönekostnader		
Resekostnader, inklusive hyrbilar och traktamenten		
Övriga kostnader (datorkostnader, material m m) Magnus Flagge 0%, Daniel Sivertsson 0%		0
Totalt		

## Ekonomisk sammanställning NILS Datafångst och Datavårdskap 2008-2012

	År 2008 sep-dec	År 2009 jan-dec	År 2010 jan-dec	År 2011 jan-dec	År 2012 jan-dec	Totalt
<b>Ingående saldo</b>	0	668 365	1 458 733	1 308 579	380 575	0
<b>Intäkter</b>						
Fakturering Naturvårdsverket (uppdrag)	1 100 000	2 300 000	1 000 000	1 000 000	700 000	6 100 000
Bidrag Naturvårdsverket	800 000	502 000	1 300 000	0	0	2 602 000
LBP-medel	0	0	0	0	300 000	300 000
<b>Summa intäkter</b>	<b>1 900 000</b>	<b>2 802 000</b>	<b>2 300 000</b>	<b>1 000 000</b>	<b>1 000 000</b>	<b>9 002 000</b>
<b>Kostnader</b>						
Lönekostnader inkl. semestertillägg	-862 612	-885 511	-744 952	-741 481	-582 401	-3 816 957
Traktamenten och förrättningstillägg	-2 509	-4 150	-435	-2 097	0	-9 191
Resekostnader och logi	-2 996	-20 286	-3 640	-3 697	0	-30 619
Konsultkostnader	0	-729 225	-1 341 464	-872 722	-598 673	-3 542 084
Övriga kostnader	-2 749	-147	-17 388	-2 692	-3 804	-26 779
Overhead inkl. lokaler	-360 769	-372 313	-342 275	-305 315	-227 681	-1 608 353
<b>Summa kostnader</b>	<b>-1 231 635</b>	<b>-2 011 632</b>	<b>-2 450 154</b>	<b>-1 928 004</b>	<b>-1 412 559</b>	<b>-9 033 984</b>
<b>Periodens resultat</b>	<b>668 365</b>	<b>790 368</b>	<b>-150 154</b>	<b>-928 004</b>	<b>-412 559</b>	<b>-31 984</b>
<b>Utgående saldo</b>	<b>668 365</b>	<b>1 458 733</b>	<b>1 308 579</b>	<b>380 575</b>	<b>-31 984</b>	<b>-31 984</b>



Sveriges lantbruksuniversitet  
Swedish University of Agricultural Sciences

## **Kravspecifikation analysapplikation**

## Innehållsförteckning

1	Syfte .....	3
2	Användare .....	3
3	Övergripande krav för urval, flöden, analysverktyg och webbportal .....	4
4	Icke-funktionella krav.....	5
5	Format.....	5
6	Standardberäkningar vs mer avancerade beräkningar.....	6
6.1	Val av statistikprogram .....	6
6.2	Vilka krav ställer beräkningarna på urvals- och beräkningsapplikationen .....	6
6.3	Hur vill vi presentera våra skattningar/göra dem tillgängliga?.....	8
7	Redovisning av kvalitet i analysverktyg och webbportal .....	9
8	Redovisning av medföljande metadata .....	10
9	Arkivering.....	11



## 1 Syfte

Analysverktyg:

Analysverktyget är en applikation med två huvudfunktioner, en urvalsfunktion och en beräkningsfunktion, och syftena med analysverktyget är att 1) underlätta sammanslagning av data och utplock ur NILSbas och 2) underlätta standardanalyser (analyser som görs regelbundet). Genomgående för byggandet av dessa båda delar är att vi börjar bygga för en liten delmängd och fortsätter sedan att utveckla och inkludera fler delar/analysmöjligheter med tiden. Urvalsfunktionen kommer i första hand bestå av vyer för att först i samband med webbportalen utvecklas vidare till ett urvalsverktyg. Analysverktyget kommer i första hand bestå av R-script som sedan utvecklas vidare till att leverera skattningar på webben.

## 2 Användare

Analysverktyget:

I FIND delades användarna av NILS-data in i specialist, professionell och publik, som har olika kompetenser och behov av servicenivå. Specialisten har kunskap om NILS data, kan förstå beräkningsscript och skriva enkla SQL-sökningar. Till professionella användare hör länsstyrelser och andra myndigheter som har behov av aggregerade data (på olika nivåer) och färdiga analyser.

Användarna av analysverktyget som helhet är i första hand samtliga analytiker på avdelningen för landskapsanalys. Även analytiker i Riksskogstaxeringen kan bli användare av analysverktyget. Den här gruppen användare kan klassificeras som specialister.

Urvalsfunktionen bör kunna användas av fler än analytiker (i förlängningen kan den ev användas även i webbapplikationen, se nästa avsnitt), t ex alla på avdelningen som behöver kunna titta på data. Även denna grupp räknas till specialisterna, då alla har god kunskap om inventeringsmetodik och variabler.

Sannolikt kommer även forskare att vilja utnyttja analysverktyget. Forskare har i allmänhet relativt liten kunskap om NILS design, variabler och beräkningsrutiner och kan därför komma att kräva en hög servicenivå. Om denna grupp ska inkluderas i användarna ställs därför andra krav på analysverktyget (liknande kraven på webbportalen), än om de inte inkluderas. Forskares behov av data från NILS kan även tillgodoses genom att analytiker tillhandahåller data i dialog med forskaren. Med den lösningen kan analysverktyget byggas så att det är maximalt flexibelt, men kräver kunskap om variabler och design.

### 3 Övergripande krav för urval, flöden, analysverktyg och webbportal

Analysverktyget består av 2 delar: urval/utplock och analys/beräkningsdel. Vi behöver sammanställa vyer och ta fram R-script för våra olika skattningar. Behovet av en beräkningsapplikation (jmf taxwebb eller kub) som automatiskt utför beräkningar är dock inte lika stort hos oss som hos RIS. Vi har inga krav på oss att leverera "samma" analysresultat varje år (typ Skogsdata), utan kommer sannolikt att få varierade uppdrag. Det kan innebära att kostnaden för att utveckla en "komplett" beräkningsapplikation blir väldigt hög i förhållande till det utbyte vi kommer att ha av den. Vad vi behöver är att script utvecklas för alla beräkningstyper och att vi lär oss använda dem. Så småningom kommer vi att märka vilka beräkningar som efterfrågas ofta och kan då ta ställning till om vi skulle ha nytta av en "tryck på knappen"-funktion. Beräkningsapplikationen skulle då utvecklas successivt över tiden och utvecklingen skulle vara behovsstyrd.

Urvalsapplikationen:

- Vyerna skall vara kopplad direkt till den normaliserade databasen
- Urvalsapplikationen som kommer att finnas på webben skall vara kopplad direkt till den normaliserade databasen
- Vilket utplock som helst skall kunna genomföras med hjälp av urvalsapplikationen med dess rullistor och boxar.
- Vyerna är första prioritet att bygga och därefter urvalsapplikationen. Enkla sammanslagningar och utplock prioriteras, följt av mer detaljerade utsökningar, till exempel visst marktäcke och visst procentintervall busktäckning.

Analysapplikationen:

- Sammanställa data från databasen
- Beräkna skattningar och standardfel (R-script)
- Genomföra tidseffektiva skattningar
- Variabelnamn ska kunna väljas på svenska eller engelska
- Urval skall kunna utföras inom valfri geografisk region. Det sätt som förordas är att avnämaren redan i förväg har sitt område avgränsat i en shapefil.
- Det skall finnas en logg över hur man gjort sammanslagningarna i urvalsapplikationen
- Användaren ska kunna skapa egna villkor och beräknade variabler
- Kunna backa till tidigare steg i uttagsapplikationen
- Kunna ändra i urvals- (SQL) och beräkningskoden (R eller SAS)
- Projekthantering, så att man kan återanvända tidigare utplock och beräkningar
- Ska koppla mot metadatabas och visa när det inte finns data för vissa variabler för vissa år eller när metodiken har ändrats över tiden (medföljande metadata ger information om vilka år som saknas, syftet med variablerna, etc.)

- Länk till variabelbeskrivning i wiki eller SP-lista
- Artdata skall vara kopplade till artdatabankens databas dynamiska taxa genom att vi använder samma ID som dyntaxa. Årligen sker en uppdatering av NILSbas gentemot databasen dyntaxa för kontinuerlig och up to date information om artnamn.
- Man skall kunna ha fler än ett utplocksfönster öppna samtidigt för att kunna korrigera och komplettera samt förhandsgranska utplocken
- Vi skall kunna välja vilken databasversion vi vill jobba med

## 4 Icke-funktionella krav

Analysverktyget:

- Hög "genomskinlighet" – kunna se koden bla
- Säkra, d.v.s. reproducerbara, beräkningar
- För att öka underhållbarheten skall inga data eller metadata sparas i applikationer utan vara hämtbara från databasen
- Utföra korrekta beräkningar – beräkningar utförs med känd variabel och kontrolleras därigenom (exv Sveriges totalarea )
- Användargränssnitt designat enligt windowsstandard med hög användbarhet

## 5 Format

Urval:

Formatet på data skall vara läsbart internt såväl som externt.

- Indata in i analysverktyget dvs format för utplock:
  - inläsning direkt från databasen
  - 'egen' xml-fil, excel-fil eller text-fil med data som laddas i analysdelen direkt
  - shape-fil för att kunna gör urval för ett godtyckligt område (se 5.1.8)

Beräkningar:

- Utdata dvs format för resultat redovisning:
  - datauttag eller beräknade data i form av en tabell
  - spara som xml, txt eller excel

## 6 Standardberäkningar vs mer avancerade beräkningar

### 6.1 Val av statistikprogram

Det statistikprogram som är mest funktionellt vid utplock och analys skall väljas. Användarvänlighet vid analys, funktionalitet gentemot databasen och eventuella synergieffekter vid en framtida statistikplattform (dvs expertis vid SLU för programvaran) skall beaktas. R har visat sig vara det program som har störst flexibilitet och flest användare vid avdelningen samt ett tilltalande survey paket som passar NILS skattningar. Programvaran är fri och utvecklas ständigt gentemot de nya behov som växer fram. R har bra kompatibilitet med geografiska data och är flexibelt och formbart. *Nackdelen är dock att kvalitetssäkringen kan hålla olika nivå för olika delar då utvecklingen av programvaran sker på ideell basis.* Det är dock viktigt att utplocken är i format som alla kan hantera då det skall vara möjligt att jobba med vilket analysprogram som helst

### 6.2 Vilka krav ställer beräkningarna på urvals- och beräkningsapplikationen

För att kunna kravspecificera denna del krävs att vi har kommit igång med beräkningarna i större utsträckning än vad vi har just nu. Många av de beräkningar som beskrivs nedan har vi aldrig genomfört. Steg 1 är att beskriva algoritmerna. Detta är beskrivet för de flesta beräkningarna och skattningarna men visst arbete återstår. Uppdateringar och beskrivningar av nya beräkningar kommer att genomföras kontinuerligt. Utöver standardanalyserna som vi väljer att bearbeta, beskriva och redovisa i första hand kommer våra användares intresse att styra vilka analyser och i vilken ordning de kommer att beräknas. Ett urval av 1) tillståndsskattningar, 2) tvåfasskattningar och 3) förändringsskattningar arbetas igenom i nämnda prioriteringsordning och utgör case studies för hur vi vill att analysverktyget i första hand men även webbportalen skall se ut. Resultat från dessa analyser presenteras i väntan på webbportalen i NILS webben.

#### Övergripande krav:

- Möjligheten till maximalt flexibla utplocka från databasen skall finnas
- Hela utplock skall gå att genomföra dvs data är kompletterade och korrigerade för missing value eller 0, saknade data är kompletterade så långt det går, variabler som bytt namn men har samma innehåll är sammankopplade etc.
- Fastställda kvalitetssäkrade beräkningsalgoritmer för de skattningar som skall genomföras skall finnas (se formelsamling och senare script biblioteket)
- Tillgång till de variabler, beräknade variabler och stöddata som krävs för att kunna genomföra beräkningarna skall finnas (inklusionssannolikheter, vikter mm).

#### Aktuella redovisningsgrupper att förhålla sig till:

I första hand:

- enskilda strata och grupper av strata
- enskilda regioner och grupper av regioner

- enskilda län och grupper av län

I andra hand:

- landskap + lappmarker
- Naturgeografisk indelning (fr. miljö tillståndet i skogen bla.)
- Biogeografiska regioner enligt habitatdirektivet
- De fem vattendistrikten och ev. huvudavrinningsområdena
- Klimatzoner

### Standardberäkningar

Följande skattningar skall vara möjliga (se formelsamlingen för mer information):

I första hand:

- Tillståndsskattningar exv. ett urval av indikatorer

I andra hand:

- Förändringsskattningar på de ovan nämnda skattningarna bla övergångsmatriser för arealer (av fr.a. marktäcke/markanvändning), kanske också för vissa kvalitetsmått inom arealer. (T.ex. för att följa dynamiken förbuskning/röjning i gräsmarker.)
- Två-fas skattningar

Följande analyser skall vara standardberäkningar (se bilaga 3):

I första hand:

- Totaler av areal- linje-, punktelement, absolut och per arealenhet.
- Medelvärden och median för provytor, t.ex. täckningsgrad av olika arter
- Medelvärden och median för attribut på linje och punktelement, t.ex. medelbredd av stigar
- Proportioner, t.ex. förekomstfrekvens av olika arter eller andel ytor med viss påverkan.
- Medelfel, konfidensintervall (95%) och ev. även andra spridningsmått (medelfel som standard i samtliga fall där det är möjligt).

I andra hand och utvecklas därmed med tiden och allteftersom:

- Ett urval av landskapsmått och index
- Framskrivningsmodeller
- Habitatmodeller
- Medelfel, konfidensintervall (95%) och ev. även andra spridningsmått (medelfel som standard i samtliga fall där det är möjligt).

## Mer avancerade beräkningar

Övriga analyser genomförs av analytikerna:

- ”scripten” för standardberäkningarna skall kunna återanvändas av analytikerna vid andra analyser än standardanalyserna. Script under utveckling sparas i scriptbibliotek som senare utgör grunden för R-paketet.

## 6.3 Hur vill vi presentera våra skattningar/göra dem tillgängliga?

Våra användare kommer att få ge förslag till hur vi bör presentera våra skattningar för att tillgodose deras behov.

### Standardskattningar:

Standardskattningar och standardberäkningar finns beskrivna under analysverktyget. Hur vi ska presentera dessa kommer att växa fram med tiden och komplettera denna del av kravspecifikationen.

1. Standardskattningar som genomförs i första hand kommer att vara tämligen enkla att presentera dels grafiskt men även i tabellform eller relaterade till en karta då det kommer att vara sammanställningar av enskilda variabler.
  - Totaler av areal- linje-, punktelement, absolut och per arealenhet.
  - Medelvärden och median för provytor, t.ex. täckningsgrad av olika arter
  - Medelvärden och median för attribut på linje och punktelement, t.ex. medelbredd av stigar
  - Proportioner, t.ex. förekomstfrekvens av olika arter eller andel ytor med viss påverkan
  - Medelfel, konfidensintervall (95%) och ev. även andra spridningsmått (medelfel som standard i samtliga fall där det är möjligt).
2. För de mer avancerade standardskattningarna såsom landskapsindex, indikatorer, framskrivningsmodeller och övergångsmatriser krävs det fler analyser och utvärderingar innan redovisningen kan beskrivas. För indikatorer och landskapsindex har vi påbörjat arbetet. För indikatorer pågår just nu utvecklingsarbete för småbiotops- och gräsmarksindikatorn inom ramen för LillNILS samt en myrindikator inom basNILS. Som ett exempel beskrivs nedan hur vi vill börja jobba med landskapsindex:
  - Enklare sammanställningar såsom tårtdiagram o dyl . (följa arealer baserade på rel. grova landskapsindelningar)
  - Urval av landskapsindex
  - Urval av övergångsmatriser
  - Habitat modeller

Vi börjar med enklare sammanställningar i ett första skede för att fortsätta bygga på med tiden. Vi behöver diskutera vidare olika klassificeringssystem och komma fram till vad vi vill presentera i första hand. Olika indelningar ger olika resultat och det är viktigt att jämföra lika med lika då

aspekter som minsta tolkningsenhet, områdets storlek och hur grov respektive finkornig klassificeringen är kan ge olika resultat i ovanstående sammanställningar.

Förslag till första presentation:

Vi väljer en enkel rättfram naturtypsklassificering till att börja med och använder denna över hela landet. Vi gör arealsammanställningar och väljer ut några index som vi anser vara intressanta och tämligen robusta och transparenta. Vi presentera en totalbild över hela Sverige där vi sen delar in i olika stratum för att kunna ge en bild över olika stratum och regioner. Detta innebär att vi får en grund att stå på och kan gå in mer i detalj och testa andra klassificeringar och specialare i ett senare skede.

- Medelfel, konfidensintervall (95%) och ev. även andra spridningsmått (medelfel som standard i samtliga fall där det är möjligt).

### **Avancerade skattningar:**

Avancerade skattningar genomförs av analytikerna. Avancerade skattningar kan med tiden komma att föras över till gruppen som utgör standardskattningar:

- Godtyckligt område
- NILS data tillsammans med olika typer och former av stöddata
- Framskrivningsmodeller
- Vissa övergångmatriser
- Vissa landskapsindex
- Vissa indikatorer

Även för dessa krävs mer analysarbete innan redovisningarna kan beskrivas och utvärderas av våra användare.

## **7 Redovisning av kvalitet i analysverktyg**

Vad som utgör tillräckligt god kvalitet i NILS måste definieras och kontinuerligt uppdateras. En första början till definition finns beskrivet i KIND-rapporten där kvalitetssäkring i framförallt datafångsten utvärderats.

För att vidmakthålla en god analyskvalitet behöver vi bl.a.:

- En kvalitetssäkrad databas - en förutsättning för att data skall hålla god kvalitet över tiden är att de komponenter som påverkar kvaliteten är kända och att det finns tydligt formulerade kvalitetsmål och rutiner för att kontinuerligt bibehålla, följa upp och öka kvaliteten i datafångst och datahantering. Vi vill ha låg och jämn personvariation vid inventerarnas bedömning, låg risk för misstag vid datainmatning och hög säkerhet vid

dataöverföring. Vidare vill vi ha lätthanterliga och funktionella databaser och att de fel som förekommer upptäcks och korrigeras innan analyserna utförs.

- Spara och kvalitetssäkra script för utplock så att de går att återanvända och kan tjäna som arbetslogg
- Låta variabler där vi känner till slutresultatet följa med i varje uträkning – ger omedelbar feedback att våra uträkningar stämmer
- Utveckla och följa väl inarbetade analysrutiner
- Gemensamma utbildningar för analytikerna främst men även för andra som analyserar data i NILS
- Få tillgång till statistikerkompetens som bl.a. sätter ramarna och utvecklar analyser, beräkningsscript och styr upp underliggande studiedesign för största möjliga styrka

Kvalitén på data i analysverktyget redovisar vi genom att alltid leverera skattningar tillsammans med variansberäkningar och storleken på samplet som analysen grundar sig på. Kontinuerliga kontrollinventeringar kommer att ge oss en möjlighet att ständigt förbättra det underliggande datats kvalitet samt göra en bedömning av hur ”bra” en variabel är dvs en variabel med stor spridning i bedömning mellan olika inventerare utgör ingen ”bra” variabel att göra analyser på utan måste först åtgärdas och förbättras.

## 8 Redovisning av medföljande metadata

### Metadata:

På datavärdsträffen 2009 var det fortfarande oklart vad som betecknas som metadata. Det ställs högre krav på datavärddar än Inspire/ISO 19115. På förslag är en metadatastandard för datavärddar, som dock inte är framtagna än. Eftersom beskrivning och standardisering av metadata fortfarande är under utveckling, behöver det finnas en viss nivå av flexibilitet.

### Syfte:

1. göra våra data sökbara
2. leverera information som behövs för att tolka data och resultat
3. leverera information som behövs för att analysera data

### Analysverktyget:

- Alla tillgängliga metadata skall vara tillgängliga



## 9 Arkivering

### Vad ska vi backa up och arkivera?

Analysverktyg:

Skattningarna kommer att utföras i realtid och kommer därför inte att sparas. Vissa uträkningar kommer att ligga färdigberäknade i databasen såsom:

- Provytearealer
- Linjekorsningskoordinater
- Landskapsindex och andra uträkningar som kräver annan programvara än vanliga statprogram och GIS

Detta skall backas up och arkiveras:

- Bibliotek för utplocksscript och beräkningsscript backas up kontinuerligt och arkiveras beständigt inom FO analysstöd
- Bibliotek för projektsammanställningar (utplocksscript, beräkningsscript, projektbeskrivningar mm) backas up och arkiveras inom FO analysstöd

### Hur ska vi göra back-up/arkivera?

Arbetsdokument/ pågående analyser lagras idag på en server där befintliga back-up rutiner är tillräckliga. Inom FO-analysstöd kommer vi att utveckla rutiner för arkivering av bibliotek för utplock, beräkningsscript och projektsammanställningar. För webbportalen kan det vara lämpligt med en separat server. Servrar driftas av IT-avdelningen, som ansvarar för backup-rutiner och säkerhet. Arkivlagen är ett styrande dokument som SLU måste förhålla sig till och som vi därför måste ta hänsyn till.

Syfte med kortsiktig backup: Kontinuerligt spara för säkerhets skull .

Syfte med långsiktig arkivering: För att ha kvar gamla versioner och kunna återskapa och återanvända.