

---

# **Arkade5 Documentation**

*Release 1.0.0*

**Arkitektum AS**

**okt. 19, 2018**



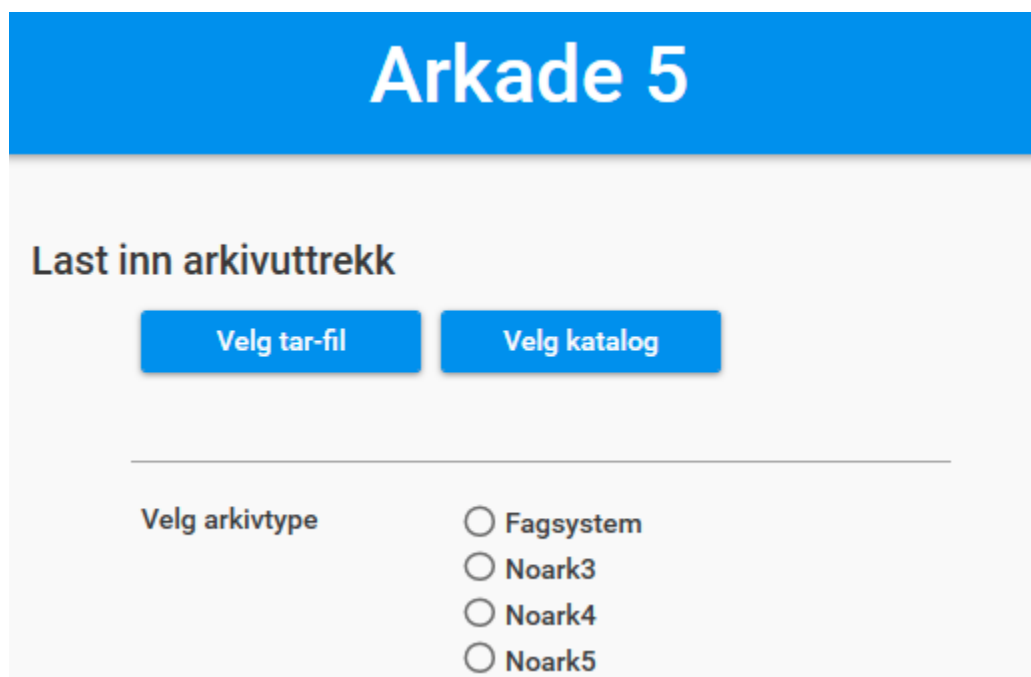
<b>1</b>	<b>Pressemelding</b>	<b>3</b>
<b>2</b>	<b>Innhold</b>	<b>5</b>
2.1	Installasjonsveiledning . . . . .	5
2.2	Brukerveiledning . . . . .	9
2.3	Systemdokumentasjon . . . . .	17
2.4	Versjonshistorikk . . . . .	23



- **Installer Arkade 5**



4. september 2017 lanserte Arkivverket versjon 1.0 av et nytt testverktøy - «Arkade 5».



Arkade vil etter hvert erstatte alle de testverktøy som Arkivverket har for å teste kvalitet av datasett generert fra arkivsystemer. Et viktig poeng er at Arkade kan teste så å si **alle** aktuelle varianter av arkivuttrekk, Noark standarder 3, 4 og 5 - pluss såkalte fagsystemer.

Arkade tester arkivuttrekk, produserer testrapporter og «pakker inn» data basert på gjeldende arkivstandarder for metadata. Testing av data baserer seg primært på Arkivverkets standard [ADDML](#) for beskrivelse av datastrukturer.

Arkade er en frittstående applikasjon som fritt kan lastes ned og enkelt installeres av alle. Programmet er svært enkelt i bruk. I prosjektfasen er det hentet innspill, kommentarer og idéer fra både kommunal og statlig sektor. Ved å senke terskelen for bruk av testverktøy håper vi å forenkle prosessene både hos arkivskapere og arkivinstitusjoner. Blant

annet vil feil og avvik kunne oppdages mye tidligere, noe som sparer både tid og kostnader for alle involverte. All kildekode og funksjonalitet i Arkade er lagt ut som åpen kildekode, så det vil også være mulig for eksterne å lage spesialtilpasninger.

Arkivverket vektlegger dessuten forvaltning og videreutvikling av programvaren. Det er allerede en løpende prosess for å legge inn mer funksjonalitet basert på registrerte ønsker under utvikling av versjon 1.0. Det vil også etableres brukerforum og tekniske støttefunksjoner, der man tidligere kanskje ikke har vært gode nok utad. Utvikling av Arkade 5 har krevd ett år med utvikling og testing. Prosjektet har vært styrt av Arkivverket, mens programvaren har vært utviklet eksternt av firmaet [Arkitektum AS](#). Samarbeidet med Arkitektum fortsetter, men forvaltning videre vil være Arkivverkets ansvar.

*«Arkade vil utgjøre en enorm tidsbesparelse ved testing av uttrekk, både for arkivskaper og depot.»* Solveig Heløe Olsen, Rådgiver, Interkommunalt Arkiv, Troms.



*Team Arkade, fra Arkitektum AS (A) og Arkivverket (AV): Fra venstre: Jørgen Tellnes (A), Henning Jensen (A), Joachim Fugleberg (AV), Hallstein Søvik (A), Erik Aaberg (AV), Terje Pettersen Dahl (AV), Tor Oskar Ova Johnsen (A), Jørgen Østengen Vik-Strandli (AV).*

*Foto: Benjamin Dehli (A)*



## 2.1 Installasjonsveiledning

### 2.1.1 Arkade 5

**Skrivebordsapplikasjon for Windows**

#### **Installer .NET**

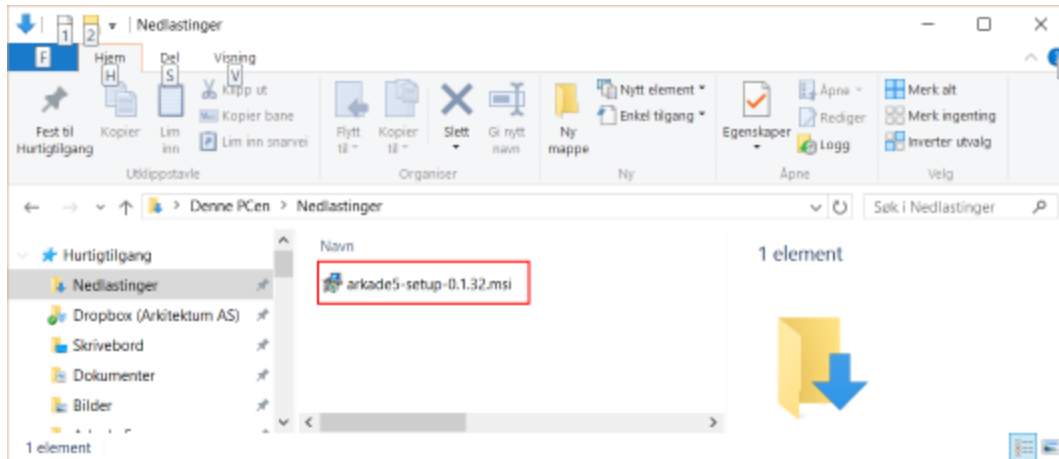
For å kjøre Arkade 5 må .NET Framework (minimum versjon 4.6.1) være installert.

Last ned siste versjon av [.NET Framework](#)

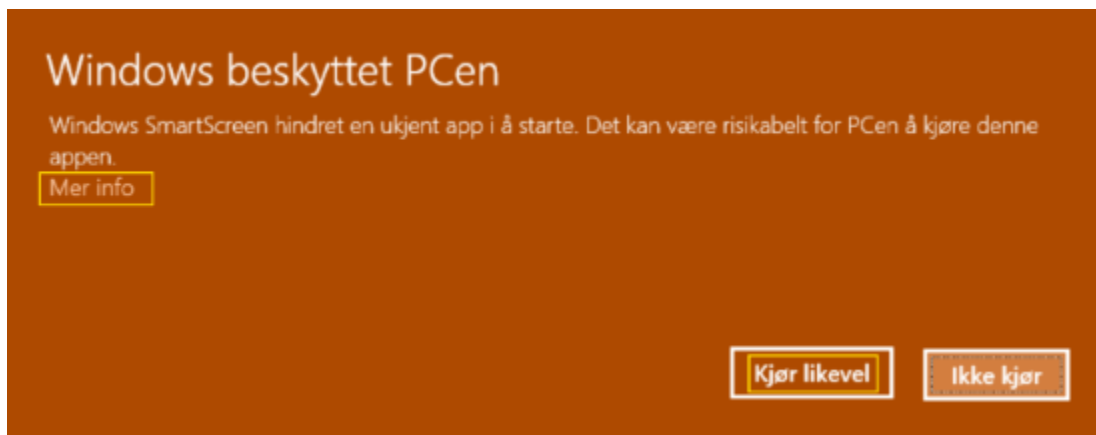
#### **Installer Arkade 5**

Last ned siste versjon av Arkade 5 - Velg filen Arkade5-<versjon>.msi

Start installasjonen ved å dobbeltklikke den nedlastede msi-filen.



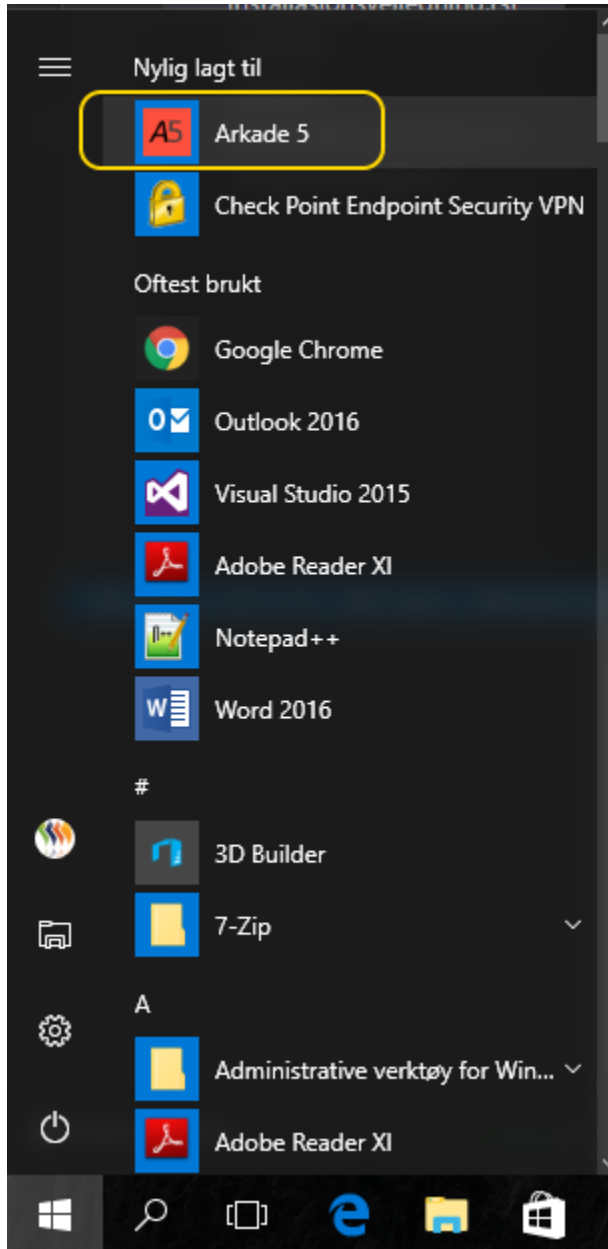
**Merk: Windows Smart Screen advarsel** Den følgende advarselen vil vises om Windows-maskinen har «Windows Smart Screen» satt på (Windows 10).



- Klikk på «Mer info»
- Klikk «Kjør likevel»

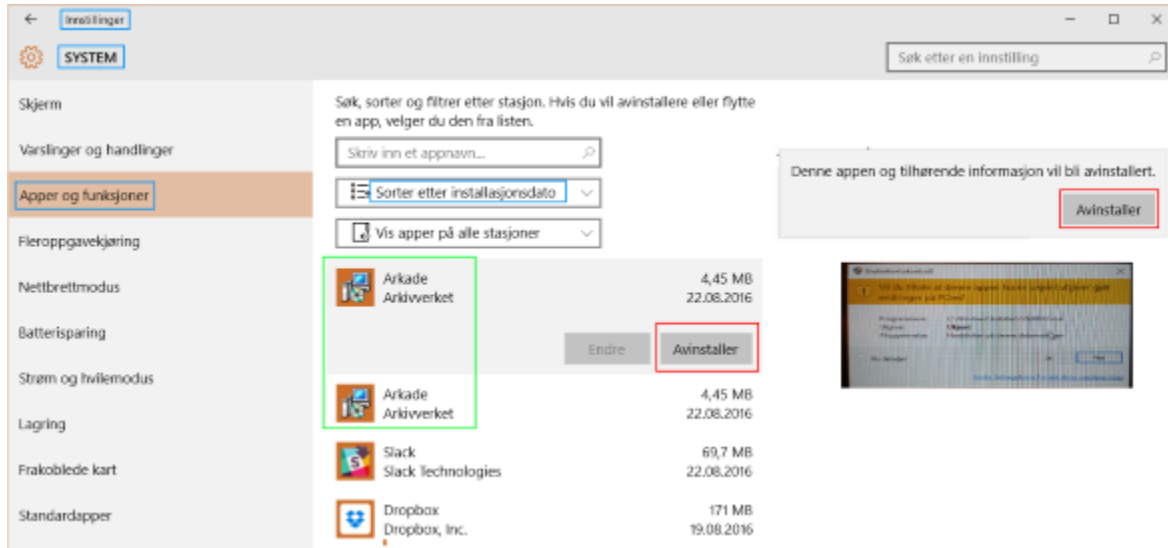
**Følg installasjonsveiviseren og aksepter alle de foreslåtte installasjonsvalgene.**

## Kjør programmet



- Start -> Alle apper
- Finn «Arkade 5» i applikasjons-listen
- Klikk på «Arkade 5» for å kjøre programmet

## Avinstallasjon av programmet (Windows 10)



- Klikk Start -> Instillinger -> System -> Apper og funksjoner
- Velg «Sorter etter installasjonsdato»
- Velg «Arkade» i listen over installerte programmer
- Klikk «Avinstaller»
- Klikk «Avinstaller» igjen i nytt vindu
- Klikk «Ja» på Brukerkontroll-advarselen fra Windows

### 2.1.2 Arkade 5 CLI

#### Frittstående kommandolinjegrensesnitt for Linux, macOS og Windows

*NB! Ved installasjon av Arkade 5 for Windows (beskrevet over) medfølger Arkade 5 CLI som filen Arkade.exe i installasjonskatalogen. På Windows behøver man ikke å følge anvisningene nedenfor med mindre man ønsker å installere Arkade 5 CLI som en frittstående applikasjon.*

#### Installer .NET Core

For å kjøre Arkade 5 CLI må .NET Core (minimum versjon 2.0) være installert.

Last ned siste versjon av .NET Core runtime for: [Linux](#), [macOS](#) eller [Windows](#).

#### Installer Arkade 5 CLI

Last ned siste versjon av Arkade 5 CLI - Velg filen Arkade5CLI-<versjon>.zip

Pakk ut den nedlastede zip-filen til der Arkade ønskes kjørt fra.

*Det spiller ingen rolle hvor i filsystemet Arkade 5 CLI kjøres fra; plassering for alle inn- og ut-data velges som parametre ved kjøring.*

Gjør filen arkade.sh kjørbær:

```
cd Arkade5CLI-<versjon>
chmod +x arkade.sh
```

Ingen ytterligere installasjon er nødvendig.

## Kjør Arkade 5 CLI

Se brukerveiledning for Arkade 5 CLI

### Avinstaller Arkade 5 CLI

- Slett katalogen Arkade5CLI-<versjon>
- Slett eventuelle gjenværende systemlogger

*Mindre enn 1 uke gamle system- og feillogger slettes ikke automatisk etter kjøring. [Les mer ...](#)*

## 2.2 Brukerveiledning

### 2.2.1 Arkade 5

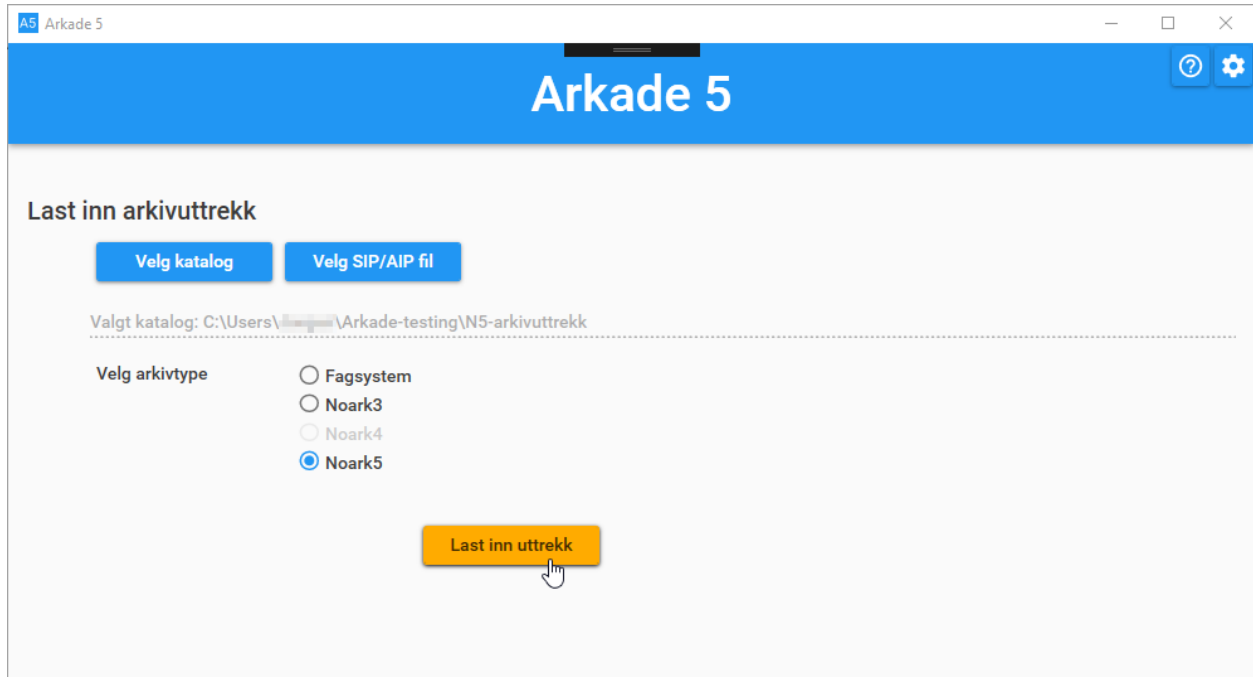
#### Skrivebordsapplikasjon for Windows

Arkade 5 brukes ved å lese inn et arkivuttrekk, utføre testing av uttrekket og/eller opprette en arkivpakke av uttrekket. Ved testing genereres det en utfyllende testrapport. Arkade 5 muliggjør også opprettelse/endring av metadata for arkivuttrekk.

#### Oppstart/avslutning

Ved oppstart av Arkade åpnes innlastingsvinduet. Dersom et tilgjengelig område for midlertidige filer ikke allerede er definert, vil Arkade først be om at dette blir oppgitt (se innstillinger). Arkade avsluttes ved å lukke hovedvinduet. Ved avslutning igangsettes sletting av midlertidige filer.

## Innlastingsvinduet



## Innlasting

Velg og last inn arkivuttrekket som skal behandles:

1. Klikk på knappen «Velg katalog» dersom uttrekket er en ordinær fil-/mappestruktur. I tilfelle velges katalogen som inneholder arkivbeskrivelse-filen (addml.xml, arkivuttrekk.xml eller NOARKIH.xml). Klikk på knappen «Velg SIP/AIP-fil» dersom uttrekket er en AIP- eller SIP-struktur pakket som en tar-fil.
2. Oppgi arkivtype for det valgte uttrekket. Arkade 5 støtter typene «Fagsystem», «Noark 3» og «Noark 5».
3. Klikk på knappen «Last inn uttrekk». Det valgte uttrekket vil lastes inn og åpnes i testvinduet.

## Testvinduet



Øverst i testvinduet vises:

- Full filsti for det valgte uttrekket
- En unik identifikator (UUID), generert for den gjeldende behandlingen av det valgte uttrekket
- Valgt arkivtype
- Hvilken fil som under testing prosesseres i øyeblikket
- Løpende informasjon om testkjøring\*

\*For uttrekk av typen Noark 5 vises antall prosesserte XML-elementer. For uttrekk basert på en ADDML-fil vises antall prosesserte filer og antall prosesserte poster.

## Testkjøring

Klikk på knappen «Start testing» for å starte testkjøring på det valgte uttrekket. Testkjøringen vil vare fra noen minutter til mange timer, avhengig av uttrekkets størrelse.

Under testkjøring vil det, i den nedre delen av vinduet, vises meldinger om innlesing, ev. strukturelle feil/mangler ved uttrekket, rapportgenerering og fullført testing.

## Testrapport

Etter fullført testing vil en rapport i HTML-format bli generert. Klikk på knappen «Vis rapport» for å åpne den i en nettleser (den vil tilpasse seg gjeldende skjermflate). Ved opprettelse av arkivpakke inkluderes alltid testrapporten. Dersom den (i tillegg) skal tas vare på et annet sted, lagres den ved hjelp av nettleseren. Med rapporten åpen kan dette, i de fleste nettlelere, gjøres ved å taste Ctrl+s.



ARKIVVERKET  
RIKSARKIVET OG STATSARKIVENE

## Testrapport

### Testsammendrag

Uuid	d42dae63-e1c8-4f0f-b63e-5d3341fe8266
Arkivskaper	Lewis Caroll, Bob Armstrong
Arkivperiode	1863-03-27 - 1865-12-11
Systemnavn	Test-arkivuttrekk
Systemtype	Sakarkiv (Noark-5)
Arkivtype	Noark5
Tidspunkt for testing	18. oktober 2018
Antall filer som er prosessert	1
Antall avvik funnet	3

### Avvik

<a href="#">N5.03 – Validering av XML i henhold til skjema</a>	1
<a href="#">N5.30 – Dokumentfilers sjekksummer</a>	1
<a href="#">N5.32 – Refererte dokumenters eksistens</a>	1

## Analyser og kontroller

### N5.02 – Validering av sjekksummer

Type: Strukturkontroll

Resultater

Ingen avvik funnet.

### N5.03 – Validering av XML i henhold til skjema

Type: Strukturkontroll

Resultater

Lokasjon	Melding
arkivstruktur.xml	Linje 4: Elementet arkiv i navneområdet <a href="http://www.arkivverket.no/standarder/noark5/arkivstruktur">http://www.arkivverket.no/standarder/noark5/arkivstruktur</a> har ugyldig underordnet element systemID i navneområdet <a href="http://www.arkivverket.no/standarder/noark5/arkivstruktur">http://www.arkivverket.no/standarder/noark5/arkivstruktur</a> . Elementet liste over mulige elementer tittel i navneområdet <a href="http://www.arkivverket.no/standarder">http://www.arkivverket.no/standarder</a>



## Arkivpakkegenerering

Klikk på knappen «Opprett pakke» for å lage en arkivpakke (AIP/SIP) av uttrekket. Dette åpner arkivpakkevinduet der valg for pakken kan gjøres før den opprettes.

*Det er mulig å opprette en arkivpakke uten først å utføre testing av det aktuelle arkivuttrekket. En slik pakke vil ikke inneholde noen testrapport.*

## Nytt uttrekk / ny kjøring

Ved klikk på knappen «Ny kjøring» avsluttes pågående arkivbehandling og Arkade returnerer til innlastingsvinduet. Dersom det inneværende arkivuttrekket endres, f.eks. som følge av feil/mangler vist i testrapporten, må uttrekket lastes inn på nytt (og ev. tester kjøres på nytt) før knappen «Opprett pakke» oppretter en pakke som inneholder endringene (og knappen «Vis rapport» åpner en gyldig testrapport). Uttrekket lastes inn på nytt ved å klikke «Ny kjøring» (eller ved å starte Arkade på nytt).

*NB! Skal det opprettes en arkivpakke som inkluderer resultatene fra inneværende testkjøring, må dette gjøres før «Ny kjøring» klikkes (eller Arkade avsluttes).*

## Arkivpakkevinduet

Arkade 5

# Arkade 5

## Registrer metadata og opprett pakke

\* = Feltet er påkrevet for å skape en gyldig arkivpakke Hent eksterne metadata

**Uttreksdato**

**Merkelapp**

Standard  Egendefinert

---

Velg pakketype

SIP  AIP

Opprett pakke Ny kjøring

Før arkivpakke opprettes registreres aktuelle metadata. For å skape en gyldig arkivpakke må påkrevde felter (merket \*) utfylles. Dersom det aktuelle arkivuttrekket i utgangspunktet var en arkivpakke (AIP/SIP) med allerede definerte metadata, vil Arkade forsøke å lese disse inn og forhåndsutfylle feltene i arkivpakkevinduet. Ved å klikke knappen «Last inn eksisterende metadata» kan det velges en fil med forhåndsdefinerte metadata, enten en METS-fil (f.eks. info.xml eller dias-mets.xml fra tidligere produsert pakke) eller metadata på JSON-format (f.eks. generert med Arkade CLI) for automatisk utfylling av feltene. Når pakken opprettes skrives den utfylte informasjonen til filen dias-mets.xml som legges ved i arkivpakken. Metadataene skrives også til filen info.xml som legges utenfor, på samme nivå som, arkivpakken.

I nedre del av vinduet velges ønsket pakketype, SIP eller AIP.

Når ønskede metadata er oppgitt, klikkes knappen «Opprett pakke». Dette åpner et dialogvindu for valg av pakkens plassering. Ved valgt plassering opprettes arkivpakken.

Arkivpakken vil opprettes som en tar-fil og filnavnet vil være UUID-en som er generert for den gjeldende arkivbehandlingen: *{uuid}.tar*. Pakken og tilhørende info.xml-fil plasseres i en katalog *Arkadepakke-{uuid}*. Når alt er ferdig generert, vises denne katalogen på den valgte plasseringen.

*TIPS: Så lenge arkivpakkevinduet ikke forlates, kan metadata endres og «Opprett pakke» klikkes på nytt. Velges samme pakkeplassering, overskrives foregående pakke og info.xml med oppdaterte metadata. Sørg bare for at filene som skal overskrives ikke er opptatt, f.eks. ved at de er åpnet i andre programmer.*

NB! Knappen «Ny kjøring» avslutter gjeldene arkivbehandling.

## Innstillinger

### Prosesseringsområde

Under kjøring benytter Arkade et filområde til plassering av midlertidige filer fra arkivprosessering, system- og feillogger samt andre systemfiler. Plassering for prosesseringsområdet velges av bruker som en katalog i filsystemet og må være definert før arkiv kan behandles. Plasseringen som velges må være egnet med tanke på størrelse, tilgjengelighet og personvern. Størrelsen må være minst den av alle uttrekk som skal behandles under samme kjøring i tillegg til plass for systemfiler. Ved avslutning av Arkade igangsettes sletting av midlertidige filer og gamle loggfiler. Plasseringen av prosesseringsområdet kan når som helst endres fra innstillingsvinduet. Ved endring av plassering, igangsettes sletting av opprinnelig prosesseringsområde. Ny plassering vil tas i bruk neste gang Arkade startes.

---

## 2.2.2 Arkade 5 CLI

### Kommandolinjegransnitt for Linux, macOS og Windows

Arkade 5 CLI tester og lager en arkivpakke av et arkivuttrekk med én kommando. Plasseringer for arkivuttrekket som skal behandles, metadata, ferdig arkivpakke og testrapport samt midlertidige filer og systemlogger, oppgis alle som parametre til kommandoen.

Se installasjonsveiledning for Arkade 5 CLI

### Kjøring

Arkade CLI kjøres med skriptet `arkade.sh` som medfølger installasjonspakken.

(Arkade CLI som medfølger en installasjon av Arkade 5 for Windows, kjøres med `Arkade.exe`)

For å kjøre skriptet slik som vist i eksemplene (`./arkade.sh`), naviger til installasjonskatalogen:

```
cd Arkade5CLI-<versjon>
```

(Ved å tilgjengeliggjøre skriptet i PATH kan det kjøres med `arkade` fra hvor som helst i filsystemet)

## Parametre

Bruk `--help` for å vise en oversikt over parametre:

```
./arkade.sh --help
```

```
-a, --archive           Archive directory or file (.tar) to process.
-t, --type             Archive type, valid values: noark3, noark5 or fagsystem
-m, --metadata-file    File with metadata to add to package.
-g, --generate-metadata-example Generate example metadata file. Argument is output file name.
-p, --processing-area  Directory to place temporary files and logs.
-o, --output-directory Directory to place created package and test report.
-s, --skip             Optional. Valid values: testing, packing
-i, --information-package-type Optional. Valid values: SIP, AIP. Default: SIP
--help                Display this help screen.
--version             Display version information.
```

Alle parametre foruten `--help` og `--version` kan oppgis i kortform og må etterfølges av aktuell verdi f.eks. `--type Noark5` eller `-t Noark5`

Parametrenes rekkefølge er likegyldig.

*NB! Alle kataloger som oppgis som må eksistere på forhånd.*

## Prosesseringsområde (CLI)

Under kjøring benytter Arkade et filområde til plassering av midlertidige filer fra arkivprosessering, system- og feillogger samt andre systemfiler. Plassering for prosesseringsområdet velges av bruker som en katalog i filsystemet og må oppgis med parameteret `--processing-area` eller `-p`. Plasseringen som velges må være egnet med tanke på størrelse, lese-/skrivehastighet og personvern. Størrelsen må være minst den av uttrekket som skal behandles, i tillegg til noen MB for systemfiler. Etter en arkivprosessering vil Arkade igangsette sletting av midlertidige filer og gamle loggfiler. Av tekniske hensyn fjernes ikke automatisk mindre enn 1 uke gamle loggfiler. Vær oppmerksom på at loggfiler kan inneholde personopplysninger. Det anbefales å bruke den samme plasseringen for prosesseringsområdet for hver kjøring. Slik kan Arkade automatisk fjerne loggfiler som er mer enn 1 uke gamle. Sørg ellers for manuelt å fjerne loggfilene.

## Eksempel på bruk

Plasseringer og navn på kataloger for inn- og utdata er valgfrie. I eksemplene som følger brukes katalogene `tmp/` og `output/` i hjemmekatalogen `~/`. Opprett de samme katalogene dersom du vil bruke eksempelkommandoene i denne veiledningen slik som de står.

```
mkdir ~/tmp
mkdir ~/output
```

### Metadata

Ved arkivprosessering må det refereres til en fil med metadata på JSON-format eller METS-format. Arkade CLI kan generere en JSON-fil med eksempel-metadata som så kan redigeres.

Oppgi full filsti for metadatafilen som skal opprettes, med `--generate-metadata-example` eller `-g`. Det må også oppgis et prosesseringsområde med `--processing-area` eller `-p`.

I kommandoen under gis metadatafilen navnet `metadata.json` og plassering i hjemmekatalogen `~/`. Prosesseringsområdet er satt til katalogen `~/tmp/`.

```
./arkade.sh -g ~/metadata.json -p ~/tmp/
```

*Obligatoriske parametre ved metadatagenerering:* `--generate-metadata-example--processing-area`

### Prosessering av arkivuttrekk

*Arkivuttrekket som skal prosesseres kan foreligge som en ordinær mappestruktur eller en AIP/SIP-struktur pakket som en tar-fil. Parameterværdien for `--archive/-a` settes til hhv. katalogen som inneholder arkivbeskrivelse-filen (`addml.xml`, `arkivuttrekk.xml` eller `NOARKIH.xml`) eller til tar-filen.*

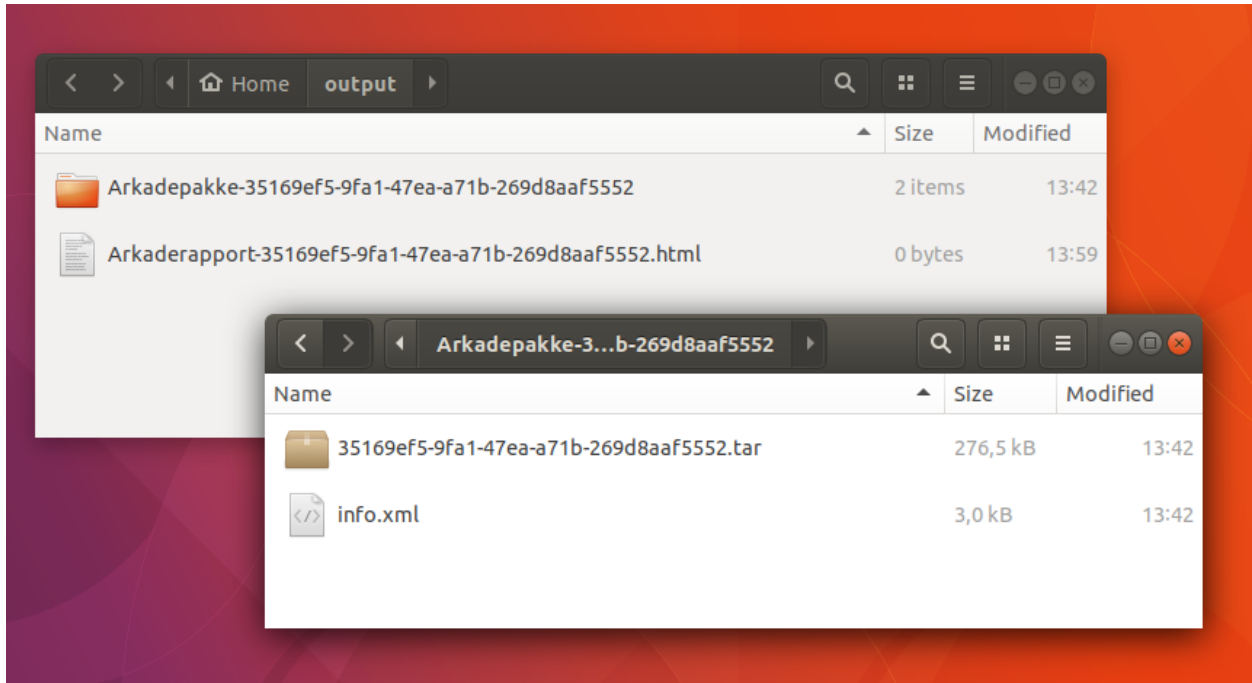
I eksempelkommandoen vist under leses det inn et arkivuttrekk fra katalogen `~/N5-arkivuttrekk/`. Arkivuttrekkets type er oppgitt til `noark5`. Metadata for uttrekket leses fra filen `~/metadata.json/`. Katalog for plassering av prosesseringsområdet er satt til `~/tmp/` og katalog for resulterende data er satt til `~/output/`.

```
./arkade.sh -a ~/N5-arkivuttrekk/ -t noark5 -m ~/metadata.json -p ~/tmp/ -o ~/output/
```

*Obligatoriske parametre ved arkivprosessering:* `--archive--type--metadata-file--processing-area--output-dir`

### Resulterende data

Kommandoen over produserer en arkivpakke (SIP) som en tar-fil, med tilhørende `info.xml` på METS-format, samlet i en katalog. Ved siden av katalogen plasseres en testrapport på HTML-format.



For hver prosessering genereres en unik UUID som bl.a. brukes i fil- og katalognavn for resultatene.

## Alternativer

Dersom det ønskes kun testing eller kun pakking av et arkivuttrekk, kan den ene av delprosessene utelates med `--skip` eller `-s` etterfulgt av parameterverdien `testing` eller `packing`. Oppgis `-s testing` produseres en utestet pakke, følgelig uten testrapport. Oppgis `-s packing` utføres kun testing og kun en testrapport blir produsert.

Med `--information-package-type` eller `-i` etterfulgt av parameterverdien `SIP` eller `AIP` kan det velges om det skal produseres en pakke av typen `SIP` eller av typen `AIP`. Når dette parameteret ikke er oppgitt (som i eksempelet over) produseres en `SIP` som standard.

## 2.3 Systemdokumentasjon

### 2.3.1 Source code

The source code is located at the GitHub-repository: <https://github.com/arkivverket/arkade5/>

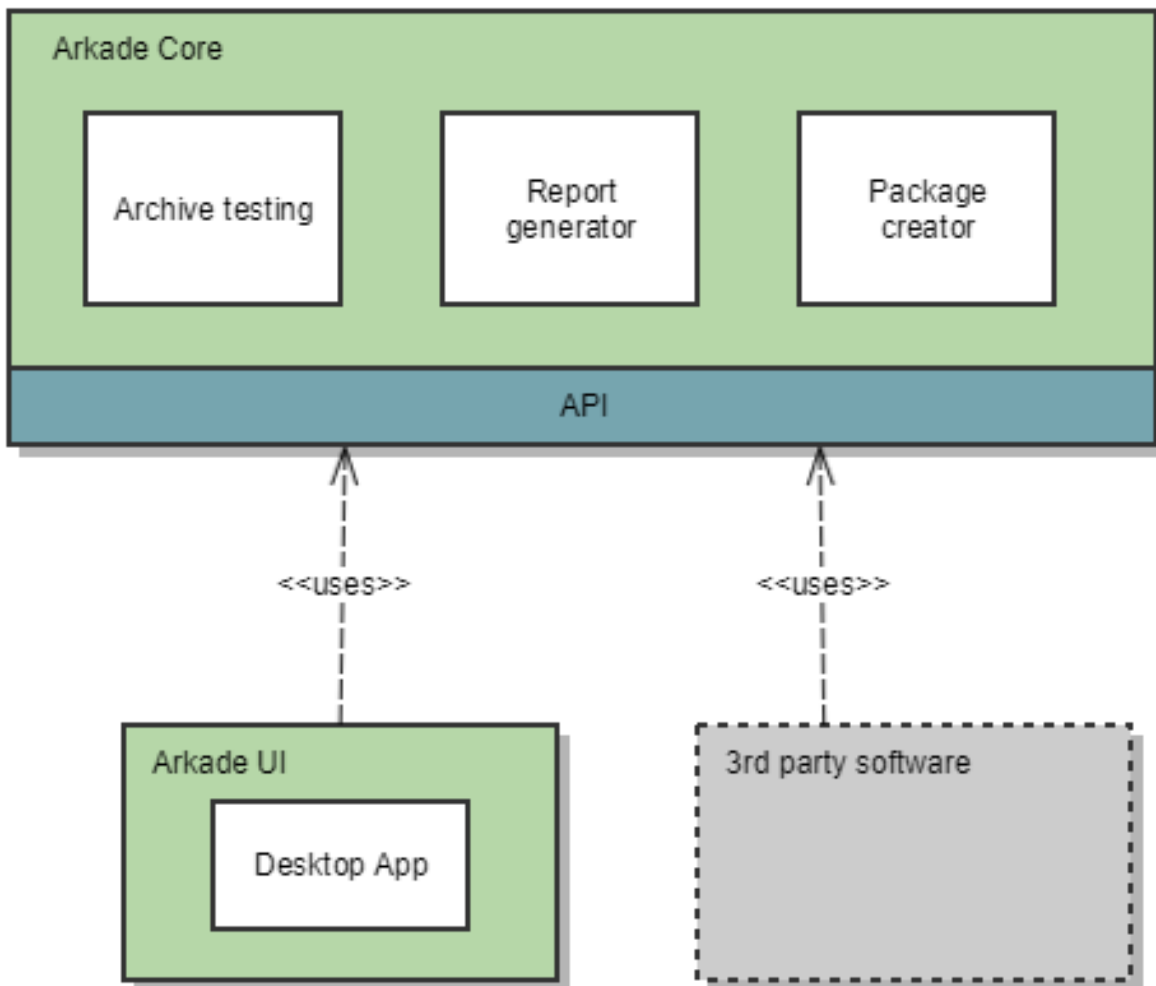
Arkade is developed with .Net and C#. The solution-file (.sln) is compatible with Visual Studio 2015 and above.

### Overview

Arkade provides mainly three different functions:

- Archive testing
- Report generator
- Package creator

These functions are exposed in the API and the graphical user interface project is also using the API-class to interact with the core functions.



Below is a brief description of each project in the solution.

### Arkivverket.Arkade

This is the core library with functions for testing archive extractions, generating reports and creating SIP/AIP-packages.

The most notable classes in the core project are the test engines, package creator and report generator:

- Arkivverket.Arkade.Core.Noark5.Noark5TestEngine
- Arkivverket.Arkade.Core.Addml.AddmlDatasetTestEngine
- Arkivverket.Arkade.Core.InformationPackageCreator
- Arkivverket.Arkade.Report.HtmlReportGenerator

A short description of the packages in the core project:

**Core** - Domain classes

**ExternalModels** - Classes generated from xml schemas

**Identify** - Identification classes for reading and identifying an archive extraction

**Logging** - Classes related to logging of events during testing

**Metadata** - Contains classes related to creating metadata files for archive extractions

**Report** - Classes for generating test reports

**Resource** - Various resource files, language files, images etc.

**Tests** - Contains test classes for testing archive extractions

**Util** - General utilities

## Arkivverket.Arkade.UI

This project provides the graphical user interface of the Arkade 5 software. It is based on WPF, Windows Presentation Foundation. Together with WPF, the application uses the [Prism](#) library for creating a loosely coupled, maintainable and testable XAML application.

[Autofac](#) is used as a dependency framework. Bootstrapping of the applications happens in **Bootstrapper.cs**. It is based on the bootstrapper provided by Prism and it loads the Autofac-module provided by the Arkade core library.

The design and layout is based on Google's [Material Design](#). This has been implemented with the help of the [MaterialDesignThemes-library](#). Note that the user interface is only inspired by the material design, not necessary strictly following it in every situation.

## Arkivverket.Arkade.Test

This project contains the unit tests and other tests classes for the project. Unit tests are created with xUnit.

## Setup

This is the setup project for creating installation binaries. You need the [Wix-toolset](#) to be able to use the Setup-project.

## Signing the installation file

In order to sign the msi file, you need the **signtool.exe** on your computer. This can be installed together with the Visual Studio. The ClickOnce Publishing package contains this tools.

Signing of the installation file is done by the continuous integration server. Signing is performed with a certificate provided by Arkivverket. The following command is run to sign the installation file:

```
"C:\Program Files (x86)\Windows Kits\8.1\bin\x64\signtool.exe" sign /f PATH_TO_
↪CERTIFICATE_FILE.pfx /p CERTIFICATE_PASSWORD src\Setup\bin\Release\Setup.msi
```

## Sample.ConsoleApp

This is a sample application, which demonstrates the use of the Arkade API.

## Porting to other platforms

For now the application is only developed for use on the Microsoft Windows platform. However, with the new [.Net Core project](#) from Microsoft, a cross platform application should be possible when the framework and tools has matured. The .Net Core platform allows running .net applications on linux and mac in addition to windows.

The Arkade project has few external dependencies and it should be possible to either update them to newer versions when they are compatible with .net core or replace them with other compatible libraries. Currently there are only three external libraries in use:

- [SharpZipLib](#)
- [Autofac](#)
- [Serilog](#)

In addition there are some of the .net packages that has been restructured, deprecated or removed that need to be fixed before the application is fully cross platform compatible.

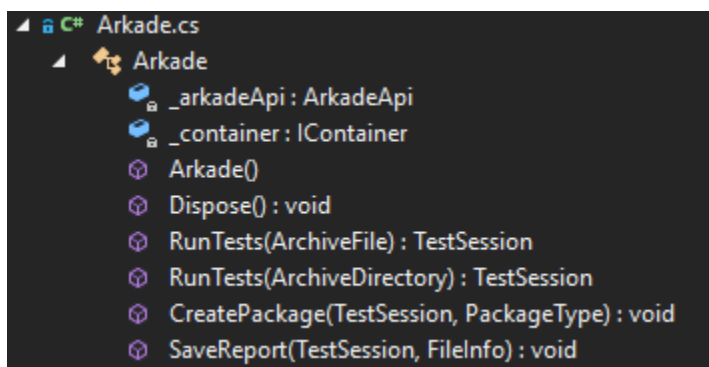
The graphical user interface can be a challenge to port, currently Microsoft has no plans for porting the Windows Presentation Framework to other platforms. This means that for creating a linux desktop app, you might have to recreate the user interface with another graphics library. Preferably a library that works on both linux and mac.

Some useful links regarding porting to .net core: \* <https://blogs.msdn.microsoft.com/dotnet/2016/02/10/porting-to-net-core/> \* <https://marketplace.visualstudio.com/items?itemName=ConnieYau.NETPortabilityAnalyzer>

### 2.3.2 Arkade API

The Arkade project provides API-classes for simplified use of the core functionality. There are two API-classes included: `Arkade.cs` and `ArkadeApi.cs`. They are located inside the namespace `Arkivverket.Arkade.Core`. Both classes provides the same functionality, the difference is that `Autofac` is used for dependency injection in the `Arkade` class. The `ArkadeApi` class must be instantiated manually. There is an `Autofac` module that can be used, `Arkivverket.Arkade.Util.ArkadeAutofacModule`, if the client software already is using `Autofac` for dependency injection.

This is the signature of the Arkade API class:



There are two `RunTests` methods that runs for a given archive, either from a directory structure or a SIP/AIP package file (.tar). After the tests are run, the api returns a `TestSession`. The `TestSession` class contains all necessary information for creating a package with tests results or generating a report.

A simple test run may look like this:

```

var arkade = new Arkade();
var testSession = arkade.RunTests(ArchiveFile.Read("c:\\tmp\\ExampleArchive.tar",
↪ ArchiveType.Noark5));
  
```

(continues on next page)



(continued from previous page)

```
arkade.SaveReport(testSession, new FileInfo("c:\\tmp\\TestReport.html"));
arkade.CreatePackage(testSession, PackageType.SubmissionInformationPackage);
```

The **TestSession** class contains various information about the testing that has been done. The **TestSuite** property contains a list of all tests that has been run and their results.

See the sample project (Sample.ConsoleApp) in the source code, for a complete example that runs testing on a Noark5 archive.

### 2.3.3 ADDML

Arkade is built to support ADDML version 8.2.

List of supported ADDML processes:

- A.01 Analyse\_CountRecords - Antall poster
- A.02 Analyse\_CountChars - Antall tegn
- A.03 Analyse\_FindExtremeRecords - Lengste og kortest post
- A.04 Analyse\_CountRecordDefinitionOccurrences - Antall poster for posttype
- A.05 Analyse\_AllFrequencyList - Koder brukt i datasettet
- A.06 Analyse\_CrossTable
- A.07 Analyse\_CountNULL - Antall null-verdi for felt
- A.08 Analyse\_FindExtremeValues - Lengste og korteste verdi for felt
- A.09 Analyse\_FindMinMaxValue - Laveste og høyeste verdi for felt
- A.10 Analyse\_FrequencyList - Antall forekomster av verdier for felt
- A.11 Control\_AllFixedLength - Kontroll av oppgitt lengde for alle posttyper
- A.12 Control\_NumberOfRecords - Antall og kontroll av antall poster
- A.13 Control\_FixedLength - Kontroll av oppgitt lengde for posttype
- A.14 Control\_NotUsedRecordDef - Kontroll om posttype benyttes
- A.15 Control\_Key - Kontroll av unik nøkkel
- A.16 Control\_ForeignKey - Kontroll av fremmednøkkel
- A.17 Control\_MinLength - Kontroll av oppgitt minste lengde
- A.18 Control\_MaxLength - Kontroll av oppgitt største lengde
- A.19 Control\_DataFormat - Kontroll av oppgitt dataformat for felt
- A.20 Control\_NotNull - Kontroll av null-verdier i felt
- A.21 Control\_Uniqueness - Kontroll av unike verdier i felt
- A.22 Control\_Codes - Koder som benyttes i felt
- A.32 Control\_Birthno - Kontroll av fødselsnummer i felt
- A.33 Control\_Organisationno - Kontroll av organisasjonsnummer i felt
- A.34 Control\_Accountno - Kontroll av kontonummer i felt
- A.35 Control\_Date\_Value - Kontroll av dato i felt

- A.36 Control\_Boolean\_Value - Kontroll av boolsk verdi i felt

## 2.3.4 NOARK 5

Arkade supports the NOARK5 standard.

List of implemented Noark5 Tests:

- N5.02 - Kontroll av sjekk-summene for filene arkivuttrekk.xml og addml.xsd
- N5.03 - Kontroll av sjekk-summene for XML-filene og XML-skjemaene i avleveringspakken
- N5.04 - Antall arkiver i arkivstrukturen
- N5.05 - Antall arkivdeler i arkivstrukturen
- N5.06 - Arkivdelen[e]s status i arkivstrukturen
- N5.07 - Antall klassifikasjonssystemer i arkivstrukturen
- N5.08 - Antall klasser i arkivstrukturen
- N5.09 - Antall klasser uten underklasser eller mapper i det primære klassifikasjonssystemet i arkivstrukturen
- N5.10 - Antall mapper i arkivstrukturen
- N5.11 - Antall mapper for hvert år i arkivstrukturen
- N5.12 - Kontroll på at mappene bare er knyttet til klasser uten underklasser i arkivstrukturen
- N5.13 - Antall mapper som er klassifisert med hver enkelt klasse i arkivstrukturen
- N5.14 - Antall mapper uten undermapper eller registreringer i arkivstrukturen
- N5.15 - Saksmappenes status i arkivstrukturen
- N5.16 - Antall registreringer i arkivstrukturen
- N5.17 - Antall forskjellige journalposttyper i arkivstrukturen
- N5.18 - Antall registreringer for hvert år i arkivstrukturen
- N5.19 - Kontroll på at registreringer bare er knyttet til klasser uten underklasser i arkivstrukturen
- N5.20 - Antall registreringer som er klassifisert med hver enkelt klasse i arkivstrukturen
- N5.21 - Antall registreringer uten dokumentbeskrivelse i arkivstrukturen
- N5.22 - Journalpostenes status i arkivstrukturen
- N5.23 - Antall dokumentbeskrivelser i arkivstrukturen
- N5.24 - Antall dokumentbeskrivelser uten dokumentobjekt i arkivstrukturen
- N5.25 - Dokumentbeskrivelsenens status i arkivstrukturen
- N5.26 - Antall dokumentobjekter i arkivstrukturen
- N5.27 - Start- og sluttdato for dokumentene i arkivstrukturen
- N5.28 - Antall dokumentfiler i arkivuttrekket
- N5.29 - Antall dokumenter i arkivuttrekket fordelt på dokumentformat
- N5.30 - Kontroll av sjekk-summen for hver dokumentfil i arkivuttrekket
- N5.32 - Kontroll på om dokumentobjektene i arkivstrukturen refererer til eksisterende
- N5.33 - Kontroll på at det ikke finnes dokumentfiler i arkivuttrekket som mangler referanse fra dokumentobjektet

- N5.34 - Antall dokumentfiler som blir referert til av flere enn ett dokumentobjekt
- N5.35 - Antall saksparter i arkivstrukturen
- N5.36 - Antall merknader i arkivstrukturen
- N5.37 - Antall kryssreferanser i arkivstrukturen
- N5.38 - Antall presedenser i arkivstrukturen
- N5.39 - Antall korrespondanseparter i arkivstrukturen
- N5.40 - Antall avskrivninger i arkivstrukturen
- N5.41 - Antall dokumentflyter i arkivstrukturen
- N5.42 - Eventuelt - antall skjerminger i arkivstrukturen
- N5.43 - Eventuelt - antall graderinger i arkivstrukturen
- N5.44 - Eventuelt - antall kassasjonsvedtak i arkivstrukturen
- N5.45 - Eventuelt - antall utførte kassasjoner i arkivstrukturen
- N5.46 - Eventuelt - antall konverterte dokumenter i arkivstrukturen
- N5.47 - Kontroll av system-identifikasjonene i arkivstrukturen
- N5.48 - Kontroll av referansene til arkivdel i arkivstrukturen
- N5.51 - Kontroll av referansene til sekundær klassifisering i arkivstrukturen
- N5.59 - Antall journalposter i arkivuttrekket
- N5.60 - Start- og sluttdato i arkivuttrekket
- N5.61 - Antall endringer i endringsloggen
- N5.62 - Kontroll av referansene i endringsloggen

## 2.4 Versjonshistorikk

Se full versjonshistorikk for Arkade 5 på [GitHub](#)