Helsedirektoratet

NPK - Teknisk dokumentasjon

Versjonslogg

Versjon	Versjonsdato	Gyldig for NPK-versjon (f.o.m.)	Kommentar
2.5	21.1.2016	1.9.109	Første ordinære versjon.
2.21	16.6.2016	1.9.132	Oppdatert med informasjon relatert til ny grupperingskomponent for Modul 8.
2.22	21.10.2016	2.0.82	Oppdatering vedr. bruk av ny DRG grouper.
2.41	16.11.2016	2.0.112	Oppdatert med informasjon om endring av logikk rundt regelgyldighetsperiode, sletteskript for resultater, konsistens-sjekk av definisjonsdata og endringer rundt integrasjon med NPK.
3.1	02.12.2016	2.0.122	Har lagt inn at filen «Regelgyldighetsperiode.txt» ikke lenger er en del av installasjonspakken og at den må fjernes fra konfigurasjonsmappen hvis man ikke skal overstyre regler.
3.3	16.12.2016	2.0.127	Lagt til installasjonsveiledning for NPK.ForCSVData.
3.4	01.03.2017	2.0.132	Oppdatert sletteskript for resultat
3.5	13.10.2017	3.0.17286.01	Oppdatert i henhold til nye felter i NPK databasen og ny logikk for å angi hvor konfigfilene skal leses fra. Feilretting i Vedlegg A (forveksling av variablene tilSted og utTilstand)
3.6	23.10.2017	3.0.17286.01	Fjernet NPK.ForNPRMelding og oppdatert informasjon om NPK.ForCSVData.
3.7	12.12.2017	3.0.17341.01	Oppdatert i forhold til ny tabell «HReseptOppgjørsdata» og nytt felt i Episode: «InnmåteHastegrad»
3.8	23.08.2018	3.0.18235.03	Feilretting knyttet til beregningsregel 1.14.
3.9	05.10.2018	4.0.	Oppgradert versjon av .Net Framework fra 4.5.1. til 4.6.1.
4.0	30.11.2018	4.1.18334.02	Oppdatert vedr. nye moduler for Tjenesteforløp
4.1	16.12.2019	5.1.19303.01	Oppdater vedr. endringer i moduler for Særtjeneste. Utvidelse av muligheter ved integrasjon mot NPK vha NPK.Start.

4.2	19.08.2020	5.3.20218	Lagt inn detaljert kolonne oversikt for input, resultat og konfigurasjonstabeller i kapittel 5.3. Oppdatert SQL skript i vedlegg D og E. Oppdatert kapittel 7.
4.3	10.09.2020	6.0	Lagt til nye felter i tjenesteforløp tabellen.
4.4	17.09.2020	6.0	Lagt til manglende felt «AntallSærtjenesterISF» i dbo.Tjenesteforløp tabellen.
4.5	28.09.2020	6.0	Lagt til nye felt i tabellene dbo.Særtjeneste, dbo.Tjenesteforløp og dbo.DRGBeregningResultat
4.6	15.10.2020	6.0	Rettet en feil i kapittel 5.3.2 dbo.Tjenesteforløp. FK var forskjøvet.
4.7	19.10.2020	6.0	Fjernet databasefelter som var plassert feil i dbo.Opphold.Poeng tabellen. (Kapittel 5.3.2)
4.8	03.09.2021	7.0	Oppdatert i hht fjerning av Systemdokumentasjon og lagt til tabellen Kommune under Brukerkonfigurasjon og nytt felt i dbo.EpisodeResultat
4.9	14.10.2021	7.0	Oppdatert kap. 5.1 i forhold til Kommuneliste i Brukerkonfigurasjon.
5.0	25.10.2021	7.0	Lagt inn beskrivelse av fremmednøkler.
5.1	14.12.2021	7.0.21347.1	Mindre teknisk endring vedrørende standard konfigurasjonsmappesti for brukerkonfigurasjon.
6.0	18.10.2022	8.0.22286.1	Oppdatert versjon for ISF 2023 med nye felt i dbo.Opphold, dbo.OppholdPoeng og dbo.Parametere
6.1	19.12.2022	8.0.22353.1	Mindre teknisk endring ifm sti til dll-er.
7.1	16.10.2023	8.0.23284.1	Dokument oppdatert ift NET6. Mindre justeringer i dokumentet.
7.2	17.12.2023	8.0.23348.01	Dokument oppdatert og beskriver løsningene for både NET 6 og .net 4.7.2.
8.1	15.10.2024	8.0.24282.2	Dokument oppdatert ift NET8. Mindre justeringer i dokumentet.
8.2	18.12.2024	8.0.24347.1	Dokument oppdatert. Mindre justeringer.
9.1	12.10.2025	8.0.25283.1	Dokument oppdatert. Mindre justeringer. Avsnitt om .net 4.7.2 er fjernet.
9.2	17.12.2025	8.0.25351.1	Dokument oppdatert.
9.3	8.01.2026	8.0.26006.1	Mindre oppdateringer.
10.1	15.10.2026	10.0.26286.5	Dokument oppdatert ift NET10. Avsnitt om NPK.ForCSVData fjernet.

Helsedirektoratet

Innhold

2 INNLEDNING 5

3 NPK DESIGN PRINSIPPER 5

4 GENERELT OM NPK 6

4.1 SYSTEMKRAV 6

5 NPK MODULER, REGLER OG KJØREJOBBSEKVENS. 6

6 INTEGRASJON MED NPK 7

6.1 INTEGRASJON AV NPK I EN WINDOWS APPLIKASJON 7
6.2 NPK KLIENT 10
6.2.1 NPK.Støtteklient 10
6.2 DATABASEGRENSESNITT 13
6.2.1 Systemtabeller 13
6.2.2 Input tabeller 13
6.2.3 Resultat tabeller 19

7 KONFIGURASJON 29

7.1 BRUKERKONFIGURASJON 29
7.1.1 Gyldighetsperiode for en regel 30
7.1.2 Brukerkonfigurasjon tabeller 32

8 INSTALLASJON 35

8.1 INSTALLASJONSPAKKEN 35
8.1.1 Krav til rettigheter ved installasjon 35
8.2 INSTALLASJON AV NPK 35

9 FEILHÅNDTERING 37

9.1 DUPLIKAT- OG KONSISTENSSJEKK AV BRUKERKONFIGURASJON 37

10 LOGGING 38

10.1 RAMMEVERK 38
10.2 MÅLFORMATER 38
10.2.1 Nivåer 38
10.2.2 Databasekonfigurasjon for logging 38
10.3 LOGG FORMAT 39
10.3.1 Logginformasjon (kort revisjon etter sjekk) 39
10.3.2 Visning av loggfiler 40

11 ANDRE TEKNISKE DETALJER 40

VEDLEGG A - MAPPING MELLOM NPK OG NPR MELDING 41

VEDLEGG B – OPPSETT OG KJØRING AV NPK.STØTTEKLIENT 43

B1 NPK 43
B1.1 INTEGRERT GRUPPERINGSKOMPONENT 43
B1.2 INNLESING AV KONFIGURASJON 43
B1.3 SJEKKE LOGGFIL 43
B1.4 HENTE RESULTAT FRA NPK BASEN 44

VEDLEGG C – SLETTESKRIPT FOR Å SLETTE KONFIGURASJON OG DATA I DATABASEN 45

C1 SLETT KONFIGURASJON 45

C2 SLETT DATA 46

VEDLEGG D – SLETTESKRIPT FOR Å SLETTE KUN RESULTATER I NPK DATABASEN 47

2 Innledning

Dette dokumentet er en overordnet teknisk dokumentasjon av NPK (Norsk pasientklassifisering). Dokumentets målgruppe er primært personer som har ansvar for å installere eller integrere NPK i ulike systemer eller driftsmiljøer.

Dokumentet beskriver NPK-løsningene som fra 2026 bygger på NET 10. Dokumentet bygger sterkt på tidligere NPK-Tekniskdokumentasjon.

Vi tar forbehold om at feil kan forekomme.

3 NPK design prinsipper

Designet av NPK følger av krav knyttet til funksjon, teknologi og bruksområde. Kravene som er definert er:

1. NPK skal kunne integreres i et databasemiljø og i Windows applikasjoner.
2. NPK skal være en tilstandsløs og idempotent kalkuleringsmotor.
3. NPK skal fungere uten behov for manuell endring av konfigurasjon.
4. NPK skal støtte sporing i resultatet tilbake til grunnlagsdataene (opphold/episoder)
5. NPK skal være åpen for endringer av regelsettet, men lukket for modifikasjon av regler. Dette betyr at regler kan utelukkes/legges til, men oppførselen til hver enkelt regel er konstant.
6. NPK skal være åpen for konfigurasjon av
    a. Logg føring av feil og ytelse.
    b. Parametere knyttet til regler
    c. Gyldig regelsett
    d. Kjørejobbsekvens
7. NPK skal avsluttes umiddelbart når feil oppstår.

NPK skal ikke validere pådragsdata *, logiske oppsettet av regler eller kjørejobbsekvens. Dette betyr at NPK vil

1. Kunne avslutte med feil
2. Kunne gi feil resultat, hvis pådragsdata ikke følger grenseverdiene definert i reglene (Se «NPK Beregningsregler» dokumentet for beskrivelse) eller hvis konfigurasjon av reglene eller kjørejobbsekvens ikke er meningsbærende.

   *NPK vil fra versjon 8.0.25283.1 kunne sette inn en episodes UtDatoTid der den mangler, men dette vil kreve en spesiell parameterinnstilling. Se også egen omtale i kapitel 7.1.Fra versjon 8.0.25351.1 er det innført en modul 90 hvor det kan settes inn i egne felt verdier for InnDatoTid og UtDatoTid til beregning av et mål LOS24TimersPerioderForAktivitetsmål for analyseformål. Dessuten at dersom parameter InnsettUtDatoTid er satt til ja, vil UtDatoTid hvor denne er innsatt settes til NULL for episoder og opphold.

4 Generelt om NPK

NPK er basert på rammeverket Microsoft NET10.

4.1 Systemkrav

NPK er en C# - løsning og bygger på .NET rammeverket. Fra 2026 er løsningen bygget på .NET 10. NET 10 løsningen krever å bli kjørt i et 64 bits miljø.

Ut ifra minneprofileringer som er gjort under utviklingsstadiet ble det lest inn ca. 1,5 millioner rader med Episoder. NPK hadde da et maks minnebruk på ca. 5,7 GB minne. Hele kjørejobben tok da ca. 26 minutter. Ut ifra denne kan man anta at minnebruken blir ca. 4 GB per 1 million Episoder.

Typisk maskinoppsett i et servermiljø (tilsvarer testmiljøet benyttet av Helsedirektoratet ved utvikling av NPK):

- Windows Server 2012 R2, 64 bit
- 32 GB minne (dedikert til NPK gir ca 8 milloner rader)
- 2,7 GHz prosessor
- 128 GB disk

Ovenstående kan skaleres opp og ned etter behov, der antall Episoder som skal inngå per kjørejobb, er viktigste faktor. Anvendelse i servermiljø er ikke en prinsipiell forutsetning; installasjon i 64 bits Windows-miljø for desktop/laptop er også mulig.

Bruk av det medfølgende databasegrensesnittet (NPK-databasen) forutsetter forutgående installasjon og tilstedeværelse av en valgfri SQL-databaseløsning. Per september 2023 er dette grensesnittet ikke testet mot andre databasesystemer enn Microsoft SQL Server 2012 – 2019.

Per versjon 5.1.19268 er databaselogikken flyttet fra NPK biblioteket til NPK.Støtteklient.

5 NPK Moduler, Regler og Kjørejobbsekvens.

NPK er organisert i et sett med funksjonelle moduler hvor hver modul er assosiert med et sett med regler. Modulenes primærfunksjon er å knytte regler opp mot funksjonelle behov. Følgende moduler er definert:

1. Episodeanalyse
2. DRG beregning av Episoder
3. Seleksjon av episoder for konstruksjon og beskrivelse av opphold
4. Oppholdskonstruksjon
5. Beskrivelse av Opphold
6. DRG beregning av Opphold
7. Poengberegning av Opphold
8. Særtjenestekonstruksjon og beskrivelse[1]
11. Konstruksjon av Særtjenester[2]
12. STG beregning av Særtjenester[2]

[1] Benyttes hvis regelsettdato er mindre enn 1.1.2018
[2] Modulene 11, 12, og 13 benyttes hvis regelsettdato er større enn eller lik 1.1.2018

13. Poengberegning av Særtjenester[2]
31. Konstruksjon av Tjenesteforløp
32. Beskrivelse av Tjenesteforløp
33. Gruppering av Tjenesteforløp
34. Poengberegning av Tjenesteforløp
90. Spesielle justeringer og beregninger

En regel kan være knyttet til 1 og bare 1 modul. Reglene og modulene er beskrevet i dokumentasjon av regler. Ved kjøring av NPK må man spesifisere en kjørejobbsekvens. En kjørejobbsekvens er den rekkefølgen modulene og tilhørende regler blir utført på pådragsdataene. I standard brukerkonfigurasjon er det definert en kjørejobbsekvens 1 – 7(8[1]), 11-13[2], 31-34,90.

6 Integrasjon med NPK

NPK er realisert som et sett med kodebibliotek utviklet i C#. Kodebiblioteket inneholder alle funksjonelle moduler og regler som er tilstrekkelig for å beregne poeng pr opphold basert på et sett med Episoder eller Opphold som pådragsdata. NPK er designet som en kalkuleringsmotor uten eget grafisk brukergrensesnitt og kan derfor kun benyttes som en integrert komponent i et eksisterende kjøremiljø. NPK har støtte for følgende integrasjoner:

1. Integrasjon med en Windows applikasjon.
   Integrasjon som en komponent i en eksisterende Windows applikasjon. NPK eksponerer et programmeringsgrensesnitt for kjøring av beregning hvor inndata er Episoder/konfigurasjon/Opphold og utdata resultatet av poengberegningen.
2. Databasegrensesnitt.
   NPK kan integreres inn i et eksisterende databasemiljø hvor både inn/ut-data utveksles gjennom en database; NPK databasen. NPK database er basert på et skjema definert av NPK prosjektet.

6.1 Integrasjon av NPK i en Windows applikasjon

Når NPK integreres inn i en Windows applikasjon har NPK ingen avhengighet til en database siden prosessering skjer i maskinminne. NPK tilbyr tre API-funksjoner for start av en kjørejobb. Forskjellen mellom disse funksjonene er kjøring med Episoder, kjøring med Episoder og Opphold eller kjøring med Episoder, Opphold, Særtjenester og Tjenesteforløp.
API funksjonene ligger i klassen er «NPK» som er definert i biblioteket «NPK.dll».

Metodene er:

```
Resultat Start(List<Episode> episoder, [Model.Konfigurasjon.Brukerkonfigurasjon brukerkonfigurasjon = null], [List<HReseptOppgjørsdata> hReseptOppgjørsdata = null])

Resultat Start(List<Episode> episoder, List<Opphold> oppholdsliste, [Model.Konfigurasjon.Brukerkonfigurasjon brukerkonfigurasjon = null], [List<HReseptOppgjørsdata> hReseptOppgjørsdata = null])

Resultat Start(List<Episode> episoder, List<Opphold> oppholdsliste, List<Særtjeneste> særtjenester, List<Tjenesteforløp> tjenesteforløpListe, [Model.Konfigurasjon.Brukerkonfigurasjon brukerkonfigurasjon = null], [List<HReseptOppgjørsdata> hReseptOppgjørsdata = null])
```

Eksempel C# kode for å starte NPK er:

```
var npk = new NPK();
npk.Initialiser(Brukerkonfigurasjon brukerkonfigurasjon);
npk.Start(…);
```

Initialiser metoden må kalles før Start metoden.

Grupperingskomponent

Gruppering av Episoder, Opphold, Særtjenester og Tjenesteforløp gjøres ved hjelp av en egen, konfigurerbar grupperingskomponent som er integrert i NPK. Denne komponenten initialiseres i NPK.Initialiser-metoden. Komponenten leser inn definisjonsdata fra en egen JSON-fil for Episoder og Opphold og en egen JSON-fil for Særtjenester. Filnavnene for JSON-filene angis i Brukerkonfigurasjon.Parametere.DefinisjonsdataforDRG (Episode og Opphold), i Brukerkonfigurasjon.Parametere.DefinisjonsdataforSTG (Særtjeneste) og i Brukerkonfigurasjon.Parametere.DefinisjonsdataforTFG (Tjenesteforløp). Se eksempel nedenfor:

Brukerkonfigurasjon.Parametere.DefinisjonsdataforDRG	DefinitionData_NOR_2016_PR1cN.json
Brukerkonfigurasjon.Parametere.DefinisjonsdataforSTG	definitiondata.json
Brukerkonfigurasjon.Parametere.DefinisjonsdataforTFG	DefinitionData_TFG_001.json

Konstruksjon av Tjenesteforløp[3]

Ved konstruksjon av Tjenesteforløp benyttes EOC.ConstructionTool, som er en integrert komponent i NPK. Komponenten leser inn definisjonsdata fra en JSON-fil. Filnavnet på JSON-filen angis i Brukerkonfigurasjon.Parametere. EOCDefinisjonsdataForTjenesteforløp. Se eksempel nedenfor:

Brukerkonfigurasjon.Parametere.EOCDefinisjonsdataForTjenesteforløp	EOCDefinitionData.json

For å kunne gjennomføre gruppering og poengberegning må man derfor sende inn Brukerkonfigurasjon til Initialiser metoden. Definisjonsdata filene må ligge under angitt Konfigurasjonsmappesti. Se avsnitt om duplikat- og konsistenssjekk av DRG- og STG-listene i brukerkonfigurasjon.

Brukerkonfigurasjonen består av et sett med klasser og ser slik ut:

```
public class Brukerkonfigurasjon
{
    public List<int> KjørejobbsekvensListe
    public Parametere Parametere
    public List<DRG> DRGListe
    public List<EpisodeseleksjonVilkår> EpisodeseleksjonVilkårListe
    public List<HelsepersonellOgDRGKombinasjoner> HelsepersonellOgDRGKombinasjonerListe
    public List<EnhetOgDRGKombinasjoner> EnhetOgDRGKombinasjoner
    public List<STG> STGListe
    public List<KodeUnntak> KodeUnntaksListe
```

[3] Gjelder ved regelsettdato større enn eller lik 1.1.2020

```
    public List<TFG> TFGListe
    public List<Kommune> KommuneListe
}
```

Se avsnitt om konfigurasjon for mer detaljert beskrivelse.

Resultatet som NPK genererer er fire lister én med EpisodeResultat, én med Opphold, én med Særtjeneste og én med Tjenesteforløp. Klassedefinisjonen er som følger:

```
public class Resultat
{
    public List<EpisodeResultat> EpisodeResultater
    public List<Opphold> OppholdListe
    public List<Særtjeneste> Særtjenester
    public List<EpisodeSærtjenesteKobling> EpisodeSærtjenesteKoblinger
    public List<Tjenesteforløp> Tjenesteforløp
    public List<OppholdSærtjenesteTjenesteforløpKobling> OppholdSærtjenesteTjenesteforløpKoblinger
}
```

Se avsnitt om databasegrensesnittet for en mer detaljer.

Klassedefinisjonene av «*Resultat», «Episode», «Opphold», «Tjenesteforløp»* samt brukerkonfigurasjonen ligger i «*NPK.Model.dll*».

6.2 NPK klient

Det finnes per 2027 hovedsakelig en måte å kjøre NPK på.

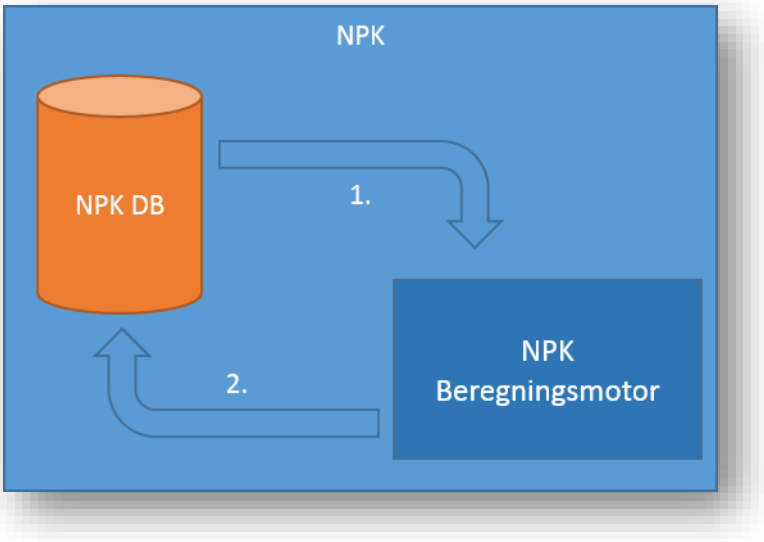
6.2.1 NPK.Støtteklient

6.2.1.1 *NPK.Støtteklient*

Dette er et eksempel på hvordan NPK er integrert i en Windows applikasjon. Når denne kalles leser den først data fra NPK databasen inn i minnet. Deretter prosesseres dataene i NPK kalkuleringsmotoren, og til slutt lagres resultatene i NPK databasen. Se figur under.

Eksempel C# kode for å kjøre denne er:

```
var npk = new NPKStaging();
npk.Start();
```



I tillegg er det også lagt inn støtte for å starte NPK.Støtteklient med flere forskjellige parametere.

Tabell 1: Oversikt over parametere ved kjøring av NPK.Støtteklient.

Parameter	Tilgjengelig fra	Eksempel	Beskrivelse
		NPK.Støtteklient.exe	Vanlig kjøring med bruk av NPK databasen
1		NPK.Støtteklient.exe 1	Oppretter NPK databasen. (Brukes under installasjon).
2		NPK.Støtteklient.exe 2	Kjører med hardkodede verdier. (Eksempelkode for integrasjon i en Windows applikasjon)
3	8.0.25283.1	NPK.Støtteklient.exe 3	Samme som ingen parameter (vanlig kjøring), men transaksjonslogg trunkeres ikke og database må eksistere fra før. Om databasen ikke eksisterer, får man feilmelding.

4	8.0.25283.1	NPK.Støtteklient.exe 4	Samme som parameter 1, men database må eksistere fra før. Om databasen ikke eksisterer, får man feilmelding.
ytelse		NPK.Støtteklient.exe ytelse 1000000 1	Kjører NPK med et oppgitt antall episoder og antall pasienter. Lager den samme episoden x antall ganger og kjører NPK med disse episodene en gang pr pasient, f. eks 1.000.000. Eksempelet vil kjøre NPK med 1 million episoder 1 gang.
brukerkonfig		NPK.Støtteklient.exe konfig «C:\NPKKonfigurasjon»	Overstyrer standardmappen for brukerkonfigurasjon.

Kommentar til tabell 1:
Fra versjon 8.0.26006.1 gjøres det en sjekk for å se om det er utestående migrasjoner før migrering kjøres. Logging reflekterer dette.

Vær oppmerksom på at brukerkonfigurasjonen må som standard eksistere under mappen for Konfigurasjon. **Det må ikke eksistere flere versjoner av samme fil i Konfigurasjonsmappen fordi NPK vil lese den første filen den finner.**

Innstillinger for plassering av konfigurasjonsmappe, loggfiler og om brukerkonfigurasjonen skal leses fra fil eller database hentes fra applikasjonens konfigurasjonsfil (appsettings.json) og kan hvis ønskelig endres.

```
{
  "KlientNavn": "NPK.Støtteklient",
  "ConnectionStrings": {
    "NpkConnectionString": "data source=localhost;initial catalog=NPK;integrated security=True;MultipleActiveResultSets=True;TrustServerCertificate=True;App=EntityFramework"
  },
  "LoggFilMappesti": "c:\\NPK.Logg",
  "KonfigurasjonsMappeSti": "c:\\NPKKonfigurasjon",
  "AppDataPath": "c:\\NPKAppData",
  "LasteBrukerkonfigurasjonFraFil": "Nei",
  "TidsavbruddgrenseForLesingfraDatabasen": "1200",
  "TidsavbruddgrenseForSkrivingtilDatabasen": "3600",
  "SqlBulkCopyBatchSize": 10000
}
```

I forbindelse med oppgradering til .NET8 for 2025 ble det lagt til nytt parameter "TidsavbruddgrenseForSkrivingtilDatabasen", for å kunne håndtere situasjoner hvor Støtteklienten feiler ved skriving av store dataset til databasen. Denne parameteren har enheten sekunder.

For NPK 2026 (8.0.25283.1) er det innført to nye parametere.

- AppDataPath muliggjør det å konfigurere hvor mellomlagring av data foregår. Merk at som default er den satt til C:\\NPKAppData.
- SqlBulkCopyBatchSize styrer antall rader som kopieres fra minne til database per batch. Som default er den satt til 10000.

Innstilling	Standardverdi	Beskrivelse
LoggFilMappesti	c:\NPK.Logg	Brukes til å overstyre standard plassering av loggfilene.
KonfigurasjonsMappeSti	c:\Program Files \Helsedirektoratet\NPK \Konfigurasjon	Brukes til å overstyre standard mappe for konfigurasjonsfilene.
LasteBrukerkonfigurasjonFraFil	Nei	Angir om NPK skal lese brukerkonfigurasjonen fra databasen eller fra filplassering. Hvis innstillingen settes til «Ja» vil den lete etter brukerkonfigurasjonsfilene i mappen som er angitt i innstillingen «KonfigurasjonsMappeSti». Eksisterende data i konfigurasjonstabellene i NPK basen vil bli overskrevet.
TidsavbruddgrenseForLesingfraDatabasen	1200	Brukes til å angi tidsavbruddgrense i sekunder for lesing (episodedata) fra databasen.
TidsavbruddgrenseForSkrivingtilDatabasen	3600	Brukes til å angi tidsavbruddgrense i sekunder for skriving til databasen.
AppDataPath	C:\NPKAppData	Muliggjør å konfigurere hvor mellomlagring av data foregår.
SqlBulkCopyBatchSize	10000	Denne styrer størrelsen på hvor store batcher NPK.Støtteklienten kopierer data fra minne til database. Som default er verdi satt til 10000. Tidligere var denne ikke konfigurerbar, men verdi internt var 1000. Denne skal forbedre ytelse noe. Verdi bør ligge i intervallet 1000 tom 50000.

Fra og med versjon 10.0.26286.5 for NPK 2027:
For steget *Leser episoder* er metoden endret fra en SQL-spørring, til å hente alle tabeller inn i minne og gjøre mapping der. Dette for å unngå EF Core kartesianske-eksplosjonsproblemer og split-query-heng ved særlig store datamengder. Det gjøres én SELECT per tabell, deretter bygges objektgrafen i minnet med Dictionary-oppslag. Det er gjort konfigurerbart om man ønsker å benytte den nye metoden eller den gamle gjennom en egen parameter i NPK_Parametere.

6.2 Databasegrensesnitt

NPK har støtte for integrasjon med ett database miljø. Det poengteres at dette ikke er definert som en kjernefunksjonalitet i NPK og at integrasjonen er dermed definert som et tillegg med de begrensninger dette gir i forhold til funksjonalitet knyttet til NPK DB.

6.2.1 Systemtabeller

I forbindelse med opprettelse av NPK databasen opprettes en tabell dbo.__EFMigrationsHistory som er en systemtabell i forbindelse med EntityFramework. Denne må være tilstede for at programmene som benytter NPK databasen skal fungere. Denne skal ikke endres manuelt.

6.2.2 Input tabeller

dbo.Avdelingsopphold

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1

dbo.Enhet

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	Type	int	4	10	True	
	OffisiellAvdelingskode	nvarchar	50	0	False	
	ISFRefusjon	int	4	10	False	
FK	Episode_Id	bigint	8	19	True	

Foreign Keys (dbo.Enhet)

Name	Column Name	Reference
FK_Enhet_Episode_Episode_Id	Episode_Id	dbo.Episode.Id

dbo.Episode

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	AlderIDager	int	4	10	False	
	EpisodeId	nvarchar	50	0	True	
	Debitor	int	4	10	True	
	FolkeregistrertKommune	nvarchar	4	0	False	
	InndatoTid	datetime2	8	3	True	
	UtdatoTid	datetime2	8	3	False	
	Omsorgsnivå	int	4	10	True	

	TilSted	int	4	10	False	
	InnTilstand	int	4	10	False	
	UtTilstand	int	4	10	False	
FK	Avdelingsopphold_Id	bigint	8	19	False	
FK	Kontakt_Id	bigint	8	19	False	
FK	Pasient_Id	bigint	8	19	True	
FK	Rapporteringsenhet_Id	bigint	8	19	True	
	InnmåteHastegrad	int	4	10	False	

Foreign Keys (dbo.Episode)

Name	Column Name	Reference
FK_Episode_Avdelingsopphold_Avdelingsopphold_Id	Avdelingsopphold_Id	dbo.Avdelingsopphold.Id
FK_Episode_Kontakt_Kontakt_Id	Kontakt_Id	dbo.Kontakt.Id
FK_Episode_Pasient_Pasient_Id	Pasient_Id	dbo.Pasient.Id
FK_Episode_Rapporteringsenhet_Rapporteringsenhet_Id	Rapporteringsenhet_Id	dbo.Rapporteringsenhet.Id

dbo.Helseperson

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	UtførendeHelsepersonell	int	4	10	True	
	Rolle	int	4	10	False	
FK	OpphavEpisode_Id	bigint	8	19	False	
FK	Kontakt_Id	bigint	8	19	False	
FK	Opphold_Id	bigint	8	19	False	

Foreign Keys (dbo.Helseperson)

Name	Column Name	Reference
FK_Helseperson_Episode_OpphavEpisode_Id	OpphavEpisode_Id	dbo.Episode
FK_Helseperson_Kontakt_Kontakt_Id	Kontakt_Id	dbo.Kontakt
FK_Helseperson_Opphold_Opphold_Id	Opphold_Id	dbo.Opphold

dbo.HReseptOppgjørsdata

Key	Name	Data Type	Length	Precision	Not Null	Identity
	EnkeltregningId	bigint	8	19	True	
	Pasientløpenummer	nvarchar	100	0	True	
	Fødselsår	int	4	10	True	
	Kjønn	int	4	10	True	
	KommuneNr	nvarchar	50	0	True	
	Utleveringsdato	datetime2	8	3	True	
	ATCKode	nvarchar	50	0	False	
	Refusjonskode	nvarchar	50	0	False	
	Rekvirentvirksomhet	int	4	10	False	
FK	Pasient_Id	bigint	8	19	False	
FK	Rapporteringsenhet_Id	bigint	8	19	False	
PK	Id	bigint	8	19	True	1 - 1

Foreign Keys (dbo.HReseptOppgjørsdata)

Name	Column Name	Reference

FK_HReseptOppgjørsdata_Pasient_Pasient_Id	Pasient_Id	dbo.pasient
FK_HReseptOppgjørsdata_Rapporteringsenhet _Rapporteringsenhet_Id	Rapporteringsenhet_Id	dbo.Rapporteringsenhet

dbo.Kode

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	KodeNr	int	4	10	True	
	Verdi	nvarchar	50	0	False	
	Kodeverk	nvarchar	50	0	False	
FK	Prosedyre_Id	bigint	8	19	False	
FK	Tilstand_Id	bigint	8	19	False	

Foreign Keys (dbo.Kode)

Name	Column Name	Reference
FK_Kode_Prosedyre_Prosedyre_Id	Prosedyre_Id	dbo.Prosedyre
FK_Kode_Tilstand_Tilstand_Id	Tilstand_Id	dbo.Tilstand

dbo.Kontakt

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	Type	int	4	10	True	
	Aktivitetssted	int	4	10	True	
	Aktivitetstype	int	4	10	False	
	IndirekteAktivitet	int	4	10	False	

dbo.Pasient

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	Nr	nvarchar	100	0	False	
	Fødselsår	int	4	10	True	
	Kjønnkode	int	4	10	True	

dbo.Prosedyre

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	ProsedyreNr	int	4	0	True	
FK	Opphold_Id	bigint	8	19	False	
FK	OpphavEpisode_Id	bigint	8	19	False	

Foreign Keys (dbo.Prosedyre)

Name	Column Name	Reference
FK_Prosedyre_Episode_OpphavEpisode_Id	OpphavEpisode_Id	**dbo.Episode**
FK_Prosedyre_Opphold_Opphold_Id	Opphold_Id	**dbo.Opphold**

dbo.Rapporteringsenhet

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	Rapporteringsenhetsnummer	nvarchar	50	0	False	
	Foretaksnummer	nvarchar	50	0	False	

dbo.Takst

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	Takstnummer	nvarchar	50	0	False	
FK	OpphavEpisode_Id	bigint	8	19	False	
FK	Kontakt_Id	bigint	8	19	False	
FK	Opphold_Id	bigint	8	19	False	
FK	Særtjeneste_Id	bigint	8	19	False	

Foreign Keys (dbo.Takst)

Name	Column Name	Reference
FK_Takst_Episode_OpphavEpisode_Id	OpphavEpisode_Id	dbo.Episode
FK_Takst_Kontakt_Kontakt_Id	Kontakt_Id	dbo.Kontakt
FK_Takst_Opphold_Opphold_Id	Særtjeneste_Id	dbo.Opphold
FK_Takst_Særtjeneste_Særtjeneste_Id	Opphold_Id	dbo.Særtjeneste

dbo.Tidspunkt

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	Verdi	datetime2	8	3	True	
	Type	int	4	10	True	
FK	Episode_Id	bigint	8	19	False	

Foreign Keys (dbo.Tidspunkt)

Name	Column Name	Reference
FK_Tidspunkt_Episode_Episode_Id	Episode_Id	dbo.Episode

dbo.Tilstand

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	ErHovedtilstand[4]	bit	1	1	False	
	Akse	int	4	10	False	
	TilstandNr	int	4	10	True	
FK	Opphold_Id	bigint	8	19	False	
FK	Særtjeneste_Id	bigint	8	19	False	
FK	OpphavEpisode_Id	bigint	8	19	False	

Foreign Keys (sbo.Tilstand)

Name	Column Name	Reference
FK_Tilstand_Episode_OpphavEpisode_Id	OpphavEpisode_Id	dbo.Episode
FK_Tilstand_Særtjeneste_Særtjeneste_Id	Særtjeneste_Id	dbo.Særtjeneste
FK_Tilstand_Opphold_Opphold_Id	Opphold_Id	dbo.Opphold

NPK databasen er strukturert ut ifra Helsedirektoratet sitt domenespråk. Det har blitt brukt norske tegn og alle feltnavn har blitt skrevet fullt ut slik at en person som ser domenemodellen for første gang enkelt skal kunne kommunisere med Helsedirektoratet på domenespråket. Dersom man ønsker en mapping mellom tabellene og NPR meldingen kan man se vedlegg A.

[4] NB! Feltet Tilstand.ErHovedtilstand skal ikke sendes inn til NPK, da dette er et felt som NPK beregner.

6.2.3 Resultat tabeller

Den samme NPK databasen brukes også til lagring av resultat. For dokumentasjon av feltnavn se dokumentet «NPK Beregningsregler». Strukturen for resultatet for EpisodeResultat, Opphold, Særtjeneste og Tjenesteforløp ser slik ut:

dbo.DRGBeregningResultat

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
FK	DRG_Id	bigint	8	19	False	
	Grupperingsregel	nvarchar	50	0	True	
	Grupperingsstreng	nvarchar	4000	0	False	
	Returkode	nvarchar	5	0	False	
	Hoveddiagnosekategori	nvarchar	50	0	False	
	CCNivå	int	4	10	True	
	AndreGrupperingsegenskaper	nvarchar	4000	0	False	
	AntallDiagnoseegenskaper	int	4	10	True	
	AntallKomplikasjonsegenskaper	int	4	10	True	
FK	STG_Id	bigint	8	19	False	
FK	TFG_Id	bigint	8	19	False	

Foreign Keys (dbo.DRGBeregningResultat)

Name	Column Name	Reference
FK_DRGBeregningResultat_DRG_DRG_Id	DRG_Id	dbo.DRG
FK_DRGBeregningResultat_STG_STG_Id	STG_Id	dbo.STG
FK_DRGBeregningResultat_TFG_TFG_Id	TFG_Id	dbo.TFG

dbo.EpisodeOppholdKobling

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Episode_Id	bigint	8	19	True	
FK	Opphold_Id	bigint	8	19	True	

Foreign Keys (dbo.EpisodeOppholdKobling)

Name	Column Name	Reference
FK_EpisodeOppholdKobling_Opphold_Opphold_Id	Opphold_Id	dbo.Opphold

dbo.EpisodeResultat

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Episode_Id	bigint	8	19	True	
	DRGGrupperingAlder	int	4	10	False	
	DRGGrupperingAvslutningsmåte	nvarchar	1	0	False	
	DRGGrupperingVarighet	int	4	10	False	
	ErLaboratorietjeneste	bit	1	1	True	
	ErDominantForOppholdsbeskrivelse	bit	1	1	True	
	ErGyldigForOppholdskonstruksjon	bit	1	1	True	
	ErGyldigForOppholdsbeskrivelse	bit	1	1	True	

	ErGyldigTelemedisinskHelsehjelp	bit	1	1	True	
	ErIndirekteHelsehjelp	bit	1	1	True	
	ErPasientBosattINorge	bit	1	1	True	
	ErPHVEllerTSB	bit	1	1	True	
	ErSærfinansiert	bit	1	1	True	
	ErUtførtAvISFGodkjentAvdeling	bit	1	1	True	
	ErDødVedAnkomst	bit	1	1	True	
	ErGyldigForOpprettelseAvSærtjeneste	Bit	1	1	True	
	ErIkkeReellPasientkontakt	bit	1	1	True	
	FørsteUtskrivningsklartidspunkt	datetime2	8	3	False	
	LOS24Timersperioder	decimal	19	3	False	
	LOSDøgnskiller	int	4	10	False	
	LOSDager	int	4	10	False	
	InneholderHjemmebasertPeriotonealdialyse	bit	1	1	True	
	InneholderPasientadministrertLegemiddelbehandling	bit	1	1	True	
	Spesialisering	int	4	10	True	
FK	DRGBeregningResultat_Id	bigint	8	19	False	
	ErSamarbeidsaktivitet	bit	1	1	True	
	ErLysbehandling	bit	1	1	True	
	HarUtDatoTidInnsattAvNPK	bit	1	1	True	
	InneholderDigitalPasientopplæring	bit	1	1	True	

Foreign Keys (sbo.EpisodeResultat)

Name	Column Name	Reference
FK_EpisodeResultat_DRGBeregningResultat _DRGBeregningResultat_Id	DRGBeregningResultat_Id	dbo.DRGBeregningResultat

dbo.EpisodeSærtjenesteKobling

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Særtjeneste_Id	bigint	8	19	True	
	Særtjenestetype	int	4	10	True	
FK	Episode_Id	bigint	8	19	False	
FK	HReseptOppgjørsdata_Id	bigint	8	19	False	

Foreign Keys (dbo.EpisodeSærtjenesteKobling

Name	Column Name	Reference
FK_EpisodeSærtjenesteKobling_Episode_Episode_Id	HReseptOppgjørsdata_Id	dbo.Episode

FK_EpisodeSærtjenesteKobling_HReseptOppgjørsdata _HReseptOppgjørsdata_Id	Episode_Id	dbo.HReseptOppgjørsdata

dbo.Helseperson

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 – 1
	UtførendeHelsepersonell	int	4	10	True	
	Rolle	int	4	10	False	
FK	OpphavEpisode_Id	bigint	8	19	False	
FK	Kontakt_Id	bigint	8	19	False	
FK	Opphold_Id	bigint	8	19	False	

Foreign Keys (sbo.HelsePerson)

Name	Column Name	Reference
FK_Helseperson_Episode_OpphavEpisode_Id	OpphavEpisode_Id	dbo.Episode
FK_Helseperson_Kontakt_Kontakt_Id	Kontakt_Id	dbo.Kontakt
FK_Helseperson_Opphold_Opphold_Id	Opphold_Id	dbo.Opphold

dbo.Kode

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	KodeNr	int	4	10	True	
	Verdi	nvarchar	50	0	False	
	Kodeverk	nvarchar	50	0	False	
FK	Prosedyre_Id	bigint	8	19	False	
FK	Tilstand_Id	bigint	8	19	False	

Foreign Keys (dbo.Kode)

Name	Column Name	Reference
FK_Kode_Prosedyre_Prosedyre_Id	Prosedyre_Id	dbo.Prosedyre
FK_Kode_Tilstand_Tilstand_Id	Tilstand_Id	dbo.Tilstand

dbo.Opphold

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	
	AntallAvdelingsopphold	int	4	10	True	
	AntallKontakter	int	4	10	True	
	InndatoTid	datetime2	8	3	True	
	UtdatoTid	datetime2	8	3	False	
	DRGGrupperingAlder	int	4	10	False	
	DRGGrupperingAvslutningsmåte	nvarchar	1	0	False	
	DRGGrupperingVarighet	int	4	10	False	
	ErLaboratorietjeneste	bit	1	1	True	

	ErPHVEllerTSB	bit	1	1	True	
	ErSærfinansiert	bit	1	1	True	
	ErUnntaksbehandlet	bit	1	1	True	
	ErUtførtAvISFGodkjentAvdeling	bit	1	1	True	
	FørsteUtskrivningsklartidspunkt	datetime2	8	3	False	
	Kommunenummer	nvarchar	4	0	False	
	LOSDøgnskillerForPoengberegning	int	4	10	False	
	LOS24TimersperioderForPoengberegning	decimal	19	3	False	
	Omsorgsnivå	int	4	10	True	
	TilSted	int	4	10	False	
	Debitor	int	4	10	True	
FK	FørsteTellendeEpisode_Id	bigint	8	19	False	
FK	SisteTellendeEpisode_Id	bigint	8	19	False	
FK	Hovedepisode_Id	bigint	8	19	False	
FK	DRGBeregningResultat_Id	bigint	8	19	False	
FK	Poeng_Id	bigint	8	19	False	
FK	Pasient_Id	bigint	8	19	True	
FK	Rapporteringsenhet_Id	bigint	8	19	True	
	InnmåteHastegrad	int	4	10	False	
	Aktivitetssted	int	4	10	False	
	EOCRegler	nvarchar		0	False	
FK	Tjenesteforløp_Id	bigint	8	19	False	
	Kontakttype	Int	4	10	False	
	HarUtDatoTidInnsattAvNPK	bit	1	1	False	
	HarInnDatoTidInnsattAvNPK	bit	1	1	False	
	InnDatoTidInnsattAvNPK	datetime2	7	3	False	
	LOS24TimersPerioderForAktivitetsmål	decimal	19	3	False	
	UtDatoTidInnsattAvNPK	datetime2	7	3	False	

Foreign Keys (dbo.Opphold)

Name	Column Name	Reference
FK_dbo.Opphold_dbo.Episode_Hovedepisode_Id	Hovedepisode_Id	dbo.Episode
FK_dbo.Opphold_dbo.Episode_SisteTellendeEpisode_Id	SisteTellendeEpisode_Id	dbo.Episode
FK_dbo.Opphold_dbo.Pasient_Pasient_Id	Pasient_Id	dbo.Pasient
FK_dbo.Opphold_dbo.Rapporteringsenhet_Rapporteringsenhet_Id	Rapporteringsenhet_Id	dbo.Rapporteringsenhet
FK_dbo.Opphold_dbo.Tjenesteforløp_Tjenesteforløp_Id	Tjenesteforløp_Id	dbo.Tjenesteforløp
FK_dbo.Opphold_dbo.Episode_FørsteTellendeEpisode_Id	FørsteTellendeEpisode_Id	Dbo.Episode
FK_dbo.Opphold_dbo.DRGBeregningResultat_DRGBeregningResultat_Id	DRGBeregningResultat_Id	Dbo.DRGBeregningResultat
FK_dbo.Opphold_dbo.Pasient_Pasient_Id	Pasient_Id	Dbo.Pasient

dbo.OppholdPoeng

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	ISFPoeng	decimal	19	3	True	
	Poengsum	decimal	19	3	True	
	DRGBasispoeng	decimal	19	3	True	
	PoengtilleggRehabiliteringPrimærVanlig	decimal	19	3	True	
	PoengtilleggRehabiliteringPrimærKompleks	decimal	19	3	True	
	PoengtilleggRehabiliteringSekundær	decimal	19	3	True	
	PoengtilleggLangLiggetid	decimal	19	3	True	
	PoengtilleggOrgandonasjon	decimal	19	3	True	
	PoengtilleggPalliativBehandling	decimal	19	3	True	
	PoengtilleggGrupperettetPasientopplæring	decimal	19	3	True	
	PoengtilleggMultipleLegekonsultasjoner	decimal	19	3	True	
	PoengtilleggBrannskadebehandlingHaukeland	decimal	19	3	True	
	PoengtilleggMedikamentellKreftbehandling	decimal	19	3	True	
	PoengfradragRituellOmskjæring	decimal	19	3	True	
	PoengfradragAssistertInseminasjon	decimal	19	3	True	
	PoengfradragSteriliseringSomHovedtilstand	decimal	19	3	True	
	ErGyldigBostedForISF	bit	1	1	True	
	ErGyldigHelsepersonOgDRGForISF	bit	1	1	True	
	ErGyldigEnhetOgDRGForISF	bit	1	1	True	
	ErGyldigOppholdsavslutningForISF	bit	1	1	True	
	ErGyldigFinansieringsmåteForISF	bit	1	1	True	
	ErGyldigTjenesteområdeForISF	bit	1	1	True	
	ErGyldigTjenesteinnholdForISF	bit	1	1	True	
	ErTellendeISFOpphold	bit	1	1	True	
	PoengtilleggSærskiltAmbulantKonsultasjon	decimal	19	3	True	
	PoengfradragKorteOpphold	decimal	18	2	False	
	ErUavhengigAvTjenesteforløpForISFBeregning	bit	1	1	False	
	ISFPoengAndelForventet	decimal	19	3	False	
	ISFPoengAndelBeregnet	decimal	19	3	False	
	PoengtilleggKonsentrerteTverrfagligeHelsetilbud	decimal	19	3	True	
	PoengtilleggEgenandel	decimal	19	3	True	

dbo.OppholdSærtjenesteTjenesteforløpKobling

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 – 1
FK	Tjenesteforløp_Id	bigint	8	19	True	
FK	Opphold_Id	bigint	8	19	False	
FK	Særtjeneste_Id	bigint	8	19	False	
	Rangeringsverdi	int	4	10	True	
	Regeltype	nvarchar		0	False	
	Regelnummer	bigint	8	19	True	

Foreign Keys (dbo.OppholdSærtjenesteTjenesteforløpKobling)

Name	Column Name	Reference
FK_OppholdSærtjenesteTjenesteforløpKobling _Opphold_Opphold_Id	Opphold_Id	dbo.Opphold
FK_OppholdSærtjenesteTjenesteforløpKobling _Særtjeneste_Særtjeneste_Id	Særtjeneste_Id	dbo.Særtjeneste
FK_OppholdSærtjenesteTjenesteforløpKobling _Tjenesteforløp_Tjenesteforløp_Id	Tjenesteforløp_Id	dbo.Tjenesteforløp

dbo.Prosedyre

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	ProsedyreNr	int	4	0	True	
FK	Opphold_Id	bigint	8	19	False	
FK	OpphavEpisode_Id	bigint	8	19	False	

Foreign Keys (dbo.Prosedyre)

Name	Column Name	Reference
FK_Prosedyre_Episode_OpphavEpisode_Id	OpphavEpisode_Id	**dbo.Episode**
FK_Prosedyre_Opphold_Opphold_Id	Opphold_Id	**dbo.Opphold**

dbo.Rapporteringsenhet

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	Rapporteringsenhetsnummer	nvarchar	50	0	False	
	Foretaksnummer	nvarchar	50	0	False	

dbo.Særtjeneste

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 – 1
FK	STG_Id	bigint	8	19	False	
	Periodetype	nvarchar	50	0	False	
	Periodenummer	int	4	10	False	
	PeriodeFraDato	date	3	10	True	
	PeriodeTilDato	date	3	10	True	
FK	Pasient_Id	bigint	8	19	True	
	Grupperingsregel	nvarchar	50	0	False	
	Grupperingsstreng	nvarchar	4000	0	False	
	ErTellendeSærtjenesteInnenPeriode	bit	1	1	True	
	ErTellendeISFSærtjeneste	bit	1	1	True	
	ISFPoeng	decimal	19	3	True	
	ABFPoeng	decimal	19	3	True	
	ErPasientBosattINorge	bit	1	1	True	
	Kommunenummer	nvarchar	10	0	False	
FK	Rapporteringsenhet_Id	bigint	8	19	False	
	ErUavhengigAvTjenesteforløpForISFBeregning	bit	1	1	False	
	ISFPoengAndelForventet	decimal	19	3	False	
	ISFPoengAndelBeregnet	decimal	19	3	False	
FK	Tjenesteforløp_Id	bigint	8	19	False	
	EOCRegler	nvarchar		0	False	
FK	DRGBeregningResultat_Id	bigint	8	19	False	
	ErGyldigEnhetForISF	bit	1	1	True	

Foreign Keys (dbo.Særtjeneste)

Name	Column Name	Reference
FK_Særtjeneste_DRGBeregningResultat_DRGBeregningResultat_Id	DRGBeregningResultat_Id	dbo.DRGBeregningResultat
FK_Særtjeneste_Pasient_Pasient_Id	Pasient_Id	dbo.Pasient
FK_Særtjeneste_Rapporteringsenhet_Rapporteringsenhet_Id	Rapporteringsenhet_Id	dbo.Rapporteringsenhet
FK_Særtjeneste_STG_STG_Id	STG_Id	dbo.STG
FK_Særtjeneste_Tjenesteforløp_Tjenesteforløp_Id	Tjenesteforløp_Id	dbo.Tjenesteforløp

dbo.Takst

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	Takstnummer	nvarchar	50	0	False	
FK	OpphavEpisode_Id	bigint	8	19	False	
FK	Kontakt_Id	bigint	8	19	False	
FK	Opphold_Id	bigint	8	19	False	
FK	Særtjeneste_Id	bigint	8	19	False	

Foreign Keys (sbo.Takst)

Name	Column Name	Reference
FK_Takst_Episode_OpphavEpisode_Id	OpphavEpisode_Id	dbo.Episode
FK_Takst_Kontakt_Kontakt_Id	Kontakt_Id	dbo.Kontakt
FK_Takst_Opphold_Opphold_Id	Særtjeneste_Id	dbo.Opphold
FK_Takst_Særtjeneste_Særtjeneste_Id	Opphold_Id	dbo.Særtjeneste

dbo.Tilstand

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	ErHovedtilstand	bit	1	1	False	
	Akse	int	4	10	False	
	TilstandNr	int	4	10	True	
FK	Opphold_Id	bigint	8	19	False	
FK	Særtjeneste_Id	bigint	8	19	False	
FK	OpphavEpisode_Id	bigint	8	19	False	

Foreign Keys (dbo.Tilstand)

Name	Column Name	Reference
FK_Tilstand_Episode_OpphavEpisode_Id	OpphavEpisode_Id	dbo.Episode
FK_Tilstand_Opphold_Opphold_Id	Opphold_Id	dbo.Opphold
FK_Tilstand_Særtjeneste_Særtjeneste_Id	Særtjeneste_Id	dbo.Særtjeneste

dbo.Tjenesteforløp

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 – 1
	StartDato	datetime2	8	3	True	
	SluttDato	datetime2	8	3	False	
FK	Pasient_Id	bigint	8	19	True	
	AntallOpphold	int	4	10	False	
	AntallSærtjenester	int	4	10	False	
	AntallOppholdISF	int	4	10	False	
	AntallSærtjenesterISF	int	4	10	False	
FK	TFG_Id	bigint	8	19	False	
	KodeUtløsendeHendelse	nvarchar	50	0	False	
FK	IdUtløsendeOpphold	bigint	8	19	False	
FK	IdUtløsendeSærtjeneste	bigint	8	19	False	
	Kommunenummer	nvarchar	4	0	False	
	År	int	4	10	False	
	Periodetype	nvarchar	50	0	False	
	Periodenummer	int	4	10	False	
FK	Poeng_Id	bigint	8	19	False	
	TFGGrupperingAlder	int	4	10	False	
	Grupperingsregel	nvarchar	50	0	False	

	TFGGrupperingVarighet	int	4	10	False	
	Grupperingsstreng	nvarchar			False	
	VarighetDager	int	4	10	False	
	VarighetDagerISF	int	4	10	False	
FK	DRGBeregningResultat_Id	bigint	8	19	False	

Foreign Keys (dbo.Tjenesteforløp)

Name	Column Name	Reference
FK_Tjenesteforløp_DRGBeregningResultat _DRGBeregningResultat_Id	DRGBeregningResultat_Id	dbo.DRGBeregningResultat
FK_Tjenesteforløp_Opphold_IdUtløsendeOpphold	IdUtløsendeOpphold	dbo.Opphold
FK_Tjenesteforløp_Pasient_Pasient_Id	Pasient_Id	dbo.Pasient
FK_Tjenesteforløp_Særtjeneste _IdUtløsendeSærtjeneste	IdUtløsendeSærtjeneste	dbo.Særtjeneste
FK_Tjenesteforløp_TFG_TFG_Id	TFG_Id	dbo.TFG
FK_Tjenesteforløp_TjenesteforløpPoeng_Poeng_Id	Poeng_Id	dbo.TjenesteforløpPoent

dbo.TjenesteforløpPoeng

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	TFGBasisPoeng	decimal	19	3	True	
	ISFPoeng	decimal	19	3	True	
	ABFPoeng	decimal	19	3	True	
	SumABFPoengDeltjenesterTotalt	decimal	19	3	True	
	SumABFPoengDeltjenesterISF	decimal	19	3	True	
	ErTellendeISFTjenesteforløp	bit	1	1	True	

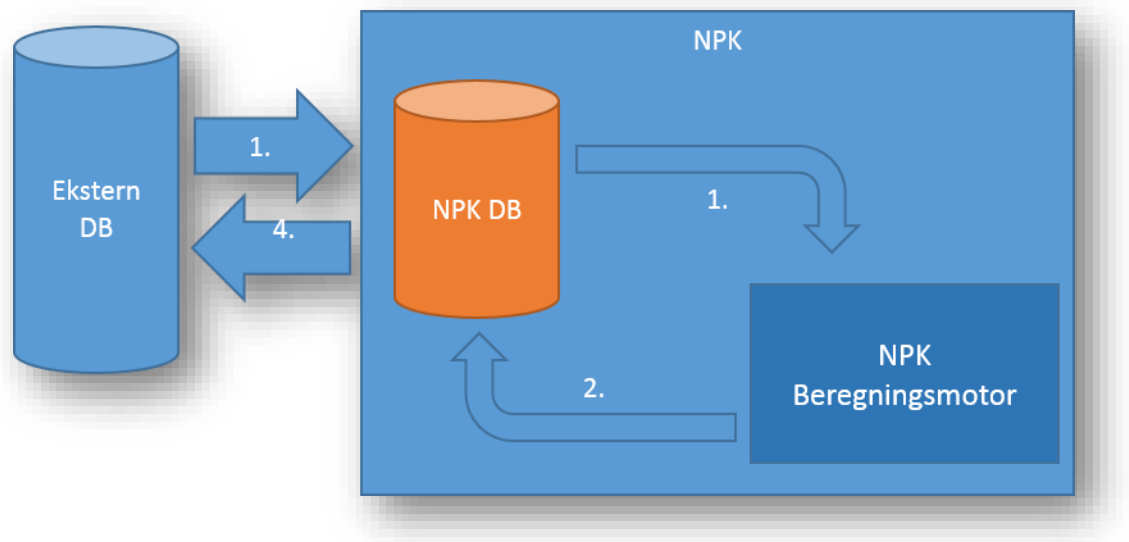
dbo.UnikeLegekontaktDRGKode

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	DRGKode	nvarchar	50	0	False	
FK	Opphold_Id	bigint	8	19	False	

Foreign Keys (dbo. UnikeLegekontaktDRGKode)

Name	Column Name	Reference
FK_UnikLegekontaktDRGKode_Opphold_Opphold_Id	Opphold_Id	dbo.Opphold

Databasegrensesnitt med ekstern database
Følgende diagram gir en oversikt over flyten av en kjørejobb:



Stegene i prosesseringen er som følger:

1. Pådragsdata (Brukerkonfigurasjon, samt Episoder eller Opphold) flyttes til NPK DB.
   Med tanke på avhengighetene mellom de ulike tabellene, må tabellene leses inn i gyldig rekkefølge. Ett av flere gyldige alternativer for rekkefølgen er:
   - Rapporteringsenhet
   - Pasient
   - Avdelingsopphold
   - Kontakt
   - Takst
   - Episode
   - Enhet
   - Helseperson
   - Prosedyre
   - Tilstand
   - Tidspunkt
   - Kode

   Brukerkonfigurasjonstabellene kan leses i vilkårlig rekkefølge.

2. Start av NPK
   NPK prosjektet har laget en referanse implementasjon for start av NPK med database kobling. Koden nedenfor viser hvordan dette kan gjøres:

```
var npk = new NPKStaging();
npk.Start();
```

Klassen «NPKStaging» er lokalisert i kodebiblioteket «NPK.dll». Det er også mulig å overstyre standard mappe for konfigurasjon ved å sende inn en egendefinert sti som en streng til metoden npk.Start().

3. Etter at prosessering er gjennomført uten feil, flyttes resultatene tilbake til NPK DB
4. Resultatene kan hentes fra NPK DB

7 Konfigurasjon

7.1 Brukerkonfigurasjon

Konfigurasjon som gir mulighet for variasjoner pr kjørejobb. Det følger med et standardkonfigurasjonssett for NPK som er knyttet til de ulike ISF regelverksversjonene. Se «NPK Beregningsregler» dokumentet for beskrivelse av innhold og struktur i brukerkonfigurasjonsfilene.

Standard brukerkonfigurasjonen består av disse filene:

- NPK_Parametere_[a-Å,0-9].csv
- NPK_DRGListe_[a-Å,0-9].csv
- NPK_Episodeseleksjon_[a-Å,0-9].csv
- NPK_EnhetOgDRGKombinasjoner_[a-Å,0-9].csv
- NPK_HelsepersonellOgDRGKombinasjoner_[a-Å,0-9].csv
- NPK_STGListe_[a-Å,0-9].csv
- NPK_Kodeunntaksliste_[a-Å,0-9].csv
- NPK_TFGListe_[a-Å,0-9].csv
- NPK_Kommuneliste_[yyyymmdd].csv

Det er laget et skript som en del av installasjonspakken som legger inn brukerkonfigurasjonen inn i NPK databasen. Se avsnitt for installasjonsvedlegg.

Vedrørende overgang for NPK_Kommuneliste fra systemkonfigurasjon til brukerkonfigursjon
Som følge av å endre NPK_Kommuneliste fra systemkonfigurasjonsfil til brukerkonfigurasjonsfil, må ved kjøring av NPK med Regelsettdato 01.01.21 eller mindre, NPK_Kommuneliste ligge blant brukerkonfigurasjonsfilene.

Vedrørende NPK_Parametere
Ved utvidelse av NPK_Parametere med en eller flere parametere, må disse parameterne inkluderes i NPK_Parametere i tidligere utgitte konfigurasjonssett, dersom man ønsker å benytte disse konfigurasjonssettene på gjeldene NPK-versjon. Kjøringen vil ellers feile.

Det er mulig å kjøre NPK på eldre konfigurasjonssett uten at nødvendigvis alle felt i NPK_Parametere er til stede.

Vedrørende spesifikke parametre i NPK_Parametere
Fra NPK versjon 8.0.25283.1 og med regelsettdato 1.1.2026, er det gjort mulig å innsette en UtDatoTid for en episode som mangler UtDatoTid. Dette krever at parameteren *InnsettUtDatoTid* i NPK_Parametere er satt til Ja. For en episode av type avdelingsopphold, vil da den datoen som er

spesifisert i parameteren *NyUtDatoTid* bli innsatt. For en episode av type kontakt vil episodens InnDatoTid bli benyttet. Dersom parameteren *InnsettUtDatoTid* er satt til Nei, vil det ikke settes inn noen ny verdi for episoder som mangler UtDatoTid.

Fra NPK versjon 10.0.26286.5 er det gjort konfigurerbart for hvilken metode man ønsker å benytte ved steg Leser episoder. I NPK_Parametere er det innført en parameter InnlesningEpisodeMinne som gir bruker mulighet til å bestemme om ny eller gammel metode benyttes. Default er gammel metode, men settes parameter til 1 benyttes ny metode.

Se eget avsnitt for duplikat- og konsistenssjekk av brukerkonfigurasjon for DRG og STG.

Definisjonsdata filene for DRGgruppering og EOCgruppering må ligge under Konfigurasjon. Systemet vil søke igjennom hele mappestrukturen under Konfigurasjon til den finner definisjonsdatafilene som er angitt i Parametere.

Standard mappesti til rotkatalogen for Konfigurasjon kan overstyres ved å sende inn mappestien til Initialiser metoden. Se eksempel under:

```
var npk = new NPK();
npk.Initialiser(Brukerkonfigurasjon brukerkonfigurasjon, "c:\\Konfigurasjon\\");
npk.Start(…);
```

Dette kan fra og med versjon 7.0.21347.1 oppnås ved å definere KonfigurasjonMappesti utenfor Initialiser metoden.

7.1.1 Gyldighetsperiode for en regel

Alle regler som er opprettet i NPK vil få et sett med attributter som vil definere hvorvidt en regel skal kjøres eller ikke. Hvilke regler som kjøres kan overstyres ved å opprette en tekstfil som må navngis «Regelgyldighetsperiode.txt» og legges i konfigurasjonsmappen. **Denne filen må fjernes igjen hvis man ikke lenger ønsker å overstyre regler.**

Bakgrunnen for denne konfigurasjonen er at man for eksempel kun skal kunne kjøre en regel for året 2018, og for det påfølgende året skal en annen versjon av regelen kjøres.

Attributtene er:
ID
Identifikator for en regel. Denne å være unik, og må generes av NPK når en ny regel opprettes. Denne må ikke endres i konfigurasjonsfilen. Da vil NPK automatisk opprette IDen på nytt. Riktig ID for regelen man ønsker å overstyre kan finnes ved å se i loggen til NPK.

BESKRIVELSE
Dette er en beskrivende tekst av regelen, og brukes for å enkelt kunne lese ut i fra filen hva regelen betyr og gjør. Det finnes ingen begrensing på hvor lang denne kan være. Det bør brukes bokstavene A-Å, og tallene 0-9. Det ikke implementert støtte for spesialtegn som for eksempel ; <, >, @. Beskrivelse kan heller ikke bestå av fnutter «». Parentes () er akseptert.

GYLDIGFRADATO
Dette er datoen som en regel skal være gyldig fra.

GYLDIGTILDATO
Dette er datoen brukes til å si hvor lagt frem i tid en regel skal være gyldig. En regel vil alltid få initial verdi 31.12.2099, og gjelder til og med denne datoen.

Skal en regel ha forskjellige gyldig fra dato eller gyldig til dato legges dette inn i fila slik:

```
{
        "Id": "d6ee847f-d9e9-41ad-b06c-519443b05670",
        "Beskrivelse": " Standard beskrivelse for dummyregel nr 1",
        "GyldigFraDato": "01.01.2014",
        "GyldigTilDato": "31.12.2014"
},
 {
        "Id": "33a4997a-7536-4737-ba94-5953f6ef559a",
        "Beskrivelse": "Standard beskrivelse for dummyregel versjon nr 2",
        "GyldigFraDato": "01.01.2015",
         "GyldigTilDato": "31.12.2099"
 }
```

Regler for json oppbyggingen:

- Det skal alltid være en [ ] rundt alt innholdet
- En regel skal alltid ha en { } rundt seg
- Regler skilles ved hjelp av komma (,)
- Navn på attributt og verdien skal alltid skrives med "" rundt seg
- Navn på attributt og verdi skal skilles med :
- Det har ingen betydning om det er mellomrom før og etter :
- Det har ingen betydelse om attributtene skrives med store eller små bokstaver
- En dato er på formatet dd.mm.åااا og det må brukes punktum i mellom dag måned og år (.)
- I beskrivelse kan man ikke benytte seg av tegnet "

7.1.2 Brukerkonfigurasjon tabeller

Brukerkonfigurasjonen ser slik ut i NPK databasen:

dbo.DRG

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	DRGKode	nvarchar	50	0	False	
	DRGNavn	nvarchar	255	0	False	
	Kostnadsvekt	decimal	19	3	True	
	TrimpunktØvre	int	4	10	True	
	ProsedyreinnholdsType	nvarchar	10	0	False	
	DRGBasispoeng	decimal	19	3	True	
	HDGKode	int	4	10	True	
	ErGyldigForSekundærRehabilitering	bit	1	1	True	
	DRGTjenestetype	nvarchar	100	0	False	
	ErGyldigForPoengtilleggAmbulantBehandling	bit	1	1	True	

dbo.EnhetOgDRGKombinasjoner

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	DRGTjenestetype	nvarchar	100	0	False	
	Organisasjonsnummer	nvarchar	50	0	False	

dbo.EpisodeseleksjonVilkår

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	VilkårNr	int	4	10	True	
	ErGyldigForOppholdskonstruksjon	bit	1	1	True	
	ErDominantForOppholdsbeskrivelse	Bit	1	1	True	

dbo.HelsepersonellOgDRGKombinasjoner

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	DRGKode	nvarchar	50	0	False	
	Helsepersonellkategori	int	4	10	True	

dbo.KodeUnntak

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	Kode	nvarchar	50	0	True	
	Unntakstype	nvarchar	100	0	True	

dbo.Kommune

Key	Name	Data Type	Length	Precision	Not Null	Identity

PK	Id	bigint	8	19	True	1 - 1
	Nummer	nvarchar	50	0	True	
	Navn	nvarchar	100	0	True	
	GyldigFraDato	date	3	10	True	
	GyldigTilDato	date	3	10	True	

dbo.Parametere

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	RegelsettDato	datetime2	8	23	True	
	TidsgrenseForEpisoderITimer	int	4	10	True	
	OrganisatoriskNivå	nvarchar	50	0	True	
	DRGKomponentProduktversjon	nvarchar	50	0	False	
	TrimpunktGrense	int	4	10	True	
	DøgnskilleTerskelMinimum	int	4	10	True	
	DøgnskilleTerskelMaksimum	int	4	10	True	
	PoengPerDøgnskille	decimal	18	2	True	
	Kjørejobbsekvens	nvarchar	50	0	True	
	GyldigPeriodeForISFFraDato	datetime2	8	23	True	
	GyldigPeriodeForISFTilDato	datetime2	8	23	True	
	Inndatatype	nvarchar	50	0	True	
	LagreDRGGrupperingStreng	bit	1	1	True	
	OppholdsgrupperingMedUtvalgteProsedyrer	bit	1	1	True	
	RituellOmskjæring_Poengfradrag	decimal	19	3	True	
	DefinisjonsdataForSTG	nvarchar	255	0	False	
	DefinisjonsdataForDRG	nvarchar	255	0	False	
	SærskiltAmbulantKonsultasjon_Poengtillegg	decimal	19	3	True	
	PalliativBehandling_Poengtillegg	decimal	19	3	False	
	DefinisjonsdataForTFG	nvarchar	Max	0	False	
	EOCDefinisjonsdataForTjenesteforløp	nvarchar	Max	0	False	
	TverrfagligeHelsetilbud_Poengtillegg	decimal	19	3	False	
	SærskiltAmbulantKonsultasjon_PoengtilleggPHVTSB	decimal	19	3	False	
	Egenandel_Poengtillegg	decimal	19	3	False	
	InnsettUtDatoTid	bit	1	1	False	
	NyUtDatoTid	datetime2	19	3	False	
	NyInnDatoTid	datetime2	19	3	False	
	InnlesningEpisodeMinne	bit	1	1	False	

dbo.STG

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	STGKode	nvarchar	50	0	False	
	STGNavn	nvarchar	255	0	False	
	Periode	nvarchar	50	0	False	
	HDGKode	int	4	10	True	
	STGBasisPoeng	decimal	19	3	True	
	STGGruppeKode	int	4	10	True	

	STGGruppeNavn	nvarchar	255	0	False	

dbo.TFG

Key	Name	Data Type	Length	Precision	Not Null	Identity
PK	Id	bigint	8	19	True	1 - 1
	TFGKode	nvarchar	50	0	False	
	TFGNavn	nvarchar	255	0	False	
	Periode	nvarchar	50	0	False	
	HDGKode	int	4	10	True	
	Kostnadsvekt	decimal	19	3	True	
	TFGBasispoeng	decimal	19	3	True	

8 Installasjon

Kapittelet beskriver hvordan installasjonsprosedyrene for NPK.

8.1 Installasjonspakken

NPK installasjonspakken leveres som en ZIP fil med følgende format:

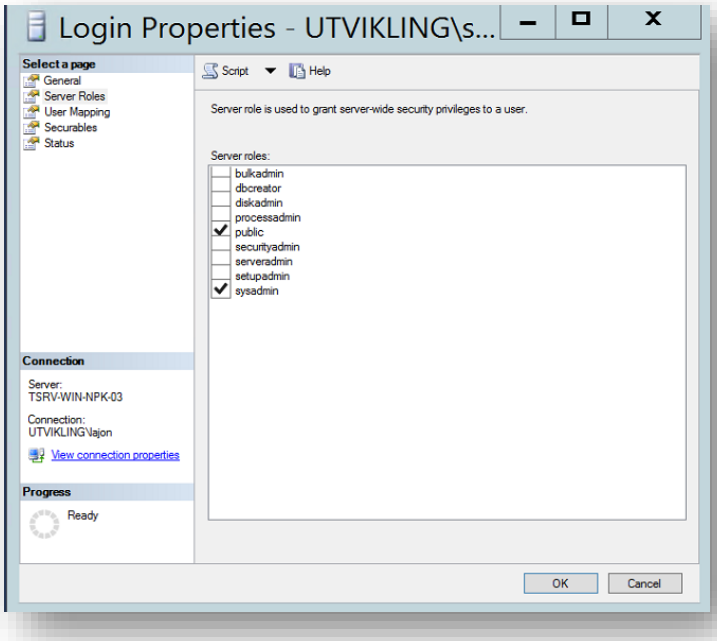
«NPK.Støtteklient.Setup.[versionsnummer].zip»

Start med å pakke ut zip filen til en temporær katalog f.eks. «*C:\temp\NPK*»

8.1.1 Krav til rettigheter ved installasjon

Det er knyttet noen krav til rettigheter for den bruker som utfører installasjon av NPK.

1. Rettigheter for installasjon av NPK komponenter:
    a. Lokal maskinadministrator.
2. Database rettigheter. Dette er kun påkrevd hvis NPK skal integreres med en lokal database instans.
    a. Sysadmin/Public på lokal database instans:



8.2 Installasjon av NPK

Installasjonssteg:

1. Skift til den katalogen NPK installasjonspakken er pakket ut.
2. Om NPK har vært installert fra før, naviger til rotnivå og slett mappe net8.0 for å unngå at gamle filer blir liggende.

3. Kjør NPK installasjonsfilen «*NPK.Støtteklient.Setup.msi*». NPK blir installert som en 64 bits Windows applikasjon under «C:\Program Files\Helsedirektoratet\NPK».
4. Opprettelse av NPK databasen:
   a. Valider at «connectionstring» feltet i «appsettings.json» er korrekt for det lokale miljøet. *«appsettings.json»* er installert i «c:\program files\Helsedirektoratet\NPK»:

```
"ConnectionStrings": {
    "NpkConnectionString": "data source=localhost;initial
catalog=NPK;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"
}
```

   b. Start kommandolinjen i Administrator modus og kjør følgende kommandoen «*NPK.Støtteklient.exe 1».* Da blir NPK databasen opprettet. MERK (gyldig fra versjon: 8.0.25283.1): Hvis man ikke har rettigheter til å opprette selve databasen, men den eksisterer fra før, kan man bruke argument 4 - «*NPK.Støtteklient.exe 4*». Da vil programmet ikke forsøke å opprette database, bare fylle med tabeller etc.
5. I appsettings.json bør det for parameter "AppDataPath" presiseres hvor AppData skal ligge (mappen brukes til mellomlagring av data og er default c:\\NPKAppData).

9 Feilhåndtering

Det er lagt til grunn at NPK skal anvende med ferdig validerte og kvalitetssikrede data, jf. krav til rapportering til NPR. Prinsippet for feilhåndtering i NPK er derfor at systemet vil avslutte en beregning ved første unntakshendelse; feile tidlig og hardt. For mer informasjon om logging av unntakshendelser se avsnitt for Logging.

9.1 Duplikat- og konsistenssjekk av brukerkonfigurasjon

NPK.Initialiser metoden har duplikatsjekk av DRG-liste og STG-liste. Hvis listene inneholder duplikate koder vil applikasjonen feile og det skrives en feilmelding til loggen med hvilke koder som det gjelder.

Denne metoden har også konsistenssjekk mellom DRG-liste og Json-definisjoner, samt mellom STG-liste og Jsondefinisjoner. Denne kjøres når Regelsettdato >= 01.01.2017. Hvis det er inkonsistens feiler applikasjonen og det skrives en feilmelding til loggen med hvilke koder som det gjelder.

NB! Sjekkene utføres ikke hvis listene ikke inneholder data.

10 Logging

10.1 Rammeverk

Løsningen bruker rammeverket log4net for logging (http://logging.apache.org/log4net/index.html). Log4net støtter logging til flere typer målformater (konsoll, database, fil m.fl.) og logg formatteringer. Logging kan styres gjennom konfigurasjon, noe som innebærer at logging kan overstyres i den lokale installasjonen. Konfigurasjonsfil for logging (log4net.config) ligger i NPK installasjonskatalogen.

10.2 Målformater

NPK kommer med støtte for følgende målformater:

- Fil, formater
- Konsoll
- Hendelseslogg
- Database

For å få logget til hendelsesloggen må klientapplikasjonen kjøres som administrator én gang. Dette er for å få opprettet klientapplikasjonen som kilde (event source) i Windows sin hendelseslogg (Event Viewer). Deretter kan klientapplikasjonen kjøres under en hvilken som helst bruker som har tilstrekkelig rettigheter til NPK databasen.

Standard katalog for loggfiler er «C:\NPK.Logg\». Det logges også til databasen i NPK.

10.2.1 Nivåer

Det er implementert logging for disse nivåene:

- DEBUG, logging under utvikling/testing. Dette nivået er slått av i installasjonspakken.
- INFO, logg av generell beregningsinformasjon, se avsnitt for logginformasjon.
- ERROR, logg av feilmeldinger.
- KONFIGURASJON (nytt fra NPK versjon 8.0.25283.1), logg av konfigurasjonsdetaljer

10.2.2 Databasekonfigurasjon for logging

Konfigurasjonsfil for logging (log4net.config) ligger i NPK installasjonskatalogen. Inne i denne filen finner du en node som heter
<appender name="AdoNetAppender" type="log4net.Appender.AdoNetAppender">
Under denne finnes en node som heter connectionString, og her må du kontrollere om den peker på riktig databaseinstans under data source.

10.2.2.1 Nærmere om connectionString i ulike NPK-versjoner

10.2.2.1.1 NET 10

Merk at log4net.config har egen connectionString. Denne kan være ulik connectionString i appsettings.json om ønskelig. Den har i utgangspunktet samme samme verdi som appsettings.json.

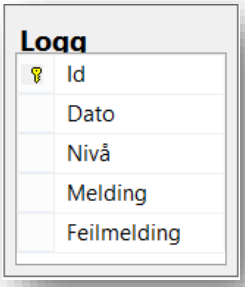
10.3 Logg format

NPK er konfigurert med følgende logg format:

```
%date{yyyy-MM-dd HH:mm:ss} %level - %message%newline
```

%level% : Logg nivå
%message%: Logg beskrivelse

Det logges akkurat det samme som det logges i loggfilen. Database tabellen for logging ser slik ut:



10.3.1 Logginformasjon (kort revisjon etter sjekk)

Følgende informasjon blir logget på INFO nivå:

- Mengde data og tidsforbruk som leses inn i minne under steg for *Leser episoder*
- Antall episoder som er lest inn*
- Antall Opphold som genereres
- Regler som blir brukt for kjørejobben
- Progresjon pr. modul (logges underveis)
- Tidsforbruk pr. modul (starttid, stopptid)
- Tidsforbruk for lagring
- Total tidsforbruk
- Minneforbruk

*Merk at fra versjon 10.0.26286.5 er det for steget Leser episoder lagt til ekstra logging.

Følgende informasjon logges på KONFIGURASJON nivå:

- Relevante konfigurasjonsparametere (RegelsettDato, Kjørejobbsekvens) mm
- NPK-versjon, DRG-versjon og EOC-versjon

10.3.2 Visning av loggfiler

Følgende verktøy kan brukes for å lese loggen:

- Notepad
  YALV, Windows viewer (https://github.com/LukePet/YALV

11 Andre tekniske detaljer

Fra NPK versjon 8.0.22334.1 er referanse noen komponenter endret.

Referanse til DRG.dll ogEOCConstructionTool.dll i NPK er utvidet, slik at full Path benyttes.

Vedlegg A - Mapping mellom NPK og NPR melding

Entitet / Tabell	Feltnavn i Nye NPK	Navn i NPR melding	Kommentar
Episode	Id		Primærnøkkel i databasen, big int
Episode	EpisodeId	Episode.episodeID	
Episode	InnDatoTid	Episode.innDatoTid	
Episode	UtDatoTid	Episode.utDatoTid	
Episode	FolkeregistrertKommune	Episode.komNrHjem	
Episode	AlderIDager	Episode.alderIDager	
Episode	TilSted	Episode.tilSted	
Episode	UtTilstand	Episode. utTilstand	
Episode	Debitor	Episode.debitor	
Episode	Omsorgsnivå	Episode.omsorgsniva	
Episode	Inntilstand	Episode.inntilstand	
Episode	InnmåteHastegrad	Episode. innmateHast	
Avdelingsopphold (AvdOpp)	Id		Primærnøkkel i databasen, big int
Enhet	Id		Primærnøkkel i databasen, big int
Enhet	Type	RefEnhet.typeEnhet	
Enhet	OffisiellAvdelingskode	Enhet.offAvdKode	
Enhet	ISFRefusjon	Enhet.isfRefusjon	
Kontakt	Id		Primærnøkkel i databasen, big int
Kontakt	Type	Kontakt.kontakttype	
Kontakt	Aktivitetssted	Kontakt.stedAktivitet	
Kontakt	Aktivitetstype	Kontakt.polkonAktivitet	
Kontakt	IndirekteAktivitet	Kontakt.polIndir	
Helseperson	Id		Primærnøkkel i databasen, big int
Helseperson	UtførendeHelsepersonell	Helseperson.polUtforende	
Helseperson	Rolle	Helseperson.Rolle	
Takst	Id		Primærnøkkel i databasen, big int
Takst	Takstnummer	Takst.takstnummer	
Rapporteringsenhet (Institusjon)	Id		Primærnøkkel i databasen, big int
Rapporteringsenhet (Institusjon)	Rapporteringsenhetsnummer	Institusjon.institusjonID	
Rapporteringsenhet	Foretaksnummer	Finnes ikke	

(Institusjon)			
Pasient	Id		Primærnøkkel i databasen, big int
Pasient	Nr	Pasient.pasientNr	
Pasient	Fødselsår	Pasient.fodselsar	
Pasient	KjønnKode	Pasient.kjonn	
Tilstand	Id		Primærnøkkel i databasen, big int
Tilstand	Akse	Tilstand.akse	
Tilstand	TilstandNr	Tilstand.tilstNr	
Prosedyre	Id		Primærnøkkel i databasen, big int
Prosedyre	ProsedyreNr	Prosedyre.prosNr	
Tidspunkt	Id		Primærnøkkel i databasen, big int
Tidspunkt	Type	Tidspunkt.typeTidspunkt	
Tidspunkt	Verdi	Tidspunkt.tidspunkt	
Kode	Id		Primærnøkkel i databasen, big int
Kode	KodeNr	Tilstand.Kode.kodeNr eller Prosedyre.Kode.kodeNr	
Kode	Verdi	Tilstand.Kode.kodeVerdi eller Prosedyre.Kode.kodeVerdi	
Kode	Kodeverk	Tilstand.Kode.Kodeverk eller Prosedyre.Kode.KodeVerk	

Vedlegg B – Oppsett og kjøring av NPK.Støtteklient

Vedlegget beskriver oppsett og kjøring av NPK.Støtteklient.

B1 NPK

Installere NPK (Se eget kapittel (0)).

B1.1 Integrert grupperingskomponent

For gruppering av episoder, opphold og særtjenester benyttes en egen grupperingskomponent som er integrert i NPK. Den krever at det ligger egne definisjonsdatafiler på JSON format under konfigurasjonsmappen. Navnet på filen angis for episoder og opphold i Brukerkonfigurasjon.Parametere.DefinisjonsdataForDRG, for særtjenester i Brukerkonfigurasjon.Parametere.DefinisjonsdataForSTG og for tjenesteforløp i Brukerkonfigurasjon.Parametere.DefinisjonsdataForTFG. I tillegg angis definisjonsdatafilen for EOC.ConstructionTool i Brukerkonfigurasjon.Parametere. EOCDefinisjonsdataForTjenesteforløp. Filene kan lastes ned her:

https://helsedirektoratet.no/finansieringsordninger/innsatsstyrt-finansiering-isf-og-drg-systemet/norsk-pasientklassifisering-npk

B1.2 Innlesing av konfigurasjon

Ved første gangs kjøring av Støtteklienten hvor denne skal opprette databasen navngitt i config filen, er det behov for å laste inn konfigurasjonsdata inn til tabellen dbo. Parametere. Dette gjøres ved å sette parameteren «LasteBrukerkonfigurasjonFraFil» i filen «C:\Program Files\NPK\net8.0\appsettings.json» til «Ja».

NPK-støtteklienten kan lese konfigurasjon direkte fra filer og fylle disse inn i NPK-databasen. Evt. kan man selv fylle NPK-databasen med konfigurasjonen. Dersom man ønsker å lese inn konfigurasjonsfilene må filene legges under:

C:\Program Files\Helsedirektoratet\NPK\Konfigurasjon\

For å kjøre NPK.Støtteklienten med data lagt inn i databasen*:

```
C:\Program Files\Helsedirektoratet\NPK>NPK.Støtteklient.exe
```

*Dersom databasen eksisterer fra før og man ikke ønsker å trunkere transaksjonsloggen, kan bruke argument 3 - «NPK.Støtteklient.exe 3» (se for øvrig kapittel 6.2.1.1).

B1.3 Sjekke loggfil

Etter kjøring kan man sjekke loggfilen som ligger der den er spesifisert i «C:\Program Files\NPK\net8.0\appsettings.json» under «LoggFilMappesti». Default vil loggfilene ligge i mappen «C:\NPK.Logg\» og hete «NpkLog.txt» og «NpkLogYalv.xml».

B1.4 Hente resultat fra NPK basen

Følgende SQL-kode kan brukes for å hente data fra NPK-databasen:

```
USE NPK
GO
SELECT * FROM dbo.DRGBeregningResultat
SELECT * FROM dbo.EpisodeOppholdKobling
SELECT * FROM dbo.EpisodeResultat
SELECT * FROM dbo.Kode
SELECT * FROM dbo.Logg
SELECT * FROM dbo.Opphold
SELECT * FROM dbo.OppholdPoeng
SELECT * FROM dbo.Prosedyre
SELECT * FROM dbo.Tilstand
SELECT * FROM dbo.Helseperson
SELECT * FROM dbo.UnikLegekontaktDRGKode
SELECT * FROM dbo.Takst
SELECT * FROM dbo.Rapporteringsenhet
SELECT * FROM dbo.Særtjeneste
SELECT * FROM dbo.EpisodeSærtjenesteKobling
SELECT * FROM dbo.Tjenesteforløp
SELECT * FROM dbo.TjenesteforløpPoeng
SELECT * FROM dbo.OppholdSærtjenesteTjenesteforløpKobling
```

Vedlegg C – Sletteskript for å slette konfigurasjon og data i databasen

C1 Slett konfigurasjon

```
ALTER TABLE dbo.DRGBeregningResultat  NOCHECK CONSTRAINT
FK_DRGBeregningResultat_DRG_DRG_Id
DELETE FROM [dbo].[DRG];
ALTER TABLE dbo.DRGBeregningResultat CHECK CONSTRAINT
FK_DRGBeregningResultat_DRG_DRG_Id
ALTER TABLE dbo.Særtjeneste NOCHECK CONSTRAINT FK_Særtjeneste_STG_STG_Id
ALTER TABLE dbo.DRGBeregningResultat  NOCHECK CONSTRAINT
FK_DRGBeregningResultat_STG_STG_Id
DELETE FROM [dbo].[STG];
ALTER TABLE dbo.Særtjeneste CHECK CONSTRAINT FK_Særtjeneste_STG_STG_Id
ALTER TABLE dbo.DRGBeregningResultat  CHECK CONSTRAINT
FK_DRGBeregningResultat_STG_STG_Id
ALTER TABLE dbo.Tjenesteforløp NOCHECK CONSTRAINT FK_Tjenesteforløp_TFG_TFG_Id
ALTER TABLE dbo.DRGBeregningResultat  NOCHECK CONSTRAINT
FK_DRGBeregningResultat_TFG_TFG_Id
DELETE FROM [dbo].[TFG];
ALTER TABLE dbo.Tjenesteforløp CHECK CONSTRAINT FK_Tjenesteforløp_TFG_TFG_Id
ALTER TABLE dbo.DRGBeregningResultat  CHECK CONSTRAINT
FK_DRGBeregningResultat_TFG_TFG_Id
TRUNCATE TABLE [dbo].[EpisodeseleksjonVilkår];
TRUNCATE TABLE [dbo].[HelsepersonellOgDRGKombinasjoner];
TRUNCATE TABLE [dbo].[EnhetOgDRGKombinasjoner];
TRUNCATE TABLE [dbo].[KodeUnntak];
TRUNCATE TABLE [dbo].[Kommune];
DELETE FROM [dbo].[Parametere];
```

C2 Slett data

```
TRUNCATE TABLE [dbo].[Kode];
DELETE FROM [dbo].[Prosedyre];
DELETE FROM [dbo].[Tilstand];
DELETE FROM [dbo].[Enhet];
DELETE FROM [dbo].[Tidspunkt];
DELETE FROM [dbo].[EpisodeOppholdKobling];
DELETE FROM [dbo].[EpisodeSærtjenesteKobling];
DELETE FROM [dbo].[Helseperson];
DELETE FROM [dbo].[Takst];
DELETE FROM [dbo].[UnikLegekontaktDRGKode];
DELETE FROM [dbo].[EpisodeResultat];
DELETE FROM [dbo].[OppholdSærtjenesteTjenesteforløpKobling];
ALTER TABLE dbo.Særtjeneste  NOCHECK CONSTRAINT
FK_Særtjeneste_Tjenesteforløp_Tjenesteforløp_Id
ALTER TABLE dbo.opphold  NOCHECK CONSTRAINT
FK_Opphold_Tjenesteforløp_Tjenesteforløp_Id
DELETE FROM [dbo].[Tjenesteforløp];
ALTER TABLE dbo.Særtjeneste  CHECK CONSTRAINT
FK_Særtjeneste_Tjenesteforløp_Tjenesteforløp_Id
ALTER TABLE dbo.opphold  CHECK CONSTRAINT FK_Opphold_Tjenesteforløp_Tjenesteforløp_Id
DELETE FROM [dbo].[TjenesteforløpPoeng];
DELETE FROM [dbo].[Opphold];
DELETE FROM [dbo].[Episode];
DELETE FROM [dbo].[Særtjeneste];
DELETE FROM [dbo].[HReseptOppgjørsdata];
DELETE FROM [dbo].[Pasient];
DELETE FROM [dbo].[Avdelingsopphold];
DELETE FROM [dbo].[Takst];
DELETE FROM [dbo].[Kontakt];
DELETE FROM [dbo].[DRGBeregningResultat];
DELETE FROM [dbo].[OppholdPoeng];
DELETE FROM [dbo].[Rapporteringsenhet];
```

Vedlegg D – Sletteskript for å slette kun resultater i NPK databasen

Hvis man ønsker å kjøre en gang til med det samme datasettet kan man bruke følgende skript for å slette kun resultatene fra forrige kjøring:

```
EXEC sp_msforeachtable 'ALTER TABLE ? NOCHECK CONSTRAINT all'
truncate table EpisodeOppholdKobling
truncate table oppholdsærtjenestetjenesteforløpkobling
delete from Kode where Tilstand_Id in (select Id from Tilstand where Opphold_Id is not
null)
delete from Kode where Prosedyre_Id in (select Id from Prosedyre where Opphold_Id is
not null)
delete from Kode where Tilstand_Id in (select Id from Tilstand where Særtjeneste_Id is
not null)
delete from Tilstand where Opphold_Id is not null
delete from Tilstand where Særtjeneste_Id is not null
update Tilstand SET ErHovedtilstand = NULL where Opphold_Id is null
delete from Prosedyre where Opphold_Id is not null
delete from Helseperson where Opphold_Id is not null
delete from Takst where Opphold_Id is not null
delete from Takst where Særtjeneste_Id is not null
delete from UnikLegekontaktDRGKode where Opphold_Id is not null
delete from tjenesteforløp
delete from tjenesteforløppoeng
delete from Opphold
delete from OppholdPoeng
truncate table EpisodeResultat
delete from DRGBeregningResultat
delete from EpisodeSærtjenesteKobling
delete from særtjeneste
EXEC sp_msforeachtable 'ALTER TABLE ? WITH CHECK CHECK CONSTRAINT all'
```