

# Handbok Okteta

**Friedrich W. H. Kossebau**  
**Alex Richardson**  
**Översättare: Stefan Asserhäll**



## Handbok Okteta

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>6</b>
<b>2</b>	<b>Grunderna</b>	<b>7</b>
2.1	Starta Okteta . . . . .	7
2.2	Användning . . . . .	7
<b>3</b>	<b>Verktyg</b>	<b>8</b>
3.1	Översikt . . . . .	8
3.1.1	Analysatorer och hanterare . . . . .	8
3.1.2	Allmänna verktyg . . . . .	9
3.2	Strukturverktyg . . . . .	9
3.2.1	Allmänt . . . . .	9
3.2.2	Installera strukturdefinitioner . . . . .	10
3.2.2.1	Installera med användning av Heta Nyheter . . . . .	10
3.2.2.2	Installerar strukturdefinitioner manuellt . . . . .	10
3.2.2.3	Använda de nyinstallerade strukturerna . . . . .	10
3.2.3	Dela strukturdefinitioner . . . . .	10
3.2.4	Skapa strukturdefinitioner . . . . .	10
3.2.5	Strukturdefinitionen med XML-filformat . . . . .	11
3.2.6	Ett exempel på en strukturdefinition både i XML och Javascript . . . . .	13
3.2.6.1	Det gemensamma steget som delas av båda tillvägagångssätten . . . . .	13
3.2.6.2	En enkel XML-strukturdefinition . . . . .	13
3.2.6.3	Den enkla strukturen i Javascript . . . . .	14
3.2.6.4	Mer komplexa strukturer . . . . .	14
3.2.6.5	Ytterligare information . . . . .	15
<b>4</b>	<b>Översikt av gränssnittet</b>	<b>16</b>
4.1	Menyalternativ . . . . .	16
4.1.1	Menyn Arkiv . . . . .	16
4.1.2	Menyn Redigera . . . . .	17
4.1.3	Menyn Visa . . . . .	17
4.1.4	Menyn Fönster . . . . .	18
4.1.5	Menyn Bokmärken . . . . .	19
4.1.6	Menyn Verktyg . . . . .	19
4.1.7	Menyn Inställningar . . . . .	19



### **Sammanfattning**

Okteta är en enkel editor för rådata i filer. Den här sortens program kallas också hexadecimal editor eller binäreditor.

## Kapitel 1

# Inledning

Okteta är en enkel editor för rådata i filer.

Data visas med två varianter: som det numeriska värdet av varje byte och som tecknen tilldelade till värdena. Värden och tecken kan antingen visas i två kolumner (det traditionella sättet i hexadecimala editorer) eller i rader med värdet ovanför tecknet. Redigering kan göras både för värdet och för tecknen.

Förutom de vanliga redigeringsmöjligheterna ger Okteta också en liten verktygsuppsättning, liksom en tabell som listar avkodning till vanliga enkla datatyper, en tabell som listar alla möjliga byte med ekvivalenta tecken och värden, en informationsvy med statistik, en checksummeräkare, ett filterverktyg och ett verktyg för att extrahera av strängar.

Alla ändringar av inläst data kan ångras eller göras om utan begränsning.

## Kapitel 2

# Grunderna

### 2.1 Starta Okteta

Skriv **okteta** på kommandoraden eller välj **Binäreditor** i gruppen **Program** → **Verktyg** i programstarten.

De vanliga kommandoradsväljarna i Qt™ och KDE Ramverk 5 är tillgängliga, och kan visas genom att skriva in **okteta --help**.

Kommandoradsväljare specifika för Okteta är:

<webbadresser> - öppna filer angivna av webbadresserna

### 2.2 Användning

Huvudfönstret i Okteta har följande komponenter: en menyrad, en verktygsrad, en statusrad, en eller flera sidorader med verktyg, och huvudvyn med datavyerna i flikar.

När en fil öppnas eller ett nytt byte-fält skapas, listas alla byte som ingår i följd på rader med ett givet antal byte per rad. De visas med två varianter: som det numeriska värdet av varje byte och som tecknen tilldelade till värdena. Värden och tecken kan antingen visas i två kolumner eller i rader med värdet ovanför tecknet. På vänster sida visas positionerna för första byten på varje rad.

Hantering av data liknar den i de flesta texteditorer: data kan redigeras, klippas ut, kopieras, klistras in, dras och släppas, på i stort sett samma sätt som text kan i editorerna. En markör anger aktuell position. Genom att trycka på tangenten **Insert** byter man mellan ersätta och infoga. Ersättning är striktare än i texteditorer, eftersom det inte tillåter någon åtgärd som ändrar byte-fältets storlek.

I motsats till texteditorer visas innehållet med två varianter. Bara en av dem är aktivt när det gäller ny inmatning. Det finns två länkade markörer som visas värden och tecken, där den aktiva markören blinkar. När tecknen är aktiva, kan tecken skrivas in på känt sätt från texteditorer. Med värden aktiva, visas en minimal editor när en siffra skrivs in för att skriva in resten av värdet.

Sökdialogrutan låter användaren söka efter en specifik byte-sträng, som kan definieras som värden (hexadecimal, decimal, oktal, binär) eller text (aktuell 8-bitars kodning eller UTF-8).

Flera byte-fält kan vara öppna samtidigt, men bara ett kan vara aktivt. Använd menyn **Fönster** för att välja vilket byte-fält som ska vara aktivt.

## Kapitel 3

# Verktyg

### 3.1 Översikt

Okteta tillhandahåller några verktyg, vissa för att analysera och hantera byte-fälten och vissa med allmänare syfte. Verktygen kan aktiveras eller inaktiveras från menyn **Verktyg** i menyraden. Varje verktyg har en liten vy, som antingen finns i en av sidoraderna eller flyttas fritt som ett fönster. Man kan docka, frigöra, arrangera om och dessutom lägga verktygen ovanpå varandra med musen. Gör det genom att hålla nere vänster musknapp på verktygsvyns namnlist, flytta omkring den som du vill, och släppa vänster musknapp för att göra färdigt åtgärden, eller annars avbryta den med tangenten Esc.

#### 3.1.1 Analysatorer och hanterare

##### Värde- eller teckentabell

Tabellen anger alla möjliga byte-värden, både som tecken och med olika numeriska kodningar.

Det valda värdet kan infogas på markörens position ett definierat antal gånger. Det kan åstadkommas genom att använda knappen **Infoga** eller genom att dubbelklicka på en rad i tabellen.

##### Binärfilter

Filtret utför binära operationer på markerade byte-värden. Efter att ha valt operation (OCH, ELLER, ROTERA, ...) kan parametrarna, om några finns, anges i rutan nedanför. Filtret tillämpas när knappen Filtrera klickas.

##### Strängar

Verktöget lokaliserar strängar i markerade byte-värden. Efter att ha valt minimal stränglängd, letas efter strängarna vid användning av knappen **Extrahera**. Listan med visade strängar kan begränsas genom att skriva in en filtreringsterm.

##### Statistik

Verktöget skapar statistik om markerade byte-värden. Statistiken anger förekomstfrekvensen för varje byte-värde i markeringen. Den kan beräknas genom att använda knappen **Bygg**.

##### Checksumma

Verktöget beräknar diverse checksummor eller kondensat för markerade byte-värden. Efter att ha valt algoritm och ställt in parametrarna, om några finns, beräknas summan när knappen **Beräkna** används.



### Avkodningstabell

Tabellen visar värdet för en eller flera byte med början vid markören för vissa vanliga enkla datatyper som heltal eller flyttal, men också UTF-8. Att dubbelklicka på en rad i tabellen visar en editor, så att värdet kan redigeras och ändras.

### Strukturer

Verktyget gör det möjligt att undersöka och redigera byte-fält baserade på användarskapade strukturdefinitioner. Detaljerade instruktioner finns i ett eget [avsnitt](#).

## 3.1.2 Allmänna verktyg

### Filsystem

Verktyget erbjuder en inbyggd filbläddrare som kan användas för att välja filer att öppna.

### Dokument

Verktyget visar alla filers om för närvarande har skapats eller lästs in. Symboler markerar filen med den för närvarande aktiva vyn, och visar också vilka filer som har osparade ändringar, eller vilka inlästa kopior som har ändrats av andra program.

### Bokmärken

Verktyget kan användas för att hantera bokmärken, som alternativ till menyn [Bokmärken](#).

#### NOT

Bokmärken är för närvarande bara tillfälliga och sparas inte om ett byte-fält eller hela programmet stängs.

### Filinformation

Verktyget visar viss information om den nuvarande filen, inklusive dess typ, platsen den är lagrad och dess storlek.

### Terminal

En inbäddad terminal, arbetskatalogen kopplas inte med den aktiva filen.

### Konvertering av teckenuppsättning

Verktyget skriver om byte så att respektive tecken är samma som med den andra teckenuppsättningen. Bara 8-bitars teckenuppsättningar stöds, och tecken som inte har någon motsvarighet ersätts för närvarande av ett värde fast kodat till 0.

## 3.2 Strukturverktyg

### 3.2.1 Allmänt

Strukturverktyget gör det möjligt att undersöka och redigera byte-fält baserade på användarskapade strukturdefinitioner, som kan skapas av fält, unioner, enkla typer och uppräkningsvärden.

Det har sin egen inställningsdialogruta, som kan nås genom att använda knappen **Inställningar**. Det finns diverse alternativ som kan ställas in, som stilen (decimal, hexadecimal eller binär) som värden visas med. Dessutom är det möjligt att välja vilka strukturdefinitioner som laddas och vilka strukturer som visas i vyn.

Strukturer definieras i Oktetas strukturdefinitionsfiler (baserade på XML, med filändelsen `.osd`). Dessutom innehåller en `.desktop`-fil metadata om strukturbeskrivningsfilen, som upphovsman, hemsida och licens.

För närvarande finns det inget stöd för att skapa eller redigera strukturdefinitioner. Därför måste det göras manuellt, som beskrivs i nästa avsnitt.

## 3.2.2 Installera strukturdefinitioner

### 3.2.2.1 Installera med användning av Heta Nyheter

Det enklaste sättet att installera nya strukturdefinitioner är genom att använda det inbyggda stödet för Heta Nyheter i Okteta. Öppna inställningsdialogrutan i strukturverktyget för att installera en befintlig struktur. Välj därefter fliken **Strukturhantering** och klicka på knappen **Hämta nya strukturer...** Dialogrutan som nu visas låter dig installera och avinstallera strukturer.

### 3.2.2.2 Installerar strukturdefinitioner manuellt

Strukturverktyget letar efter strukturbeskrivningar i underkatalogen `okteta/structures/` i användarens katalog för programdata (den kan hittas genom att köra `qtpaths --paths GenericDataLocation`). Katalogen kanske måste skapas om inga strukturdefinitioner har installerats ännu.

Det finns två filer i varje strukturdefinition: En fil för själva definitionen och en `.desktop`-fil för metadata (upphovsman, version, etc.).

I den katalogen finns en underkatalog för varje strukturdefinition, som både innehåller `.desktop`-filen och `.osd`-filen eller filen `main.js` för den definitionen.

Exempelvis med programdatakatalogen `qtpaths --paths GenericDataLocation` och en strukturdefinition vid namn Exempelstruktur, finns katalogen `okteta/structures/Exempelstruktur` som innehåller filen `Exempelstruktur.desktop` och filen `Exempelstruktur.osd`.

### 3.2.2.3 Använda de nyinstallerade strukturerna

Efter att ha installerat en ny strukturdefinition, måste du starta om Okteta innan den kan användas. När Okteta väl har startat, öppna strukturverktygets inställningsdialogruta. Välj där fliken **Strukturhantering** och försäkra dig om att relevant strukturdefinition är markerad. Byt därefter till fliken **Strukturer** och försäkra dig om att önskat element listas på höger sida.

## 3.2.3 Dela strukturdefinitioner

För vanliga strukturer kanske du inte behöver skapa en egen definition, utan istället kan återanvända en redan befintlig definition från ställen som [store.kde.org](https://store.kde.org).

Du kanske också vill dela med dig av en egen definition. För att göra det, skapa ett filarkiv (t.ex. ett komprimerat tar-arkiv, `.tar.gz`) som bara innehåller underkatalogen med `.desktop`-filen och strukturdefinitionsfilen. Med exemplet i föregående sektion skulle det vara underkatalogen `Exempelstruktur` med hela sitt innehåll. Att använda det här formatet för att dela strukturdefinitioner gör det möjligt att installera dem inne i Okteta, och kräver ingen manuell inställning.

## 3.2.4 Skapa strukturdefinitioner

### NOT

En mer aktuell, men dock icke fullständig handledning om hur man skriver strukturdefinitioner finns [på KDE:s användarbas Wiki](#).

Det finns två olika sätt att skapa strukturdefinitioner. Det första är att skriva definitionen i XML, det andra är att använda Javascript. Metoden med Javascript låter dig skapa mer komplexa strukturer med funktioner som exempelvis validering av strukturen. Med XML får du mindre funktioner, men om en statisk struktur är allt som behövs kan det vara det enklaste sättet. Om du behöver

en dynamisk struktur, dvs. där fältlängder beror på andra värden i strukturen eller strukturens layout är annorlunda när vissa ingående värden ändras, måste strukturdefinitionen skrivas i Javascript. Det finns ett undantag till regeln: Om du har ett fält där längden förväntas vara **exakt** likadan som ett annat värde i strukturen kan du använda XML. Men om den är något i stil med *längd - 1* måste Javascript användas.

### 3.2.5 Strukturdefinitionen med XML-filformat

#### NOT

En mer aktuell, men dock icke fullständig handledning om hur man skriver strukturdefinitioner finns [på KDE:s användarbas Wiki](#).

XML-filen `.osd` har ett rotelement: `<data>` utan egenskaper. Inne i detta element måste ett av följande element finnas:

#### `<primitive>`

För att skapa en primitiv datatyp, som t.ex. *int* och *float*. Elementet accepterar inte några delelement och kan ha följande egenskaper:

##### type

Den primitiva typens typ. Den måste vara en av följande:

- *char* för ett 8-bitars ASCII-tecken
- *int8*, *int16*, *int32*, *int64* för ett heltal med tecken av den storleken
- *uint8*, *uint16*, *uint32*, *uint64* för ett heltal utan tecken av den storleken
- *bool8*, *bool16*, *bool32*, *bool64* för ett Booleskt värde utan tecken (0 = falskt, alla andra värden = sant) av den storleken
- *float* för ett 32-bitars IEEE754 flyttal
- *double* för ett 64-bitars IEEE754 flyttal

#### `<bitfield>`

För att skapa ett bitfält. Elementet accepterar inte några delelement och kan ha följande egenskaper:

##### width

Antal bitar som används av bitfältet. Måste vara mellan 1 och 64.

##### type

Det här bitfältets typ. Den måste vara en av följande:

- *unsigned* för ett bitfält där värdet tolkas som ett värde utan tecken (värdet går från 0 till  $2^{\text{bredd}} - 1$ )
- *signed* för ett bitfält där värdet tolkas som ett värde med tecken (värdet går från  $-2^{\text{bredd} - 1}$  till  $2^{\text{bredd} - 1} - 1$ )
- *bool* för ett bitfält där värdet tolkas som ett Booleskt värde

#### NOT

Kom alltid ihåg att lägga till utfyllnad efter `<bitfield>`, eftersom annars börjar nästa element i mitten av en byte (utom för strängar och fält, eftersom de lägger till utfyllnad automatiskt). Utfyllnad är förstås inte nödvändig om beteendet är det önskade.

#### `<enum>`

För att skapa en primitiv typ, men där värdena visas som medlemmar i en uppräkningsstyp om möjligt. Elementet accepterar inga delelement (dock behövs taggen `<enumDef>` i filen för att referera till den). Den har följande egenskaper:

**enum**

Värdets underliggande uppräkningsstyp. Måste matcha egenskapen *name* för en av taggarna `<enumDef>` i filen.

**type**

Uppräkningsstypens typ. Se typegenskaperna för `<primitive>`. Enda skillnaden är att *Double* och *Float* inte är vettiga.

**<flags>**

Det här är samma sak som `<enum>` med den enda skillnaden att värdena representeras som *logiskt* eller *på bitnivå* av alla värden i uppräkningsstypen.

**<struct>**

För att skapa en struktur. Alla andra element (inklusive `<struct>`) kan ingå under det och blir då en del av strukturen som skapas.

**<union>**

För att skapa en union. I stort sett samma sak som `<struct>` utom att alla ingående element börjar från samma position. Användbart för att tolka samma byte-sekvens på olika sätt.

**<array>**

För att skapa ett fält. Elementet accepterar exakt ett underliggande objekt (fältets typ), som kan vara vilket som helst av elementen, till och med `<array>` själv. Det har också följande egenskaper:

**length**

Antal element i fältet som ett decimaltal. Som ett alternativ kan det också vara en sträng som motsvarar namnegenskapen hos ett tidigare definierat element `<primitive>`, `<enum>` eller `<flags>`. Då blir längden det elementets värde. För närvarande är det begränsat till 10000, eftersom större fält skulle använda för mycket minne och slöa ner verktyget för mycket.

**<string>**

För att skapa en sträng med diverse olika kodningar. Normalt skapas en C-liknande sträng terminerad med *NULL*. Dock kan annorlunda typer av strängar skapas med följande egenskaper:

**terminatedBy**

Egenskapen avgör vilket kodvärde i Unicode som avslutar strängen. Det måste vara ett hexadecimalt tal (med valfritt inledande *0x*). När kodningen är *ASCII*, är bara värden upp till *0x7f* meningsfulla. Om varken denna, *maxCharCount* eller *maxByteCount* har angivits, antas den vara inställt till 0 (strängformat enligt C).

**maxCharCount**

Det maximala antalet tecken som strängen kan ha. Om *terminatedBy* också har angivits slutar strängen beroende på vilket som inträffar först. Det är ömsesidigt uteslutande för *maxByteCount*.

**maxByteCount**

Det maximala antalet byte som strängens längd kan vara. Om *terminatedBy* också har angivits slutar strängen beroende på vilket som inträffar först. Detta är ömsesidigt uteslutande med *maxCharCount*. Med kodningar som *ASCII* är det samma sak som *maxCharCount*.

**type**

Strängens kodning. Kan vara ett av följande:

- *ASCII*
- *LATIN-1*
- *UTF-8*
- *UTF-16-LE* eller *UTF-16-BE*. Om varken suffixet *-LE* eller *-BE* anges, antas *LE* (little endian).

- *UTF-32-LE* eller *UTF-32-BE*. Om varken suffixet *-LE* eller *-BE* anges, antas *LE* (little endian).

Varje element accepterar också egenskapen *name* som därefter är synligt i strukturvyn.

### 3.2.6 Ett exempel på en strukturdefinition både i XML och Javascript

#### NOT

En mer aktuell, men dock icke fullständig handledning om hur man skriver strukturdefinitioner finns [på KDE:s användarbas Wiki](#).

#### 3.2.6.1 Det gemensamma steget som delas av båda tillvägagångssätten

Vår metadatafil ser ut på följande vis:

```
[Desktop Entry]
Icon=arrow-up<:\coref{1}{icon}>
Type=Service
ServiceTypes=KPluginInfo

Name=Enkel teststruktur
Comment=En mycket enkelt teststruktur som bara innehåller två ←
      objekt

X-KDE-PluginInfo-Author=Alex Richardson
X-KDE-PluginInfo-Email=foo.bar@email.org
X-KDE-PluginInfo-Name=enkelstrukt
X-KDE-PluginInfo-Version=1.0
X-KDE-PluginInfo-Website=https://www.plugin.org/
X-KDE-PluginInfo-Category=structure
X-KDE-PluginInfo-License=LGPL
X-KDE-PluginInfo-EnabledByDefault=false
```

- ❶ Ikonen visas i Okteta för strukturen. Kan vara vad som helst som hittas genom att köra **kdiallog --geticon** eller en sökväg till en ikon.

Fälten bör vara ganska självklara, utom *X-KDE-PluginInfo-Name*. Värdet på detta fält måste motsvara katalogen som innehåller filen, samt namnet på *.desktop*-filen. När XML-strukturdefinitionerna skapas måste också namnet på *.osd*-filen motsvara namnet.

I det här exemplet skulle vi ha en katalog kallad *enkelstrukt* som innehåller filen *enkelstruktur.desktop*. När strukturer definieras med XML skulle katalogen också innehålla en fil kallad *enkelstrukt.osd*. Med Javascript skulle vi istället ha en fil kallad *main.js*.

#### 3.2.6.2 En enkel XML-strukturdefinition

Till att börja med skapar vi en definition för en mycket enkel teststruktur som bara innehåller enkla datatyper (ett tecken, ett 32-bitars heltal och ett bitfält). De skulle uttryckas i C/C++ som:

```
struct simple {
    char aChar;
    int anInt;
    bool bitFlag :1;
    unsigned padding :7;
};
```

Det första steget är att skriva `.osd`-filen, som vi kallar `enkelstrukt.osd`:

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
  <struct name="simple">
    <primitive name="aChar" type="Char"/>
    <primitive name="anInt" type="Int32"/>
    <bitfield name="bitFlag" type="bool" width="1"/>
    <bitfield name="padding" type="unsigned" width="7"/>
  </struct>
</data>
```

som är ganska lik definitionen i C/C++.

Skapa nu katalogen `enkelstrukt` inne i strukturens installationskatalog (se manuell installation av strukturdefinitioner), och kopiera de två filerna till katalogen. Nu kan du starta om Okteta och använda den nya strukturen.

### 3.2.6.3 Den enkla strukturen i Javascript

För att implementera ovanstående struktur med Javascript, skapa en fil kallad `main.js` istället för `enkelstrukt.osd` och ändra `X-KDE-PluginInfo-Category=structure` till `X-KDE-PluginInfo-Category=structure/js`. Filens innehåll ska vara:

```
function init() {
  var structure = struct({
    aChar : char(),
    anInt : int32(),
    bitFlag : bitfield("bool", 1),
    padding : bitfield("unsigned", 7),
  })
  return structure;
}
```

Strukturen som visas av Okteta är alltid returvärdet från funktionen `init`.

Följande funktioner kan anropas för att skapa en enkel typ:

- `char()`
- `int8()`, `int16()`, `int32()` eller `int64()`
- `uint8()`, `uint16()`, `uint32()` eller `uint64()`
- `bool8()`, `bool16()`, `bool32()` eller `bool64()`
- `float()`
- `double()`

Funktionen `bitfield` har två parametrar, där den första är en sträng som består av `bool`, `signed` eller `unsigned`. Den andra parametern är ett heltal som anger bredden i bitar.

### 3.2.6.4 Mer komplexa strukturer

Härnäst skapar vi en definition av en mer komplex struktur, som vi kallar "komplex" och sparar i en fil som heter `komplex.osd`. Strukturen innehåller två fält (ett med fast längd och ett där längden bestäms vid körningstillfället) samt en nästlad struktur och en union.

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
  <struct name="complex">
    <primitive name="size" type="UInt8" />
    <union name="aUnion">
      <array name="fourBytes" length="4">
        <primitive type="Int8" />
      </array>
    </union>
    <struct name="nested">
      <array name="string" length="size"> <!-- references the ↔
        field size above -->
        <primitive type="Char" />
      </array>
    </struct>
  </struct>
</data>
```

Det skulle motsvara följande i pseudo-C/C++

```
struct complex {
    uint8_t size;
    union aUnion {
        int8_t fourBytes[4];
    };
    struct nested {
        char string[size] // inte giltig C++, referensvärde för storleken ↔
        uint8
    };
};
```

#### NOT

Uppenbarligen kan du bara låta fält med dynamisk längd referera till fält innan fältet.

Därefter skapar vi filen `komplex.desktop` precis som i det tidigare exemplet (försäkra dig om att du anger `X-KDE-PluginInfo-Name` riktigt) och gör också samma sak för att installera båda filerna.

#### 3.2.6.5 Ytterligare information

Några exempel på strukturdefinitioner finns i [Git-arkivet](#). Det omfattar till exempel filhuvud för PNG-filer, filhuvud för ELF-filen. Ett XML-schema som beskriver strukturen hos `.osd`-filen finns [här](#). Om mer information behövs kontakta mig gärna via [arichardson.kde@gmail.com](mailto:arichardson.kde@gmail.com).

## Kapitel 4

# Översikt av gränssnittet

### 4.1 Menyalternativ

Förutom de vanliga menyerna i KDE som beskrivs i kapitlet [Meny](#) i dokumentet KDE:s grunder, har Okteta följande programspecifika menyalternativ:

#### 4.1.1 Menyn Arkiv

##### Arkiv → Ny (Ctrl+N)

Skapa ett nytt byte-fält...

- **Tomt:** ... som tomma.
- **Från klippbordet:** ... enligt klippbordets nuvarande innehåll.
- **Mönster:** ... med ett givet mönster.
- **Slumpmässig data:** ... med slumpmässig data.
- **Följd:** ... med alla byte-värden från 0 till 255.

##### Arkiv → Exportera

Exportera markerade byte-värden till en fil...

- **Värden:** ... kodade som byte-värden. Normalt åtskiljs värdena av ett blanktecken. Tecknen som används som **Avdelare** kan ändras i dialogrutan **Exportera**.
- **Tecken:** ... kodade som vanlig text.
- **Base64:** ... kodad med [Base64](#)-format.
- **Base32:** ... kodad med [Base32](#) format.
- **Ascii85:** ... kodat med formatet [Ascii85](#).
- **Uuencoding:** ... kodat med formatet [Uuencoding](#).
- **Xxencoding:** ... kodat med formatet [Xxencoding](#).
- **Intel Hex:** ... kodat med formatet [Intel Hex](#).
- **S-Record:** ... kodat med formatet [S-Record](#).
- **C-fält:** ... definierade som ett fält i programspråket C.
- **Visa som vanlig text:** ... som i datavyn med position, byte-värden och tecken.

##### Arkiv → Rättigheter → Aktivera skrivskydd

Om vald, kan inte några ändringar göras av det inlästa byte-fältet.

##### Arkiv → Stäng alla andra

Stäng alla utom det aktuella byte-fältet.



### 4.1.2 Menyn Redigera

#### Redigera → Kopiera som

Kopiera de markerade värdena med ett av olika format till klippbordet. Se menyalternativet **Arkiv → Exportera** för en lista med tillgängliga format.

#### Redigera → Infoga

##### Infoga mönster...

Infoga en angiven bytesträng vid markören.

Alternativ i dialogrutan låter dig ange antal gånger som mönstret infogas och dess format (Hexadecimal, Decimal, Oktal, Binär, Tecken eller UTF-8).

#### Redigera → Avmarkera (Ctrl+Skift+A)

Avmarkerar den aktuella markeringen.

#### Redigera → Markera intervall... (Ctrl+E)

Öppnar en inbäddad dialogruta för att ange intervallet att markera.

#### Redigera → Ersättningsläge (Ins)

Byt mellan infoga och ersätt.

##### NOT

Överskrivningsläge är implementerat att vara mycket strikt. Det är inte möjligt att ändra datastorleken (inget tillägg eller borttagning av data).

#### Redigera → Sök... (Ctrl+F)

Sök efter ett angivet mönster i dokumentet. Hexadecimal, decimal, oktal, binär eller textmönster kan sökas efter.

Alternativ i dialogrutan låter dig ange startpunkten, riktningen och området för sökningen.

#### Redigera → Gå till position... (Ctrl+G)

Flytta markören till en angiven position.

### 4.1.3 Menyn Visa

#### Visa → Visa radposition (F11)

Växla visning av radposition i en ruta till vänster på eller av.

#### Visa → Visa värden eller tecken

Välj vilka tolkningar av byte-värdena som visas. Möjligheterna är:

- Värden
- Tecken
- Värden och tecken

#### Visa → Värdekodning

Välj kodningen av värdena från:

- Hexadecimal
- Decimal

- Oktal
- Binär

**Visa → Teckenkodning**

Välj kodningen av tecknen från undermenyn.

**Visa → Visa tecken som inte kan skrivas ut**

Växla visning av tecken som inte kan skrivas ut på eller av. Om visningen är avstängd, visas ett ersättningstecken istället på motsvarande platser i teckenkolumnen.

**Visa → Ställ in byte per rad**

Markera visade byte per rad i dialogrutan, standardvärdet är 16 byte.

**Visa → Ställ in byte per grupp**

Normalt visas hexadecimala värden i grupper om 4 byte. Genom att använda det här menyalternativet kan du anpassa det till dina önskemål i en dialogruta.

**Visa → Dynamisk layout**

Ställ in reglerna för layout av datavisningen. Det definierar hur många byte som visas per rad, beroende på vyns bredd. Möjliga regler är:

- **Av:** Layouten är låst till aktuellt antal byte-värden per rad och ändras inte när vyns storlek ändras.
- **Radbryt bara fullständiga byte-grupper:** Placerar så många byte-värden per rad som möjligt, under förutsättning att grupper av värden är fullständiga.
- **På:** Samma som föregående, men tillåter också ofullständiga grupper av byte-värden.

**Visa → Vyläge**

Välj utläggning av vyn från:

- **Kolumner:** Värden och tolkningen som tecken visas i den klassiska utläggningen där var och en anges i en separat kolumn.
- **Rader:** Tolkningen av en byte som tecken visas direkt under värdetolkningen.

**Visa → Dela horisontellt (Ctrl+Skift+T)**

Dela vyområdet med vyn som för närvarande har fokus i två delar och lägg till en kopia av den nuvarande vyn i det nya nedre området.

**Visa → Dela vertikalt (Ctrl+Skift+L)**

Dela vyområdet med vyn som för närvarande har fokus i två delar och lägg till en kopia av den nuvarande vyn i det nya högra området.

**Visa → Stäng vyområde (Ctrl+Skift+R)**

Stänger vyområdet med vyn som för närvarande har fokus.

**Visa → Vyprofil**

Vyinställningar kan sparas separat som vyprofiler. Profilen som för närvarande är vald kan direkt uppdateras från de nuvarande vyinställningarna, eller så kan en ny skapas från dem. Alla vyprofiler kan hanteras i en dialogruta tillgänglig med **Inställningar → Hantera vyprofiler...**

#### 4.1.4 Menyn Fönster

Tillhandahåller en lista med aktuella vyer. Välj aktivt fönster.

### 4.1.5 Menyn Bokmärken

Flera bokmärken kan anges för ett enskilt byte-fält. Varje byte-fält har sin egen uppsättning bokmärken, och lämplig uppsättning visas längst ner i menyn **Bokmärken**. Välj ett bokmärke i menyn för att flytta markören i vyn till det.

#### NOT

Bokmärken är för närvarande bara tillfälliga och sparas inte om ett byte-fält eller hela programmet stängs.

#### **Bokmärken → Lägg till bokmärke (Ctrl+B)**

Lägg till ett bokmärke för en plats i byte-fältet.

#### **Bokmärken → Ta bort bokmärke (Ctrl+Skift+B)**

Ta bort aktuellt bokmärke. Kommandot är bara tillgängligt om markören är på en plats med ett bokmärke.

#### **Bokmärken → Ta bort alla bokmärken**

Rensa bokmärkeslistan.

#### **Bokmärken → Gå till föregående bokmärke (Alt+Up)**

Flytta markören till föregående bokmärke.

#### **Bokmärken → Gå till nästa bokmärke (Alt+Neråtpil)**

Flytta markören till nästa bokmärke.

### 4.1.6 Menyn Verktyg

Tillhandahåller en lista med installerade verktyg. Visa eller dölj varje verktyg. En detaljerad beskrivning av varje verktyg hittar du i avsnittet [Verktyg](#).

### 4.1.7 Menyn Inställningar

#### **Inställningar → Hantera vyprofiler...**

Visar en dialogruta för att skapa, redigera, ta bort och ställa in en förvald vyprofil.

## Kapitel 5

# Tack till och licens

Okteta

Program Copyright 2006-2012 Friedrich W. H. Kossebau [kossebau@kde.org](mailto:kossebau@kde.org)

Dokumentation Copyright 2008, 2010 Friedrich W. H. Kossebau [kossebau@kde.org](mailto:kossebau@kde.org), Alex Richardson [arichardson.kde@gmail.com](mailto:arichardson.kde@gmail.com)

Översättning Stefan Asserhäll [stefan.asserhall@bredband.net](mailto:stefan.asserhall@bredband.net)

Den här dokumentationen licensieras under villkoren i [GNU Free Documentation License](#).

Det här programmet licensieras under villkoren i [GNU General Public License](#).