

# Het handboek van Okteta

**Friedrich W. H. Kossebau**

**Alex Richardson**

**Vertaler/Nalezer: Freek de Kruijf**

**Vertaler: Ronald Stroethoff**



## Het handboek van Okteta

# Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>5</b>
<b>2</b>	<b>Grondbeginselen</b>	<b>6</b>
2.1	Okteta opstarten . . . . .	6
2.2	Gebruik . . . . .	6
<b>3</b>	<b>Hulpmiddelen</b>	<b>7</b>
3.1	Overzicht . . . . .	7
3.1.1	Analyzers en manipulatoren . . . . .	7
3.1.2	Algemene hulpmiddelen . . . . .	8
3.2	Hulpmiddel voor structuren . . . . .	9
3.2.1	Algemeen . . . . .	9
3.2.2	Structuurdefinities installeren . . . . .	9
3.2.2.1	Met KNewStuff installeren . . . . .	9
3.2.2.2	Structuurdefinities handmatig installeren . . . . .	9
3.2.2.3	De nieuwe geïnstalleerde structuren gebruiken . . . . .	9
3.2.3	Structuurdefinities delen . . . . .	10
3.2.4	Structuurdefinities maken . . . . .	10
3.2.5	Structuurdefinitie XML-bestandsformaat . . . . .	10
3.2.6	Een voorbeeld van een structuurdefinitie in zowel XML als JavaScript . . . . .	12
3.2.6.1	De gezamenlijke stap gedeeld door beide benaderingen . . . . .	12
3.2.6.2	Een eenvoudige XML-structuurdefinitie . . . . .	13
3.2.6.3	De eenvoudige structuur in JavaScript . . . . .	14
3.2.6.4	Complexere structuren . . . . .	14
3.2.6.5	Meer informatie . . . . .	15
<b>4</b>	<b>Overzicht van het interface</b>	<b>16</b>
4.1	Menu-items . . . . .	16
4.1.1	Menu Bestand . . . . .	16
4.1.2	Menu Bewerken . . . . .	17
4.1.3	Menu Beeld . . . . .	17
4.1.4	Menu Vensters . . . . .	19
4.1.5	Het menu Bladwijzers . . . . .	19
4.1.6	Menu hulpmiddelen . . . . .	19
4.1.7	Menu Instellingen . . . . .	19
<b>5</b>	<b>Verdiensten en licentie</b>	<b>20</b>

### **Samenvatting**

Okteta is een eenvoudige bewerker voor de ruwe data van bestanden. Dit type programma wordt ook wel hex-editor of binaire editor genoemd.

# Hoofdstuk 1

## Introductie

Okteta is een eenvoudige bewerker voor de ruwe gegevens van bestanden.

De gegevens worden weergegeven in twee varianten: als de numerieke waarde van de bytes en als het teken dat behoort bij de waarden. Waarden en tekens kunnen getoond worden in twee kolommen (de traditionele weergave in hex-editors) of in rijen met de waarde boven het teken. Bewerken kan gedaan worden zowel in de waarden als in de tekens.

Naast de gebruikelijke bewerkingsmogelijkheden, biedt Okteta ook een kleine set hulpmiddelen, zoals een tabel die decodings laat zien in algemene eenvoudige typen gegevens, een tabel die alle mogelijke bytes laat zien met hun teken en overeenkomende waarden, statistische informatie, een rekenmachine voor een controlesom, een filter en een hulpmiddel voor extractie van tekenreeksen.

Alle wijzigingen aan de geladen gegevens kunnen eindeloos teruggezet of opnieuw gedaan worden.

## Hoofdstuk 2

# Grondbeginselen

### 2.1 Okteta opstarten

Type **okteta** op een commandoprompt of selecteer **Hexbewerker** uit de groep **Toepassingen** → **Hulpmiddelen** in het toepassingenstartpaneel.

De standaard Qt™ en KDE-frameworks; 5 commandoregeloptyes zijn beschikbaar en kunnen worden getoond door **okteta --help** in te voeren.

Opdrachtregel-opties specifiek voor Okteta zijn:

<URL ('s) > - open bestand(en) uit de gespecificeerde URL('s)

### 2.2 Gebruik

Het hoofdvenster van Okteta heeft de volgende componenten: een menubalk, een werkbalk, een statusbalk, één of meer zijbalken met hulpmiddelen en het hoofddeelvenster met de tabbladen met gegevensweergaven.

Wanneer een bestand is geopend of een nieuwe reeks bytes wordt aangemaakt, dan worden de bytes, die er in zitten, achtereenvolgend getoond in regels met een gegeven aantal bytes per regel. Ze worden getoond in twee varianten: als de numerieke waarde van de bytes en als het teken toegekend aan de waarde. Waarden en tekens kunnen gescheiden van elkaar worden getoond in twee kolommen of bij elkaar met de waarde boven het teken. Aan de linkerzijde staan de offsets van het eerste byte in elke regel.

De behandeling is vergelijkbaar met die in de meeste tekstbewerkers: de gegevens kunnen bewerkt worden, geknipt, gekopieerd, geplakt, versleept en losgelaten zoals dat met tekst kan. Een cursor markeert de huidige positie. Op de **Insert**-toets drukken schakelt tussen de modi overschrijven en invoegen. De modus overschrijven is strikter dan in tekstbewerkers, omdat het geen bewerking toestaat die de lengte van de reeks bytes wijzigt.

Anders dan in tekstbewerkers wordt de inhoud in twee varianten getoond. Alleen een hiervan is actief met betrekking tot nieuwe invoer. Er worden twee gekoppelde cursors getoond voor de waarde en het tonen van het teken, de cursor van de actieve knippert. Met de tekens actief, kunnen tekens ingevoerd worden zoals bekend uit tekstbewerkers. Met de waarden actief opent het typen van een cijfer een minimale editor om de rest van de waarde in te voeren.

De zoekdialoog stelt de gebruiker in staat om naar een specifieke reeks bytete zoeken, te definiëren als waarden (hexadecimaal, decimaal, octaal, binair) of tekst (huidige 8-bit codering of UTF-8).

Meerdere byte-arrays kunnen tegelijk open zijn, maar slechts één kan er actief zijn. Gebruik het menu **Venster** om te selecteren welk byte-array actief zal zijn.

## Hoofdstuk 3

# Hulpmiddelen

### 3.1 Overzicht

Okteta biedt enige hulpmiddelen voor analyse en manipulatie van de byte-arrays en enige voor meer algemene doelen. Deze hulpmiddelen kunnen geactiveerd of gedeactiveerd worden vanuit het item **Hulpmiddelen** in de menubalk. Elk hulpmiddel heeft een klein venster, dat zich vastzet ofwel in een van de zijbalken of vrij zweeft als een venster. U kunt vastzetten, losmaken, opnieuw arrangeren en de hulpmiddelvensters ook met de muis stapelen, door met de linkermuisknop op de titel balk van een hulpmiddelvenster te drukken, het naar wens te verplaatsen en de linkermuisknop los te laten om de actie te voltooien, annuleer het door op de **Esc**-toets te drukken.

#### 3.1.1 Analyzers en manipulators

##### Tabel met waarde/teken

De tabel laat alle mogelijke byte-waarden zien, zowel als teken als in de verschillende numerieke coderingen.

De geselecteerde waarde kan op de cursorpositie ingevoegd worden voor een gedefinieerd aantal bytes. Dit kan bereikt worden door de knop **Invoegen** te gebruiken of dubbel te klikken op de regel in de tabel.

##### Binaire filter

Het filter voert binaire bewerkingen uit op de geselecteerde bytes. Na het kiezen van de bewerking (EN, OF, ROTATIE..) kunnen de parameters, indien aanwezig, ingesteld worden in het onderstaande vak. Het filter wordt uitgevoerd door de knop **Filter** te gebruiken.

##### Tekenreeksen

Dit hulpmiddel lokaliseert de tekenreeksen in de geselecteerde bytes. Na het kiezen van de minimale lengte van de tekenreeks, worden de tekenreeksen opgezocht door de knop **Extraheren** te gebruiken. De lijst met de getoonde tekenreeksen kan kleiner gemaakt worden door een filter in te voeren.

##### Statistieken

Dit hulpmiddel bouwt een statistiek voor de geselecteerde bytes. De statistiek geeft de frequentie van voorkomen van elke bytewaarde in de selectie. Het kan berekend worden door de knop **Bouwen** te gebruiken.

### Controlesom

Dit hulpmiddel berekent verschillende controlesommen of hashsummen voor de geselecteerde bytes. Na het kiezen van het algoritme en het instellen van de parameter, indien nodig, zal de som berekend worden bij gebruik van de knop **Berekenen**.

### Decoderingstabel

De tabel toont de waarden van het byte of de bytes beginnend bij de cursor voor enkele algemene eenvoudige gegevenstypen zoals geheel getal of drijvende komma, maar ook UTF-8. Dubbelklikken op een regel in de tabel opent een editor, zodat de waarde bewerkt en gewijzigd kan worden.

### Structuren

Dit hulpmiddel schakelt onderzoeken en bewerken van byte-arrays in gebaseerd op door de gebruiker gemaakte structuurdefinities. Gedetailleerde instructies staan in een eigen [sectie](#).

## 3.1.2 Algemene hulpmiddelen

### Bestandssysteem

Dit hulpmiddel biedt een ingebedde bestandsbrowser die gebruikt kan worden om te openen bestanden te selecteren.

### Documenten

Dit hulpmiddel toont alle nu aangemaakte of geladen bestanden. Symbolen markeren het bestand met de nu actieve weergave en tonen ook welke bestanden niet opgeslagen wijzigingen of welke opgeslagen kopie gewijzigd is door een ander programma.

### Bladwijzers

Dit hulpmiddel kan worden gebruikt om de bladwijzers te beheren, als alternatief voor het menu [Bladwijzers](#).

#### OPMERKING

Bladwijzers zijn nu slecht voorbijgaand en worden niet opgeslagen als u een byte-array of het gehele programma sluit.

### Bestandsinformatie

Dit hulpmiddel toont enige informatie over het huidige bestand, inclusief het type, de locatie van opslaan en de grootte.

### Terminal

Een ingebedde terminal, de werkmap is niet gekoppeld aan het actieve bestand.

### Conversie van tekenset

Het hulpmiddel herschrijft de bytes zodat de respectievelijke tekens hetzelfde zijn als in de andere tekenset. Alleen 8-bits tekensets worden ondersteund en tekens zonder overeenkomst worden op dit moment vervangen door een waarde hard gecodeerd op 0.



## 3.2 Hulpmiddel voor structuren

### 3.2.1 Algemeen

Het hulpmiddel Structuren schakelt analyseren en bewerking van byte-arrays in gebaseerd op door de gebruiker gemaakte structuurdefinities, die kunnen worden gebouwd uit arrays, unions, primitieve typen en een lijst met waarden.

Het heeft een eigen instellingendialoog, die bereikt wordt de knop **Instellingen**. Er zijn verschillende opties die ingesteld kunnen worden, zoals de stijl waarin (decimaal, hexadecimaal of binair) de waarden getoond worden. Verder is het mogelijk te kiezen welke structuurdefinities geladen moeten worden en welke structuren getoond worden in het voorbeeld.

Structuren worden gedefinieerd in Okteta-structuurdefinitiebestanden (gebaseerd op XML, met de bestandsextensie `.osd`). Verder kan een `.desktop` bestand metagegevens bevatten over dat structuurbeschrijvingsbestand, zoals auteur, homepage en licentie.

Op dit moment is er geen ingebouwde ondersteuning voor maken of bewerken van structuurdefinities, dit moet daarom handmatig gedaan worden zoals beschreven in de volgende secties.

### 3.2.2 Structuurdefinities installeren

#### 3.2.2.1 Met KNewStuff installeren

De gemakkelijkste manier om nieuwe structuurdefinities te installeren is door de ingebouwde KNewStuff-ondersteuning in Okteta te gebruiken. Om een bestaande structuur te installeren opent u de instellingendialoog van het hulpmiddel Structures. U selecteert daar het tabblad **Beheer van structuren** en drukt op de knop **Nieuwe structuren ophalen...** De dialoog die verschijnt stelt u nu in staat om structuren te installeren en te deïnstalleren.

#### 3.2.2.2 Structuurdefinities handmatig installeren

Het hulpmiddel voor structuren zoekt naar structuurdefinities in de map `okteta/structures/` van de map voor programmagegevens van de gebruiker (u kunt deze vinden via de opdracht `qtpaths --paths GenericDataLocation`). U moet deze map misschien nog aanmaken als er nog geen structuurdefinities zijn geïnstalleerd.

Er bestaan twee bestanden voor elke structuurdefinitie: een bestand voor de actuele definitie en een `.desktop`-bestand voor de metagegevens (auteur, versie, etc.).

In deze map is een submap voor elke structuurdefinitie, waarin zowel het `.desktop`-bestand en het `.osd` of `main.js` bestand van deze definitie aanwezig zijn.

Bijvoorbeeld met de programmagegevensmap `qtpaths --paths GenericDataLocation` en een structuurdefinitie genaamd `VoorbeeldStructuur` is de map `okteta/structures/VoorbeeldStructuur`, die bestanden `VoorbeeldStructuur.desktop` en `VoorbeeldStructuur.osd` bevat.

#### 3.2.2.3 De nieuwe geïnstalleerde structuren gebruiken

Nadat u nieuwe structuurdefinities heeft geïnstalleerd, moet u Okteta opnieuw starten voordat u deze kunt gebruiken. Nadat Okteta opnieuw is gestart, opent u de instellingendialoog van het hulpmiddel voor structuren en selecteert u vervolgens het tabblad **Beheer van structuren** en controleert of de relevante structuurdefinitie is ingeschakeld. Ga vervolgens naar het tabblad **Structuren** en controleer dat het verlangde element aan de rechterkant is te zien.

### 3.2.3 Structuurdefinities delen

Het is waarschijnlijk niet nodig voor standaard structuren dat u de definitie zelf aanmaakt, in plaats daarvan kunt u een al bestaande definitie gebruiken van een plaats zoals [store.kde.org](http://store.kde.org).

U kunt ook een eigengemaakte definitie met anderen delen. Om dit te doen maakt u een archiefbestand aan (bijv. een gezippt tar-archief, `.tar.gz`) met daarin de map met het `.desktop`-bestand en het structuurdefinitiebestand. In het voorbeeld in het laatste sectie van dit hoofdstuk zal dit zijn de map `ExampleStructure` met al zijn inhoud. Door gebruik van dit formaat voor het delen van de structuurdefinities is het mogelijk ze automatisch in Okteta te installeren en is handmatige installatie niet nodig.

### 3.2.4 Structuurdefinities maken

#### OPMERKING

een meer bijgewerkte, maar niet geheel gerede handleiding om structuurdefinities te schrijven is te vinden in [op de KDE UserBase Wiki](#).

Er zijn twee verschillende manieren om structuurdefinities aan te maken. De eerste is het schrijven van de definitie in XML de andere is het gebruik van JavaScript. Het gebruik van JavaScript geeft u de vrijheid om meer complexere structuren met mogelijkheden zoals structuurcontrole te creëren. Het gebruik van XML geeft u minder mogelijkheden maar als statistische structuren alles is wat u nodig heeft dan kan dit de makkelijkste manier zijn. Als u een dynamische structuur nodig heeft bijv. waar array lengtes van andere waarden in de structuur afhangen of de structuurindeling verandert wanneer sommige waarden wijzigingen, dan zult u de structuurdefinitie in JavaScript moeten schrijven. Er is een uitzondering op die regel: als u een array heeft waarvan de lengte **exact** gelijk is aan een andere waarde in de structuur, dan kan u ook XML gebruiken. Maar als het iets is zoals `lengte - 1` dan moet het JavaScript zijn.

### 3.2.5 Structuurdefinitie XML-bestandsformaat

#### OPMERKING

een meer bijgewerkte, maar niet geheel gerede handleiding om structuurdefinities te schrijven is te vinden in [op de KDE UserBase Wiki](#).

Het `.osd` XML-bestand heeft één basiselement: `<data>` zonder attributen. Binnen dit element moet er één van de volgende elementen zijn:

#### `<primitive>`

Om een 'primitive' gegevenstype zoals bijv. `int` en `float` aan te maken. Dit element accepteert geen subelementen en kan de volgende attributen hebben:

#### type

Het type van deze primitieve type. Het moet een van de volgende zijn:

- `char` voor een 8-bits ASCII-teken
- `int8`, `int16`, `int32`, `int64` voor een geheel getal met teken met die grootte
- `uint8`, `uint16`, `uint32`, `uint64` voor een geheel getal zonder teken van die grootte
- `bool8`, `bool16`, `bool32`, `bool64` voor een logische waarde (boolean) (0 = false, elke andere waarde = true) van die grootte
- `float` voor een 32-bits IEEE754 drijvende-komma getal

- *double* voor een 64-bits IEEE754 drijvende-komma getal

### <bitfield>

Om een 'bitfield' aan te maken Dit element accepteert geen subelementen en kan de volgende attributen hebben:

#### **breedte**

Het aantal bits gebruikt door dit 'bitfield'. Moet liggen tussen 1 en 64.

#### **type**

Het type van dit 'bitfield'. Het moet één van de volgende zijn:

- *unsigned* voor een bitfield waar de waarde geïnterpreteerd zal worden als een waarde zonder teken (waardereeks van 0 tot  $2^{\text{breedte}} - 1$ )
- *signed* voor een bitfield waar de waarde geïnterpreteerd zal worden als een waarde met teken (waardereeks van  $-2^{\text{breedte} - 1}$  tot  $2^{\text{breedte} - 1} - 1$ )
- *bool* voor een 'bitfield' waar de waarde als een booleaanse waarde zal worden geïnterpreteerd

#### OPMERKING

Denk er aan om padding toe te voegen na een <bitfield>, omdat het volgende element anders in het midden van een byte zou beginnen (behalve voor tekenreeksen en arrays, omdat zij automatisch padding toevoegen). Als u dit gedrag wenst dan is padding natuurlijk niet nodig.

### <enum>

Om een type primitive te maken, maar waar de waarden getoond worden, indien mogelijk, als leden van een enumeratie. Dit element accepteert geen subelementen (maar u hebt mogelijk een tag <enumDef> in het bestand nodig om er naar te refereren). Het heeft de volgende attributen:

#### **enum**

De onderliggende 'enum' voor deze waarde. Moet overeenkomen met het attribuut *name* van één van de tags <enumDef> in dit bestand.

#### **type**

Het type hiervan is enum. Zie type attribuut van <primitive>. Het enige verschil is dat *Double* en *Float* geen betekenis hebben.

### <flags>

Dit is hetzelfde als <enum> met het enige verschil dat waarden gerepresenteerd worden als een *bitwise-or* van alle waarden van de enumeratie (opsomming).

### <struct>

Hiermee creëert u een structuur. Alle andere elementen (inclusief een <struct>) kunnen hiervan onderdeel uitmaken (child)

### <union>

Hiermee creëert u een union. In wezen hetzelfde als een <struct>, behalve het feit dat alle child-elementen vanaf dezelfde offset starten. Dit is handig voor het op verschillende manieren onderzoeken van dezelfde volgorde van bytes.

### <array>

Hiermee creëert u een array. Dit element accepteert precies een child (het onderliggende type array), waarin u elk soort element kan plaatsten, zelfs een <array>. Het heeft de volgende attributen:

### lengte

Het aantal elementen in dit array als decimaal getal. Als alternatief kan het ook een tekenreeks zijn waarvan de naam overeenkomt met een eerder gedefinieerde *<primitive>*, *<enum>* of *<flags>* element. De lengte zal dan overeenkomen met de waarde van dat element. De lengte is op dit moment beperkt tot 10000, omdat grotere arrays teveel geheugen gebruiken en teveel vertragen.

### *<string>*

Hiermee kun u een tekenreeks in een codering naar keuze creëren. Standaard is dit een tekenreeks in de C-style beëindigt door een *NULL*. Maar met de volgende attributen kunt u andere soorten tekenreeksen creëren:

#### **terminatedBy**

Dit attribute bepaalt door welk unicode karakter de tekenreeks wordt beëindigd. Dit moet een hexadecimaal nummer zijn (optioneel met een leidende *0x*). Wanneer de codering in ASCII is dan hebben alleen waarden tot *0x7f* een betekenis. Als u zowel dit of *maxCharCount* of *maxByteCount* niet heeft ingesteld dan is dit standaard ingesteld op 0 (C-style string)

#### **maxCharCount**

Het maximum aantal tekens dat deze tekenreeks kan hebben. Als *terminatedBy* ook is ingesteld dan eindigt de tekenreeks door datgene wat als eerste is bereikt. Dit gaat niet samen met *maxByteCount*

#### **maxByteCount**

Het maximum aantal bytes dat in deze tekenreeks past. Als *terminatedBy* ook is ingesteld, dan wordt de tekenreeks afgesloten door datgene wat het eerst bereikt wordt. Dit gaat niet samen met *maxCharCount*. Alleen bij decoderingen zoals *ASCII* komt dit overeen met *maxCharCount*.

### type

De codering van deze tekenreeks. Kan een van de volgende zijn:

- *ASCII*
- *LATIN-1*
- *UTF-8*
- *UTF-16-LE* of *UTF-16-BE*. Als geen *-LE* of *-BE* achtervoegsel is gegeven, wordt 'little endian' aangenomen.
- *UTF-32-LE* of *UTF-32-BE*. Als geen *-LE* of *-BE* achtervoegsel is gegeven, wordt 'little endian' aangenomen.

Elk element accepteert ook een attribuut *name* dat dan zichtbaar is in de structurenweergave.

## 3.2.6 Een voorbeeld van een structuurdefinitie in zowel XML als JavaScript

### OPMERKING

een meer bijgewerkte, maar niet geheel gereede handleiding om structuurdefinities te schrijven is te vinden in [op de KDE UserBase Wiki](#).

### 3.2.6.1 De gezamenlijke stap gedeeld door beide benaderingen

Ons bestand met metagegevens ziet er zo uit:

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

Name=Simpele teststructuur
Comment=Een erg simpele teststructuur met slechts twee items

X-KDE-PluginInfo-Author=Alex Richardson
X-KDE-PluginInfo-Email=foo.bar@email.org
X-KDE-PluginInfo-Name=simplestruct
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 Het pictogram getoond in Okteta voor deze structuur kan alles zijn wat gevonden wordt door **kdialog --geticon** uit te voeren of een pad naar een pictogram

Deze velden zouden allemaal zichzelf moeten verklaren, behalve voor `X-KDE-PluginInfo-Name`. De waarde van dit veld moet overeenkomen met de naam van de map met het bestand evenals de naam van het `.desktop`-bestand. Bij het maken van XML-structuurdefinities zal de naam van het `.osd` bestand ook met de naam overeen moeten komen.

In dit voorbeeld willen we een map met de naam `simplestruct` met daarin het bestand `simplestruct.desktop` hebben. Als we de structuren definiëren in XML dan is in de map ook een bestand genaamd `simplestruct.osd` aanwezig. Bij gebruik van JavaScript hebben we in plaats daarvan een bestand met de naam `main.js`.

### 3.2.6.2 Een eenvoudige XML-structuurdefinitie

Om te beginnen creëren we een definitie voor een eenvoudige teststructuur met alleen integrale gegevenstypes (een char, een 32-bit signed integer en een bitfield). Dit is in C/C++ uitgeschreven als:

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

De eerste stap is het schrijven van het `.osd`-bestand overeenkomstig het bestandsformaat zoals in de vorige sectie beschreven. We geven het de naam `simplestruct.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>
```

wat overeenkomt met de C/C++ definitie.

Maak nu een map aan met de naam `simplestruct` in de structuur installatiemap (lees handmatig structuurdefinities installeren) en kopieer de twee bestanden naar deze map. Na het opnieuw opstarten van Okteta kunt u de nieuwe structuur gebruiken.

### 3.2.6.3 De eenvoudige structuur in JavaScript

Om bovengenoemde structuren in JavaScript te implementeren, creëert u een bestand met de naam `main.js` in plaats van `simplestruct.osd` en wijzigt u `X-KDE-PluginInfo-Category=structure` in `X-KDE-PluginInfo-Category=structure/js`. De inhoud van dat bestand zou het volgende moeten zijn:

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

De structuur die Okteta toont is altijd de "return value" van de `init`-functie.

U kunt de volgende functies aanroepen om een primitief type te creëren:

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

De `bitfield`-functie heeft twee parameters nodig, de eerste is een tekenreeks bestaande uit `bool`, `signed` of `unsigned`. De tweede parameter is een integer die de breedte in bits instelt.

### 3.2.6.4 Complexere structuren

Vervolgens creëren we een definitie voor een complexere structuur waaraan we de naam "complex" geven en we slaan dit op in een bestand genaamd `complex.osd`. Deze structuur bestaat uit twee arrays (een met een vaste lengte en een waarvan de lengte tijdens de uitvoering wordt bepaald), een geneste structuur en een verzameling.

```
<?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">
```

## Het handboek van Okteta

```
        <array name="string" length="size"> <!-- verwijzing naar de ↔  
            field-size boven -->  
        <primitive type="Char" />  
    </array>  
</struct>  
</struct>  
</data>
```

Dit zou overeen moeten komen met het volgende in pseudo-C/C++

```
struct complex {  
    uint8_t size;  
    union aUnion {  
        int8_t fourBytes[4];  
    };  
    struct nested {  
        char string[size] //geen geldige C++, refereert waarde van de ↔  
        grootte uint8  
    };  
};
```

### OPMERKING

U kunt uiteraard de referentie-velden voor arrays met dynamische lengte alleen vóór het array hebben,

Vervolgens maken we het bestand `complex.desktop` net als in het vorige voorbeeld (ga na dat u `X-KDE-PluginInfo-Name` juist instelt) en ook hetzelfde doet om beide bestanden te installeren.

### 3.2.6.5 Meer informatie

Enkele voorbeelden voor structuurdefinities kunt u vinden in deze [Git repository](#). Inclusief voorbeelden voor de bestand-header van PNG-bestanden en de ELF bestand-header. Een XML schema voor de structuur van een `.osd`-bestand kunt u [hier](#) vinden. Als u meer informatie wilt dan kunt u met mij contact opnemen via [arichardson.kde@gmail.com](mailto:arichardson.kde@gmail.com)

## Hoofdstuk 4

# Overzicht van het interface

### 4.1 Menu-items

Naast de standaard KDE-menu's zoals beschreven in het hoofdstuk [Menu](#) van de documentatie in KDE Fundamentals heeft Okteta specifieke menu-onderdelen van het programma:

#### 4.1.1 Menu Bestand

##### Bestand → Nieuw (Ctrl+N)

Een nieuw byte-array aanmaken...

- **Leeg:** ... zoals een lege.
- **Van klembord:** ... door de huidige inhoud van het klembord.
- **Patroon...:** ... met een gegeven patroon.
- **Willekeurige gegevens...:** ...met willekeurige gegevens.
- **Reeks:** ... met alle bytes van 0 tot 255.

##### Bestand → Exporteren

De geselecteerde bytes naar een bestand exporteren...

- **Waarden:** ... gecodeerd als byte-waarden. Standaard zijn de waarden gescheiden door één witruimte. Het **Scheidings**-teken kunt u wijzigen in het dialoogvenster **Exporteren**.
- **Tekens:** ... gecodeerd als platte tekst.
- **Base64:** ... gecodeerd in het [Base64](#)-formaat.
- **Base32:** ... gecodeerd in het [Base32](#)-formaat.
- **Ascii85:** ... gecodeerd in het formaat [Ascii85](#).
- **Uuencoding:** ... gecodeerd in het formaat [Uuencoding](#).
- **Xxencoding:** ... gecodeerd in het formaat [Xxencoding](#).
- **Intel Hex:** ... gecodeerd in het formaat [Intel Hex](#).
- **S-Record:** ... gecodeerd in het [S-Record](#)-formaat.
- **C array:** ... gedefinieerd als een array in de programmeertaal C.
- **In platte tekst bekijken:** ... zoals in de gegevensweergave met offset, waarden van bytes en tekens.



**Bestand** → **Toegangsrechten** → **Instellen op alleen-lezen**

Er kunnen mogelijk geen wijzigingen worden gemaakt in het geladen byte-array, indien geactiveerd.

**Bestand** → **Alle anderen sluiten**

Sluit alles behalve het huidige byte-array.

## 4.1.2 Menu Bewerken

**Bewerken** → **Kopiëren als**

De geselecteerde bytes in één van de andere data-formaten naar het klembord kopiëren. Ga voor de lijst met beschikbare data-formaten naar het menu-item **Bestand** → **Exporteren**

**Bewerken** → **Invoegen**

**Patroon invoegen...**

Voeg een gespecificeerde tekenreeks met bytes in bij de cursor.

Opties in het dialoogvak stellen u in staat het aantal keren invoegen van het patroon te specificeren en het formaat (hexadecimaal, decimaal, octaal, binair, teken(s) of UTF-8).

**Bewerken** → **Deselecteren (Ctrl+Shift+A)**

Maak de huidige selectie ongedaan.

**Bewerken** → **Bereik selecteren... (Ctrl+E)**

Opent een ingebedde dialoog om het te selecteren bereik in te voeren.

**Bewerken** → **Overschrijfmodus (Ins)**

Schakelen tussen de modi Invoegen en Overschrijven.

**OPMERKING**

De modus overschrijven is erg strikt geïmplementeerd, het is niet mogelijk om de grootte van de gegevens te wijzigen (geen toevoegen of verwijderen van bytes).

**Bewerken** → **Zoeken... (Ctrl+F)**

Zoek naar een gespecificeerd patroon in het document. Er kan naar hexadecimale, decimale, octale, binaire of tekstpatronen worden gezocht.

Opties in het dialoogvak stellen u in staat om het beginpunt, richting en zoekgebied te specificeren.

**Bewerken** → **Ga naar offset... (Ctrl+G)**

De cursor naar een gespecificeerde offset verplaatsen.

## 4.1.3 Menu Beeld

**Beeld** → **Regeloffset tonen (F11)**

Aan/uitzetten van het tonen van de regeloffset in een paneel links.

**Beeld** → **Waarden of tekens tonen**

Selecteer welke van de interpretaties van een byte getoond wordt. Mogelijk zijn:

- **Waarden**
- **Tekens**
- **Waarden en tekens**

**Beeld → Waardecodering**

Selecteer de codering van de waarden uit:

- **Hexadecimaal**
- **Decimaal**
- **Octaal**
- **Binair**

**Beeld → Tekencodering**

Selecteer de codering van de tekens uit het submenu.

**Beeld → Niet-afdrukbare karakters tonen**

Aan/uitzetten van het tonen van niet-afdrukbare tekens. Als de weergave is uitgezet, wordt in plaats daarvan, op de overeenkomstige plaatsen in de kolom voor het teken, een vervangend teken getoond.

**Beeld → Stel bytes per regel in**

Selecteer het te tonen aantal bytes per regel uit de dialoog, de standaard waarde is 16 bytes.

**Beeld → Stel bytes per groep in**

Standaard worden de hexadecimale waarden getoond in groepen van 4 bytes. Door gebruik van dit menu-item kunt dit aan uw voorkeuren aanpassen in een dialoogvenster.

**Beeld → Dynamische opmaak**

Stel de regels in voor de opmaak van de gegevensweergave. Dit definieert hoeveel bytes per regel er worden getoond, afhankelijk van de breedte van het beeld. Mogelijke regels zijn:

- **Uit:** De opmaak is vast en gelijk aan het huidige aantal bytes per regel en niet aangepast aan de wijziging van de grootte van het beeld.
- **Breek alleen af op volledige bytegroepen:** Zet zoveel mogelijk bytes op een regel, zolang groepen bytes compleet zijn.
- **Aan:** Hetzelfde als de vorige, maar staat ook onvolledige groepen bytes toe.

**Beeld → Weergavemodus**

Selecteer de opmaak van het beeld uit:

- **Kolommen:** De waarden en tekens worden op de klassieke manier getoond met elk in een aparte kolom.
- **Rijen:** Het bijbehorende teken van een byte wordt direct onder de waarde getoond.

**Beeld → Horizontaal splitsen (Ctrl+Shift+T)**

Splits het weergavegebied met het huidige beeld met focus in twee delen en voeg een kopie van het huidige beeld in het nieuwe onderste gedeelte.

**Beeld → Verticaal splitsen (Ctrl+Shift+L)**

Splits het weergavegebied met het huidige beeld met focus in twee delen en voeg een kopie van het huidige beeld in het nieuwe rechter gedeelte.

**Beeld → Weergavegebied sluiten (Ctrl+Shift+R)**

Sluit het weergavegebied met het huidige beeld met focus.

**Beeld → Profiel weergeven**

Beeld-instellingen kunt u apart als weergaveprofielen opslaan. Het huidige geselecteerde profiel kunt u bijwerken met de huidige instellingen maar u kunt hier ook een nieuw profiel van aanmaken. Alle weergaveprofielen kunt u via een dialoogvenster beheren dat beschikbaar is via **Instellingen → Weergaveprofielen beheren...**

#### 4.1.4 Menu Vensters

Levert een lijst met de huidige beelden. Selecteer het actieve venster.

#### 4.1.5 Het menu Bladwijzers

Meerdere bladwijzers kunnen worden ingesteld voor een enkele reeks bytes. Elke reeks bytes heeft zijn eigen set bladwijzers en de bijbehorende set is onderaan het menu **Bladwijzers** weergegeven. Kies een bladwijzer uit het menu om de cursor te verplaatsen en het te bekijken.

##### OPMERKING

Bladwijzers zijn nu slecht voorbijgaand en worden niet opgeslagen als u een byte-array of het gehele programma sluit.

##### **Bladwijzers** → **Bladwijzer toevoegen (Ctrl+B)**

Locatie in een aantal bytes als bladwijzer instellen.

##### **Bladwijzers** → **Bladwijzer verwijderen (Ctrl+Shift+B)**

De huidige bladwijzer verwijderen. Dit commando is alleen beschikbaar als de cursor op een locatie in een bladwijzer staat.

##### **Bladwijzers** → **Alle bladwijzers wissen**

De lijst met bladwijzers wissen.

##### **Bladwijzer** → **Ga naar de vorige bladwijzer (Alt+Omhoog)**

Beweeg de cursor naar de vorige bladwijzer.

##### **Bladwijzer** → **Ga naar de volgende bladwijzer (Alt+Omlaag)**

Beweeg de cursor naar de volgende bladwijzer.

#### 4.1.6 Menu hulpmiddelen

Levert een lijst met geïnstalleerde hulpmiddelen. Schakel de weergave van elk hulpmiddel aan of uit. Een gedetailleerde beschrijving van elk hulpmiddel is te vinden in de sectie [Hulpmiddelen](#).

#### 4.1.7 Menu Instellingen

##### **Instellingen** → **Weergaveprofielen beheren...**

Open een dialoog om een standaard profiel aan te maken, te bewerken, te verwijderen of in te stellen.

## Hoofdstuk 5

# Verdiensten en licentie

Okteta

Programma Copyright 2006-2012 Friedrich W. H. Kosebau [kosebau@kde.org](mailto:kosebau@kde.org)

Documentatie Copyright 2008,2010 Friedrich W. H. Kosebau [kosebau@kde.org](mailto:kosebau@kde.org), Alex Richardson [arichardson.kde@gmail.com](mailto:arichardson.kde@gmail.com)

Op- of aanmerkingen over de vertalingen van de toepassing en haar documentatie kunt u melden op <http://www.kde.nl/bugs>.

Dit document is vertaald in het Nederlands door Freek de Kruijf [freekdekruijf@kde.nl](mailto:freekdekruijf@kde.nl).

Dit document is vertaald in het Nederlands door Ronald Stroethoff [stroet43@zonnet.nl](mailto:stroet43@zonnet.nl).

De vertaling werd nagelezen door Freek de Kruijf [freekdekruijf@kde.nl](mailto:freekdekruijf@kde.nl).

Deze documentatie valt onder de bepalingen van de [GNU vrije-documentatie-licentie](#).

Deze toepassing valt onder de bepalingen van de [GNU General Public License](#).