Das Handbuch zu Okteta

Friedrich W. H. Kossebau
Alex Richardson
Übersetzung: Ingo Malchow

Das Handbuch zu Okteta

Inhaltsverzeichnis

Zusammenfassung

Okteta ist ein einfacher Editor für die Rohdaten von Dateien. Diese Programme werden auch Hexeditor oder Binäreditor genannt.

Kapitel 1

Einführung

Okteta ist ein einfacher Editor für die Rohdaten von Dateien.

Die Daten werden auf zwei Arten dargestellt: Als numerische Werte der Bytes und als zugehörige Zeichen. Werte und Zeichen können in zwei Spalten (die traditionelle Anzeige von Hexeditoren) oder in Reihen mit der Anzeige der Werte über den Zeichen dargestellt werden. Die Bearbeitung der Daten ist sowohl als Werte wie auch als Zeichen möglich.

Neben den üblichen Bearbeitungsmöglichkeiten verfügt Okteta auch über eine kleine Auswahl an Werkzeugen, wie eine Dekodierungs-Tabelle für typische einfache Datentypen, eine Byte-Tabelle mit allen möglichen Zeichen und ihren zugehörigen Werten, eine Informationsansicht mit einer Statistik; eine Prüfsummenberechnung, ein Filterwerkzeug und ein Werkzeug um Zeichenfolgen herausziehen.

Alle Änderungen an den geladenen Daten können beliebig oft rückgängig gemacht und wiederhergestellt werden.

Kapitel 2

Grundlagen

2.1 Okteta starten

Geben Sie **`okteta`** in einer Befehlszeile ein, oder wählen Sie **Hex-Editor** aus der Gruppe **Programme** → **Dienstprogramme** im Startmenü.

Die Standard-Qt™- und KDE Frameworks 5-Befehlszeilenoptionen sind verfügbar, sie werden aufgelistet mit **`okteta --help`**.

Es gibt folgende spezielle Befehlszeilenoptionen für Okteta:

`<URL(s)>` - öffnet Dateien mit den angegebenen URL(s)

2.2 Benutzung

Das Hauptfenster von diese Anwendung besteht aus folgende Bereichen: Eine Menüleiste, eine Werkzeugleiste, eine Statusleiste, eine Seitenleiste mit Werkzeugen und dem Hauptbereich mit der Ansicht von Daten in Unterfenstern.

Wenn eine Datei geöffnet oder ein Byte-Feld erstellt wird, werden die enthaltenen Bytes zeilenweise fortlaufend mit einer vorgegebenen Anzahl von Bytes je Zeile angezeigt. Die Daten werden auf zwei Arten dargestellt, als numerische Werte derBytes und als zugehörige Zeichen. Werte und Zeilen können entweder nebeneinander in zwei Spalten oder übereinander mit den Werten oberhalb der Zeichen gezeigt werden. Links wird in der Anzeige der Versatz (Offset) des ersten Byte der Zeile angegeben.

Die Bearbeitung funktioniert ähnlich wie in den meisten Texteditoren, die Daten können geändert, kopiert, eingefügt, gezogen und abgelegt werden. Ein Cursor markiert die aktuelle Position. Das Drücken der Taste **Einfg** schaltet zwischen Überschreiben und Einfügen um. Der Überschreiben-Modus ist strikter als in Texteditoren, eine Bearbeitung, die die Länge des Byte-Felds ändert, ist nicht möglich.

Anders als in Texteditoren wird der Inhalt in zwei Ansichtsfenstern auf verschiedenen Art dargestellt. Eine Eingabe ist nur für eine dieser Ansichten möglich. Die beiden Ansichten haben miteinander verbundene Cursor, der Cursor in der aktiven Ansicht blinkt. In der Zeichenansicht können Zeichen wie im Texteditor eingegeben werden. Geben Sie in der Zahlenansicht eine Ziffer ein, wird ein minimaler Editor für die weitere Eingabe aktiviert.

Im Suchdialog können Sie nach einer bestimmten Folge von Bytes suchen, das Suchmuster kann als Wert (hexadezimal, dezimal, oktal, binär) oder als Text (kodiert in 8-Bit oder UTF-8) eingegeben werden.

Mehrere Byte-Felder können zur gleichen Zeit geöffnet werden, aber nur eines kann aktiv sein. Benutzen Sie das Menü **Fenster** zum Auswählen des aktiven Byte-Feldes.

Kapitel 3

Werkzeuge

3.1 Übersicht

Okteta enthält einige Werkzeuge, um Byte-Felder zu untersuchen und zu bearbeiten und auch Werkzeuge für einen allgemeineren Verwendungszweck. Diese Werkzeuge können im Menü **Extras** in der Menüleiste aktiviert werden. Für jedes Werkzeug gibt es eine kleine Ansicht, die entweder in den Seitenleisten oder freischwebend als Fenster angeordnet werden kann. Die Werkzeugansichten lassen sich in die Seitenleiste einfügen und wieder lösen, sie können neu angeordnet und gestapelt werden. Halten Sie dazu die linke Maustaste auf der Titelleiste der Werkzeugansicht gedrückt, verschieben Sie sie an den neuen Platz und lassen die linke Maustaste los, um die Aktion abzuschließen. Sie können diese Aktion durch Drücken der **Esc**-Taste abbrechen.

3.1.1 Daten untersuchen und verändern

Werte/Zeichen-Tabelle

Diese Tabelle listet alle möglichen Byte-Werte auf, sowohl als Zeichenwerte wie auch als verschiedene numerische Kodierungen.

Der ausgewählte Wert kann an der Cursor-Position mit einer definierten Anzahl an Bytes eingefügt werden. Dazu drücken Sie den Knopf **Einfügen** oder doppelklicken Sie in der entsprechenden Zeile der Tabelle.

Binärfilter

Dieser Filter führt Binäroperationen an den ausgewählten Bytes durch. Nach der Wahl der Operation (UND, ODER, Rotieren...) können die Parameter, falls vorhanden, im darunterliegenden Abschnitt eingestellt werden. Klicken Sie auf dem Knopf **Filtern**, um ihn anzuwenden.

Zeichenfolgen

Dieses Werkzeug findet die Zeichenfolgen in den ausgewählten Bytes. Nachdem Sie die minimale Zeichenfolgenlänge eingestellt haben, werden die Zeichenfolgen beim Drücken von **Extrahieren** durchsucht. Der Umfang der angezeigten Zeichenfolgenliste kann durch Einstellung eines Filters eingeschränkt werden.

Statistik

Dieses Werkzeug erstellt eine Statistik der ausgewählten Bytes an. Die Statistik gibt die Häufigkeit der einzelnen Bytewerte in der Auswahl an. Sie kann durch den Knopf **Erstellen** neu berechnet werden.

Prüfsumme

Mit diesem Werkzeug können verschiedene Prüfsummen oder Hashsummen für die ausgewählten Bytes berechnet werden. Nach der Auswahl des Berechnungsverfahrens und Einstellung der zugehörigen Parameter, falls erforderlich, wird die Summe durch Klicken auf den Knopf **Berechnen** ermittelt.

Dekodierungs-Tabelle

Diese Tabelle zeigt den Wert des Bytes oder der Bytes ab der Position des Cursors in einer der bekannten Datentypen wie Ganzzahl, Gleitkomma oder auch als UTF-8-Zeichen an. Durch Doppelklicken auf eine Zeile in der Tabelle wird ein Editor geöffnet, sodass der Wert bearbeitet und geändert werden kann.

Strukturen

Diese Werkzeug ermöglicht die Untersuchung und Bearbeitung von Byte-Feldern auf der Grundlage von benutzerdefinierten Strukturdefinitionen. Ausführlichen Anleitungen dazu finden Sie in diesem Abschnitt.

3.1.2 Allgemeine Werkzeuge

Dateisystem

Dieses Werkzeug ist ein Datei-Browser, damit können zu öffnende Dateien ausgewählt werden.

Dokumente

Dieses Werkzeug zeigt alle aktuell erstellten oder geladenen Dateien. Symbole zeigen die Datei mit der aktiven Ansicht, Dateien mit ungespeicherten Änderungen oder die Änderung der Datei durch andere Programme gegenüber der geladenen Kopie im Arbeitspeicher.

Lesezeichen

Hiermit können die Lesezeichen alternativ zum Menü **Lesezeichen** verwaltet werden.

> ANMERKUNG
> Lesezeichen sind zurzeit nur für die aktuell ausgeführte Sitzung gültig, sie werden nicht gespeichert, wenn ein Byte-Feld oder das Programm geschlossen wird.

Datei-Information

Dieses Werkzeug zeigt einige Informationen über die geöffnete Datei an, wie den Typ, den Speicherort und die Dateigröße.

Terminal

Ein eingebettetes Terminal, dessen Arbeitsordner nicht mit der aktiven Datei gekoppelt ist.

Zeichensatzumwandlung

Schreibt die Bytes so um, das die entsprechenden Zeichen dem anderen Zeichensatz entsprechen. Nur 8-Bit-Zeichensätze werden unterstützt, nicht übereinstimmende Zeichen werden zur Zeit durch den Wert 0 ersetzt.

3.2 Strukturen-Werkzeug

3.2.1 Allgemein

Das Strukturen-Werkzeug erlaubt die Untersuchung und Bearbeitung von Byte-Feldern auf der Basis von benutzerdefinierten Strukturdefinitionen, die aus Feldern, Verbunden, einfachen Typen und Aufzählungswerten erstellt werden können.

Dieses Werkzeug hat einen eigenen Einrichtungsdialog, der durch den Knopf **Einstellungen** geöffnet wird. In dem Dialog kann unter anderem der Anzeigestil (Dezimal, Hexadezimal, Binär) der Werte festgelegt werden. Außerdem kann ausgewählt werden, welche Strukturdefinition geladen und welche Strukturen in der Ansicht dargestellt werden.

Strukturen werden in Okteta in Strukturdefinitions-Dateien auf der Basis von XML-Dateien mit der Erweiterung `.osd` definiert. Zusätzlich gibt es eine `.desktop`-Datei mit Metadaten über die Strukturdefinitions-Datei wie zum Beispiel Autor, Webseite oder Lizenz.

Zurzeit ist es nicht möglich, Strukturdefinitionen in Okteta zu erstellen oder zu bearbeiten. Diese Definitionen müssen daher wie im nächsten Abschnitt beschrieben manuell erstellt werden.

3.2.2 Strukturdefinitionen installieren

3.2.2.1 Installation von Strukturen mit der Funktion „Neue Erweiterungen“ herunterladen

Am einfachsten lassen sich neue Strukturen mit der Funktion „Neue Erweiterungen“ herunterladen in Okteta installieren. Dazu öffnen Sie den Einrichtungsdialog des Strukturwerkzeugs. Gehen Sie zur Seite **Strukturen-Verwaltung** und drücken den Knopf **Neue Strukturen holen**. Im Dialog, der dann geöffnet wird, können Strukturen installiert und deinstalliert werden.

3.2.2.2 Strukturdefinitionen manuell installieren

Das Werkzeug Strukturen sucht die Definitionen der Strukturen im Unterordner `okteta/structures/` im Datenordner für Programme im Persönlichen Ordner des Benutzers. Diesen Ordner finden Sie mit der Eingabe von **`qtpaths --paths GenericDataLocation`** auf der Konsole. Wenn noch keine Strukturdefinitionen installiert wurden, muss dieser Ordner angelegt werden.

Für jede Strukturdefinition gibt es zwei Dateien: Eine Datei mit der tatsächlichen Definition und eine `.desktop`-Datei mit den Metadaten (Autor, Version usw.).

In diesem Ordner gibt es für jede Strukturdefinition einen eigenen Unterordner, der sowohl die `.desktop`-Datei und auch die `.osd`-Datei oder die `main.js`-Datei mit der Strukturdefinition.

Ist zum Beispiel der Datenordner der Programme `qtpaths --paths GenericDataLocation`, dann finden Sie eine Strukturdefinition namens BeispielStruktur im Unterordner `okteta/structures/BeispielStruktur`, darin dann die Dateien `BeispielStruktur.desktop` und `BeispielStruktur.osd`.

3.2.2.3 Neu installierte Strukturen benutzen

Wenn Sie eine neue Strukturdefinition installiert haben, müssen Sie Okteta neu starten, um sie benutzen zu können. Nach den Neustart von Okteta den Einrichtungsdialog der Strukturen öffnen. Wählen Sie dort die Seite **Strukturen-Verwaltung** und überprüfen Sie, ob die neue Strukturdefinition ausgewählt ist. Wechseln Sie dann zur Seite **Strukturen** und überprüfen Sie, ob das gewünschte Element in der rechten Liste enthalten ist.

3.2.3 Strukturdefinitionen gemeinsam nutzen

Allgemein gebräuchlich Strukturen müssen Sie wahrscheinlich nicht selbst definieren, sondern können zum Beispiel vorhandene Definitionen von der Seite store.kde.org benutzen.

Möchten Sie eine Strukturdefinition im Internet zur Verfügung stellen, packen Sie den Unterordner mit der `.desktop`-Datei und der Strukturdefinitions-Datei in ein Archiv, zum Beispiel ein gezipptes Tar-Archiv (`.tar.gz`). Bei dem Beispiel im vorigen Abschnitt ist das der Unterordner `BeispielStruktur` mit allen Dateien darin. Verwenden Sie diese Format, dann können Strukturdefinitionen in Okteta installiert werden. Eine manuelle Installation ist dann nicht erforderlich.

3.2.4 Strukturdefinitionen erstellen

> ANMERKUNG
> Eine aktuellere aber unvollständige Anleitung zum Schreiben von Strukturdefinitionen finden Sie im KDE UserBase Wiki.

Es gibt zwei Möglichkeiten, um eine Strukturdefinition zu erstellen, entweder im XML-Format oder als JavaScript. Mit JavaScript können Sie komplexere Strukturen mit Funktionen wie zum Beispiel der Überprüfung der Gültigkeit der Struktur schreiben. Mit XML haben Sie weniger Funktionen, wenn Ihnen aber eine statische Struktur reicht, ist das der einfachste Ansatz. Brauchen Sie dynamische Strukturen, bei denen zum Beispiel eine Feldlänge oder das Strukturlayout von anderen Werten in der Struktur abhängt, dann müssen Sie die Strukturdefinition in JavaScript schreiben. Es gibt eine Ausnahme für diese Regel: Haben Sie ein Feld, dessen Länge **genau** wie ein anderer Wert in der Struktur angenommen wird, dann können Sie auch XML verwenden. Ist dieser Wert aber so ähnlich wie *Länge -1*, dann müssen Sie JavaScript benutzen.

3.2.5 XML-Dateiformat einer Strukturdefinition

> ANMERKUNG
> Eine aktuellere aber unvollständige Anleitung zum Schreiben von Strukturdefinitionen finden Sie im KDE UserBase Wiki.

Die `.osd`-XML-Datei hat ein oberstes Element: <*data*> ohne Attribute. Innerhalb dieses Elements muss eines der folgenden Elemente enthalten sein:

<primitive>

Erstellt einen einfachen Typ wie z. B. *int* und *float*. Dieses Element kann keine untergeordneten Elemente enthalten und darf nur folgende Attribute haben:

type

Der Typ dieses einfachen Typs. Folgende Typen sind erlaubt:

- *char* für ein 8 Bit ASCII-Zeichen
- *int8*, *int16*, *int32*, *int64* für Ganzzahlwerte dieser Größen mit Vorzeichen
- *uint8*, *uint16*, *uint32*, *uint64* für Ganzzahlwerte dieser Größen ohne Vorzeichen
- *bool8*, *bool16*, *bool32*, *bool64* für einen vorzeichenlosen Boole'schen Wert (0 = falsch, jeder andere Wert = wahr) dieser Größe
- *float* für eine 32 Bit Fließkommazahl nach IEEE754
- *double* für eine 64 Bit Fließkommazahl nach IEEE754

<bitfield>

Zur Definition eines Bitfelds. Dieses Element kann keine untergeordneten Elemente enthalten und darf nur folgende Attribute haben:

width
Die Anzahl der von diesem Bitfeld benutzten Bits. Der Wert muss zwischen 1 und 64 liegen.

type
Der Typ dieses Bitfelds. Folgende Werte sind erlaubt:

- *unsigned* für ein Bitfeld, dessen Wert als vorzeichenlos interpretiert wird (Wertebereich von 0 bis $2^{width} - 1$)
- *signed* für ein Bitfeld, dessen Wert als Wert mit Vorzeichen interpretiert wird (Wertebereich von $-2^{width - 1}$ bis $2^{width - 1} - 1$)
- *bool* für ein Bitfeld, dessen Wert als Boolescher Wert interpretiert wird

> ANMERKUNG
> Denken Sie daran für „padding“ nach einem *<bitfield>* einen Wert anzugeben, sonst beginnt das nächste Element (ausgenommen Zeichenfolgen (strings) und Felder (arrays), die „padding“ automatisch einfügen) mitten in einem Byte. Ist dieses Verhalten gewünscht, brauchen Sie offensichtlich kein „padding“ anzugeben.

<enum>

Zur Definition eines einfachen Typs bei dem alle Werte als Elemente einer Aufzählung dargestellt werden. Diese Element kann keine untergeordneten Elemente enthalten, Sie brauchen jedoch ein Tag *<enumDef>* in der Datei zur Referenzierung. Folgende Attribute sind erlaubt:

enum
Die zurückliegende Aufzählung für diesen Wert. Muss zu dem Attribut *name* in einer der Tags *<enumDef>* in dieser Datei passen.

type
Der Typ dieser Aufzählung, mögliche Typen wie beim Element *<primitive>*. Nur die Typen *Double* und *Float* sind hier nicht sinnvoll.

<flags>

Dies ähnelt dem Element *<enum>*, mit dem einen Unterschied, dass die Werte als *bitweises Oder* aller Werte der Aufzählung dargestellt werden.

<struct>

Zur Definition einer Struktur. Alle anderen Element einschließlich *<struct>* können als untergeordnetes Element eingefügt werden und sind dann Bestandteil der gesamten Struktur.

<union>

Zur Definition eines Verbund-Datentyps. Ein Verbund ähnelt einem Datentyp *<struct>*, aber alle untergeordneten Elemente beginnen am gleichen Versatz. Nützlich für die Darstellung der gleichen Bytefolge auf verschiedene Arten.

<array>

Zur Definition eines Felds. Diese Element kann nur ein untergeordnete Element - den Typ des Felds - enthalten. Dieser Typ keine ein beliebiges Element sein, sogar ein *<array>* selbst. Es hat folgende Attribute:

length
Die Anzahl der Elemente in diesem Feld als Dezimalzahl. Alternativ kann hier auch eine Zeichenfolge für das Attribut „Name“ eines bereits definierten Elements *<primitive>*, *<enum>* or *<flags>* angeben werden, Der Wert dieses Elements gibt dann die Länge an. Dieser Wert ist zurzeit auf 10000 begrenzt, da größere Felder zu viel Speicher belegen und dieses Werkzeug verlangsamen.

<string>

Zur Definition einer Zeichenfolge in verschiedenen Kodierungen. Standard ist eine durch *NULL* begrenzte C-Zeichenfolge. Andere Arten von Zeichenfolgen können mit folgenden Attributen erzeugt werden:

terminatedBy

Dieses Attribut legt fest, welcher Unicode-Codepunkt die Zeichenfolge begrenzt. Der Wert muss eine hexadezimale Zahl sein, wahlweise mit führendem *0x*. Ist die Zeichenfolge in ASCII kodiert, sind nur Werte bis *0x7f* von Bedeutung. Ist weder eine Kodierung, noch die Attribute *maxCharCount* oder *maxByteCount* angegeben, wird für diesen Wert 0 wie bei einer C-Zeichenfolge angenommen.

maxCharCount

Die maximale Anzahl der Zeichen in dieser Zeichenfolge. Ist zusätzlich auch *terminatedBy* angegeben, begrenzt der kleiner der beiden Werte die Zeichenfolge. Die Attribute *maxByteCount* und *maxCharCount* schließen sich gegenseitig aus.

maxByteCount

Die maximale Länge der Zeichenfolge in Byte. Ist zusätzlich auch *terminatedBy* angegeben, begrenzt der kleiner der beiden Werte die Zeichenfolge. Die Attribute *maxByteCount* und *maxCharCount* schließen sich gegenseitig aus. Mit Kodierungen wie *ASCII* sind beide Attribute gleichwertig.

type

Die Kodierung dieser Zeichenfolge, folgende Kodierungen sind möglich:

- *ASCII*
- *LATIN-1*
- *UTF-8*
- *UTF-16-LE* oder *UTF-16-BE*. Wenn weder die Nachsilbe *-LE* oder *-BE* angegeben ist, wird die Byte-Reihenfolge *-LE* (Little Endian) angenommen.
- *UTF-32-LE* oder *UTF-32-BE*. Wenn weder die Nachsilbe *-LE* oder *-BE* angegeben ist, wird die Byte-Reihenfolge *-LE* (Little Endian) angenommen.

Jedes Element kann außerdem ein Attribut *name*. Dieser Name wird dann in der Strukturansicht angezeigt.

3.2.6 Ein Beispiel einer Strukturdefinition sowohl in XML als auch in JavaScript

> ANMERKUNG
> Eine aktuellere aber unvollständige Anleitung zum Schreiben von Strukturdefinitionen finden Sie im KDE UserBase Wiki.

3.2.6.1 Gemeinsame Schritte für beide Ansätze

Die Metadatendatei hat diesen Inhalt:

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

Name=Simple test structure
Comment=A very simple test structure containing only two 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 Das in Okteta benutzte Symbol für die Anzeige dieser Struktur kann durch Ausführung des Befehls **`kdialog --geticon`**. Alternativ kann der Pfad zu einem Symbol eingetragen werden.

Die Bedeutung der Felder erschließt sich leicht aus Ihrem Namen, ausgenommen für das Feld `X-KDE-PluginInfo-Name`. Der Wert dieses Felds muss mit dem Namen des Ordners, in dem diese Datei enthalten ist, und auch dem Namen der `.desktop`-Datei übereinstimmen. Bei XML-Strukturdefinitionen muss der Wert dieses Feldes außerdem den gleichen Namen wie die `.osd`-Datei haben.

In diesem Beispiel befindet sich die Datei `simplestruct.desktop` im Ordner `simplestruct`. Bei XML-Strukturdefinitionen enthält der Ordner auch noch eine Datei namens `simplestruct.osd`. Wird JavaScript verwendet, gibt es stattdessen die Datei `main.js`.

3.2.6.2 Eine einfache XML-Strukturdefinition

Zuerst wird die Definition einer einfachen Teststruktur erstellt, die nur ganzzahlige Datentypen wie ein „char“, einen vorzeichenbehafteten 32-bit Integerwert und ein Bitfeld enthält. In C/C++ wird das so geschrieben:

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

Als erstes wird die `.osd`-Datei mit dem im vorherigen Abschnitt definierten Dateiformat geschrieben. Diese erhält den Namen `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>
```

Dies ist ähnlich wie die C/C++ Definition.

Erstellen Sie nun einen Ordner `simplestruct` im Strukturinstallationsordner wie im Abschnitt manuelle Installation beschrieben. Kopieren Sie diese beiden Dateien in diesen Ordner. Starten Sie Okteta neu und benutzen Sie die neue Struktur.

3.2.6.3 Die einfache Struktur in JavaScript

Um die oben gezeigte Struktur in JavaScript zu schreiben, erstellen Sie eine Daten namens `main.js` anstelle der Datei `simplestruct.osd` und ändern Sie X-KDE-PluginInfo-Category=structure zu X-KDE-PluginInfo-Category=structure/js. Die Datei sollten folgenden Inhalt haben:

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

Die in Okteta angezeigte Struktur ist immer der Rückgabewert der Funktion `init`.

Die folgenden Funktionen können benutzt werden, um einen primitiven Typ zu erzeugen:

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

Die Funktion „bitfield“ benötigt zwei Parameter, als erstes einen String wie `bool`, `signed` oder `unsigned`. Der zweite Parameter ist ein Integer-Wert, der die Breite in Bits angibt.

3.2.6.4 Komplexere Strukturen

Als nächstes wird eine komplexere Struktur definiert. Sie wird ″complex″ genannt und in der Datei `complex.osd` gespeichert. Diese Struktur enthält zwei Felder, eines mit fester Länge und ein anderes, dessen Länge erst zur Laufzeit bestimmt wird, außerdem noch eine verschachtelte Struktur und einen Verbund.

```
<?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>
```

Dies entspricht folgendem Pseudocode in C/C++

```
struct complex {
            uint8_t size;
            union aUnion {
              int8_t fourBytes[4];
            };
            struct nested {
              char string[size] //not valid C++, references value of the ↩
                  uint8 size
            };
          };
```

> ANMERKUNG
> Referenzfelder für dynamische Felder müssen vor dem Feld definiert werden.

Als nächstes wird die Datei `complex.desktop` wie im vorher gezeigten Beispiel erstellt. Benutzen Sie für `X-KDE-PluginInfo-Name` den richtigen Namen und installieren Sie beide Dateien.

3.2.6.5 Weitere Informationen

Einige Beispiele von Strukturdefinitionen finden Sie im Git-Archiv. Dort ist zum Beispiel der Vorspann für PNG- und ELF-Dateien enthalten. Ein XML-Schema, das die Struktur von `.osd`-Dateien beschreibt, finden Sie hier. Brauchen Sie weitere Informationen, kontaktieren Sie arichardson.kde@gmail.com.

Kapitel 4

Benutzeroberfläche

4.1 Menüeinträge

Außer den bekannten KDE-Menüeinträgen, die im Kapitel Menüs der KDE-Grundlagen beschrieben werden, gibt es folgende spezielle Menüeinträge für Okteta:

4.1.1 Das Menü Datei

Datei → Neu (Strg+N)

Legt ein neues Byte-Feld an.

- **Leer**: ... als leeres Byte-Feld.
- **Aus der Zwischenablage**: ... mit dem aktuellen Inhalt der Zwischenablage.
- **Muster ...**: ... mit einem vorgegebenen Muster.
- **Zufällige Daten ...**: ... mit zufälligen Daten.
- **Zeichenfolge**: ... einf Folge aller Bytes von 0 bis 255.

Datei → Exportieren

Exportiert die ausgewählten Bytes in eine Datei...

- **Werte**: .. kodiert als Byte-Werte. Als Standard werden die werte mit einem Leerzeichen getrennt. Das **Trennzeichen** kann im Dialog **Exportieren** geändert werden.
- **Zeichen**: ... kodiert als einfacher Text.
- **Base64**: ... kodiert im Format Base64.
- **Base32**: ... kodiert im Format Base32.
- **Ascii85**: ... kodiert im Format Ascii85 oder Base85.
- **Uu-Kodierung**: ... kodiert im Format Uuencoding.
- **Xx-Kodierung**: ... kodiert im Format Xxencoding.
- **Intel Hex**: ... kodiert im Format Intel HEX.
- **S-Record**: ... kodiert im Format S-Record.
- **C-Feld**: ... definiert als ein Feld in der Programmiersprache C.
- **Offset, Werte und Text**: ... wie in der Datenansicht mit Offset, Byte-Werten und Zeichen.

Datei → Zugriffsrechte → Nur-Lesen-Modus

Mit dieser Einstellungen können die Werte im geladenen Byte-Feld nicht geändert werden.

Datei → Alle anderen schließen

Schließt alle Felder außer dem aktuellen Byte-Feld.

4.1.2 Das Menü Bearbeiten

Bearbeiten → Kopieren als

Kopiert die ausgewählten Bytes in verschiedenen Formaten in die Zwischenablage. Eine Liste der verfügbaren Formate finden Sie im Menüeintrag **Datei → Exportieren**.

Bearbeiten → Einfügen

Muster ...

Fügt eine angegebenes Muster von Bytes an der Cursor-Position ein.

In diesem Dialogfenster können Sie das Format (**Hex**, **Dez**, **Okt**, **Bin**, **Zeichen** oder **UTF-8**) der Eingabe einstellen und vorgeben, wie oft das Muster eingefügt wird.

Bearbeiten → Auswahl aufheben (Strg+Umschalt+A)

Hebt die vorhandene Auswahl auf.

Bearbeiten → Bereich auswählen ... (Strg+E)

Öffnet einen Dialog als Teil des Hauptfenster, in dem ein Bereich ausgewählt werden kann.

Bearbeiten → Einfügen-Modus (Einfg)

Schaltet im Editor zwischen Einfügen und Überschreiben um.

> ANMERKUNG
> Im Modus Überschreiben ist es nicht möglich, die Größe der Daten zu ändern (kein Anhängen oder Entfernen von Bytes).

Bearbeiten → Suchen ... (Strg+F)

Findet ein eingegebenes Muster in dem Dokument. Es kann nach Hexadezimal-, Oktal-, Binär- oder Textmustern gesucht werden.

Mit den Einstellungen im Dialogfenster können Sie den Startpunkt, die Richtung und die Reichweite der Suche bestimmen.

Bearbeiten → Gehe zu Offset ... (Strg+G)

Bewegt den Cursor zu einem angegebenen Offset.

4.1.3 Das Menü Ansicht

Ansicht → Zeilen-Offset anzeigen (F11)

Schaltet die Anzeige des Zeilen-Offsets auf der linken Seite ein und aus.

Ansicht → Werte oder Zeichen anzeigen

Legt fest, welche Byte-Interpretationen angezeigt werden. Es sind folgende Einstellungen möglich:

- **Werte**
- **Zeichen**
- **Werte und Zeichen**

Ansicht → Werte-Kodierung

Wählt die Kodierung der Werte aus:

- **Hexadezimal**

- **Dezimal**
- **Oktal**
- **Binär**

Ansicht → **Zeichen-Kodierung**

Wählt die Kodierung für die Zeichen aus dem Untermenü.

Ansicht → **Nicht druckbare Zeichen anzeigen**

Schaltet die Anzeige von nicht-druckbare Zeichen ein oder aus. Wenn diese Option deaktiviert ist, werden an den entsprechenden Stellen in der Zeichenspalte Ersatzzeichen dargestellt.

Ansicht → **Bytes pro Zeile festlegen**

Auswahl der je Zeile angezeigten Bytes in einem Dialog, der Standardwert beträgt 16 Byte.

Ansicht → **Bytes pro Gruppe festlegen**

In der Voreinstellung werden die hexadezimalen Werte in Gruppen von 4 Byte angezeigt. Mit diesem Menüeintrag können Sie diese Einstellung in einem Dialog anpassen.

Ansicht → **Dynamischer Umbruch**

Legt die Regeln für die Darstellung der Daten fest. Dadurch wird bestimmt, wieviele Bytes pro Reihe dargestellt werden, abhängig von der Breite der Ansicht. Mögliche Regeln sind:

- **Aus**: Die Anzeige bleibt unverändert bei der aktuell vorhandenen Anzahl von Byte je Zeile und wird bei einer Größenänderung des Ansichtsfensters nicht angepasst.
- **Nur vollständige Bytegruppen umbrechen**: Stellt möglichst viele Bytes pro Reihe dar, Gruppen von Bytes werden nicht getrennt.
- **An**: Stellt möglichst viele Bytes pro Reihe dar, trennt dabei aber Gruppen von Bytes.

Ansicht → **Anzeigemodus**

Wählt das Layout für die Ansicht aus:

- **Spalten**: Die Interpretationen als Werte und Zeichen werden im klassischen Layout angezeigt. In diesem werden beide in getrennten Spalten aufgelistet.
- **Zeilen**: Die Interpretation eines Bytes als Zeichen wird direkt neben der als Wert angezeigt.

Ansicht → **Geteilte Ansicht** → **Waagerecht teilen (Strg+Umschalt+T)**

Teilt den Bereich der aktuellen Ansicht in zwei Teile und zeigt eine Kopie dieser Ansicht in der unteren Hälfte an.

Bearbeiten → **Senkrecht teilen (Strg+Umschalt+L)**

Teilt den Bereich der aktuellen Ansicht in zwei Teile und zeigt eine Kopie dieser Ansicht in der rechten Hälfte an.

Ansicht → **Ansicht schließen (Strg+Umschalt+R)**

Schließt den Ansichtsbereich, die gerade aktiv ist.

Ansicht → **Ansichtsprofil**

Die Einstellungen einer Ansicht können getrennt als Ansichtsprofil gespeichert werden. Das aktuell ausgewählte Profil kann direkt mit den Einstellungen der aktuellen Ansicht aktualisiert oder eine neues Profil kann damit erstellt werden. Alle Ansichtsprofile können in einem Dialog verwaltet werden, der mit **Einstellungen** → **Ansichtsprofile verwalten ...** geöffnet wird.

4.1.4 Das Menü Fenster

Zeigt eine Liste von aktuellen Ansichten an. Wählen Sie daraus das aktive Fenster.

4.1.5 Das Menü Lesezeichen

Mehrere Lesezeichen können für ein einzelnes Byte-Feld gesetzt werden. Jedes Byte-Feld hat seinen eigenen Satz an Lesezeichen. Die vorhandenen Lesezeichen werden im Menü **Lesezeichen** am Ende angezeigt. Wählen Sie ein Lesezeichen aus dem Menü, um den Cursor auf die entsprechende Stelle zu setzen.

> ANMERKUNG
> Lesezeichen sind zurzeit nur für die aktuell ausgeführte Sitzung gültig, sie werden nicht gespeichert, wenn ein Byte-Feld oder das Programm geschlossen wird.

Lesezeichen → **Lesezeichen hinzufügen (Strg+B)**

Fügt ein Lesezeichen innerhalb eines Byte-Feldes ein.

Lesezeichen → **Lesezeichen löschen (Strg+Umschalt+B)**

Entfernt das aktuelle Lesezeichen. Dieser Befehl nur verfügbar, wenn der Cursor an einer Position mit Lesezeichen steht.

Lesezeichen → **Alle Lesezeichen löschen**

Löscht die gesamte Lesezeichenliste.

Lesezeichen → **Zum vorherigen Lesezeichen gehen (Alt+Pfeil hoch)**

Bewegt den Cursor zu dem vorherigen Lesezeichen.

Lesezeichen → **Zum nächsten Lesezeichen gehen (Alt+Pfeil runter)**

Bewegt den Cursor zu dem nächsten Lesezeichen.

4.1.6 Das Menü Extras

Das Menü zeigt eine Liste mit allen installierten Werkzeuge an und erlaubt es Ihnen, die Anzeige der einzelnen Werkzeuge ein- und auszuschalten. Eine ausführliche Beschreibung für jedes Werkzeug finden Sie in dem Kapitel Werkzeuge.

4.1.7 Das Menü Einstellungen

Einstellungen → **Ansichtsprofile verwalten ...**

Öffnet einen Dialog, in dem Sie Ansichtsprofile erstellen, bearbeiten, löschen und ein Profil als Standard setzen können.

Kapitel 5

Danksagungen und Lizenz