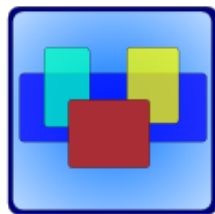


Het handboek voor de editor voor reguliere expressie

Jesper K. Pedersen
Vertaler/Nalezer: Freek de Kruijf



Het handboek voor de editor voor reguliere expressie

Inhoudsopgave

1	Inleiding	5
2	Wat is een reguliere expressie?	6
3	De reguliere expressie editor gebruiken	9
3.1	De organisatie van het scherm	9
3.2	Hulpmiddelen voor bewerkingen	10
3.2.1	Selectiehulpmiddel	10
3.2.2	Tekstgereedschap	10
3.2.3	Lettertekengereedschap	10
3.2.4	Gereedschap voor willekeurig teken	11
3.2.5	Herhaalgereedschap	11
3.2.6	Alternatievengereedschap	11
3.2.7	Samenstellingengereedschap	12
3.2.8	Gereedschappen voor regelbegin en geleide	12
3.2.9	Gereedschappen voor (niet-)woordgrenzen	12
3.2.10	Gereedschappen voor positief en negatief vooruitkijken	13
3.3	Gebruikergedefinieerde reguliere expressies	13
4	Fouten melden en suggesties voor functies	14
5	Veel voorkomende vragen	15
5.1	Ondersteunt de editor voor reguliere expressies achterwaartse referenties?	15
5.2	Ondersteunt de editor voor reguliere expressies het tonen van gevonden overeenkomsten?	15
5.3	Ik ben de auteur van een KDE-programma; hoe kan ik dit grafische element in mijn applicatie gebruiken?	15
5.4	Ik kan de knop <i>Editor voor reguliere expressies</i> niet vinden in bijvoorbeeld konqueror in een andere KDE3 installatie, waarom?	15
6	Dankbetuigingen en licentie	16

Samenvatting

Dit handboek beschrijft de editor voor reguliere expressies

Hoofdstuk 1

Inleiding

De editor voor reguliere expressies is een editor voor het grafisch bewerken van reguliere expressies (in plaats van via de ASCII-syntaxis). Traditioneel werden reguliere expressie getypt in ASCII-syntaxis, die er bijvoorbeeld uitzag als $\hat{.}^*kde\backslash b$. Grote nadelen van deze stijl zijn:

- Hij is lastig te begrijpen voor niet-programmeurs.
- U dient voor bepaalde symbolen een *stuurcode* te gebruiken. Om een ster te vinden, dient u bijvoorbeeld \backslash^* te typen.
- U dient regels te onthouden voor de *evaluatievolgorde* (Waarmee komt $x|y^*$ overeen? Een enkele x of een aantal keer y , OF een aantal keer x en y door elkaar?)

De editor voor reguliere expressies, echter, laat u uw reguliere expressie op eenduidige wijze *tekenen*. de editor lost item twee en drie uit bovengenoemde opsomming op. Dit zorgt er misschien niet voor dat reguliere expressies beschikbaar worden voor niet-programmeurs, maar dat kunnen alleen tests door gebruikers uitwijzen. Dus, als u een niet-programmeur bent die de kracht van de reguliere expressies hebt bemachtigd met deze editor, [laat het mij dan alstublieft weten](#).

Hoofdstuk 2

Wat is een reguliere expressie?

Een reguliere expressie is een manier om *condities* te specificeren waaraan voldaan dient te worden voor een beoogde situatie. Wanneer u normaal gesproken in een teksteditor zoekt, specificeert u *letterlijk* de tekst waarnaar gezocht moet worden. Wanneer u daarentegen gebruik maakt van reguliere expressies, geeft u aan hoe een gegeven zoekresultaat eruit dient te zien. Voorbeelden hiervan zijn onder andere *ik zoek naar het woord KDE, maar alleen aan het begin van de regel*, of *ik zoek naar het woord de, maar het moet een losstaand woord zijn*, of *ik ben opzoek naar bestanden die beginnen met het woord test, gevolgd door een aantal getallen, bijvoorbeeld test12, test107 en test007*

U construeert een reguliere expressie op basis van sub-expressies, zoals groot Lego speelgoed van kleinere onderdelen gemaakt worden. Evenals in de Lego-wereld, is er een aantal basisbouwstenen. In het volgende beschrijf ik elk van deze basisbouwstenen aan de hand van een aantal voorbeelden.

Example 2.1 Zoeken naar normale tekst.

Als u op zoek bent naar een gegeven tekst, dan is een reguliere expressie zeker geen goede keuze. De reden hiervoor is dat reguliere expressies een speciale betekenis toekennen aan sommige karakters. Hiertoe behoren onder andere de volgende karakters: *|\$. Dus als u wilt zoeken naar de tekst `kde.` (m.a.w. de karakters `kde` gevolgd door een punt), dan zou u dit als `kde\.` dienen te specificeren.¹ Het schrijven van `\.` in plaats van gewoon `.` wordt *escaping* genoemd.

Example 2.2 URL's zoeken

Wanneer u iets dat op een URL lijkt selecteert in KDE, dan biedt het programma **klipper** aan **konqueror** te starten met de geselecteerde URL.

Klipper doet dit door de selectie te vergelijken met een aantal reguliere expressies, en wanneer één van de reguliere expressies ermee overeenkomt, wordt het bijbehorende commando uitgevoerd.

De reguliere expressie voor URL's zegt (onder andere), dat de selectie moet beginnen met de tekst `http://`. Dit wordt beschreven met reguliere expressies door `http://` voor de tekst te plaatsen met een dakje (het teken `^`).

Het bovenstaande is een voorbeeld van posities zoeken met behulp van reguliere expressies. Op overeenkomstige wijze kan naar de positie *regeleinde* gezocht worden met het teken `$` (m.a.w. het dollarteken).

Example 2.3 Zoeken naar het woord `the`, maar niet naar `there`, `breathe` of `another`

Twee extra types positie kunnen op bovenstaande wijze gezocht worden, namelijk *de positie op een woordgrens*, en *de positie op een niet-woordgrens*. De posities worden gespecificeerd met de tekst `\b` (voor woordgrens) en `\B` (voor niet-woordgrens)

Dus, zoeken naar het woord `the` kan gedaan worden met de reguliere expressie `\bthe\b`. Dit geeft aan dat we zoeken naar `the` zonder letters aan elke kant ervan (m.a.w. met een woordgrens aan elke kant)

De vier reguliere expressies om op positie te zoeken worden in de reguliere expressie editor toegevoegd met [vier verschillende positiehulpmiddelen](#)

Example 2.4 Zoeken naar ofwel `dit` of `dat`

Stelt u zich voor dat u een document wilt doorzoeken voor ofwel het woord `dit` of het woord `dat`. Met een normale zoekmethode kunt u dit in twee zoekacties, de eerste keer, zou u naar `dit` zoeken, en de tweede maal zou u zoeken naar `dat`.

Met behulp van reguliere expressies zou u naar beide zoeken in dezelfde zoekactie. Dit doet u door te zoeken naar `dit|dat`. bijv. door de twee woorden van elkaar te scheiden met een verticale balk.²

In de reguliere expressie editor schrijft u de verticale balk niet zelf, maar in plaats daarvan selecteert u [hulpmiddel alternatieven](#), en voegt u sub-reguliere expressies in boven elkaar.

Example 2.5 Iets zoeken

Reguliere expressies worden vaak vergeleken met het zoeken met wildcards in de shell - dat is de mogelijkheid een aantal bestanden te specificeren met een asterisk. U kent zoeken met wildcard waarschijnlijk van de volgende voorbeelden:

- `rm *~` - hier is `*~` de shell wildcard die overeenkomt met ieder bestand dat eindigt met `~`
- `cat test?.res` - komt overeen met elk bestand dat begint met `test` gevolgd door twee willekeurige tekens, gevolgd door extensie `.res`

In de shell komt de asterisk overeen met een willekeurig teken en een willekeurig aantal malen. Met andere woorden, de asterisk komt overeen met *alles*. Dit wordt geschreven als `.*` met de syntaxis voor reguliere expressies. De punt komt overeen met een willekeurig afzonderlijk teken, m.a.w. slechts *één* teken, en de asterisk zegt dat de reguliere expressie die eraan voorafgaat een willekeurig aantal malen gezocht dient te worden. Samen betekent het een willekeurig afzonderlijk teken een willekeurig aantal malen.

Dit ziet er misschien onnodig ingewikkeld uit, maar u zult de kracht ervan zien wanneer u het overzicht van het geheel krijgt. Hier is een andere eenvoudige reguliere expressie: `a`. De letter `a` op zichzelf is een reguliere expressie die overeenkomt met een enkele letter, namelijk de letter `a`. Als u deze combineert met de asterisk, bijv. `a*`, dan hebben we een reguliere expressie die een willekeurig aantal keren de letter `a` zoekt.

We kunnen verscheidene reguliere expressies opeenvolgend met elkaar combineren, bijvoorbeeld `ba(na)*`.³ Stelt u zich voor dat u deze reguliere expressie in het zoekveld van een tekst-editor getypt had, dan had u (onder andere) de volgende woorden gevonden: `ba`, `bana`, `banana`, `banananananana`

Gegeven bovenstaande informatie, is het hopelijk niet moeilijk voor u de shell wildcard `test?.res` als een reguliere expressie te schrijven. Antwoord: `test\.\res`. De punt alleen is een willekeurig teken. Om alleen een punt te zoeken dient u `\.` te schrijven.⁴ Met andere woorden, de reguliere expressie `\.` komt overeen met een punt, terwijl een punt op zichzelf overeenkomt met een willekeurig teken.

In de reguliere expressie editor wordt de herhaalde reguliere expressie gemaakt met het [hulpmiddel herhaalde inhoud](#)

Example 2.6 Het vervangen van & door & in een HTML-document

In HTML dient het speciale teken & geschreven te worden als & - dit is vergelijkbaar met gebruik van stuurcodes in reguliere expressies.

Stelt u zich voor dat u een HTML-document heeft geschreven in een normale editor (bijvoorbeeld XEmacs of Kate), en u hebt deze regel helemaal vergeten. Wat u zou doen wanneer u zich uw vergissing realiseerde, is elke & vervangen door &.

Dit kunt u eenvoudig doen met het normale zoeken en vervangen, er ligt echter een adder onder het gras. Stelt u zich voor dat u zich deze regel wel *een beetje* hebt herinnerd - en het op een aantal plaatsen goed gedaan hebt. Onvoorwaardelijk vervangen zou resulteren in dat & vervangen wordt door & amp;

Wat u eigenlijk wilt is dat & alleen vervangen dient te worden als het *niet* gevolgd wordt door de letters amp;. U kunt dit doen door gebruik te maken van reguliere expressies met *positief vooruitkijken*.

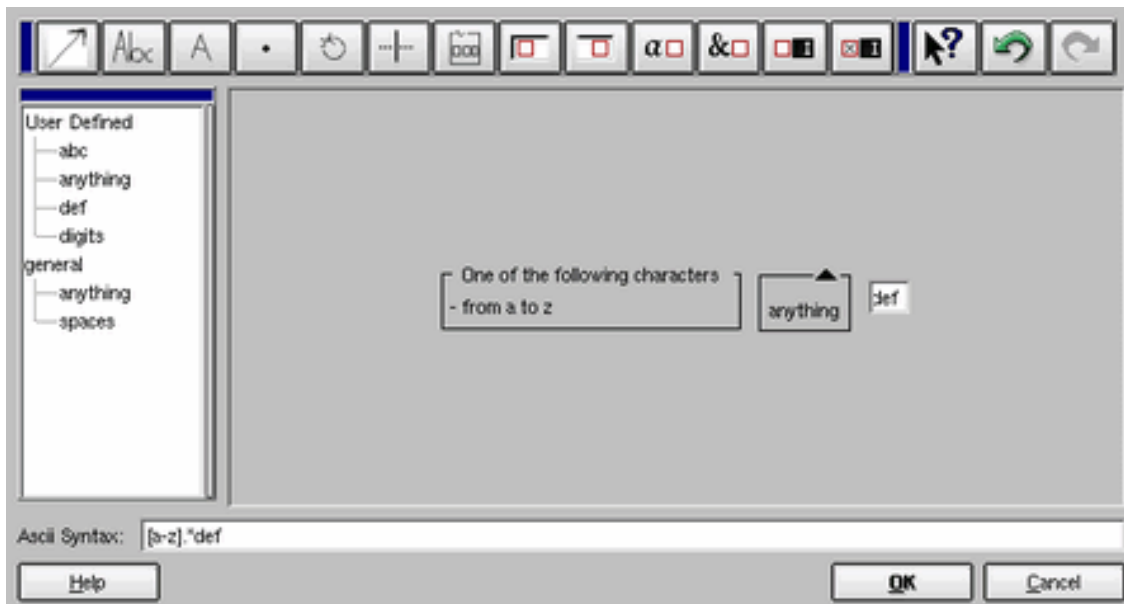
De reguliere expressie, die alleen overeenkomt met een en-teken als die niet gevolgd wordt door de letters amp; ziet er als volgt uit: &(?!amp;). Dit is natuurlijk eenvoudiger te lezen door de reguliere expressie editor te gebruiken, waar u de [hulpmiddelen voor vooruitkijken](#) zou gebruiken.

Hoofdstuk 3

De reguliere expressie editor gebruiken

Dit hoofdstuk vertelt hoe de reguliere expressie editor werkt.

3.1 De organisatie van het scherm



Het belangrijkste onderdeel van de editor is natuurlijk de ruimte voor bewerkingen; dit is de ruimte waar u uw reguliere expressies kunt tekenen. Dit gebied is het grote grijze middendeel.

Boven de bewerkingsruimte hebt u twee werkbalken, de eerste bevat [bewerkingsacties](#) - deze lijken erg op tekenhulpmiddelen in een tekenprogramma. De tweede werkbalk bevat de knop *Wat is dit?*, en knoppen om bewerkingen ongedaan te maken en opnieuw uit te voeren.

Onder de bewerkingsruimte vindt u de huidige reguliere expressie, in de zogenoemde ascii syntaxis. De ascii syntaxis wordt bijgewerkt terwijl u de reguliere expressie in de grafische editor bewerkt. Indien u liever de ascii-syntaxis bijwerkt dan kan dit, de grafische editor wordt direct bijgewerkt om uw wijzigingen weer te geven.

Ten slotte vindt u links van de bewerkingsruimte een aantal voorgebouwde reguliere expressies. Deze hebben twee doelen: (1) Wanneer u de editor laadt met een reguliere expressie, dan wordt deze reguliere expresse *opgeschoond* of begrijpelijker door veelvoorkomende reguliere expressies te vervangen. In de schermafbeelding boven, kunt u zien hoe de ascii-syntaxis `.*` vervangen is door een veld `"iets"`. (2) Wanneer u reguliere expressies invoegt, kunt u bouwstenen voor uw eigen reguliere expressies vinden in de verzameling voorgebouwde reguliere expressies. Zie de paragraaf over [gebruikergedefinieerde reguliere expressies](#) om te leren hoe u uw eigen reguliere expressies kunt opslaan.

3.2 Hulpmiddelen voor bewerkingen

De tekst in deze paragraaf verwacht dat u het hoofdstuk over [wat een reguliere expressie is](#) heeft gelezen, of voorgaande kennis over dit onderwerp heeft.

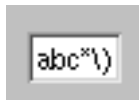
Al de hulpmiddelen voor bewerkingen bevinden zich in de menubalk boven het bewerkingsgebied. Elk van hen zal beschreven worden in het volgende.

3.2.1 Selectiehulpmiddel



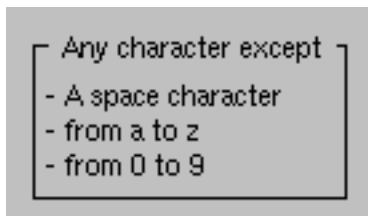
Het selectiegereedschap wordt gebruikt om elementen te markeren voor knippen-en-plakken en drag-and-drop. Dit is vergelijkbaar met het selectiegereedschap in elk tekenprogramma.

3.2.2 Tekstgereedschap



Met dit hulpmiddel kunt u normale tekst invoeren die gezocht dient te worden. Er wordt naar de letterlijke tekst gezocht, bijv. dat u zich niet hoeft te bekommeren om gebruik van stuurcodes voor speciale tekens. In het voorbeeld boven wordt de volgende reguliere expressie gebouwd:
`abc*\\`

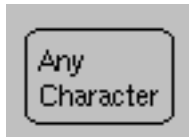
3.2.3 Lettertekengereedschap



Met dit gereedschap kunt u speciale bereiken voor lettertekens invoeren. Voorbeelden zijn onder andere ascii-teksten `[0-9]`, `[^a-zA-Z, _]`. Wanneer u een item toevoegt met met dit hulpmiddel, verschijnt een dialoog, waarin u de tekenbereiken kunt specificeren.

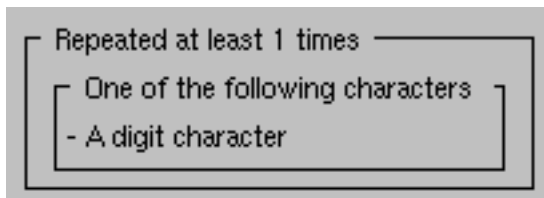
Zie de beschrijving van [herhaalde reguliere expressies](#).

3.2.4 Gereedschap voor willekeurig teken



Dit is een reguliere expressie "dot" (.). Deze komt overeen met een willekeurig enkel letterteken.

3.2.5 Herhaalgereedschap



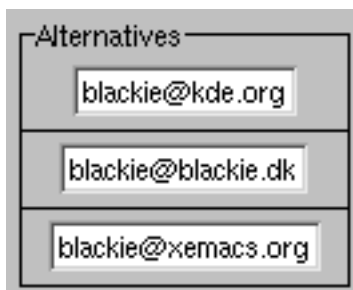
Dit is het gereedschap voor herhaalde elementen. Het bevat wat in ascii syntaxis gerepresenteerd wordt door een asterisk (*), een plusteken (+), een vraagteken (?), en bereiken ({3,5}). Wanneer u een item invoegt met dit gereedschap, verschijnt een dialoogvenster waarin u gevraagd wordt naar het aantal malen om het element te herhalen.

U geeft aan wat herhaald dient te worden door de herhaalde inhoud te tekenen in een veld dat dit hulpmiddel invoegt.

Herhaalde elementen kunnen zowel van buiten naar binnen als van binnen naar buiten gebouwd worden. Dat betekent dat u eerst datgene kunt tekenen dat u wilt herhalen, het selecteert en het herhaalgereedschap gebruikt. Of eerst het herhaalelement kunt invoegen, en datgene dat herhaald dient te worden erin plaatst.

Zie de beschrijving van [herhaalde reguliere expressies](#).

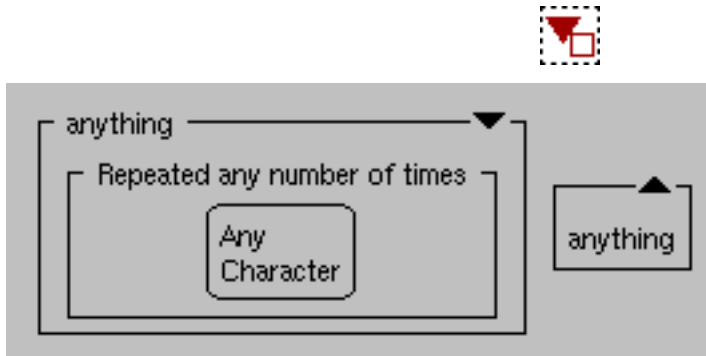
3.2.6 Alternatievengereedschap



Dit is het gereedschap voor alternatieve reguliere expressies (|). U kunt de alternatieven specificeren door ze boven elkaar te tekenen in het kader dat dit hulpmiddel invoegt.

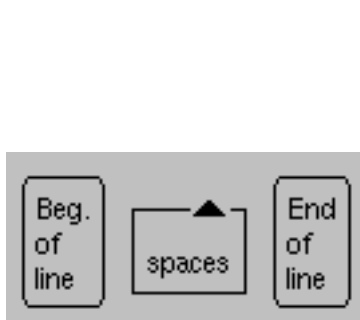
Zie de beschrijving van [alternatieve reguliere expressies](#)

3.2.7 Samenstellingengereedschap



Het samenstellingengereedschap representeert geen reguliere expressies. Het wordt gebruikt om andere onderdelen te groeperen in een kader, dat eenvoudig ingeklapt kan worden zodat alleen de titel getoond wordt. Deze is te zien in het rechterdeel van bovenstaande schermafdruk.

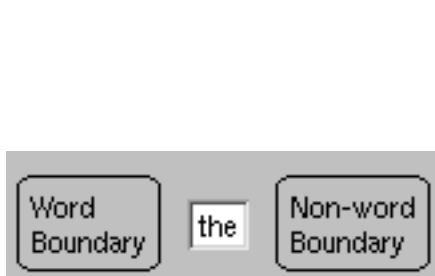
3.2.8 Gereedschappen voor regelbegin en regeleinde



De gereedschappen voor regelbegin en regeleinde zoeken respectievelijk naar overeenkomsten van het begin van de regel, en het eind van de regel. De reguliere expressies in de schermafdruk boven komen dus overeen met regels die alleen spaties bevatten.

Zie de beschrijven van [reguliere expressies voor posities](#)

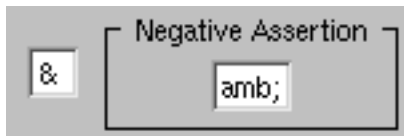
3.2.9 Gereedschappen voor (niet-)woordgrenzen



De grensgereedschappen zoeken naar overeenkomsten van respectievelijk woordgrens en niet-woordgrens. De reguliere expressie in de schermafdruk zoekt dus naar woorden die beginnen met the. Het woord the zelf, echter, wordt niet gezocht.

Zie de beschrijving van [reguliere expressies voor grenzen](#)

3.2.10 Gereedschappen voor positief en negatief vooruitkijken



De vooruitkijkgereedschappen specificeren een positieve of negatieve reguliere expressie waarnaar gezocht dient te worden. De overeenkomst maakt echter geen deel uit van de totale gezochte overeenkomst.

N.B.: U kunt vooruitkijkgereedschappen alleen aan het einde van een reguliere expressie plaatsen. De grafische Reguliere expressie editor controleert dit echter niet.

Zie de beschrijving van [reguliere expressies voor vooruitkijken](#)

3.3 Gebruikergedefinieerde reguliere expressies

Links van het bewerkingsgebied bevindt zich een lijst die gebruikergedefinieerde reguliere expressies bevat. Sommige reguliere expressies worden van tevoren geïnstalleerd met uw KDE-installatie, terwijl u andere zelf kunt opslaan.

Deze reguliere expressies hebben twee doelen ([zie de gedetailleerde beschrijving](#), namelijk (1) u een verzameling bouwstenen bieden en (2) reguliere expressies er mooier uit te laten zien.

U kunt uw eigen reguliere expressies opslaan door in het bewerkingsgebied op de rechtermuisknop te drukken, en te kiezen voor `Reguliere expressie opslaan`.

Als de reguliere expressie die u opslaat zich binnen een [container voor samenstellingen](#) bevindt, dan wordt de reguliere expressie gebruikt worden om volgende reguliere expressies mooier te maken.

Gebruikergedefinieerde reguliere expressies kunnen verwijderd worden of hernoemd worden door boven de desbetreffende reguliere expressie in de lijst de rechtermuisknop in te drukken.

Hoofdstuk 4

Fouten melden en suggesties voor functies

Fouten en gewenste functies kunnen gemeld worden via het [KDE Bug Tracking System](#). Controleer **voordat** u een fout meldt of een suggestie doet, of deze nog niet is [gemeld/gesuggereerd](#).

Hoofdstuk 5

Veel voorkomende vragen

5.1 Ondersteunt de editor voor reguliere expressies achterwaartse referenties?

Nee, momenteel wordt dit nog niet ondersteund. Dit is gepland voor de volgende versie.

5.2 Ondersteunt de editor voor reguliere expressies het tonen van gevonden overeenkomsten?

Nee, hopelijk zal dit in de volgende versie beschikbaar zijn.

5.3 Ik ben de auteur van een KDE-programma; hoe kan ik dit grafische element in mijn applicatie gebruiken?

Zie [De documentatie voor de klasse KRegExpEditorInterface](#).

5.4 Ik kan de knop *Editor voor reguliere expressies* niet vinden in bijvoorbeeld konqueror in een andere KDE3 installatie, waarom?

De grafische component voor reguliere expressies bevindt zich in het pakket KDE-utils. Indien u dit pakket niet geïnstalleerd heeft, dan verschijnen de knoppen *reguliere expressie bewerken* niet in het programma.

Hoofdstuk 6

Dankbetuigingen en licentie

Documentatie is copyright 2001, Jesper K. Pedersen blackie@kde.org

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.

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

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