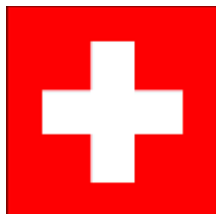


GSAK 301!

GSAK automatiseren: Beginnen met macro's



Dit is een basishandleiding voor hen die GSAK willen automatiseren door middel van macro's. Alle onderstreepte tekst in deze handleiding is een link naar een web site - klik op die tekst om direct naar die web site te gaan.

Let op:

Twee dingen:

1. Deze handleiding is in het Nederlands, maar het programma zelf is Engels-talig! Er zal vaak gebruik gemaakt worden van engels-talige uitdrukkingen. Meestal is dat omdat er dan een bepaalde handeling of beeld wordt bedoeld en die hoeft niet omschreven te worden omdat u het zo in het programma aantreft.

2. Deze handleiding gaat over het automatiseren van GSAK. Het is de bedoeling dat u hem van begin tot einde doorwerkt. Dat hoeft niet in één keer, maar het een bouwt voort op het voorgaande.

3. *Als u direct naar een alinea gaat zonder het voorgaande doorgewerkt te hebben kan het zijn dat u wezenlijke info mist en daardoor in de war raakt.* Bovendien, er wordt van uit gegaan dat u de eigenschappen en functies van GSAK reeds kent. Als dat niet zo is, lees dan **GSAK 101 Dutch - Nederlands** en **GSAK 201 Dutch - Nederlands**.

'GSAK 101 Dutch - Nederlands' gaat over de installatie en het algemene gebruik, 'GSAK 201 Dutch - Nederlands' over het aanpassen van de diverse instellingen van GSAK om het aan te passen aan uw eigen wensen.

Veel mogelijkheden binnen GSAK blijven hier onbesproken. Kijk, om daarover meer aan de weet te komen, in het GSAK Help-bestand (in GSAK, ga naar het Help-menu, klik dan op 'Contents', of toets de F1-functietoets van het toetsenbord). Voor hulp in relatie tot de macro's, ga in GSAK naar het 'Macro'-menu en klik op 'Help'.

Voor gebruikers met ervaring in computer-programmering zullen deze handleiding te simplistisch vinden. Hebt u ervaring met scripting, duik er in, gebruikmakend van de GSAK Help-files als uw gids.



Wat zijn macro's ?

Nee, we hebben het hier niet over macrons - kokoskoeken. 'Macro' is een technische uitdrukking voor een serie instructies die **een functie of eigenschap** automatisch uitvoert/gebruikt. GSAK heeft automatisering ofwel 'macro'-support waardoor gebruikers in staat zijn een serie GSAK-instructies in een tekstbestand samen te brengen. Dat kan met elke willekeurige tekstbewerker, ook met de 'Macro Editor' in GSAK. (In GSAK ga naar 'Macro > Edit/Create' om de editor te openen.)

Waarom macro's gebruiken?

Macro's worden gebruikt om veel gebruikte handelingen te automatiseren, waardoor een aantal vaste gebruikershandelingen worden teruggebracht tot één handeling. Maar ook om krachtige opdrachten uit te voeren die niet via de normale gebruikersmenu's ter beschikking staan.

Laten we er eens van uit gaan dat u vaak een Pocket Query ophaalt, het GPX-file in GSAK laadt, de caches sorteert aan de hand van de 'Last GPX'-datum, controleert of caches - die niet werden ververs - zijn gearchiveerd, vervolgens filtert aan de hand van de afstand tot uw thuislocatie en de 500 meest nabije caches exporteert naar een kaartprogramma en uw GPS-ontvanger.

In plaats van steeds weer al deze handelingen te doen, elke keer als u een 'Pocket Query' ontvangt, kunt u hiervoor een macro gebruiken die al deze handelingen met één muisklik voor u regelt. Een voorbeeld dus van een aantal vaste gebruikershandelingen die worden teruggebracht tot één handeling.

Of, een ander voorbeeld, u wilt aan de hand van meerdere kolommen van het tabelvenster sorteren. Bijvoorbeeld 'cache-owner' - 'date placed' - 'last found', of gebruik maken van een stukje informatie dat in GSAK beschikbaar is maar waarvoor geen specifieke aanduiding beschikbaar is.

Deze dingen kunnen *uitsluitend* worden gerealiseerd door middel van een macro, een voorbeeld van krachtige functies die niet door middel van de normale menu's ter beschikking staan.

Zijn macro's moeilijk in het gebruik?



Macro's zijn makkelijk in het gebruik. Clyde (de maker van GSAK) en GSAK-gebruikers hebben er al veel gemaakt en vrijwillig ter beschikking gesteld van alle GSAK-gebruikers in de Macro Library.

De 'Macro Library' is onderverdeeld in algemene categorieën die gebaseerd zijn op het eindresultaat zoals de gebruiker die ziet.

Als u in een bepaalde categorie niet kunt vinden wat u zoekt of u hebt geen zin om alle categorieën te doorzoeken, dan kunt u gebruik maken van de zoekfunctie binnen het forum om de hele 'Macro Library' te doorzoeken.

Let op : Met ingang van versie 6.6 zijn er belangrijke wijzigingen ingevoerd ten aanzien van het macro-gebruik en later ook in GSAK versie 7. Als u nog niet de meest recente versie van GSAK gebruikt, haalt u dan de update op van de GSAK Homepage en installeer deze voordat u probeert om deze macro's te gebruiken!

Een macro uit de 'Macro Library' gebruiken:

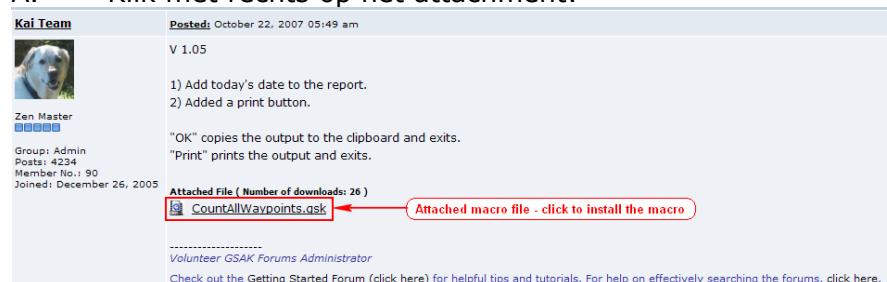
1) **Lees de beschrijving:** Klik op een onderwerp dat u interesseert en kijk wat de macro doet. Als het u wat lijkt :

2) **Kijk of er voor de macro aanpassingen vereist zijn:**

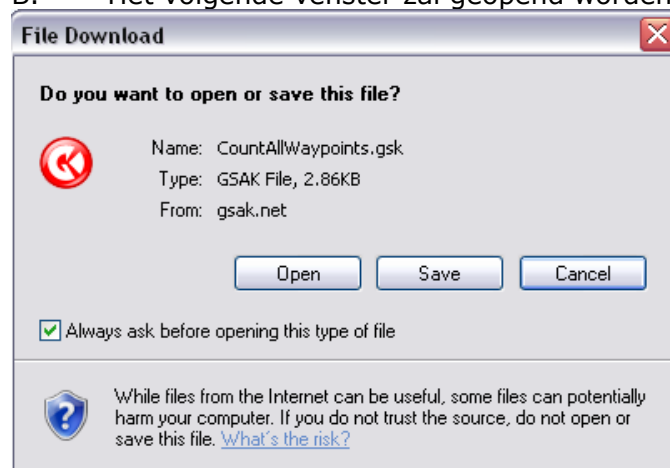
De meeste macro's vereisen geen aanpassingen, maar sommige wél (bijv. het opzetten van een filter dat vervolgens eerst gesaved moet worden voordat de macro de eerste keer gebruikt wordt). De beschrijving in het forumdeel "Macro Library" zal uitwijzen of u eventueel nog wijzigingen aan moet brengen in de macro voor specifieke eisen die u hebt. Als u de macro wilt gebruiken:

3) **Installeer de macro op (GSAK Versie 7) :** De meest recente versie van de macro staat gewoonlijk als attachment in een van de laatste of de laatste posting binnen het onderwerp. Hierin kunnen fouten uit voorgaande versies zijn hersteld of nog weer nieuwe mogelijkheden zijn opgenomen.

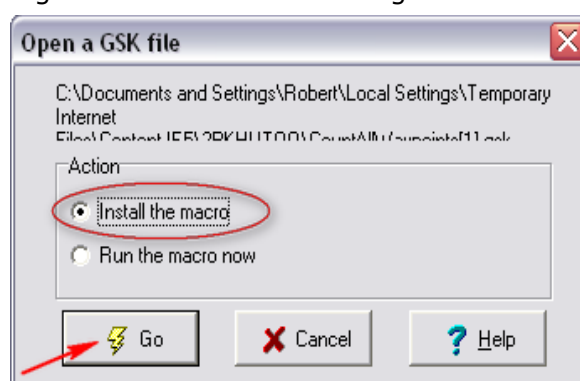
A. Klik met rechts op het attachment:



B. Het volgende venster zal geopend worden – klik op “Open”:



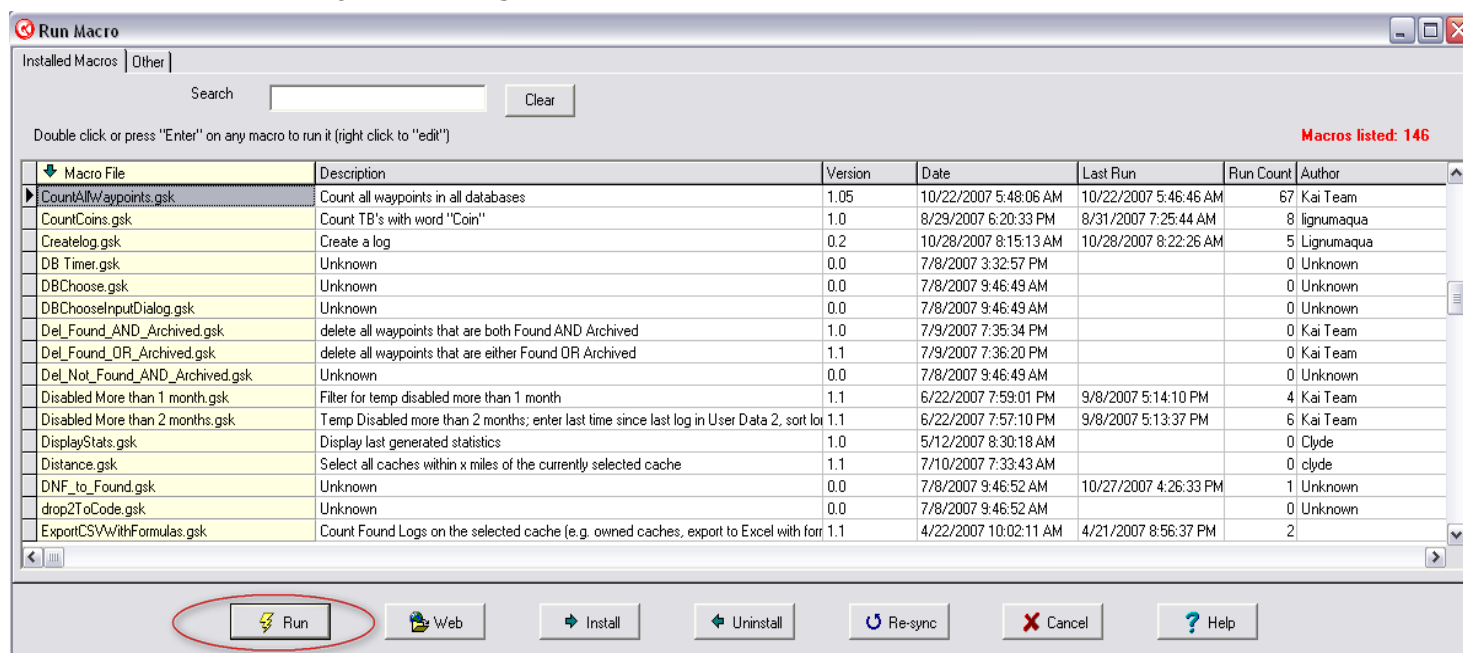
C. GSAK zal – indien het nog niet is opgestart – worden geopend en het volgende venster zal worden getoond:



D. Selecteer : “Install the macro” en klik op de “Go”- button.
De macro zal worden geïnstalleerd.

4) **Aanpassen (indien noodzakelijk - zie stap 2):** Als het nodig is om instellingen van de macro aan te passen (bijvoorbeeld opgeslagen filterinstellingen, bestandsnamen of directory-paden), open dan GSAK en ga naar het 'Macro>Edit\Create' -menu. Klik op 'File>Open' in de macro-editor en kies de macro die u zojuist hebt geïnstalleerd. Breng de nodige wijzigingen aan en sla het bestand op.

5) **Voer de macro uit:** Om de macro te gebruiken gaat u naar 'Macro>Run' in het GSAK-menu (of op het toetsenbord: 'Ctrl+M'). Het "Run Macro" dialoogvenster zal geopend worden en een lijst van alle geïnstalleerde macro's tonen:



Klik op de macro die u wilt gebruiken onder "Macro File" en klik op de RUN-button onderaan het dialoogvenster (of dubbelklik op de filenaam). De macro zal uitgevoerd worden en de acties uitvoeren waarvoor hij ontworpen was.

Voor informatie over het "Run/Manage"- dialoog vind u in het GSAK helpfile (Help>Contents>Using GSAK>Macros (automating GSAK)>Managing and Running).

GSAK-gebruikers, die macro's posten, onderhouden deze ook. Denkt u er alstublieft aan dat dit gewone medegebruikers zijn die vrijwillig hun werk aan anderen ter beschikking stellen. Als u vragen heeft post die dan in het [Macros](#) support forum, met een link of een aanduiding in verbinding met de geposte macro.

Noot: Als dit een macro is die u vaak zult gebruiken, dan kunt u hem toevoegen aan de 'Tool Bar'.

Dan is het niet meer dan een muisklik om hem te gebruiken!

Is het maken van macro's moeilijk?

Of u het 'bouwen' van een macro moeilijk vindt of niet zal veel afhangen van de taken - en de complexiteit ervan - die u wilt automatiseren. Ook is het afhankelijk van uw aandacht voor details, uw geduld en uw aanleg voor logisch denken.

Veel schrijvers van macro's beginnen met het automatiseren van een relatief eenvoudige taak (Hieronder geven we voorbeelden), of ze bestuderen, kopieëren of passen macro's van iemand anders aan. Alle macro's in de GSAK [Macro Library](#) mogen vrijelijk worden gekopieerd en gewijzigd. Een goede manier om te weten te komen hoe macro's werken is door te kijken hoe andere gebruikers dingen in *hun* macro's hebben opgelost.

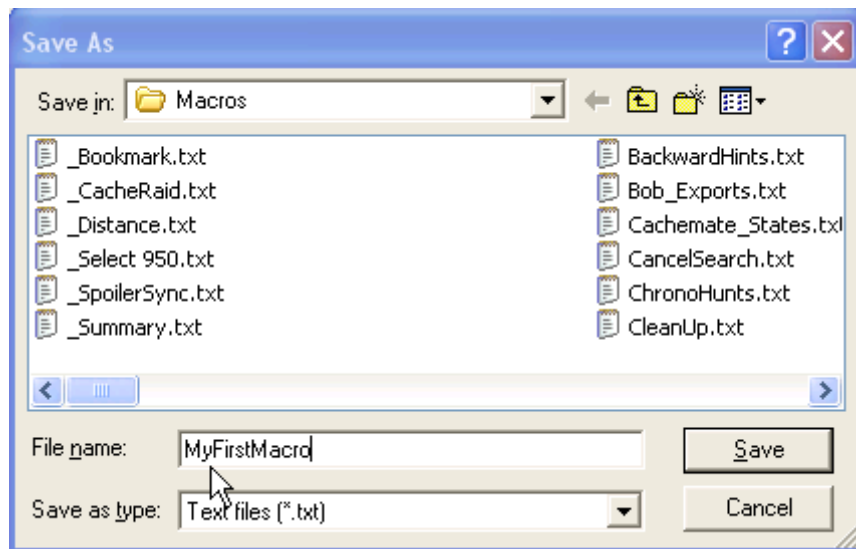
Het schrijven van Uw eerste macro.

Eenvoudige macro's.

In zijn simpelste vorm kan een macro bestaan uit één enkele regel. Om er even aan te proeven zullen we een eenvoudige macro schrijven en uitvoeren. Deze macro zal GSAK de opdracht geven om te pauzeren met wat het op dat moment aan het doen is en in plaats daarvan een dialoogvenster te tonen met een bericht dat wij hebben opgesteld. En het bericht zal de volgende 'diepzinnige' inhoud hebben: 'Mijn eerste macro is een succes!'. Als eerste, ga in GSAK naar 'Macro>Edit/Create'. Type nu (of kopieer en plak) de rose tekst precies zoals die hier wordt getoond (in de macro-editor zal die niet meer rose zijn, dat geeft niet):

PAUSE Msg="My first macro is a succes!"

Ga vervolgens naar het 'File menu' in de macro-editor en kies 'Save and Run'. Er zal een dialoogvenster verschijnen dat u vraagt om uw macro een naam te geven. Type 'MyFirstMacro' in het invulvak (in plaats van 'New Macro') en klik op de 'Save'-knop:



GSAK zal de macro opslaan en uitvoeren. Er zal een dialoogvenster worden getoond met uw bericht:



Klik op de "Continue" knop om de macro af te sluiten. U bent nu erkend GSAK macro-schrijver!

Het plannen van een complexere macro – het logische proces.

De eerste stap bij het schrijven van een complexere macro is het duidelijk vastleggen van wat u er van verwacht. Niet alleen het gewenste resultaat, maar ook de logische volgorde van stappen die gemaakt moeten worden om dit te bereiken. Ofwel wat moet er gebeuren voordat een volgende stap genomen kan worden?

Als u een geocache wilt vinden (het gewenste resultaat) moet u eerst in uw auto stappen en gaan rijden, waarbij u op het juiste moment moet afslaan (logische volgorde van stappen) om op de juiste plaats uit te komen.

Als u op jacht gaat naar een cache begint u niet als eerste met na te denken over waar u gaat zoeken als u hem hebt bereikt. Als eerste denkt u na over hoe u vanaf het punt waar u zich bevindt naar daar waar de cache zou moeten liggen komt. Dan denkt u pas na over waar u precies zult zoeken.

Het schrijven van een macro is eigenlijk een soortgelijk, stap voor stap proces. Als u een afslag mist als u op weg bent naar de cachelokatie, zult u er niet aankomen. Laat u na een bepaalde handeling in te bouwen bij het schrijven (en daardoor bij het uitvoeren) van een macro, dan zal die niet het gewenste resultaat opleveren!

Om op gang te komen, gaan een relatief simpele maar nuttige macro plannen en schrijven. Het resultaat dat deze op moet leveren is het selecteren van 200 caches die het dichtst bij een bepaald punt ('Location') liggen. Vervolgens moeten we die caches kunnen bekijken, naar een GPS-ontvanger versturen of kunnen exporteren naar een bestand dan wel een ander programma (b.v. een navigatie programma of Cachemate).

De logische stappen die nodig zijn om dit te bereiken zijn:

1. Selecteer de database die de gewenste caches bevat;
2. Sorteer de caches aan de hand van hun afstand tot 'Location';
3. Selecteer de eerste 200 caches (die, die het dichtst bij 'Location' liggen);
4. Zodanig filteren dat alleen de caches die aan de criteria voldoen nog zichtbaar zijn.

Deze macro komt overeen met het uitvoeren van de volgende menu-keuzes in GSAK:

1. Selecteer een database (Database>Select)
2. Sorteer de database aan de hand van afstand tot de huidige 'Location' (klik op de "Miles" or "Kms" kolom-kop)
3. Verwijder alle 'user flags' (User Flags>Clear All User Flags)
4. Vink de 'user flags' van de volgende 'nn' rijen (User Flags>Set for Next nn)
5. Activeer een filtering met als criterium: user flags = set'(T)

Let op. Hét grote verschil is dat de macro een 'macro flag' activeert en gebruikt in plaats van de 'user flag'. Als gevolg hiervan worden eerder ingestelde 'user flags' niet beïnvloed door het gebruik van de macro.

De Macro Editor van GSAK.

U kunt iedere willekeurige tekstverwerker (editor) gebruiken om een macro te schrijven (b.v. Windows Kladblok), maar GSAK heeft een eigen macro editor die gemakkelijk is in het gebruik omdat die in GSAK is geïntegreerd. U kunt uw macro's uittesten vanuit deze editor (File>Save and Run of CTRL-R) en, als de macro een fout meldt, u kunt via een druk op de knop terugkeren naar de editor en de regel waar de fout ontstond.

Om een nieuwe macro te schrijven met behulp van de GSAK-editor gaat u in het GSAK menu naar 'Macro>Edit/Create'.

Het schrijven van een macro – gebruik maken van de 'macro help files'.

Als u een macro schrijft is het GSAK macro help file net zoals als een woordenboek voor een vreemde taal – het geeft de aanwijzingen die nodig zijn omdat wat u wilt om te zetten in uitdrukkingen die GSAK begrijpt.

Om het macro help file te kunnen raadplegen gaat u in GSAK naar 'Macro>Help'.

Net zoals in een gewone taal zijn er in de macro taal verschillende categorieën uitdrukkingen.

Die worden als links aangeduid aan het begin van 'Macro Help' en klikken op een van de links brengt u naar de lijst met beschikbare uitdrukkingen, hun definities en hun syntaxis (de opties gerelateerd aan die uitdrukking).

Doe (nog) geen poging om al deze informatie uit het hoofd te leren (er is zo veel dat er in GSAK gedaan kan worden dat het je er duizeling van kan worden). We zullen hier een overzicht geven van de meest gebruikte uitdrukkingen, maar u kunt altijd weer in de Help-file nakijken wat een bepaalde uitdrukking is, mocht u het vergeten zijn. We zullen ook uitleggen hoe u in de praktijk deze uitdrukkingen gebruikt in ons macro-voorbeeld hieronder.

Nou hier zijn dan de categorieën uitdrukkingen die u vindt in het Help-file (uw vertaalwoordenboek voor de macrotaal):

Commands & Functions – zij vertellen GSAK iets te ondernemen! Dit zijn de 'werkwoorden' van de macro-taal (computer programmeurs zullen zeggen dat er een verschil is tussen 'commands' en 'functions'. Weliswaar hebben ze gelijk, maar daar zullen we hier niet op in gaan. Als u alle 'commands'(opdrachten/commando's) wilt zien die in GSAK beschikbaar zijn, [klik dan hier](#) en klik dan op 'command summary'. Voor een lijst met de in GSAK beschikbare 'functions', kunt u [hier klikken](#) en vervolgens op 'function summary'.

Variables: de 'dingen' waarom de handelingen draaien. Er zijn meerdere typen variabelen:

User Created Variables: zijn uitdrukkingen die u zelf definieert om informatie in op te slaan, dan wel voor gebruik in de macro of voor gebruik in het resultaat. Alle 'user created variables' beginnen met een dollar-teken (\$) en kunnen vrijwel elke naam hebben die u wilt (b.v. \$MijnVariabele is een geldige 'user created variable'), als die maar begint met \$. Uitzonderingen hierop zijn namen die gereserveerd zijn voor een databasevariabele of systeemvariabele (denk aan woorden als PRINT, COM, PORT).

Database Variables: zijn gereserveerde uitdrukkingen ten behoeve van de informatie (data) die in GSAK is opgeslagen. Alle 'Database variables' beginnen met "\$d_". Als voorbeeld: de GC (Geocaching) of andere unieke code voor een cache of waypoint is opgeslagen in de database variabele '\$d_Code'. Een lijst met database-variabelen die in GSAK beschikbaar zijn vindt u door [hier te klikken](#). U mag '\$d_' **nooit** gebruiken om een 'user created variable' te benoemen omdat dit voorzetsel is gereserveerd voor database variabelen.

System Variables: zijn gereserveerde uitdrukkingen voor informatie die aanwezig is in uw computersysteem als GSAK actief is, maar wordt niet gebruikt in de database zelf (ofwel: het is geen waypoint of cache informatie). Alle 'system variables' beginnen met '\$_'. Zo is de database, die geselecteerd is, opgeslagen in '\$_CurrentDatabase', en de locatie waar u GSAK installeerde op uw computer is vastgelegd in de systeemvariabele '\$_Install'. Voor een lijst met alle beschikbare systeem variabelen [klikt u hier](#) en vervolgens op 'system variables'. En weer: u mag het voorzetsel

'\$_' **nooit** gebruiken voor een 'user created variable' omdat het gereserveerd is voor systeem variabelen.

Expressions: Expressies bieden mogelijkheden om rekenkundige of logische bewerkingen uit te voeren, maar ook om variabelen of data te toetsen. Met expressies kunt u optellen, aftrekken, vermenigvuldigen, vergelijken met (gelijk aan, niet gelijk aan, groter dan, kleiner dan) of meerdere uitdrukkingen logisch met elkaar vergelijken (AND, OR, NOT) . [Klik hier](#) om de lijst met beschikbare expressies te openen en scroll naar beneden naar 'Variable Operands', 'Relational Operators' en 'Logical Operators' (geen angst, u hoeft geen rekenwonder te zijn, trek u niets aan van de exotische namen!).

Genoeg nu! We gaan een macro schrijven:

OK, we beginnen met het voorbeeld dat we hierboven noemden. De bedoeling was dat we de 200 meest nabije caches in relatie tot onze locatie willen tonen en de vereiste stappen zijn:

1. Selecteer de database die de gewenste caches bevat;
2. Sorteer de caches aan de hand van hun afstand tot 'Location';
3. Selecteer de eerste 200 caches (die, die het dichtst bij 'Location' liggen.
4. Zodanig filteren dat alleen de caches die aan de criteria voldoen nog zichtbaar zijn.

Mocht u het nog niet gedaan hebben, open dan de GSAK macro editor (in GSAK, Macro> Edit/Create) en we gaan beginnen.

Noot: De 'macro code' (instructies) zoals die in de macro wordt genoteerd wordt in deze handleiding getoond in **rose** zodat die tekst makkelijk te onderscheiden is van de bijbehorende uitleg. U kunt elke zin/regel gewoon uittypen of door middel van kopiëren en plakken de volledige macro (die aan het einde van dit deel is afgedrukt) overbrengen naar de Macro Editor.

Verder is de voorbeeldmacro inhoudelijk niet vertaald.

Eerste stap: Het is een goede gewoonte om commentaar in uw macro's op te nemen. Een commentaar is elke zin/regel die begint met het karakter # , en iedere regel/zin die begint met # wordt genegeerd door GSAK. En dus, de eerste regel in onze macro is een commentaar dat ons aanwijzingen geeft wat de macro doet:

Macro to select a number of caches that are closest to a location

ofwel:

Macro om een aantal caches te selecteren dat het dichtst bij een locatie ligt.

U kunt commentaren op elke plek in een macro gebruiken en het advies is om dit ook echt te doen! Goed gedocumenteerde macro's (die dus voorzien zijn van goede commentaren) helpen u om uw macro te 'debuggen' (het herstellen van fouten) en helpen u en anderen te begrijpen wat een bepaald deel van de macro doet. (Het is verbazingwekkend hoe snel u kunt vergeten wat u hebt gedaan of waarom u juist dát hebt gedaan als u later weer eens een macro bekijkt die u zelf schreef!)

Stap Twee: We weten nu wat de bedoeling is dat deze macro gaat doen en we moeten nu de database gaan kiezen die de caches bevat waarin we geïnteresseerd zijn. Hier in deze 'les' zullen we de 'Default' database selecteren omdat GSAK altijd een 'Default' database heeft.

De opdracht om een database te selecteren is:

DATABASE Name="Default" Action=Select

Het eerste deel van de regel (DATABASE) is de opdracht (command) , en laat GSAK weten dat we iets willen doen met een database. Het tweede deel (Name=) vertelt GSAK met welke database u wilt gaan werken. Het derde deel (Action=) vertelt GSAK wat u wilt doen

met die database. In dit geval vertellen we GSAK om de database 'Default' te selecteren of er naar over te schakelen.

Als u naar het macro Help-file gaat (Macro>Help, of online [klik hier](#)), klik op de "Command Summary" link, klik dan op 'DATABASE' en u ziet de volgende syntaxis voor de opdracht 'DATABASE':

DATABASE <Name="Database"> [<Action=select|create|delete>]

In de Macro Help-files, wordt de opdracht als eerste genoemd, gewoonlijk in hoofdletters. "Tokens" vullen de opdracht aan, staan tussen < > (deze <>tekens worden niet getyped in de opdracht zelf) en zijn - indien ze vermeld worden - verplicht. Gegevens tussen vierkante haken ([...]) zijn optioneel en dus niet verplicht. Toegestane opties worden gescheiden door een | (een verticale streep). Geen van de opdrachten, tokens en opties(parameters) is (hoofd-)letter gevoelig, d.w.z dat zowel hoofdletters als kleine letters zijn toegestaan.

In dit geval is de opdracht (het 'command') 'DATABASE'. NAME is een verplicht 'token' en "database" (de naam van de te gebruiken database) moet worden vermeld **tussen dubbele aanhalingstekens** (daarom staat het tussen dubbele aanhalingstekens in het Help-file). 'Action' (tussen de vierkante haken) is optioneel - u kunt het naar wens gebruiken of weglaten. Is 'Action' niet vermeld, dan zal GSAK er van uit gaan dat 'select' wordt bedoeld. De actie waar GSAK van uit gaat (default) voor een optioneel token is altijd onderstreept in het Help-file. Omdat dit voor de opdracht DATABASE 'select' is, zouden we de token 'Action=' weg hebben kunnen laten en toch hetzelfde resultaat hebben bereikt.

Stap Drie: We hebben GSAK in de vorige stap verteld met welke database de macro zal moet werken. Nu moeten we de caches in die database gaan sorteren aan de hand van de afstand tot de lokatie die we gebruiken. Van de kleinste naar de grootste afstand. Om dat te doen schrijven we:

SORT By="distance" Sequence=A

SORT is de opdracht, 'By=' is de GSAK kolom waarop we willen sorteren (zie de [SORT](#) opdracht in het Help-file voor een lijst met kolom namen) en 'Sequence=' bepaalt voor GSAK of we willen sorteren van laagste naar hoogste (dit is Ascending = A) of van hoogste naar laagste (dit is Descending = D). 'Sequence=' is een optionele voorwaarde, en de default-instelling is ascending (A). We hadden deze token dus weg kunnen laten en zouden toch hetzelfde resultaat hebben gekregen.

Stap Vier: OK, we hebben de goede database en we hebben die gesorteerd op afstand. Nu gaan we de gebruiker vragen hoeveel caches hij wilt selecteren. De volgende regel in onze macro is:

INPUT Msg="How many waypoints do you want to select" Default="200"

De opdracht 'INPUT' staat de gebruiker toe om informatie in te voeren (= "input"). Dit is een erg handige opdracht als u de mogelijkheid wilt bieden om een 'willekeurig' antwoord te geven op een vraag. 'Msg=' is het 'message token' - dat wat u schrijft na 'Msg=' zal op het scherm worden getoond als de macro wordt uitgevoerd. In dit geval wordt aan de gebruiker verteld wat voor soort informatie er van hem verwacht wordt. (Het lijkt een beetje op de opdracht 'PAUSE' die we in de eerste macro gebruikten, met dit verschil dat INPUT de gebruiker de kans geeft iets in te voeren). 'Default=' is optioneel. 'Default=' vertelt GSAK welke waarde er moet worden gebruikt als de gebruiker niets invoert maar alleen maar op de 'Enter'-toets drukt. We zeggen hier tegen GSAK de waarde 200 aan te houden als de gebruiker geen waarde invult. Een andere optie voor de opdracht 'INPUT' is "BROWSE". In dat geval kan de gebruiker door middel van aanwijzen en klikken een file of folder (map, directory) selecteren, maar hier hebben we alleen behoefte aan een getal dus gebruiken we die optie hier niet.

Stap Vijf: De INPUT-opdracht vangt de informatie die wordt ingetypt op en zet deze in een standaard variabele met de naam '\$Result' (U kunt de INPUT-opdracht dwingen het antwoord in een andere variabele op te slaan met behulp van de 'VarName' optie, maar dat is hier niet nodig).

Omdat de informatie die de INPUT-opdracht ontvangt via het toetsenbord wordt ingetypt neemt de opdracht aan dat het resultaat een waarde is die 'string' variabele wordt genoemd. Een string variabele is elke combinatie van letters, cijfers en symbolen en wordt door GSAK gezien als *tekst*. Een computer kan niet tellen aan de hand van tekst of een andere berekening er op baseren. Onze volgende stap in onze macro moet GSAK dus instrueren om die *tekst* '200' om te zetten in het *getal* '200' en dit getal toe te wijzen aan een 'user created variable' die we '\$Number' noemen (die zou ook '\$Komkommer kunnen heten als u dat leuk vindt, maar het is handig om variabelen een naam te geven die duidt op hun functie). De volgende regel in onze macro is:

\$Number = Val(\$Result)

Deze regel bestaat in feite uit twee delen. Het eerste deel, \$Number =, geeft GSAK de opdracht om een variabele in te stellen met de naam '\$Number' en om alles wat na het = - teken volgt in die variabele op te slaan. Het tweede deel, Val(\$Result), gebruikt de 'Value' functie om een string (tekst) om te zetten in een getal (waarde = value). We zeggen hier: zet de variabele \$Result (die afkomstig is van de INPUT-opdracht) om in een getal.

Stap Zes: Aangezien we de caches die het dichtst bij de locatie liggen willen tellen moeten we GSAK nu instrueren om de 'pointer' (de interne aanwijzer die GSAK laat weten naar welke cache in de database het 'kijkt') naar de top van de gesorteerde lijst met caches te verplaatsen (de cache die het dichtst bij de locatie ligt). De volgende regel is dan ook:

GOTO Position=Top

De 'GOTO'-opdracht instrueert GSAK om de pointer binnen de database te verplaatsen. De opties zijn: "top" (de eerste cache in de huidige lijst), "bottom" (de laatste cache in de lijst), "next" (de eerstvolgende cache onder die waar de pointer nu is) en "nn" (een bepaald aantal caches onder de cache waar de pointer nu naar wijst – b.v. "10" verplaatst de pointer 10 plaatsen naar beneden ten opzichte van de huidige plaats).

Stap Zeven: Zoals boven al opgemerkt gaan we de 'Macro Flag' gebruiken. Die werkt net zoals een 'user flag' maar uitsluitend binnen een macro. Gebruik van de 'Macro Flag' laat een eerder ingestelde 'user flag' onaangetast.

Voordat we de 'macro flags' gebruiken, moeten we ons er van verzekeren dat er niet reeds 'macro flags' zijn ingesteld.

Zodoende is onze volgende regel:

MacroFlag Type=Clear Range=All

De opdracht is 'MacroFlag', Type= is de token die GSAK instrueert wat we met de 'macro flag' willen doen (hier: 'clear' = wissen), en 'Range=' vertelt GSAK welke 'macro flags' we willen wissen (hier: 'All' = allemaal).

Stap Acht: Nu weten we zeker dat alle 'macro flags' zijn verwijderd en kunnen we het aantal gaan activeren (=Set) dat is gevraagd door de gebruiker door het intypen van het gewenste aantal caches (via de INPUT-opdracht).

Onze volgende regel is:

MacroFlag Type=Set Range=\$number

Dit is de zelfde opdracht als in stap zeven, maar nu geven we GSAK de opdracht de 'macro flags' te activeren gelijk aan het getal dat is opgeslagen in de variabele '\$number' (denk er aan dat '\$number' de getalswaarde is van het antwoord dat de gebruiker gaf naar aanleiding van de vraag die werd gesteld door de opdracht 'INPUT!').

Stap Negen: Als laatste gaan we een filtering instellen die de caches uitfiltert die gemarkeerd zijn door middel van een 'macro flag':

MFILTER If=\$d_MacroFlag

MFILTER is de opdracht die binnen een macro GSAK instrueert om een filtering toe te passen. Dit komt overeen met de menu-keuze 'Search>Filter' in de GSAK menu's, maar met de extra mogelijkheid dat u kunt filteren op basis van *elke* database variabele die u wenst (zie 'Macro>Help' voor een lijst van alle *database variabelen* die u kunt gebruiken). In dit geval vragen we GSAK om te filteren op grond van de database variabele '\$d_MacroFlag', de variabele waarin GSAK de informatie opslaat of een cache een 'flag' heeft of niet.

Dat is alles! De complete macro ziet er zo uit (u kunt de roze tekst hieronder kopieëren en plakken naar de macro editor):

Macro to select the first nn caches in a filter

**DATABASE Name="Default" Action=Select
SORT By="distance" Sequence=A
INPUT Msg="First nn waypoints to select" Default="200"
\$Number = Val(\$Result)
Goto Position=Top
MacroFlag Type=Clear Range=All
MacroFlag Type=Set Range=\$number
MFILTER If=\$d_MacroFlag**

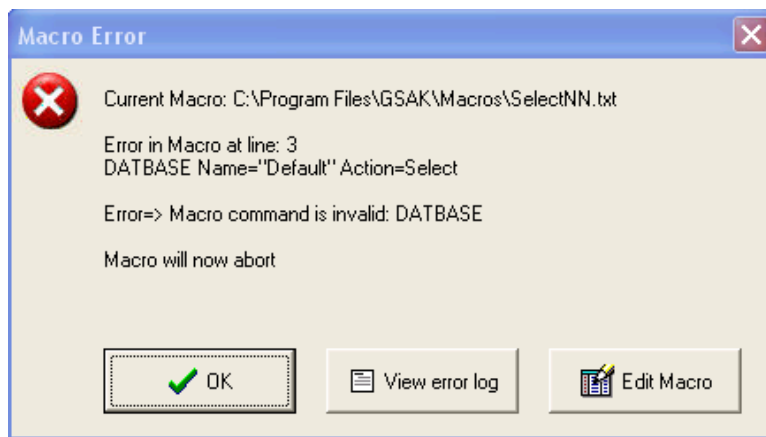
Wanneer het allemaal in de macro editor staat, ga dan naar 'File>Save and Run'. Geef de macro een naam (bijvoorbeeld: "SelectNN", zonder de aanhalingstekens!) en klik op 'Save'. GSAK zal nu automatisch de macro uitvoeren en als hij afgewerkt is zal de database gesorteerd zijn en ziet u de uitgefilterde lijst met 200 caches die het dichtst bij de gekozen locatie liggen, klaar om verzonden te worden naar de GPS-ontvanger!

Testen en het 'Debuggen' van een Macro: Het maakt niet uit hoe ervaren u bent in het schrijven van macro's, de kans is groot dat er 'bugs' (fouten) in de macro zitten bij de eerste opzet (gewoonlijk zijn dat type-fouten in een opdracht, een functie of variabele). GSAK controleert de macro code ('het programma') voordat het met de uitvoering begint en is slim genoeg om dit soort fouten te vinden en te signaleren. Dit wordt 'debuggen' genoemd en is een noodzakelijke stap tijdens het proces.

Laten we een voorbeeld nemen, stel we hebben een typefout gemaakt toen we begonnen met het schrijven van onze macro. Dat was toen we de eerste regel na het commentaar schreven. In plaats van DATABASE hebben we DATBASE getyped:

DATABASE Name="Default" Action=Select

Als we dan de macro testen zal GSAK deze fout herkennen en een foutboodschap tonen die er zo uitziet:



GSAK laat ons weten dat de uitgevoerde macro ('SelectNN.txt') een fout heeft op regel 3 en toont de regel met de fout ('DATABASE Name=...'). Vertelt ons wat de fout is ('DATABASE' is geen geldige opdracht - Error =>Macro command is invalid: DATABASE), - want de geldige opdracht is DATABASE) en stopt de uitvoering ('Macro will now abort!') zodat de ongeldige opdracht geen problemen kan veroorzaken.

Het mooie van het gebruiken van de ingebouwde editor is dat als u op 'Edit Macro' klikt, GSAK de macro editor opent en de cursor op de regel met de fout plaatst. Het aanbrengen van een correctie is dan makkelijk, waarna de macro kan worden opgeslagen en weer worden gestart voor een volgende test.

Dat mag allemaal erg simpel zijn bij een fout in regel 3 maar zou de fout in regel 73 zitten is het een toffe eigenschap van de editor!

De 'DEBUG' opdracht (Debug command): GSAK heeft ook een ingebouwde mogelijkheid om fouten in een macro op te sporen – de DEBUG opdracht. De Help file syntaxis ziet er zo uit:

`DEBUG <STATUS=on|off> [<Height=nnn>] [<Width=nnn>] [<Left=nnn>] [<Top=nnn>]`

De DEBUG-opdracht biedt de mogelijkheid om de macro stap voor stap af te werken. Vóór een programma-opdracht wordt uitgevoerd worden dan in een dialoogvenster de opdracht en alle variabelen getoond.

In dit venster hebt u de mogelijkheid om de opdracht te accepteren of over te slaan, dan wel om de macro te stoppen. Binnen de macro kunt U de 'DEBUG'-opdracht in- en uitschakelen met behulp van de STATUS token ('on' dan wel 'off').

Hoogte (Height), breedte (Width), links (Left) en bovenzijde (Top) zijn optioneel en bieden de mogelijkheid om de afmetingen en positie van het dialoogvenster aan te passen. Alle afmetingen (nnn) zijn in beeldpixels.

Het is een goed idee om, zeker als u net begint met macro's, de 'DEBUG Status=on' - opdracht aan het begin van elk nieuw blok van het programma, dat u schrijft, te plaatsen. Als het allemaal werkt als gehoopt, verwijder dan de 'DEBUG'-opdracht (of wijzig de status in 'off') zodat de macro niet bij elke programma opdracht stopt.

Een test database gebruiken: De macro die we hierboven gebruikt hebben is valt onder de categorie 'niet destructief' omdat hij geen permanente veranderingen in de gebruikte database aanbrengt.

Er is geen risico aan een dergelijke macro verbonden want zelfs als het fout zou gaan zal er geen data verloren gaan. In dit geval zou het uitschakelen van het Filter voldoende zijn om alles ongedaan te maken.

Maaarr, de macro-taal is erg krachtig en kan worden gebruikt om dingen te doen die wél destructief van aard zijn zoals het vervangen van databasevariabelen (cache gegevens), verwijderen van caches of zelfs complete databases! Het is dan ook beter om een nieuwe database ten behoeve van GSAK te creëren met de naam 'Test' en een aantal waypoints vanuit een andere database hierheen te kopiëren (niet 'Move' =verplaatsen) en dan deze testdatabase te gebruiken bij het debuggen van een macro. Zou de macro per abuis cachedata verminken of caches verwijderen, dan is er nog niets aan de hand. Daarvoor is het een testdatabase. Kopieër gewoon weer een aantal caches en vervolg het testen na het aanbrengen van de nodige wijzigingen. Bij een échte database zou het wel even anders zijn als alle data verdwenen of verminkt zou zijn!

Meer bruikbare macro zaken en Tips

Verbindt een Macro met een knop (button): In het 'Customize Speedbar' venster (zie [GSAK 201 Dutch - Nederlands](#) voor meer gedetailleerde informatie over het aanpassen van de GSAK 'SpeedBar'), is de voorlaatste categorie: 'Macros'. (Let op: er is ook een categorie Macro, dus zonder 's') Als u op die categorie klikt, wordt een aantal knoppen getoond, met de namen 'All Macros' en 'Macro1', 'Macro2', etc. Als u een macro download of zelf schrijft die u vaak zult gebruiken denk er dan aan om die macro aan een knop te verbinden. Sleep hem vervolgens naar de 'Speedbar', en uw aangepaste instructies voor GSAK zijn slechts één klik verwijderd!

Als u uw macro's met knoppen verbindt en dan de 'All Macros'-knop op de 'SpeedBar' plaatst, wordt de 'All Macros' knop veranderd in een uitklapmenu dat u direct uit de totale lijst van beschikbare macro's laat kiezen! In dit menu wordt in plaats van de macronaam de macro-omschrijving getoond en is dus meer informatieve tekst zichtbaar dan op een knop zichtbaar is

Interactie met de gebruiker: We gebruikten twee soorten interactie met de gebruiker (de 'PAUSE' en 'INPUT'opdracht) in onze voorbeeld macro's, maar er zijn er nog een aantal:

SHOWSTOP & SHOWSTATUS: De 'SHOWSTOP'-opdracht toont een klein stop dialoogvenster als de macro wordt uitgevoerd. Normaal gesproken zou u dit als het de eerste opdracht in een macro met veel opdrachten plaatsen. Dit geeft u de mogelijkheid de macro af te breken door op de STOP-knop te klikken. Als u een macro schrijft en de 'DEBUG'-opdracht niet gebruikt is het aanbevelenswaardig om de 'SHOWSTOP'-opdracht aan het begin van de macro op te nemen omdat dit de mogelijkheid biedt om de macro af te breken mocht u onverhoopt een 'loop' gecreëerd hebben zodat de computer telkens in hetzelfde rondje blijft rond draaien en niet meer te onderbreken is. Als de macro eenmaal foutloos draait (ge-debugged is) kunt u de 'SHOWSTOP'-opdracht verwijderen.

'SHOWSTATUS' lijkt op 'SHOWSTOP', maar het toont de voortgang die GSAK maakt bij het verwerken van grote hoeveelheden data (b.v. tijdens het bewerken van een grote database). Zie het 'Macro Help'- file voor meer informatie over het gebruik van de 'SHOWSTATUS'-opdracht.

CHOOSE (kiezen): 'CHOOSE' lijkt op 'INPUT' maar geeft de mogelijkheid om de gebruiker een aantal keuzemogelijkheden aan te bieden. Dit is handig om typefouten te voorkomen maar ook om de keuzemogelijkheden te beperken. Zo kan de gebruiker bijv geen invoer doen die geen zin heeft of een keuze die niet bestaat.

SHOWFORM (toon formulier): 'SHOWFORM' biedt schrijvers van macros de mogelijkheid om 'formulieren', zeg maar invulvensters, te bouwen. Daar kunnen allerlei zaken worden ingevoegd. Zoals: Teksten (instructies, waarschuwingen, vragen, etc), diverse keuzemiddelen (knoppen, aanvinkhokjes, datumvelden, wijzigingsvelden, selectieknoppen (radiobuttons)) en afbeeldingen. De kleuren kunnen naar wens worden ingesteld evenals de lay-out. Voor meer informatie over 'ShowForm', ga naar 'Macros>Help>Commands and Functions, scroll naar beneden en klik op "ShowForm".

Het gebruik van 'conditional statements' 'loops' en 'subroutines' : Er zijn in GSAK meerdere opdrachten beschikbaar die de mogelijkheid bieden om iets te testen alvorens door te gaan of een actie of een set van acties te herhalen zonder dezelfde opdrachten telkens in te moeten typen:

IF/ELSE/EndIF (Indien/anders/eindig als): Deze opdracht biedt de mogelijkheid een stukje programma op te nemen waarbij aan een voorwaarde moet worden voldaan voordat een ander deel van het programma wordt uitgevoerd. 'IF' (indien) iets waar is doe dát. ELSE (anders) - dus als dat **niet** waar is - doe dát.

WHILE/ENDWHILE (zo lang als/beëindig) : -opdracht gaat door met het uitvoeren van een aantal acties net zo lang als een bepaalde voorwaarde 'waar' is - er aan die voorwaarde wordt voldaan. 'WHILE' wordt bijvoorbeeld gebruikt om door een groep caches 'te lopen', ze één voor één te bekijken om te zien of ze aan een voorwaarde voldoen (bijv of ze gevonden zijn), totdat het einde van de lijst bereikt is. De algemene schrijfwijze hiervoor is (de systeem variabele \$_EoL laat GSAK weten wanneer het einde van de lijst (End of List=EoL) is bereikt):

```
GOTO Position=Top
  WHILE not($_EoL)
    IF [condition]
      [do something]
    ELSE
      [do something else]
    EndIF
  GOTO Position=Next
EndWhile
```

GOSUB: De 'BeginSub', 'EndSub' en 'GOSUB' opdrachten geven de mogelijkheid om een stukje programma te schrijven (een subroutine) dat meer dan eens in dezelfde macro wordt gebruikt. U kunt hiervan ook gebruik maken om een complex stukje programma 'apart' te zetten binnen de macro en zo de werking te volgen en het debuggen te vereenvoudigen.

Kijk eventueel in 'Macro Help' voor meer info over het gebruik van 'IF', 'WHILE' en 'GOSUB'.

Dit is wel genoeg voor een 'Beginnen met..' handleiding (sommigen zullen misschien zelfs zeggen dat het te veel was!). Macro's kunnen in eerste instantie verwarrend werken, maar zoals zo veel dingen, als je er eenmaal mee begint wordt het duidelijker.

Een tip: start met het opzetten van een relatief eenvoudige taak of bestudeer, kopieër stukjes van andere macro's of pas de macro van een ander aan. Alle macro's in de GSAK [Macro Library](#) mogen vrijelijk worden gekopieëerd en gewijzigd worden. Bovendien is het een prachtige manier om te leren hoe macro's werken!