

Skript Studio I.

Projekt GRDWE 2012-18

MANUÁL SKRIPT STUDIO I.

TÉMA: TVORBA SKRIPTU
VYTVOŘENO: 08.06.2018 09:45
VERZE: 2.0.022.
EXPORT DO: .PDF
PUBLIKOVÁNO: WEB PROJEKT GROWE

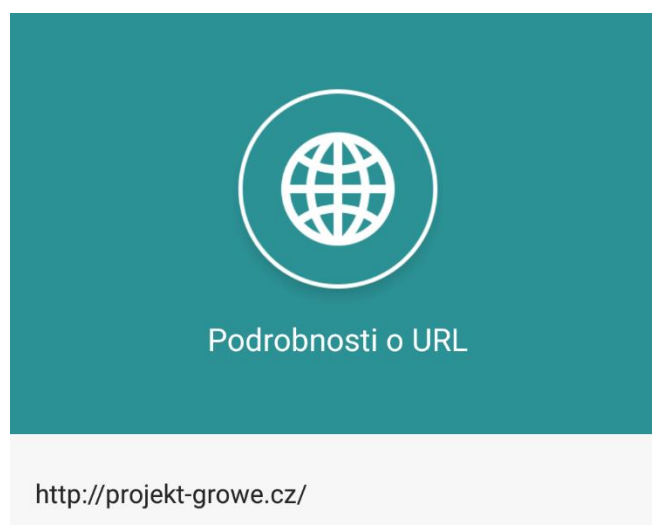
DATUM: 09.06.2018 15:35
DATUM:

Skript Studio I. Uživatelský manuál

Projekt GROWE © 2012-18

COPYRIGHT Projekt GROWE© 2012-18

Vydání druhé, **ver. II. - User Type**, sestavení **#024 BETA I.** Poslední změna: **6.8.2023**



Nejnovější verzi manuálu **Skript Studio I.** naleznete na adrese:

<http://projekt-growe.cz/wp-content/uploads/2017/05/Manual-Skript-Studio-I.pdf>

Nejnovější verzi manuálu **Projekt GROWE** naleznete na adrese:

http://projekt-growe.cz/wp-content/uploads/2017/01/Manual_GROWE-UserVersion.pdf

Screenshots (obrázky obsahu obrazovek) v manuálu se mohou lišit od fyzické podoby programu **Skript Studio I.**, v závislosti na verzi vývoje při tvorbě manuálu. Tyto změny jsou zpravidla minoritní, pokud je změna rozsáhlejšího rázu, je manuál po implementaci změny a BETA testu programu aktualizován.

Veškeré dotazy týkající se distribuce tohoto manuálu směřujte na E-mailovou adresu: support@projekt-growe.cz

Webová adresa projektu: www.projekt-growe.cz

Žádná část tohoto manuálu nesmí být publikována a šířena žádným způsobem a v žádné podobě bez výslovného souhlasu vydavatele – **Projekt GROWE**.

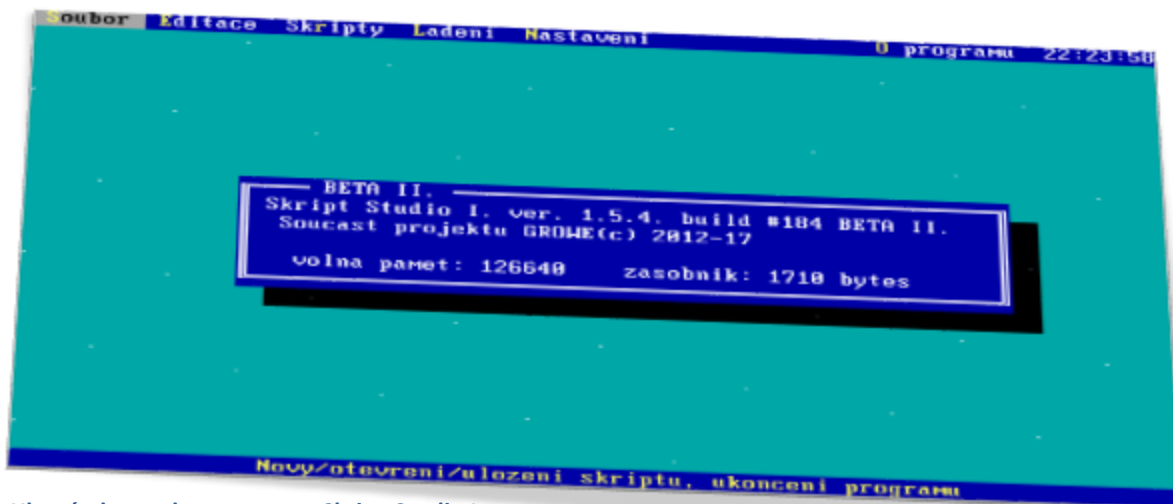
OBSAH

1.1. Popis programu	5
1.1.1. Co je to řídicí skript	6
1.2. Ovládání programu	8
1.3. Nový skript.....	8
1.4. Tvorba / Otevření již existujícího skriptu.....	10
1.5. Kontrola kriptu	12
1.5.1. Zobrazení protokolu s výsledky kontroly skriptu	12
1.5.2. Chybové stavy při kontrole skriptu	12
Chyba #0.....	13
Chyba #1 až #499.....	13
Chyba #504	13
Chyba #505	13
Chyba #507	13
Chyba #510	13
Chyba #511	14
Chyba #520	14
Chyba #530	14
Chyba #535	14
Chyba #540	14
Chyba #550	14
Chyba #560	15
Chyba #596	15
Chyba #599	15
Chyba #600	15
1.6. Inicializace skriptu	16
1.7. Obnova inicializace skriptu	16
1.8. Digitální podpis skriptu.....	17
1.9. Nízko úroňová editace skriptu.....	17
1.10. Simulace řídicího procesu pomocí skriptu	18
1.11. Skript Storage – interní úložiště skriptů	18
1.11.1. Přímé spouštění simulace z interního úložiště	19
1.12. Nastavení a konfigurace prostředí.....	19
1.12.1. Nastavení - barevná schémata	19
1.12.2. Konfigurace programu.....	20
1.12.3. Obnova konfiguračních souborů.....	21

1.13. Náповěda a elektronický manuál	22
2. Řídící skript	23
2.1. Popis skriptu	23
2.2. Použití řídícího skriptu	23
2.3. Historie formátu .DBF	24
2.3.1. Vznik dBase	24
2.3.2. dBase jako objektově orientovaný jazyk	24
2.3.3. xBase	24
2.3.4. Soubor databáze – obecný formát .dbf	24
2.3.5. Vývoj a současnost	25
2.4. Struktura skriptu	25
2.5. Rozšířená hlavička souboru .DBF ver. 4.x.x.	25
2.6. Nastavení ochrany zápisu skriptu	28
2.6.1. Program UnLockSC.EXE	28
3.1. Instalace	29
3.2. Aktualizace programů	30
3.3. Zálohování	30
4. Licence a autorská práva	31
4.1. Autorská práva	31
4.2. Licenční ujednání	31
4.3. Záruky a odpovědnosti	31
5. Adresy a kontakty	32
6. Slovníček pojmů	32

1.1. POPIS PROGRAMU

Program **Skript Studio I.** vznikl sjednocením jednotlivých programových nástrojů pro správu řídicích skriptů. Dosud bylo potřeba skript vytvořit v samostatném programu, poté jej zkontrolovat pomocí jiného, následně



Hlavní obrazovka programu Skript Studio I.

provést jeho inicializaci, a pokud bylo všechno v pořádku, bylo možno skript načíst do hlavního programu pro řízení procesů, nebo jeho simulace.

Vznikem programu **Skript Studio I.** se všechny potřebné programy sjednotily do jednoho celku, kde lze provést všechny operace, od vzniku skriptu, přes jeho editaci, ladění, inicializaci, až po možnost jej uložit do interního úložiště **Skript Storage**.

MEZI HLAVNÍ MOŽNOSTI SKRIPT STUDIA I. PATŘÍ:

- Tvorba a editace řídicího skriptu v editoru s pokročilými funkcemi
- Nízko úroňová editace databáze skriptu s přímým zápisem
- 5. fázová kontrola skriptu:
 - Kontrola formátu **.DBF/.DBG**
 - Kontrola syntaktické správnosti řídicích příkazů
 - Kontrola logické správnosti řídicích příkazů
 - Kontrola úplnosti skriptu (existence řídicích příkazů pro všechny dny cyklu)
 - Kontrola existence a syntaktické správnosti datumové řady – inicializace skriptu
- Inicializace, mazání inicializace, optimalizace skriptů
- Vkládání, mazání, aktualizace řídicích skriptů ve vnitřním úložišti **Skript Storage**
- Spouštění skriptů v procesu simulace uložených na disku i přímo z úložiště
- Testování, ladění, simulace běhu vytvořených skriptů
- Simulace procesu **GROWE**, s grafickým znázorněním celého procesu a výpočtem spotřeby el. energie
- Přístup k elektronickým manuálům, popisu formátu, příkazů řídicího skriptu, ovládání programu a další.

1.1.1. CO JE TO ŘÍDÍCÍ SKRIPT

Skript je soubor, obsahující příkazy, kterými se řídí automaticky ovládaný **Proces GROWE**. Pro řízení tohoto procesu je navržen program **Řízení MINI I.**, který čte příkazy v řídicím skriptu a tyto pak provádí v určeném čase, tz. spíná jednotlivá zařízení pomocí centrální spínací jednotky. Tato řídicí jednotka je připojena k počítači přes rozhraní **LPT**, přes který je ovládána a jako zpětná vazba také čten stav zařízení.¹

Příkazy pro celý cyklus jsou uloženy ve skriptu v databázi **GBase .DBG**, po jednotlivých větách. Věta je seznam příkazů pro určitý den. Příkazy ve větě jsou uloženy v polích, které mají specifické vlastnosti jako pozice ve větě, délka, typ informace, který obsahují a jiné. Této definici polí se říká tabulka databáze a fyzicky je v souboru uložena v hlavičce. Protože pro všechny věty platí stejná definice, jsou i všechny stejně dlouhé. Umožňuje to tedy snadné listování, snadnou úpravu vět i jednotlivých polí, přidávání, záměna, třídění, mazání, a to jak nevratně, tak s možností obnovy. Ne všechny uvedené vlastnosti a možnosti jsou využity, pro možnost, aby databáze nesla "další rozšiřující informace", byl původní **dBase .DBF** formát upraven a rozšířen. Zpětná kompatibilita byla zachována, **pouze pro prohlížeče obsahu dBase ver. 3.xx** kompatibilní. Použití **dBase ver. 3.xx** editorů, se pro editaci skriptu důrazně **nedoporučuje**, po uložení databáze těmito editory se nezachovají rozšiřující informace formátu **GBase**.

Rozšiřující informace, které jsou do skriptu vloženy, jsou např. ochrana skriptu proti zápisu, vložení kontrolního součtu CRC, kryptování obsahu databáze a další. Popis rozšiřující hlavičky je popsán v sekci: [2.4. STRUKTURA SKRIPTU](#)

Program pro tvorbu skriptu je vytvořen pro rychlou tvorbu tohoto skriptu, kdy pro pochopení, jak jej správně vytvořit, je třeba vysvětlit, jak řídicí program chápe skript. Skript obsahuje tedy hlavičku databáze, v níž je vložena tabulka plus další informace, dále jednotlivé věty, v počtu dnů procesu **GROWE**.

TABULKA DATABÁZE:

<i>pozice</i>	<i>název</i>	<i>typ</i>	<i>pozice</i>	<i>délka</i>
1.	DATUM	D	2	8
2.	DAY	C	10	6
3.	RELE_1	C	16	21
4.	RELE_2	C	37	21
5.	RELE_3	C	58	21
6.	RELE_4	C	79	21
7.	RELE_5	C	100	21
8.	RELE_6	C	121	21
9.	RELE_7	C	142	21
10.	RELE_8	C	163	21

První pole obsahuje datum (**typ D**), fyzický formát je **RRRRMMDD**, **RRRR** - rok, **MM** - měsíc a **RR** - rok. Druhé pole je pořadové číslo dne procesu **GROWE**. Tyto první dvě pole jsou do skriptu doplňovány automaticky při následných procesech, které budou popsány později. Dalších osm polí je určeno pro příkazy. Z výše uvedené tabulky databáze vyplývá, že prostor pro příkazy má předimenzovanou délku, i když jejich fyzická délka může být i 2 znaky. Do plné délky jsou doplněny mezerami na konci, které jsou při zpracování ignorovány a při ukládání opět doplňovány, není potřeba je ručně dopisovat.

Program je navržen tak, aby tvorba skriptu a ukládání jednotlivých příkazů byla co nejjednodušší, dotazové dialogy kontrolují zadané údaje, pokud se rozhodnete psát příkazy přímo, bude kontrolovat obsah po zadání příkazu. Změní barvu textu na červenou a při ukládání věty na chybný příkaz upozorní. Přesto pokud budete na chybném obsahu trvat, program umožní tento uložit, v následných procesech budete na tuto chybu upozorněni

¹ Program existuje také pro platformu Win10 + RaspberryPi, kde je připojení na řídicí modul realizován pomocí sběrnice GPio.

a bez opravy nebude možné skript tzv. inicializovat, tj. vložení finalizačních informací a vložit kontrolní součet. Bez těchto informací, nebo při jejich neplatnosti hlavní řídicí program takový skript odmítne vykonat. Proto, a i ve snaze o maximální stabilitu běhu procesu by mělo být v zájmu uživatele o vytvoření bezchybného skriptu. Pokud jej budete vytvářet pomocí dialogového menu, je možnost vytvoření chyby sníženo na minimum.

Pro snadnou tvorbu skriptu je třeba podrobněji vysvětlit, jak jej chápe řídicí program. Ten jej kontroluje na několika úrovních (úroveň ochrany lze nastavit v konfiguraci tohoto programu). Při prvotním otevření souboru nejdříve čte hlavičku souboru a dále hledá rozšiřující informace. Dále kontroluje platnost **CRC** součtu, pak kontroluje logickou platnost, nejprve prvního a pak druhého pole v každé větě. Poté hledá větu s aktuálním datem, tuto načte a započne s vykonáváním jednotlivých příkazů.

Příkazy (podle popisu v tabulce) jsou určeny pro **RELÉ 1 až 8**. Tyto relé, spínají napětí na zásuvkách hlavního napájecího rozvaděče. Jednotlivá zařízení mají přesně stanovenou zásuvku pro zapojení, z této konfigurace je potřeba vycházet při zadávání příkazů.

Vycházejme z konfigurace, kdy na rozvaděči jsou zapojeny zařízení takto:

1. zásuvka	zvlhčovač
2. zásuvka	pomocné osvětlení
3. zásuvka	hlavní osvětlení
4. zásuvka	základní odsávání
5. zásuvka	přídavné odsávání
6. zásuvka	vnitřní cirkulace vzduchu
7. zásuvka	zálivka (čerpadlo nebo jiné zařízení)

Relé 8 slouží pouze k systémovým účelům, proto pro něj nelze zadat jiný příkaz, než vypnuto - **OFF**. Nicméně, rozšíření konfigurace o další zásuvku je teoreticky možné, v konečném návrhu s ním ale není počítáno.

Některá zařízení, v tomto případě uvedme odsávání je řešeno jedním spotřebičem se dvěma nezávislými napájeními, nebo dvěma nezávislými zařízeními. V prvním případě, kdy ventilátor má dva režimy odsávání, pomalý s nízkým výkonem a hlučností a maximální s vysokým výkonem a hlučností. Oba režimy nemohou jet současně, a to z důvodu, že motor ventilátoru má dva druhy vinutí, které nemohou být napájeny současně. Program má ochranu na úrovni řídicího modulu, kdy nedovolí jejich současné sepnutí, a to jak v automatickém, tak manuálním režimu ovládání. Proto lze pro **relé 5** možno použít pouze příkaz **ON**, **OFF** nebo **AUTO**. Toto relé (zařízení) má ve své funkci v procesu **GROWE** vyšší prioritu, pokud použijete příkaz **ON**, sepne se přídavné odsávání a tím se odpojí odsávání základní. Tento stav se nezmění, dokud **relé 5** nedostane příkaz **OFF** a nedojde k jeho vypnutí. Teprve potom bude **relé 4** akceptovat příkaz pro sepnutí.

Pro jednotlivá relé lze použít tyto příkazy:

Relé 1	OFF, ON, TO, WHIT
Relé 2	OFF, ON, TO, WHIT
Relé 3	OFF, ON, TO, WHIT
Relé 4	OFF, ON
Relé 5	OFF, ON, AUTO
Relé 6	OFF, ON, TO, WHIT
Relé 7	OFF, ON, TO, WHIT
Relé 8	OFF

Ted' k samotnému návrhu skriptu. Nejdříve je si třeba ujasnit proces **GROWE**, druh a perioda světla, typ závlahy, délka jednotlivých fází procesu. Předpokládáme, že pro různé fáze procesu požadujeme jinou světelnou periodu, s jiným druhem světla, s jinou hodnotou vlhkosti.

Pro první dny proto navrhujeme pro I. fázi 18-ti hodinovou světelnou periodu, základní druh světla a maximální vlhkost. Postupně přidáváme druhé osvětlení až do plného nahrazení základního světla. Po celou dobu přizpůsobujeme požadavkům další funkcionality jako zálivka, vnitřní ventilace...

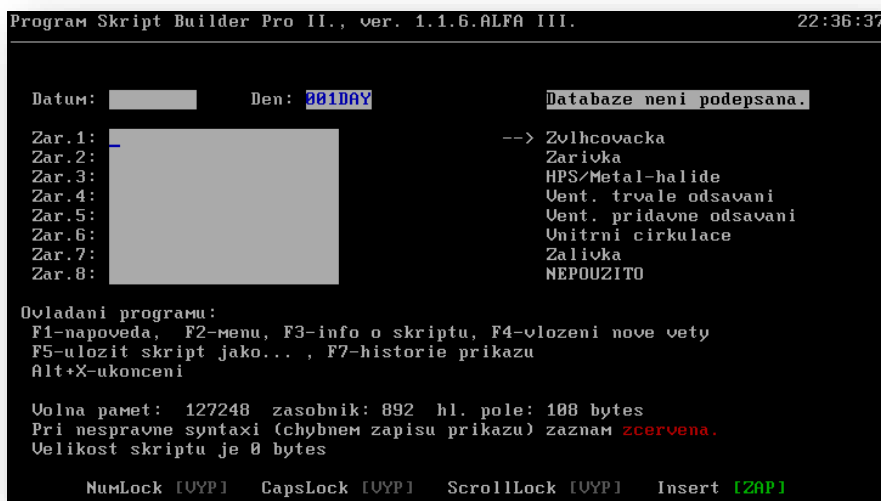
Po požadované délce fáze I. nastavíme přechod na 12-ti hodinovou světelnou periodu, (fáze II.), změníme četnost zvlhčování, které se postupně může úplně vyřadit. Jako ukončení lze nastavit všechny příkazy posledního dne na **OFF**, proces se neukončí přechodem přes půlnoc. Při ukončování procesu **GROWE** dochází k uložení veškerých dat disk, pro pozdější analýzu, proto nemusí být vždy žádoucí končit proces uprostřed noci.

Dá se říct, že vytvořit skript lze pouze přidržením klávesy **ENTER**. Ve skutečnosti to tak jednoduché není, pokud si ale návrh skriptu dobře promyslíte, bude Vám program **Skript Studio I.** dobrým pomocníkem při jeho vytvoření.

1.2. OVLÁDÁNÍ PROGRAMU

Program má sjednocené ovládání klávesami, šipkovými klávesami pohyb po menu, **ENTER** výběr volby, **F1** nápověda, **ESC** nebo **Alt+X** - ukončení editace nebo dialogu. Pokud jsou jednotlivé editační, konfigurační funkce v programu ovládány dalšími klávesami, jsou tyto na obrazovce vždy uvedeny. Součástí programu **Skript Studio I.** je funkce nápovědy, pomocí klávesy **F1**, kde je možné nalézt podrobné informace o jednotlivých funkcích a jejich ovládání a použití.

1.3. NOVÝ SKRIPT



Skript Builder II. - tvorba nového skriptu

Výběrem volby Nový skript se spustí editor, pomocí kterého vytvoříte celý příkazový skript, po jednotlivých dnech. Do editačních řádků lze zadávat příkazy pro jednotlivá zařízení. Podrobný popis je v kapitole 2. **Řídící Skript.**

Pro tvorbu i editaci skriptu se používá původní program **Skript Builder Pro II.**, který je i součástí programu **Skript Studio I.** Design, ovládání programu i způsob tvorby skriptu bylo zachováno, některé původně ruční volby byly zautomatizovány, jako např. vkládání digitálního podpisu, který se aktualizuje na platný při každém ukládání skriptu.

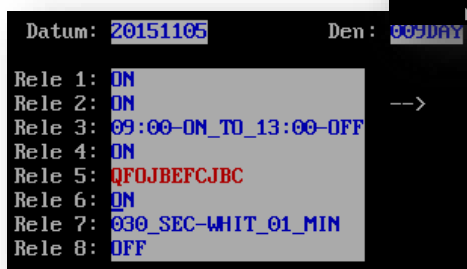
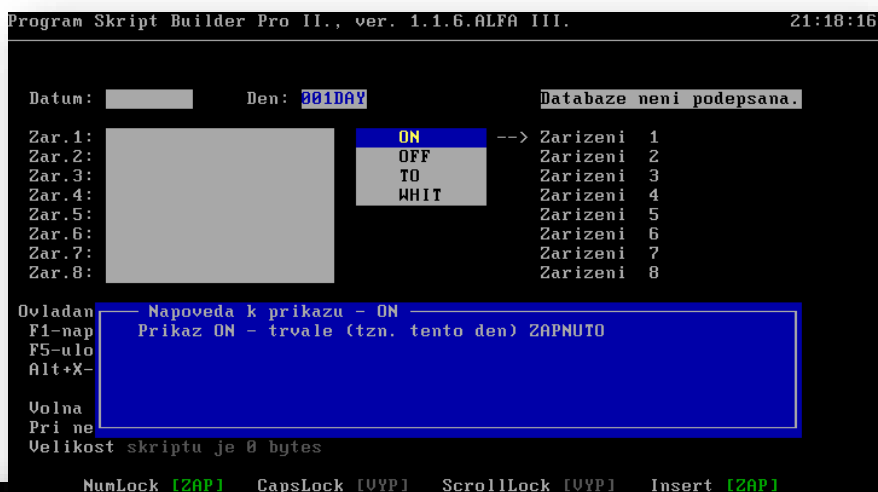
PŘÍKAZY, KTERÉ LZE POUŽÍT PŘI TVORBĚ SKRIPTU JSOU:

OFF	- spotřebič vypnut
ON	- spotřebič zapnut
AUTO	- automaticky režim řízení (ovládá se za základě měření stavu teplot/y)
TO hh:mm-ON_TO_hh:mm-OFF	- v hh:mm (hodina:minuta) zapnout , v hh:mm (hodina:minuta) vypnout
WHIT sss_SEC-WHIT_mm_MIN	- zapnout na sss (sekund), co mm (minut), příkaz se cyklicky opakuje

Pozn.:

Cyklus WHIT začíná vždy zapnutím na sss (sekund), poté následuje čekací stav v délce mm(minut), přesná definice logického významu je: po uplynutí "mm" minut zapnout na "sss" sec., neustále se opakující cyklus.

Příkazy lze vpisovat přímo, pro usnadnění je možné použít klávesu **F2**, která vyvolá menu, ve kterém lze požadovaný příkaz vybrat a ihned vložit. Vložené příkazy jsou ihned kontrolovány na syntaktickou správnost, kdy formát příkazu je pevně dán a pokud není dodržen, změní



Chybně zadaný příkaz

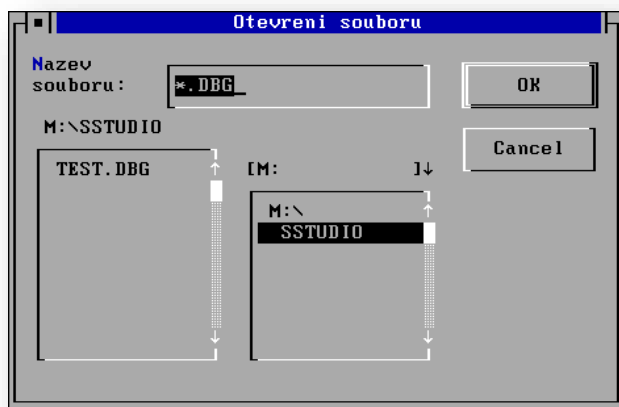
Příkazové menu s nápovědou k příkazu

svou barvu na červenou.

Chybné příkazy jsou takto označeny, nicméně program nezamezuje jejich uložení do výsledného skriptu.

1.4. TVORBA / OTEVŘENÍ JIŽ EXISTUJÍCÍHO SKRIPTU

Při výběru otevření existujícího skriptu se zobrazí dialog pro výběr souboru, jinak program zobrazí hlavní obrazovku. V levé části je editační část pro zápis, v pravé jsou doplňující informace. Ve spodní části je nápovědný text, úplně dole je zobrazen stav systémových funkcí klávesnice. Pro okamžitou nápovědu lze kdykoli v programu použít klávesu **F1**, která zobrazí text vztahující se k aktuálně prováděné činnosti.

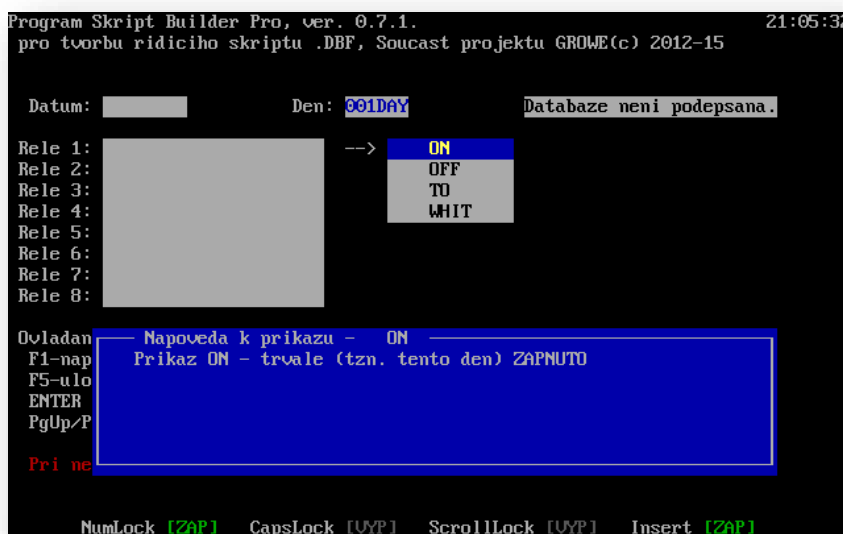


Dialog Otevření souboru

Editační okna **Datum** a **Den** nejsou přístupná pro zápis, jak bylo již dříve uvedeno, tyto informace se doplní automaticky. Pole **Den** se vkládá automaticky jako číselná řada +1 pro každý nový den, pole **Datum** se naplní datumovou řadou při inicializaci skriptu. Datumová řada je posloupnost datumů, kdy jako první se vloží aktuální ve chvíli inicializace skriptu, další následují dle kalendáře. Veškeré odchylky jako např. přestupný rok jsou akceptovány a program s nimi počítá.

Po výběru Nový skript je kurzor umístěn v poli **Relé 1** na první pozici a je možno hned zadávat příkazy. Po stisku **ENTER** se zadaný příkaz zkontroluje, pokud je správný, přesune se kurzor na první pozici druhého okna **Relé 2**. Pokud není příkaz syntakticky správný, změní se barva textu na červenou. Pokud je příkaz psán malými písmeny, změní se na velká. Mezi jednotlivými editačními okny se lze pohybovat šipkovými klávesami, nebo kl. **TAB** (další) a **SHIFT+TAB** (předchozí). Mezi jednotlivými větami se lze pohybovat klávesami **PGUp** a **PGDown**.

Klávesou **F2** lze vyvolat menu, které nabídne dostupné příkazy pro pole, na kterém je umístěn kurzor. Výběrem a potvrzením **ENTER** se příkaz vloží do pole. Pokud je potřeba doplnit další údaje do příkazu, přesune se kurzor na toto místo a očekává jeho vložení. Do příkazu lze vložit pouze povolené znaky (např. nelze vložit písmena do údaje čas), v opačném



Hlavní obrazovka s výběrovým menu příkazů

případě program tyto znaky smaže a zvukovou signalizací upozorňuje na chybné vložení. Pokud je údaj akceptován, přesune se kurzor na další údaj. Při ukládání se dále kontroluje logická správnost zadávaných údajů, např. není možné ve stejný čas provést zapnutí i vypnutí relé, druhý zadávaný údaj se v tomto případě smaže a požaduje vložení jiného údaje

(příkaz **TO**). Po vložení příkazu do pole **relé 8** se zobrazí dotaz na uložení právě vytvořené věty a vytvoření další v pořadí.

Věta se dá kdykoliv uložit klávesou **F5**, pokud ale obsahuje chybné příkazy označené červenou barvou, uložení se neprovede a zobrazí se chybové hlášení.

Po vytvoření celého skriptu lze klávesou **F6** vložit digitální podpis. Klávesa **F3** zobrazí podrobné informace o vytvářeném skriptu, verze, datum vytvoření, tabulka databáze, **CRC** součet a jiné.

Pouze pro čtení: NE

Informace o příznaku zákazu

Program lze ukončit stiskem **Alt+X**, pokud během editace došlo ke změně v databázi, zobrazí se dotaz na uložení změn do souboru před ukončením. Pokud je ale nastaven příznak **Pouze čtení**, není zápis změn do databáze možný. Proto neměňte obsah souboru takto označený (nejdříve přesvědčte se o možnosti zápisu, informace získáte pomocí klávesy **F3**), změny se neuloží a po ukončení programu budou ztraceny. Je důležité nejprve odstranit příznak **Pouze čtení**, teprve poté provádět změny v obsahu.

```

Informace o databázi                                     21:18:35

Verze: 4.xx, GROWE type
Poslední aktualizace: 12/28/15
Pocet zaznamu: 0   delka zaznamu: 183   pocet poli: 10
Rozsirena hlavicka (ver. 4.xx a vyssi): *0000000000000000G
Pouze pro cteni: NE   CRC32: nelze provest, nenalezen zadny zaznam.

+-----+
| Tabulka databaze |
+-----+
| Nazev      | Typ | Pozice | Delka | Roz. cis. |
+-----+
| DATUM      | D   | 2       | 8      | 0          |
| DAY        | C   | 10      | 6       | 0          |
| RELE_1     | C   | 16      | 21      | 0          |
| RELE_2     | C   | 37      | 21      | 0          |
| RELE_3     | C   | 58      | 21      | 0          |
| RELE_4     | C   | 79      | 21      | 0          |
| RELE_5     | C   | 100     | 21      | 0          |
| RELE_6     | C   | 121     | 21      | 0          |
| RELE_7     | C   | 142     | 21      | 0          |
| RELE_8     | C   | 163     | 21      | 0          |
+-----+

klavesa pro pokracovani

NumLock [ZAP]   CapsLock [UYP]   ScrollLock [UYP]   Insert [ZAP]

```

Informace o struktuře databáze

Součástí programu je několik již hotových skriptů, navržených pro univerzální použití. Tyto můžete jakkoli zkoumat, upravovat. Nicméně doporučujeme uchovávat použité skripty prověřené provozem, kdy každý úspěšně dokončený proces **GROWE** je do skriptu zapsán a tato informace pak slouží jako rozlišení mezi ověřeným a neověřeným skriptem. Tato informace je pouze informativní, její hodnota slouží pouze pro uživatele. Toto značení může být z důvodu ochrany soukromí zakázáno.

```

Program Skript Builder Pro II., ver. 1.1.6.ALFA III.                                     22:46:03

Datum: [ ] Den: 281DAY   Podpis databaze neni platny.

Zar.1: DN   --> Zvolhcovacka
Zar.2: DN   Zarioka
Zar.3: DN   HPS/Metal-halide
Zar.4: DN   Vent. trvale odsavani
Zar.5: DN   Vent. pridane odsavani
Zar.6: DN   Unitrni cirkulace
Zar.7: DN   Zalioka
Zar.8: DN   NEPOUZITO

Ovladani programu:
F1-napoveda, F2-menu, F3-info o skriptu, F4-vlozeni nove vety
F5-ulozit skript jako... , F7-historie prikazu
Alt+X-ukonceni

Volna pamet: 124816   zasobnik: 538   hl. pole: 324 bytes
Pri nespravne syntaxi (chybnem zapisu prikazu) zaznam zcervena.
Velikost skriptu je 0 bytes

NumLock [UYP]   CapsLock [UYP]   ScrollLock [UYP]   Insert [ZAP]

```

Skript Builder II. – editace existujícího skriptu

UPOZORNĚNÍ:

Pokud budete zaměňovat skript pomocí dálkové správy, je platný digitální podpis vyžadován VŽDY!!!

1.5. KONTROLA KRIPTU

Pro proces inicializace skriptu se používá program **.DBF Reader Pro**, soubor **DBF_READ.EXE**. Tento program postupně analyzuje celý skript v několika krocích, kdy kontroluje platnost všech údajů v hlavičce souboru, dále syntaktická a logická správnost všech příkazů, pokud jsou všechny testy úspěšné, provede zálohu skriptu a jeho inicializaci. V opačném případě ohlásí nalezenou chybu a nabídne možnost opravy. Po uložení opravených údajů se celý kontrolní proces provede znovu. Pokud opět najde chybu, opět ji ohlásí, po opravě znovu spustí test a toto probíhá až do chvíle, kdy skript vyhoví testům. Teprve tehdy program umožní jeho inicializaci.

```

Uytvoreno pomocí programu DBF_READ ver. 3.9.5. build #270 BETA XU.
Soucast projektu GROWE(c) 2012-17

Nazev skriptu: TEST.DBG          Datum skriptu:
Verze databaze: 3.00            Den cyklu: 003DAY
Ulozeno (d/m/r): 17/5/17        Rele 1 :
Pocet vet: 3                    Rele 2 :
Delka vet: 183                  Rele 3 :
Slov ve vete: 10                Rele 4 :
Slov celkem: 30                 Rele 5 :
Pouze pro cteni: NE             Rele 6 :
                                  Rele 7 :
                                  Rele 8 :

Vysledky testu:
Inicializace skriptu... NENI
Uplnost skriptu... PRAZDNE PRIKAZY
Syntaxe skriptu... CHYBA

Datum inicializace:
DataBase Table Type: GROWE
Volna pamet: 118656 bytes
Zasobnik: 538 bytes

Detekovane chyby skriptu:
Detekovane chyby skriptu: #530, #511
Podrobne informace o nalezenych chybach jsou v souboru TESTSCR.TXT

PAUZA

```

DBG Reader – kontrola vytvořeného skriptu

1.5.1. ZOBRAZENÍ PROTOKOLU S VÝSLEDKY KONTROLY SKRIPTU

O provádění všech fází testu skriptu se vytváří v souboru **TestSCR.TXT** záznam. Na začátku souboru je uvedena celá struktura databáze a její vlastnosti, dále pak záznamy o jednotlivých testech a jejich výsledky. Pokud při testu byla nalezena chyba ve skriptu, je zde i uvedeno číslo a popis této chyby a postup při jejím řešení. Soubor zůstává na disku přítomen i po ukončení programu **Skript Studio I.**, je možné jej přejmenovat a použít k popisu skriptu. Novým spuštěním testu skriptu se soubor výsledků přepíše aktuálními hodnotami.

```

Radek: 22 Soubor: testscr.txt Konec: ESC Pohyb: ↑↓ PgUp PgDn Home End
Detekovana chyba #530:
Skript DATABASE.DBG není inicializovan.

Popis chyby:
Při kontrole skriptu bylo zjištěno, že není INICIALIZOVAN, což znamená,
že neobsahuje datovou radu. Při kontrole je tato operace nahrazena
automaticky, s tím, že do první příkazové vety se vloží aktuální datum
do dalších pak daty následující. Při použití v procesu GROWE(c) daty
slouží k jednoznačnému určení příkazu pro určitý den.

Skript není VALIDNI pro použití v procesu GROWE(c).

Uplnost skriptu...PRAZDNE PRIKAZY

Detekovana chyba #511:
Cast skriptu DATABASE.DBG je prazdna.

Popis chyby:
Byla detekovana nekonzistence skriptu, jeho cast je prazdna.
Pouzijte program Skript Studio a doplnite chybejici casti. Pokud ma skript
jiz vami pozadovany pocet prikazu a na jeho konci je pouze prazdny prostor,
pro jeho odstraneni pouzijte program Skript Studio, funkci OPTIMALIZACE SKRIPTU

```

Výsledek kontroly skriptu

1.5.2. CHYBOVÉ STAVY PŘI KONTROLE SKRIPTU

Při kontrole skriptu může dojít k chybovým stavům, jejich číslo prezentuje chybové hlášení a stručný návod k odstranění těchto stavů. Chyby se vztahují k jednotlivým druhům testu, proces se při jejím nalezení nepřerušuje, pouze se vytvoří záznam o chybě a test pokračuje. Proto je možné, že je během testu nalezeno více chyb. Pro odstranění chyb ve skriptu použijte volbu Editace skriptu, pokud jste zkušený uživatel, můžete použít editor s přímým přístupem. Poté proveďte optimalizaci databáze a spusťte test znovu. Teprve až bude výsledek testu bez chyb, bude možné provést jeho inicializaci.

```

Radek: 1 Soubor: testscr.txt Konec: ESC Pohyb: ↑↓ PgUp PgDn Home End
AUTOMATICKY UYGENEROVANY TEXT, vysledky kontroly skriptu TEST.DBG

Uytvoreno pomocí programu DBF_READ ver. 3.9.5. build #270 BETA XU.
Soucast projektu GROWE(c) 2012-17

05-17-2017 22:57:42
Nazev skriptu: TEST.DBG
Verze databaze: 3.00
Ulozeno (d/m/r): 17/5/17
Pocet vet: 3
Delka vet: 183
Slov ve vete: 10
Slov celkem: 30
Pouze pro cteni: NE

Probíha kontrola struktury databaze skriptu TEST.DBG
Probíha I. faze kontroly skriptu TEST.DBG
Vysledky testu:
Inicializace skriptu... NENI
Probíha II. faze kontroly skriptu TEST.DBG

Detekovana chyba #530:
Skript DATABASE.DBG není inicializovan.

```

Zobrazení obsahu souboru TESTSCR.TXT s výsledky testu

CHYBA #0

POPIS CHYBY:

Během práce s programem nedošlo k žádné chybě v běhu programu, ukončení proběhlo normálně. Podrobnější informace jsou uloženy v běhových .LOG souborech (pokud je v konfiguraci nastaveno, aby byly vytvářeny).

Skript je VALIDNÍ pro použití v procesu GROWE.

CHYBA #1 AŽ #499

POPIS CHYBY:

Závažná systémová chyba, prosím informujte autora o této chybě na E-mailové adrese: support@projekt-growe.cz

CHYBA #504

POPIS CHYBY:

Zápis skriptu je v zastaralém formátu databáze ver3.00. Pro konverzi na rozšířenou verzi 4.x.x. použijte program **Skript Studio I.** (sstudio.exe), volbu **OPTIMALIZACE DATABASE.**

Skript není VALIDNÍ pro použití v procesu GROWE.

CHYBA #505

POPIS CHYBY:

Při otevírání skriptu se vyskytla chyba, a to z těchto možných příčin: Soubor má nulovou délku, právě jej používá jiný proces, nebo je závažným způsobem poškozen. Pro jeho opravu použijte program Skript Studio (sstudio.exe). Pokud se vyskytuje tato chyba a soubor má nulovou délku, byl pravděpodobně přepsán. Při používání přenosných úložišť, používejte vždy funkci odpojení od **OS** před jeho vyjmutím z počítače. Předejdete tak ztrátám dat, jejichž uložení bylo odloženo systémem na pozdější dobu a nekorektním vyjmutím paměťového media nebylo uložení dat dokončeno.

Skript není VALIDNÍ pro použití v procesu GROWE.

CHYBA #507

POPIS CHYBY:

Nebyla detekována tabulka databáze skriptu dle specifikace **GROWE**. Toto není závažná chyba, programy **Projektu GROWE** umí s takovouto databází pracovat, nicméně, tento skript pravděpodobně nevznikl v programu **Skript Studio I.** a nemusí obsahovat rozšířenou hlavičku - **ExtHead**, která obsahuje dodatečná data. Bez těchto informací nemusí být skript akceptován v procesu GROWE, pokud jsou nastavena přísná pravidla pro zabezpečení procesu. Pro konverzi na rozšířenou verzi databáze použijte program **Skript Studio I.** (sstudio.exe), volbu **OPTIMALIZACE DATABASE.**

Skript plně neodpovídá specifikaci GROWE.

CHYBA #510

POPIS CHYBY:

Nebyl detekován formát skriptu **.DBG dBase ver.4.x.x.** typ **GROWE(c)**. Soubor pravděpodobně neobsahuje řídicí skript, pokud jste přesvědčen, že ano, použijte program **Skript Studio I.** (sstudio.exe) pro jeho opravu.

Skript není VALIDNÍ pro použití v procesu GROWE.

CHYBA #511

POPIS CHYBY:

Byla detekována nekonzistence skriptu, jeho část je prázdná. Použijte program Skript Studio I. a doplňte chybějící části. Pokud má skript již vámi požadovaný počet příkazů a na jeho konci je pouze prázdný prostor, pro jeho odstranění použijte program **Skript Studio I.**, funkci **OPTIMALIZACE SKRIPTU**.

POZOR!!!

Pokud jsou některé věty skriptu označené příznakem * (**DELETED**) - určeny pro smazání, budou tyto při použití této funkce smazány **TRVALE**. Skript **není VALIDNÍ** pro použití v procesu GROWE.

CHYBA #520

POPIS CHYBY:

Zadaný skript nebyl nalezen v aktuálním adresáři, zkontrolujte, zda jste zadali správný název, nebo jestli jste umístili soubor do požadovaného adresáře.

CHYBA #530

POPIS CHYBY:

Při kontrole skriptu bylo zjištěno, že není **INICIALIZOVÁN**, což znamená, že neobsahuje datumovou řadu. Při kontrole je tato operace nabídnuta automaticky, s tím, že do první příkazové věty se vloží aktuální datum, do dalších pak datumy následující. Při použití v procesu GROWE datumy slouží k jednoznačnému určení příkazů pro určený den.

Skript není VALIDNÍ pro použití v procesu GROWE.

CHYBA #535

POPIS CHYBY:

Při kontrole skriptu bylo zjištěno, že **INICIALIZACE** není platná, což znamená, že je buďto smazána a chybně obnovena, nebo je nekonzistentní. Pro opravu skriptu smažte tuto neplatnou **INICIALIZACI** a podle potřeby proveďte novou.

Skript není VALIDNÍ pro použití v procesu GROWE.

CHYBA #540

POPIS CHYBY:

Při kontrole skriptu byly nalezeny chyby, které musí být opraveny. Dokud nebudou odstraněny, nebude možné skript inicializovat pro použití v procesu GROWE.

Skript není VALIDNÍ pro použití v procesu GROWE.

CHYBA #550

POPIS CHYBY:

Při kontrole skriptu bylo zjištěno závažné poškození skriptu. Pokud to bude možné, použijte pro jeho editaci program **Skript Studio I.** (sstudio.exe). Opravte poškozené věty, popř. je označte příznakem (***DELETED**) a použijte funkci **OPTIMALIZACE DATABAZE** pro jejich odstranění. Poté doplňte příkazy skriptu do požadovaného stavu, pro použití v procesu GROWE jej inicializujte.

Skript není VALIDNÍ pro použití v procesu GROWE.

CHYBA #560

POPIS CHYBY:

Obsah skriptu je chráněn příznakem **POUZE PRO ČTENÍ**, který je součástí rozšířené hlavičky souboru. Pokud je příznak nastaven, není možné provádět ve skriptu žádné změny. Pokud se přesto o toto pokusíte, změny se neuloží. Pro odstranění příznaku použijte pomocnou utilitu **Skript UnLocker** (unlocksc.exe), nebo změňte v **HEXA** editoru **18-bit** skriptu na **00**. Ochranu opět aktivujete změnou tohoto bitu na hodnotu **F0**, nebo opětovně použijte utilitu **Skript UnLocker**. Tento příznak, ať už aktivní nebo neaktivní není považován za chybu, hlášení je pouze informativní o restrikci v přístupu k souboru. Tedy v případě nalezení chyby kontrolním programem, ji nebude možné pro tuto ochranu automaticky opravit.

CHYBA #596

POPIS CHYBY:

Při vytváření zálohových souborů byl překročen jejich maximální počet. Prosím odstraňte nepotřebné zálohové soubory (**ZALOHA00.DBG** až **ZALOHA99.DBG**).

CHYBA #599

POPIS CHYBY:

Při běhu programu uživatel stiskl klávesu **ESC**, čímž přerušil jeho běh. Kontrola skriptu nebyla dokončena a nebude možné ho použít v procesu GROWE.

CHYBA #600

POPIS CHYBY:

Program lze použít ke smazání již existující datumové rady – INICIALIZACE. Ta spočívá ve smazání datumu pouze v první větě, kdy takový skript je pak detekován jako dosud neinicializován. K mazání pouze prvního záznamu dochází proto, že pokud byla inicializace smazána omylem, lze ji např. **HEXA** editoru obnovit. Pokud ale bude po smazání skript znovu inicializován, je původní datumová řada přepsána a tím i trvale ztracena.

Výsledky, v závislosti na nalezených chybách se řadí do třech skupin:

Skript je VALIDNÍ

Skript plně NEODPOVÍDÁ SPECIFIKACI (je validní s výhradami)

Skript není VALIDNÍ.

1.6. INICIALIZACE SKRIPTU

Inicializace skriptu je proces, při němž se do všech vět přidají k příkazům i jednotlivé datumy, v postupné řadě za sebou podle kalendáře (datumová řada). Podle nich se budou načítat příkazy pro jednotlivé dny, v případě přerušení běhu řídicího programu, tento ví, kde pokračovat. Procesu inicializace předchází kompletní kontrola skriptu, pokud nejsou žádné chyby nalezeny, je provedení inicializace automaticky nabídnuto. V případě odmítnutí jsou další operace přerušeny a program ukončen. Pokud ale provedení inicializace odsouhlasíte, provede se nejdříve záloha původního souboru, poté se do souboru vloží datumová řada. První datum v řadě je aktuální datum, kdy se inicializace provedla. Po úspěšné inicializaci je skript připraven pro použití v procesu GROWE.

```
DBG_READ ver. 3.9.5. build #270 BETA XV.
Soucast projektu GROWE(c) 2012-17

Nazev skriptu: TEST.DBG          Datum skriptu: 20170524
Verze databaze: 3.00            Den cyklu: 004DAY
Ulozeno (d/m/r): 21/5/17        Rele 1 : ON
Pocet vet: 4                    Rele 2 : ON
Delka vety: 183                 Rele 3 : ON
Slov ve vete: 10                Rele 4 : ON
Slov celkem: 40                 Rele 5 : ON
Pouze pro cteni: NE             Rele 6 : ON
                                Rele 7 : ON
                                Rele 8 : ON

Vysledky testu:
Inicializace skriptu... OK
Uplnost skriptu... OK          Datum inicializace: 21.05.2017
Syntaxe skriptu... OK          DataBase Table Type: GROWE
                                Volna pamet: 146224 bytes
                                Zasobnik: 872 bytes

Detekovane chyby skriptu:
Podrobne informace o nalezenych chybach jsou v souboru TESTSCR.TXT

Pokracovani po stisku klavesy <p - pauza> <z - zobrazeni .LOGu> < 8 sec >
```

Test po provedení inicializace

```

+ + + [001DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]
+ + + [002DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]
+ + + [003DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]
+ + + [004DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]
+ + + [005DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]

```

Náhled na neinicializovaný skript (první slovo v hranatých závorkách chybí)

1.7. OBNOVA INICIALIZACE SKRIPTU

Program DBF_READ.EXE umožňuje i smazání již existující inicializace, kdy smaže datum v první větě, ostatní programy tento skript budou považovat za neinicializovaný, to znamená, že na něj pohlížejí jako na nově vytvořený, kdy datumová řada ještě nebyla do něj vložena. Skutečnost je ale taková, že se maže jen první záznam, což umožňuje obnovení původní datumové řady. Operace obnovení smazané datumové řady je poměrně nestandardní, vyžadující znalost problematiky přímé editace, tzv. "že uživatel ví, co dělá". Její poměrná nesnadnost, vyžadující nízkou úroveň editace v HEXA editoru a přesné vložení datumové řady do prvního záznamu ve větě do souboru. Další možnost použití funkce nízké úrovně editace, která je součástí **Skript Studio I**. Podrobný popis této funkce je také v samostatném souboru, v elektronické. Existuje jistá pravděpodobnost, že se Vám tato operace nemusí podařit a dojde k porušení konzistence databáze. Proto si vždy před "operací" vytvořte zálohu souboru, ať neztratíte možnost návratu k funkční verzi skriptu. Při závažném porušení konzistence není zaručeno, že bude možná obnova obsahu databáze. Zvláště zranitelný v tomto je skript s kryptovaným obsahem, kde jakákoli změna způsobí porušení integrity a trvalou ztrátu obsahu.

```

[ ] [ ] [ ] [001DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]
[ ] [ ] [ ] [002DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]
[ ] [ ] [ ] [003DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]
[ ] [ ] [ ] [004DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]
[ ] [ ] [ ] [005DAY] [023_SEC-WHIT_01_MIN] [030_SEC-WHIT_01_MIN] [010_SEC-WHIT_01_MIN] [ON] [AUTO] [020_SEC-WHIT_01_MIN] [042_SEC-WHIT_01_MIN] [065_SEC-WHIT_01_MIN]

```

Smazaná inicializace skriptu, červeně je označen první smazaný záznam

Pokud byla inicializace skriptu smazána, nebo pokud ještě nebyla vložena, použijte funkci Inicializace skriptu. Dojde k vložení celé datumové řady, počínaje aktuálním datem přiřazeným k první příkazové větě skriptu. Po vložení nové datumové řady obnova původní již není možná.

1.8. DIGITÁLNÍ PODPIS SKRIPTU

Digitální podpis je druh ochrany obsahu řídícího skriptu, kdy po jeho vytvoření se do hlavičky skriptu uloží tento podpis, obsahující cyklický redundantní součet obsahu, označovaný také **CRC** (zkratka z anglického názvu **Cyclic Redundancy Check**). Je to speciální hašovací funkce, používaná k detekci chyb během přenosu či ukládání dat. Výsledek tohoto kontrolního součtu je uložen v hlavičce, a pokud tento existuje (tedy není ve výchozím stavu **00000000**), je programem pro práci se skriptem a hlavním řídícím programem čten, následně proveden nový kontrolní součet s aktuálními daty a výsledek je s uloženým porovnán. Pokud se shodují, je digitální podpis uznán jako **PLATNÝ**. V opačném případě je předpoklad, že byl obsah od vložení podpisu modifikován a digitální podpis je označen jako **NEPLATNÝ**.

1.9. NÍZKO ÚROVŇOVÁ EDITACE SKRIPTU

Program Skript Studio I. obsahuje mocný nástroj, a to editor pro přímou editaci skriptu. Pomocí přímého přístupu

```
Record 1/ 3

DATUM
DAY      001DAY
RELE_1   ON
RELE_2   ON
RELE_3   ON
RELE_4   ON
RELE_5   ON
RELE_6   ON
RELE_7   ON
RELE_8   ON

Next Prev Edit Delete Undelete Add Refresh Quit

Volna pamet: 122416 bytes, zasobnik: 538 bytes
Velikost souboru TEST.DBG je 903 Bytes
```

Editor s přímým přístupem k databázi

původní, což by systém vyhodnotil jako duplicitu záznamů. Věty s nastaveným příznakem ***(DELETED)** jsou ale systémem ignorovány, proto je možné jejich zachování, aniž by došlo k narušení integrity databáze.

Program se ovládá pomocí kláves vyznačených ve spodní části červenou barvou. Další informace obsahují hodnoty o volné paměti, velikosti volného zásobníku pro proměnné (stack) a velikosti editovaného souboru.

Upozornění:

Důrazně doporučujeme před editací skriptu si provést jeho zálohu. Kromě nastavení příznaku pro smazání věty jsou všechny provedené editační zásahy nevratné.

do databáze umožňuje listování po větách, editaci jednotlivých příkazů, přidávání prázdných vět na konec, nebo vkládání do těla databáze, označovat věty příznakem pro smazání. Jakékoli změny jsou ihned uloženy, bez předchozího dotazu nebo varování, neprovádí se ani kontrola na syntaktickou správnost příkazů. Věty označené pro smazání ***(DELETED)** zůstávají součástí databáze, až do provedení operace **OPTIMIZE DATABÁZE**. Poté jsou smazány trvale. Příznak pro mazání slouží k označení věty, která má být nahrazena novou, obsahující stejný datum a pořadové číslo jako věta

1.10. SIMULACE ŘÍDÍCÍHO PROCESU POMOCÍ SKRIPTU

Vytvořený skript je možné vyzkoušet v simulovaném procesu řízení **GROWE**. Simulace probíhá ve zrychleném režimu, kdy jsou všechny kroky řízení prováděny a kontrolní procedurou ihned překontrolovány na shodnost výsledných stavů. Rychlost simulace se odvíjí od rychlosti procesoru, na kterém je program spuštěn. I přes toto zrychlení může simulace cyklu o délce např. **60-ti** dnů trvat několik hodin. V horní části obrazovky jsou údaje o skriptu, ukazatel vytižení systému, ve spodní části je výpis stavů simulovaného řídicího modulu. Sloupec **Test1** obsahuje stavy řídicí procedury, sloupec **Test2** výsledky kontrolní procedury, které se neustále porovnávají.

Sloupec **Valid** pak zobrazuje výsledek porovnávání a stav validačního procesu. Ve sloupci **RAW prikaz** je příkaz načtený z řídicího skriptu. Sloupec **LED** signalizuje čtený stav z řídicího hardwarového modulu (v režimu simulace nemá uplatnění). Ve sloupci **Spotreba** je odběr jednotlivých spotřebičů, ve sloupci **Cena** se pak vypočítává cena za spotřebovanou el. energii. Údaje o spotřebě a ceně za 1KWh je třeba zadat do konfiguračního souboru **SSTUDIO.INI**. Popis nastavení hodnot je popsán v kapitole [1.12.2. KONFIGURACE PROGRAMU](#)

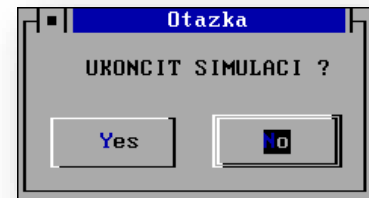
Validate ver. 2.2.5. Soucast projektu GROWE(c) 2012-17
simulace behu ridiciho skriptu .DBG

Simulace cas: 00:09:40 Start testu: 05-17-2017 23:00:40 82840.16 CPUc: 68
MYTIMER: 588 RETIMER: 588 Mod behu: Simulace
Cislo Dne: 1 / 3 Skript : TEST.DBG

	Test1	Test2	Valid	RAW prikaz	LED	Spotreba	Cena
Rele 1	ON	ON	OK	ON	■	18 W	0.01 Kc
Rele 2	ON	ON	OK	ON	■	20 W	0.01 Kc
Rele 3	ON	ON	OK	ON	■	40 W	0.02 Kc
Rele 4	ON	ON	OK	ON	■	250 W	0.33 Kc
Rele 5	ON	ON	OK	ON	■	50 W	0.03 Kc
Rele 6	ON	ON	OK	ON	■	80 W	0.05 Kc
Rele 7	ON	ON	OK	ON	■	40 W	0.02 Kc
Rele 8	ON	ON	OK	ON	■	0 W	0.00 Kc
Celkem							0.47 Kc

Volna pamet: 119616 Bytes Zasobnik: 538 Bytes
Ukonceni programu po stisku ESC

Simulace řídicího skriptu



Dotaz na ukončení simulace

Proces lze kdykoli přerušit stiskem klávesy **ESC**, který vyvolá dialog, kterým je třeba přerušení potvrdit. Přerušovaný proces není možné znovu obnovit od pozice, kde byl přerušen, ale pokaždé se simulace spustí od začátku.

1.11. SKRIPT STORAGE – INTERNÍ ÚLOŽIŠTĚ SKRIPTŮ

Funkce Skript Storage je funkce úložiště, do kterého lze vložit až 20 existujících skriptů. Úložiště se ovládá z hlavního menu. Mezi hlavní funkce úložiště patří vkládání skriptů, spouštění simulace běhu skriptu přímo z výběrového menu, uložení skriptu na disk, mazání skriptů z úložiště. Celý obsah úložiště je fyzicky na disku uložen v souboru **FILESTOR.FFS**, kde jsou jednotlivé



Zobrazení obsahu úložiště v menu



Zobrazení obsahu úložiště v menu – spuštění simulace

skripty uloženy v komprimované podobě. Pravidelnou zálohou tohoto souboru pak zajistíte, že obsah celého úložiště bude zabezpečen před ztrátou.

Soubor **FILESTOR.FFS**, lze také editovat pomocí programu pro příkazový řádek **FileStor.EXE**. Tento program umožňuje přidávat, mazat, nahrazovat, extrahovat skripty a zobrazovat obsah úložiště a tento seznam ukládat do souboru **FileStor.TXT**.

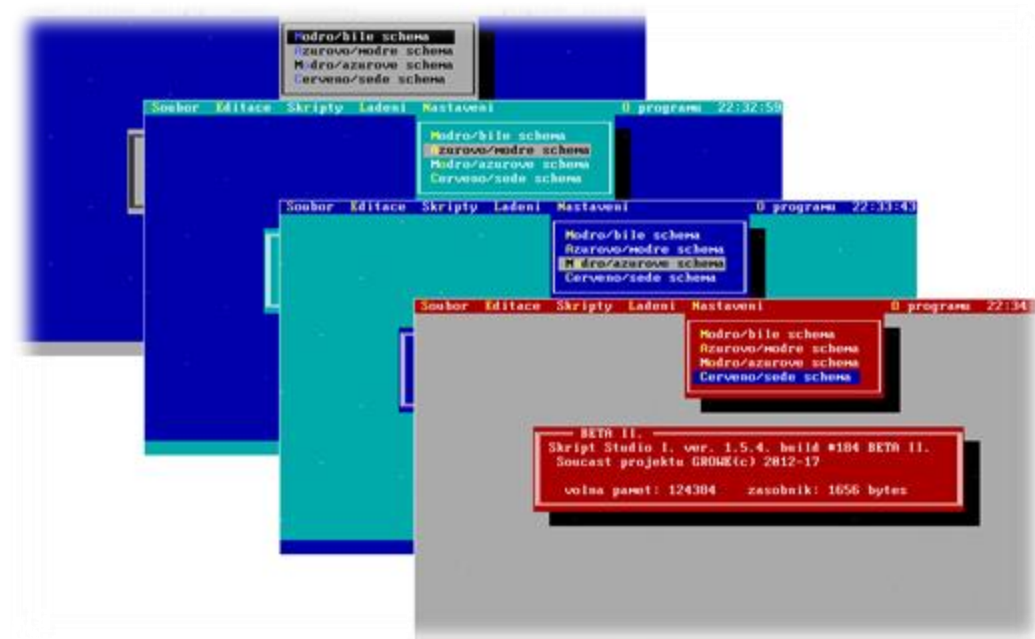
Tento program je součástí instalace **Skript Studio I.**, nebo jej můžete stáhnout z WEBu: www.projekt-growe.cz, sekce **Ke stažení**.

1.11.1. PŘÍMÉ SPOUŠTĚNÍ SIMULACE Z INTERNÍHO ÚLOŽIŠTĚ

Proces simulace řídicího procesu je shodný s procesem popsaným v kapitole **1.10. Simulace řídicího procesu pomocí skriptu**, s tím, že skripty uložené v interním úložišti lze takto přímo spouštět bez nutnosti extrahování souboru na disk, poté jeho otevření a spuštění v procesu simulace. Po spuštění simulace, běží tato od začátku až po konec skriptu. Ukončení je oznámeno zvukovým signálem a hlášením o výsledku operace. Stiskem klávesy se poté navrátíte do hlavního menu.

1.12. NASTAVENÍ A KONFIGURACE PROSTŘEDÍ

1.12.1. NASTAVENÍ - BAREVNÁ SCHÉMATA



Barevná schémata prostředí

Ve volbě **Nastavení** lze snadno změnit barevné schema prostředí. Tato volba se zapamatuje a při dalším spuštění se bude automaticky zobrazovat, až do příští změny.

1.12.2. KONFIGURACE PROGRAMU

Konfiguraci prostředí je uložena v souboru **STUDIO.INI**, ze kterého je čtena při spuštění programu. Formát souboru je dle specifikace **Microsoft (Win 3.11) .INI** FileType.

POPIS STRUKTURY *.INI SOUBORU:

.INI soubor se skládá ze sekcí a klíčů. Název sekce v ***.INI** souboru je vždy uveden v hranatých závorkách. Každá sekce potom obsahuje několik klíčů (klíč musí být vždy přiřazen k sekci, která je uložena v souboru před tímto klíčem). Obsah jedné sekce končí označením další sekce. Název klíče je vždy na samostatném řádku a znakem = (rovná se) a má přiřazen obsah. Obsah konfiguračního souboru je **CaSe SeNsItIvE**, což znamená, že jsou rozlišována **malá** a **VELKÁ** písmena. Identifikátor je ve tvaru **VELKÝCH** písmen **INI** v hranatých závorkách, název sekce obsahuje první písmeno **VELKÉ**, ostatní malá, název klíče je shodné s maďarskou notací označování názvů proměnných, což je název, složený z několika slov, vyjadřující význam nesoucí informace, s prvním písmenem **VELKÝM**, ostatní malými písmeny a bez mezer.

PŘÍKLAD SOUBORU .INI:

```
[INI]
[Files]
OneFile=True
Pi=3.14
[Item]
User=Harry
```

VÝKLAD OBSAHU:

V příkladu souboru **.INI** jsou tedy obsahuje:

- návěstí **INI**, používá se jako jednoznačný identifikátor konfiguračního souboru, pokud není nalezen, je obsah celého souboru odmítnut
- sekce **Files**, obsahuje klíč **OneFile**, jehož obsahem je "**True**" a klíč **Pi**, jehož obsahem je hodnota "**3.14**"
- sekce **Item** obsahuje klíč **User** s obsahem "**Harry**"

Pozn.:

Výstupem čtení klíče je vždy textová (**STRING**) hodnota, a to i v případě, že obsahem je číslo.

Konfigurační soubor **STUDIO.INI** obsahuje několik sekcí. V první sekci **Configuration** jsou klíče pro aktuální barevné schema, nastavení délky nečinnosti před spuštěním spořiče obrazovky a počet hvězd zobrazovaných během jeho běhu (má vliv na rychlost animace spořiče), klíč **RestrictDataBaseType** obsahující pravidlo pro akceptování nebo odmítnutí skriptu, který nemá tabulku databáze shodnou dle specifikace GROWE Script Specification **2.4. Struktura skriptu**. Klíč **VersionINI** je pouze informativní, označuje verzi **.INI** souboru, pro identifikaci verze konfigurace, při vývoji programu. Další sekce popisují kompletní funkcionalitu **MENU Systému** a nedoporučuje se, jejich hodnoty měnit. Sekce **ColorScheme** obsahuje barvy pro jednotlivá barevná schémata **MENU Systemu**, sekce **Validation** maximální počet počet ovládaných spotřebičů v **Procesu GROWE** (max. hodnota je 8, s použitím hardwarového modulu ver. 1.0 je max. hodnota 7), hodnota **PriceKW** obsahuje cenu el. energie 1 KW v Kč. Sekce **Electricity_consumtion** obsahuje odběr jednotlivých spotřebičů **Device_1** až **8** ve **W (Watt)**.

Konfigurační soubor **BUILDER.INI** obsahuje pravidla pro chování editoru skriptu, klíč **AutoUpperCase** obsahuje hodnotu pro povolení/zakázání pravidla, kdy vložené příkazy do editoru jsou automaticky převáděny na **VELKÁ** písmena.

Konfigurační soubor **RELDESK.INI** pak obsahuje názvy jednotlivých spotřebičů, použitých ve funkci simulace běhu řízení. Tento soubor je shodný se souborem, který používá hlavní řídicí program **MINI I.** pro pojmenování svých spotřebičů, zobrazovaných na hlavní obrazovce.

1.12.3. OBNOVA KONFIGURAČNÍCH SOUBORŮ

Pokud je konfigurační soubor pozměněn do té míry, kdy program **Skript Studio I.** při spuštění a načítání neporozumí údajům konfigurace, je načítání přerušeno chybovým hlášením a program se nespustí. Pro tento případ obsahuje program opravnou funkci, kdy dojde k přepsu konfiguračního souboru **SSTUDIO.INI** výchozími hodnotami. Program se spustí s argumentem **/REPAIR**, příkaz pro spuštění je pak ve formátu **SSTUDIO.EXE /REPAIR**. Program se spustí nouzovém módu, nastaví výchozí hodnoty konfigurace a program spustí v normal módu.

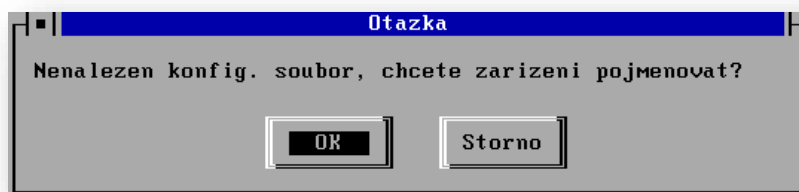


Obnova konfiguračních souborů

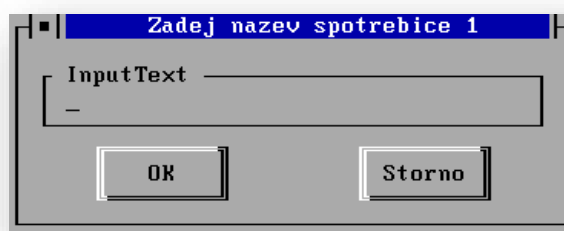
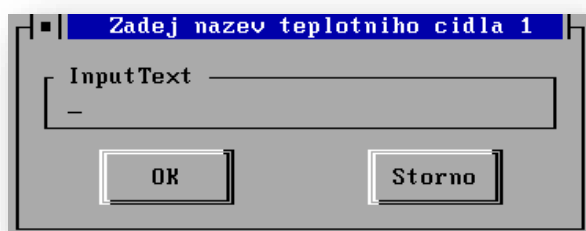
Pozn.:

Pokud i po spuštění funkce obnovy konfiguračních souborů se nezdaří spuštění programu Skript Studio I., zkontrolujte, zda máte přístup k zápisu do souboru konfigurace SSTUDIO.INI, zda máte na disku dostatečný prostor pro vytvoření tohoto souboru, nebo zda soubor nevyužívá jiný běžící proces (např. editor).

Pokud je spuštěna funkce **Nový skript**, nebo **Editace skriptu** a není v aktivním adresáři soubor **RELDESK.INI**, obsahující názvy teplotních čidel a spotřebičů, je uživatel vyzván k jejich pojmenování.



Dotaz pro pojmenování teplotních čidel a spotřebičů



Dialogy pro pojmenování teplotních čidel a spotřebičů

Následně, pokud zvolí volbu pojmenovat, bude postupně vyzván k pojmenování čidel 1 a 2 a spotřebičů 1 až 8. Pokud je pojmenování odmítnuto, provede se jejich automatické pojmenování.

1.13. NÁPOVĚDA A ELEKTRONICKÝ MANUÁL

Program **Skript Studio I.** má implementovanou funkci interaktivní nápovědy, tedy téma nápovědy se přizpůsobuje tomu, ve které části programu se nacházíte. Tato nápověda se spouští klávesou **F1**. Kromě této nápovědy je z **Menu** systému dostupný elektronický manuál, a to, stručný návod v textovém formátu, obsahující popis procedur a popis vývoje programu a podrobný manuál popisující jednotlivé funkce programu podrobně.

Pokud je program **Skript Studio I.** volá

```

Radek: 1   Soubor: Manual.txt   Konec: ESC   Pohyb: ↑↓ PgUp PgDn Home End
Manual k programu Skript Studio I.
komplexní nástroj pro správu řídicích skriptů, .DBF/.DBG
Urceno pro program Řízení MINI(c)
Součást projektu GROWE(c) 2012-17
WEB: www.projekt-growe.cz   E-mail: support@projekt.growe.cz
ver. 0.0.1. ALFA
Program slouží ke tvorbě, editaci, testování, ladění, ukládání a dalším
operacím pro správu řídicího skriptu .DBF/.DBG

Obsah:

```

Zobrazení obsahu nápovědy

s argumentem **/VER**, tedy **SSTUDIO.EXE /VER**, jsou na obrazovku vypsány názvy jednotlivých výkonných modulů, verzí a statusu vývoje.

Kompletní manuál je přiložen v souboru .PDF, který je součástí instalace, nebo je v [On-line podobě](#) na Webu Projektu GROWE.

```

Skript Studio I., ver. 1.5.4. build #184 BETA II. .... [OK]
3D Stars ScreenSaver, ver. 1.0.5. RC-II. .... [OK]
Box, ver. 2.0.2. BETA .... [OK]
ClearKeybBuffer, ver. 1.0.1. BETA .... [OK]
CopyFile, ver. 1.0.6. BETA .... [OK]
CRC32, ver. 1.0.4. BETA I. .... [OK]
DB Access, ver. 5.0.6. ALFA .... [OK]
DBG_READ, ver. 3.9.5. build #270 BETA XU. .... [OK]
Skript Builder Pro II., ver. 1.1.6. ALFA III. .... [OK]
ERROR_Code, ver. 0.3.2. ALFA IV. .... [OK]
FileExist, ver. 2.0.3. RC-I .... [OK]
.INIConfig I/O, ver. 2.0.3. BETA VII. .... [OK]
Keyboard Flags Service, ver. 1.0.3. CR-I. .... [OK]
Menu System, ver. 2.0.4. ALFA III. .... [OK]
PathName, ver. 1.0.3. RC-I .... [OK]
STButton, ver. 2.0.1. BETA .... [OK]
TRIM, ver. 1.1.2. RC-I. .... [OK]
ViewFile, ver. 1.0.8. BETA II. .... [OK]
WaitState, ver. 2.0.4. BETA .... [OK]
File Storage System, ver. 1.1.4. BETA II. .... [OK]

Kód je validní dle [Valid_G-3.0.5]
Pokracovani po stisku klavesy.

```

Zobrazení výkonných modulů

```

Radek: 1   Soubor: ver_info.txt   Konec: ESC   Pohyb: ↑↓ PgUp PgDn Home End
Program Skript Studio I.
komplexní nástroj pro správu řídicích skriptů, .DBF/.DBG
Urceno pro program Řízení MINI(c)
Součást projektu GROWE(c) 2012-17
WEB: www.projekt-growe.cz   E-mail: support@projekt.growe.cz

[posledni modifikace 17.05.2017 19:41]

HISTORIE VERZI:
■Ver. 1.5.4. BETA II.
Dokonceni vyvoje a zakladniho BETA testu. Prechod do faze BETA II., bude dale
provadeno pouze ladeni a tvorba el. manualu v .TXT, .XDOC a .PDF formatu.
Kod je validni dle formatu [Valid_G-3.0.6.1]
■Ver. 1.5.3. BETA I.
Prepracovan modul Builder pro zmenu tvorby skriptu - tvorba a editace v poli,
teprve finalni podoba je ulozena do souboru, pridana ExtendHead a DiGiSing
Dopracovan modul DBG_Read, nyni pocita vsechny chyby behem vseh testu,
uklada do souboru vysledku, vsetne napovednych textu.
Uyladeni chyb, aktualizace modulu, redukce deklaraci.
■Ver. 1.5.2. BETA I.
Prepracovan zpusob kompilace, nyni je tvorena pomocí davkového souboru, IDE
prostředí BU-DOS není možné použít z důvodu nedostaku pameti. Pridana funkce
automaticke inkrementace cisla buildu a aktualizace datumu pri kompilaci.

```

Zobrazení historie vývoje

2. ŘÍDÍCÍ SKRIPT

2.1. POPIS SKRIPTU

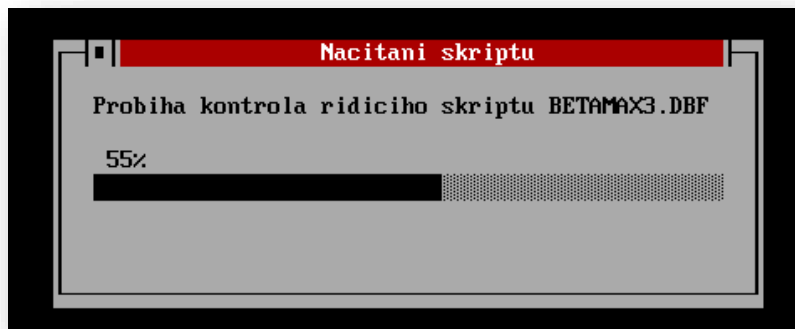
Řídící skript je soubor příkazů, kterými se program řídí po celou dobu automatického běhu procesu **GROWE**. Při spuštění programu se načtou příkazy pro aktuální den a program je provádí až do půlnoci, kdy jsou načteny nové, aktuální pro další den.



Náhled na textovou podobu řídicího skriptu

2.2. POUŽITÍ ŘÍDÍCÍHO SKRIPTU

Řídící příkazy jsou uloženy v databázovém souboru **.DBG** ve formátu **GBase**, který vychází z formátu **dBase ver. 3.xx** s modifikovanou hlavičkou souboru tzv. **eXtHead**. Tato obsahuje dodatečné informace, rozšiřující možnosti tohoto formátu, ale zároveň způsobuje zpětnou nekompatibilitu s původním formátem.



Načítání a kontrola řídicího skriptu při startu programu Řízení MINI

Upozornění:

Skript (soubor databáze) není z důvodu přidáním **eXtHead** plně kompatibilní s formáty **dBase ver. 3.xx, 4.xx**. V případě použití standardních nástrojů pro editaci **.DBF** formátu, může dojít k jejímu přepsání, nebo změně obsahu.

2.3. HISTORIE FORMÁTU .DBF ²

dBase byl první masově rozšířený systém řízení báze dat (**SŘBD**, **DBMS**) pro mikropočítače, který byl publikován americkou firmou **Ashton-Tate** pro **CP/M**. A později pro **Apple II**, **Apple Macintosh**, **UNIX**[1], **VMS**[2] a **IBM PC** pod **DOS**, kde se na mnoho let stal jedním z nejprodávanejších softwarových titulů. **dBase** pomalu úspěšně přecházel pod **Microsoft Windows** a pozvolna uvolňoval podíl na trhu svým konkurentům **Paradox**, **Clipper**, **FlagShip**, **FoxPro** a **Microsoft Access**. **dBase** byl prodán **Borlandu** v roce 1991, který prodal práva k této sortimentní skupině v roce 1999 tehdy nově vzniklé společnosti **dBase Inc.** V roce 2004 **dBase Inc.** změnila svůj název na **dataBased Intelligence, Inc.**

2.3.1. VZNIK DBASE

Původním vývojářem **dBase** byl **Wayne Ratliff**. V roce 1978 **Ratliff** napsal databázový program v assembleru pro mikropočítače založené na **CP/M**, aby mu pomohl vyhrát kancelářskou fotbalovou sázku. Založil ho na **JPLDIS (Jet Propulsion Laboratory Display Information System)** Jebe Longa a nazval ho **Vulcan** po panu Spockovi ze seriálu **Star Trek**. Podle **Ratliffa** byl jazyk použitý v **JPLDIS** jednoduchý. Jsou zde známky toho, že **JPLDIS** byl ovlivněn mainframovským databázovým produktem zvaným **RETRIEVE**.

Na počátku 80. let 20. století, **George Tate** z **Ashton-Tate** uzavřel s **Ratliffem** obchodní dohodu. **Vulcan** se přejmenoval na **dBase** a software rychle vzrostl na popularitě.

2.3.2. DBASE JAKO OBJEKTIVĚ ORIENTOVANÝ JAZYK

dBase byl posléze zařazen mezi moderní objektově orientované jazyky, které běží na 32 bitových **Windows**. Může být použit k výstavbě aplikací se širokým využitím, např. webové aplikace provozované na **Windows** serveru. **dBase** má přístup k většině moderních databází pomocí rozhraní **ODBC**.

2.3.3. XBASE

Počátkem poloviny osmdesátých let 20. století mnoho jiných společností přišlo s vlastními variantami daného produktu a jazyka. Tyto zahrnovaly **FoxPro** (nyní **Visual FoxPro**), **Arago**, **Force**, **dbFast**, **dbXL**, **Quicksilver**, **Clipper**, **Xbase++**, **FlagShip**, **Recital Terminal Developer**, **CodeBase**, **MultiBase** a **Harbour/XHarbour**. Dohromady jsou označovány jako **xBase**.

2.3.4. SOUBOR DATABÁZE – OBECNÝ FORMÁT .DBF

Výchozím formátem pro **dBase** je soubor **.DBF**. Ten se hojně využívá v mnoha dalších aplikacích, kde je potřeba jednoduchý formát pro uchování strukturovaných dat. V současné době mu ale v tomto ohledu konkuruje **SQLite**.

Pro každou tabulku byl jeden soubor **.DBF**. Extra soubory **dBase** vytvářela pro sloupce typu memo (v souboru **.DBT**; viz dále) a pro index (**.IDX**) nebo indexy (**.MDX**). Organizace tabulek do databází na úrovni **SŘBD** neexistovala, stejně jako např. pohledy, cizí klíče, triggerů a jiné prvky pozdějších databázových systémů. Přístup více uživatelů řešily jednotlivé aplikace na úrovni operačního systému. Přístupové údaje pro přístup k tabulkám chyběly.

Jméno tabulky bylo určeno jménem souboru. Soubor na začátku obsahoval definici sloupců, následovanou jednotlivými řádky. Sloupce mohly mít název dlouhý maximálně 11 znaků a mohly být těchto typů (první písmeno je kód typu, používaný v souboru):

- C** – řetězec (v **SQL** typ **char**) se specifikovanou délkou (maximálně 255 znaků)
- L** – logická hodnota (s hodnotami **Y** nebo **T** pro pravdu, **N** nebo **F** pro nepravdu)
- N** – číslo, u kterého lze specifikovat maximální počet číslic
- F** – číslo, u kterého lze specifikovat maximální počet desetinných míst
- D** – datum (v **SQL** typ **date**) od roku 0000 do roku 9999

² V článku byly použity citace z internetové encyklopedie **Wikipedie**, při zachování licence **Creative Commons 3.0 Unported** a případných dalších podmínek. Zdroj: <https://cs.wikipedia.org/wiki/DBase>

M – memo (v **SQL** typ text) text do délky 64 Kb

Řádky měly stejnou délku (kratší řetězce se doplňovaly \0 zprava a čísla nulami zleva), takže šly rychle procházet. Každý záznam obsahoval indikátor platnosti. Mazání záznamů sestávalo samotné indikace neplatnosti. Tím ale vznikala fragmentace. Pro optimalizaci velikosti se tedy dal soubor zabalit/spakovat (program v něm prošel všechny záznamy a ty platné záznamy srotil k začátku souboru a zbytek souboru „usekl“).

Obsah souborů **.DBT** je rozsekaný na základní bloky o délce 512 bajtů. Tyto bloky jsou interně číslovány (počínaje nulou) a v souboru **.dbf** je u sloupců typu memo uloženo, kde (na bloku jakého čísla) začíná odpovídající text. Text je vždy doplněný na nejbližší násobek čísla 512.

2.3.5. VÝVOJ A SOUČASNOST

Vývoj formátu pokračoval pod taktovkou **Microsoftu** až do 2007, kdy byl vydán, jako součást programu **Visual FoxPro 9.0 SP2** – objektový nástroj k vytvoření a správě vysoce výkonných 32bitových databázových aplikací a komponent. Jeho další vývoj byl následně ukončen.

2.4. STRUKTURA SKRIPTU

Délka všech polí ve větě databáze je **VŽDY** stejná, i když neobsahuje žádnou hodnotu, proto jsou i jednotlivé věty databáze stejně dlouhé. Jsou to tzv. sekvence, odtud název sekvenční databáze. Jednotlivé věty jsou skládány jedna za druhou, proto je snadné načítání jednotlivých vět a dle délky pole, definované v tabulce rozsekat větu na jednotlivá pole.

PŘÍKLAD VĚTY STRUKTURY DATABÁZE:

Hlavička databáze 32 bytes (16 bytes dBase def. + 16 bytes eXtHead def.)									
DATUM	DAY	RELE_1	RELE_2	RELE_3	RELE_4	RELE_5	RELE_6	RELE_7	RELE_8
Typ D	Typ C	Typ C	Typ C	Typ C	Typ C	Typ C	Typ C	Typ C	Typ C
Pozice	Pozice	Pozice	Pozice	Pozice	Pozice	Pozice	Pozice	Pozice	Pozice
2	10	16	37	58	79	100	121	142	163
+	+	+	+	+	+	+	+	+	+
Délka	Délka	Délka	Délka	Délka	Délka	Délka	Délka	Délka	Délka
8 =	6 =	21 =	21 =	21 =	21 =	21 =	21 =	21 =	21 =

celková délka věty

Náhled na hlavičku souboru databáze odhalí fyzické uložení. Jak bylo již uvedeno, hlavička je dlouhá 32 bytes, z toho je 16 bytes **dBase** základní definice + 16 bytes **eXtHead**. Za ní je uložena tabulka databáze. Je definována proměnnými, kterým je přiděleno pořadové číslo, název, typ, offset a délka.

2.5. ROZŠÍŘENÁ HLAVIČKA SOUBORU .DBF VER. 4.X.X.

Řídící skript obsahuje hlavičku, což je prostor na začátku souboru, kde prvních 32 bytes souboru obsahuje konfigurační informace o skriptu.

```
00000000: 03 0F 0A 1A 6F 00 00 00|61 01 6C 00 00 00 00 00 | ....o...a.1.....
00000010: 04 00 00 00 00 00 46 45|31 35 42 46 43 32 00 20 | .....FE15BFC2.
```

Hlavička souboru

HLAVIČKA DATABÁZE DLE SPECIFIKACE DBASE (PŘEVZATO ZE SPECIFIKACE FORMÁTU):

Pozice	obsah
1	dBASE verze (3, nebo &H83 jestli je soubor typu memo)
2	Rok
3	Měsíc
4	Den posledního update
5-8	Celkový počet záznamů v souboru (long integer)
9-10	Počet záznamů, uložených v hlavičce (integer)
11-12	Délka záznamu v souboru (integer)

Hlavička skriptu	pozice	délka	typ proměnné	popis
Header.Version = ASC(Buffer\$)	1	1 byte	Version AS INTEGER	Verze skriptu .DBF
Header.Rok = ASC(MID\$(Buffer\$, 2, 1))	2	1 byte	Year AS INTEGER	Datum souboru rok, formát rr
Header.Mesic = ASC(MID\$(Buffer\$, 3, 1))	4	1 byte	Day AS INTEGER	Datum souboru den, formát dd
Header.TRecs = CVL(MID\$(Buffer\$, 5, 4))	5-8	4 bytes	TRecs AS LONG	Počet vet v databázi
NEPOUZITO	9-10	2 bytes	TFields AS INTEGER	Počet polí ve větě
Header.Reclen = CVI(MID\$(Buffer\$, 11, 2))	11-12	2 bytes	RecLen AS INTEGER	Délka vety
REZERVOVÁNO pro zpětnou kompatibilitu	13-15	3 bytes	Memo AS INTEGER	Příznak není využit
Header.ExtendHEAD = MID\$(Buffer\$, 17, 16)	16-32	16 bytes	ExtendHead AS STRING*16	Rozšířená hlavička

Rozšířená hlavička **Extend Head** (eXtHead), obsahující dodatečné a rozšiřující informace je součástí základní hlavičky. Využívá nepoužívaných 16-ti bytes ze spodní části hlavičky.

Pozice	obsah
1	dBASE verze 3, nebo &H83 typ memo
2	Rok
3	Měsíc
4	Den posledního update
5-8	Celkový počet záznamů v souboru (long integer)
9-10	Počet záznamů, uložených v hlavičce (integer) ³
11-12	Délka záznamu v souboru (integer)
13-32	Rozšířená hlavička - Extend Head

Hlavička včetně tabulky databáze

³ Číslo vyjadřující počet záznamů (vět) v databázi, je uloženo v hlavičce a v určitých případech se může lišit od fyzického počtu v databázi. Proto na tento údaj není brán zřetel a vždy se počítá fyzický počet záznamů.

Rozšířená hlavička
Extend Head

00000000:	03 0F 0A 1A 6F 00 00 00	61 01 6C 00 00 00 00 00	o...a.l....
00000010:	04 00 00 00 00 00 46 45	31 35 42 46 43 32 00 20	FE15BFC2.
00000020:	44 41 54 55 4D 00 00 00	00 00 00 44 00 00 00 00	DATUM.....D....
00000030:	08 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000040:	44 41 59 00 00 00 00 00	00 00 00 43 00 00 00 00	DAY.....C....
00000050:	06 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000060:	52 45 4C 45 5F 31 00 00	00 00 00 43 00 00 00 00	RELE_1....C....
00000070:	03 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000080:	52 45 4C 45 5F 32 00 00	00 00 00 43 00 00 00 00	RELE_2....C....
00000090:	1D 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000A0:	52 45 4C 45 5F 33 00 00	00 00 00 43 00 00 00 00	RELE_3....C....
000000B0:	1D 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000C0:	52 45 4C 45 5F 34 00 00	00 00 00 43 00 00 00 00	RELE_4....C....
000000D0:	03 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
000000E0:	52 45 4C 45 5F 35 00 00	00 00 00 43 00 00 00 00	RELE_5....C....
000000F0:	04 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000100:	52 45 4C 45 5F 36 00 00	00 00 00 43 00 00 00 00	RELE_6....C....
00000110:	03 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000120:	52 45 4C 45 5F 37 00 00	00 00 00 43 00 00 00 00	RELE_7....C....
00000130:	13 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000140:	52 45 4C 45 5F 38 00 00	00 00 00 43 00 00 00 00	RELE_8....C....
00000150:	03 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	
00000160:	0D 20		.

POPIS ROZŠÍŘENÉ HLAVIČKY:

Hodnoty v hlavičce jsou rozděleny na příznaky a r_znaky (řídící znaky). Příznaky určují vlastnosti databáze, zatímco řídící (r_znaky) přímo mění vlastnosti databáze (např. v tomto případě zamezují zápis na úrovni modulu **DBAccess**).

1. příznak	verze databáze volba CHR\$(3) jiný příznak	- příznak pro formát verze 3.00 - NENÍ AKCEPTOVÁN
2. r_znak	ochrana proti zápisu do databáze volba CHR\$(240) jiná hodnota	- zápis ZAKAZAN - zápis POVOLEN
3. příznak	kryptování obsah databáze volba CHR\$(1) jiná hodnota	- obsah JE zakrytován - obsah NENÍ zakrytován
4. příznak	inicializace skriptu, volba CHR\$(1) jiná hodnota	- skript JE inicializován - skript NENÍ inicializován
5. příznak	originalita skriptu, volba CHR\$(0) volba +=1	- originální skript - zaměněný skript (hodnota zn. pořadí, např. záměna dálkovou správou)
6. příznak	hodnota CHR\$(1) jiná hodnota	- použití skriptu v procesu RIZENI - proces GROWE spuštěn/restartován - skript dosud nepoužit
7. příznak	ukončení procesu GROWE volba CHR\$(0) volba CHR\$ +=1	- příznak ověřeného skriptu - skript dosud nepoužit - počet dokončených cyklů
8.-15. znak	8. Bytes CRC	- kontrolní součet
16. znak	G A	- databáze GROWE - (any) všeobecná databáze

Příklad *eXtendHead*:

04	00	00	00	00	00	46	45	31	35	42	46	43	32	00	20
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

POZN.:

Při kontrole platnosti je testován pouze obsah databáze. Může tedy docházet ke změnám v hlavičce bez zneplatnění podpisu.

2.6. NASTAVENÍ OCHRANY ZÁPISU SKRIPTU

Do skriptu lze přidat příznak **POUZE ČTENÍ**, který aktivně zamezuje jakémukoli zápisu do obsahu. Tato ochrana je aktivní pouze u programů projektu **GROWE**, kde je tato ochrana na úrovni fyzického přístupu k souboru databáze. Toto však neplatí pro editaci na úrovni **OS DOS/Windows 98**, kdy provedení fyzických změn v souboru prakticky není možné zamezit. Pro ochranu obsahu proto slouží digitální podpis, kdy jakákoli změna v datech jej zneplatní společně s příznakem pro zákaz zápisu. Jediná možnost, jak provést změnu v datech a vložit platný podpis, je použít některý z programů projektu **GROWE**, které ale akceptují příznak **POUZE ČTENÍ** a provedení změn tedy neumožní. Kombinace ochrany obsahu skriptu je dostatečná pro zamezení použití náhodně přepsaného obsahu skriptu, pro který je i určena.

Header.ReadOnly	2.byte	CHR\$(240)	- aktivní příznak pouze čtení
	jiný znak		- zápis povolen

2.6.1. PROGRAM UNLOCKSC.EXE

Program UnLockSc.EXE (Unlock Script) slouží nastavení nebo odstranění příznaku Read Only (pouze čtení) popsaneého v předchozí kapitole.

```

Program UnLock Script, ver. 1.0.1. Beta IV.
Soucasti projektu GROWE(c) 2012-18

Slouzi k rychlemu nastaveni/odstraneni priznaku ReadOnly v ridicim skriptu.
Napoveda k programu UnLock Script:

volani programu:
UnLOCKSC.EXE <FileName.DBG> </LOCK> nebo </UNLOCK>

argumenty uzavrene v zavorkach <> jsou povinne !

Program de/aktivuje priznak ReadOnly ve trech krocich:
1. Cte stav priznaku
2. Zmeni priznak na pozadovany
3. Opetovne cte stav priznaku

Pouzijte prikaz UNLOCKSC.EXE /? nebo /HELP pro zobrazeni napovedy.
Konec programu

```

Program UnLockSc.EXE

3.1. INSTALACE

Instalace je operace, kdy se provede zkopírování všech potřebných souborů do příslušného adresáře a nastaví se výchozí konfigurační soubory. Po stažení instalačního programu v komprimované podobě **WinZIP**, který naleznete na Webu viz. kapitola [5. Adresy a kontakty](#), se celý obsah „rozbalí“ do dočasného adresáře. Poté instalaci spustíte příkazem **Install.EXE**, která vás provede celou operací po jednotlivých krocích.

```

Radek: 1   Soubor: licence.txt   Konec: ESC   Pohyb: ↑↓ PgUp PgDn Home End
Licence a autorska prava

1.Autorska prava
Autorska prava, pokud nejsou vyslovne uvedena jinak, nalezi spolecnosti
Projekt GROWE(c). Autorska prava nalezejici k programovym modulum, ktere
jsou pouzity, jsou ve zdrojovych kodech uvedeny, vcetne jejich vlastniku.

2.Licencni ujednani
Toto licencni ujednani je pravni smlouvou mezi Vami, koncovym spotrebitelem
(dale jen uzivatele) softwaroveho produktu Skript Studio I. dale oznacovan
jako "produkt" a spolecnosti Projekt GROWE(c)
Produkt jako takovy se neprodava, pouze se udeluje licence k jeho uzivani.
Instalaci a pouzivanim produktu uzivatele potvrzuje, ze souhlasí s podmínkami
tohoto ujednani a zavazuje se jimi ridit.

Na zaklade teto licencni smlouvy se udeluji nasledujici prava:

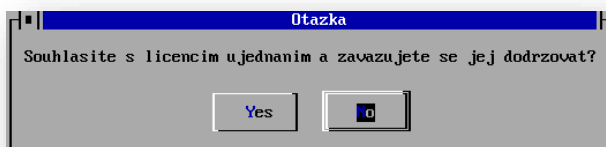
Produkt je poskytnut uzivateli ZDARMA, pocet instalaci, nebo kopii neni nijak
omezen. Muze jej bez omezeni sirit, pokud dodrzi pravidlo, ze program bude
siren jako celek, vcetne podpurnych, konfiguračních a dalsich souboru, uplne
a nepozmenene dokumentace a elektronickych manualu.

Popis dalsich prav a omezeni:

```

Licenční ustanovení

Jako první krok před započítím instalace je zobrazení **licenčního ujednání**. Po přečtení a stisku **ESC** se zobrazí dotaz na jejich dodržování. Pokud s licenčním ujednáním souhlasíte, zvolte odpověď ANO/YES, instalační proces bude pokračovat. V opačném případě se instalace ukončí.



Dotaz na souhlas s licenčním ujednáním

```

Cesta: C:\GROWE\RIZENI\UTIL
Dekomprese sstudio.ex# -> SSTUDIO.EXE      ulozeno 26845 Bytes      [OK]
Instalace souboru SSTUDIO.EXE do C:\GROWE\RIZENI\UTIL
Prerusit instalaci (Ano/Ne): _

```

Přerušování instalačního procesu

```

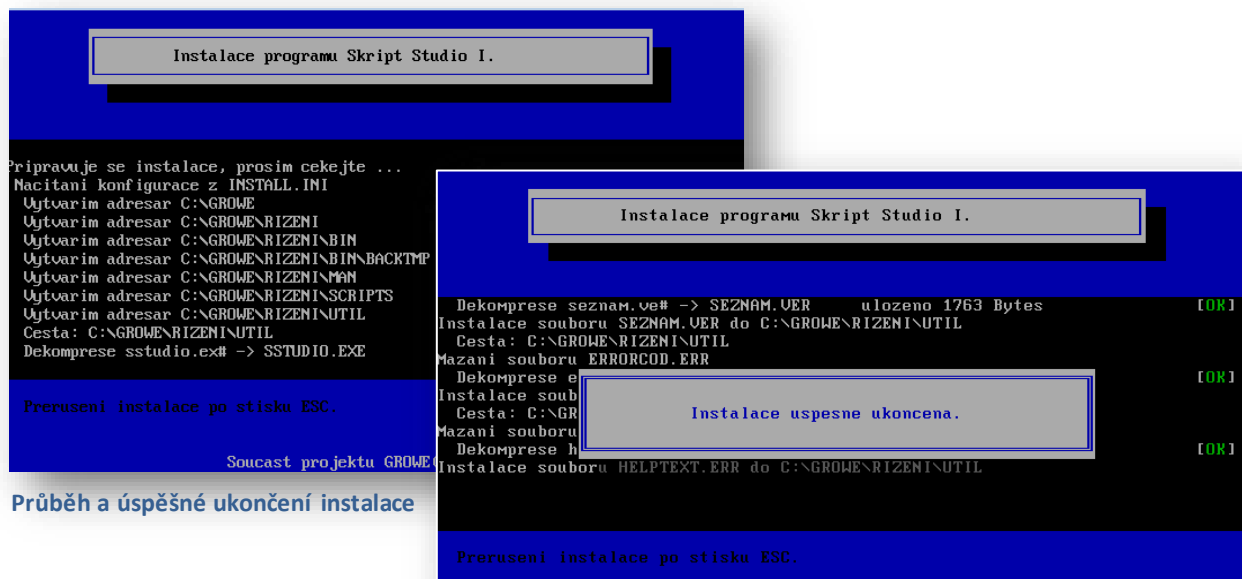
Instalace souboru SSTUDIO.EXE do C:\GROWE\RIZENI\UTIL
Prerusit instalaci (Ano/Ne): a
Instalace prerusena na zadost uzivatele.

C:\TEMP\SSTUDIO>_

```

SEZNAM INSTALAČNÍCH SOUBORŮ:

NÁZEV SOUBORU	VELIKOST	TYP	NÁZEV INSTALOVANÉHO SOUBORU
INSTALL.EXE	150 591	APLIKACE	-
INSTALL.INI	1 650	KONFIGURAČNÍ SOUBOR	-
ERRORCOD.ER#	3 324	SOUBOR INSTALACE	ERRORCOD.ERR
HELPTXT.ER#	11 224	SOUBOR INSTALACE	HELPTXT.ERR
FILESTOR.FS#	977	SOUBOR INSTALACE	FILESTOR.FSS
BUILDER.IN#	68	SOUBOR INSTALACE	BUILDER.INI
RELDESK.IN#	387	SOUBOR INSTALACE	RELDESK.INI
SSTUDIO.IN#	3 708	SOUBOR INSTALACE	SSTUDIO.INI
MANUAL.TX#	623	SOUBOR INSTALACE	MANUAL.TXT
SEZNAM.VE#	20 281	SOUBOR INSTALACE	SEZNAM.VER



Průběh a úspěšné ukončení instalace

POZN.:

Seznam souborů se může lišit od obsahu aktuálního instalačního balíčku, jak jsou přidávány podpůrné programy, manuály a další podpůrné soubory.

3.2. AKTUALIZACE PROGRAMŮ

Aktualizace programů se provádí jako manuální zásah uživatele, neprobíhá tedy automaticky. Aktualizační balíčky jsou k dispozici na Webu projektu. K aktualizacím souborům je přiložen podrobný popis aktualizace v textové formě. Instalační a aktualizací balíčky jsou v komprimované podobě ve formátu **WinZIP**.

Jako aktualizace **nejvyšší priority** jsou ty, které opravují chyby nebo dis-funkcionality hlavního nebo pomocných programů. Důrazně se doporučuje je nainstalovat, jsou označeny písmeny **RC** (Release Candidate) v kombinaci s číslicí, např.: Skript Studio I. ver. 1.5.7. **RC-1**.

Další v žebříčku jsou aktualizace **střední priority**, přinášející nová rozšíření a funkce, tzv. **BETA** verze. U nich byl vývoj nových funkcionalit již ukončen a nyní probíhá jejich testování. V případě, že budete tyto aktualizace testovat, je potřeba se držet návodu, který je součástí aktualizací balíčku. Je pravděpodobné, že tyto novinky budou s sebou přinášet i změny v klíčových částech programu. Doporučuje se, provést úplnou zálohu důležitých souborů, která je popsána v kapitole **3.3. Zálohování**

Nejnižší prioritu, mají aktualizace, přinášející nové funkce **ve fázi vývoje**, označeny jako **ALFA**. Jsou určeny pouze pro testovací účely, funkcionality, které obsahují, mohou být nasazeny pouze experimentálně, ve finální verzi se mohou objevit v pozměněné podobě, nebo mohou být vypuštěny úplně. Nedoporučuje se, tyto programy používat v ostrém provozu, nebo v nich upravovat hotové klíčové skripty. Pokud přesto chcete otestovat nové funkcionality, nezapomeňte si provést zálohu těchto souborů.

3.3. ZÁLOHOVÁNÍ

Pro předejítí ztráty důležitých souborů, je potřeba provádět jejich pravidelnou zálohu. Zvláště důležité jsou soubory řídících skriptů **.DBG**, soubor úložiště skriptů **FILESTOR.FSS**, soubory výsledků testu skriptu **TESTSCR.TXT**, výsledky kalkulace ceny spotřeby el. energie **VYPIS.TXT**

4. LICENCE A AUTORSKÁ PRÁVA

4.1. AUTORSKÁ PRÁVA

Autorská práva, pokud nejsou výslovně uvedena jinak, náleží společnosti **Projekt GROWE**. Autorská práva náležející k programovým modulům, které jsou použity, jsou ve zdrojových kódech uvedeny, včetně jejich vlastníků.

4.2. LICENČNÍ UJEDNÁNÍ

Toto licenční ujednání je právní smlouvou mezi Vámi, koncovým spotřebitelem (dále jen uživatel) softwarového produktu **Skript Studio 1.**, dále v kapitole **3. Licence a autorská práva** označován jako produkt a společností **Projekt GROWE®** (v textu použito jako **Projekt GROWE**). Produkt jako takový se neprodává, pouze se uděluje licence k jeho užívání. Instalací a používáním produktu uživatel potvrzuje, že souhlasí s podmínkami tohoto ujednání a zavazuje se jimi řídit.

NA ZÁKLADĚ TÉTO LICENČNÍ SMLOUVY SE UDĚLUJÍ NÁSLEDUJÍCÍ PRÁVA:

Produkt je poskytnut uživateli **ZDARMA**, počet instalací, nebo kopií není nijak omezen. Může jej bez omezení šířit, pokud dodrží pravidlo, že program bude šířen jako celek, včetně podpůrných, konfiguračních a dalších souborů, úplné a nepozměněné dokumentace a elektronických manuálů.

POPIS DALŠÍCH PRÁV A OMEZENÍ:

Uživatel není oprávněn produkt prodávat, pronajímat, půjčovat nebo poskytovat produkt za úplatu. Zpětná analýza produktu není povolena. Jednotlivé komponenty nebo části produktu nelze oddělovat pro samostatné použití. Je zakázáno, upravovat chod programu prostřednictvím produktů třetích stran. Zasahovat do databáze (import/export nebo zpracovávání dat) je povoleno, avšak s přihlédnutím na možného zásahu do autorských práv jejich vlastníků.

Projekt GROWE si vyhrazuje právo na změnu licenční politiky bez předchozího upozornění.

4.3. ZÁRUKY A ODPOVĚDNOSTI

PROJEKT GROWE PAK ZVLÁŠTĚ NEZARUČUJE BEZVADNÝ CHOD PRODUKTU, POKUD:

- a) je provozován spolu s programy jiných výrobců, které znemožňují bezchybný chod programů
- b) je provozován na chybně nakonfigurovaném počítači nebo na chybně nastavené počítačové síti
- c) uživatel provedl zásah do produktu, do jeho databázových nebo konfiguračních souborů pomocí jiných nástrojů, nebo prostředků, než je samotný program nebo nástroje, které jsou součástí programového vybavení a pro tento zásah určeny.

Produkt poskytován bez jakýchkoli záruk.

Projekt GROWE nenese odpovědnost za případné následné škody, ať jsou jakékoli, které by vznikly na základě použití jeho produktů. **Projekt GROWE** nenese odpovědnost za použití softwarového vybavení k jiným účelům, než je zde výslovně uvedeno. V případě požadavku na satisfakci za možné vyniklé škody, náleží uživateli částka do maximální výše nákladů, prokazatelně uhrazených za softwarový produkt k rukám společnosti **Projekt GROWE**.

5. ADRESY A KONTAKTY

WEB stránky projektu:

www.projekt-growe.cz

E-mail:

support@projekt-growe.cz

6. SLOVNÍČEK POJMŮ

Aktualizace programu - proces instalace, při kterém dochází ke změně výkonných souborů na nejnovější verzi

Databáze - systém ukládání dat do souboru s pevnou strukturou záznamů

dBase - Databázový soubor společnosti dBase Inc.

GBase - Databázový soubor společnosti Growe DataBase

Hlavička database ExtHead - rozšířená hlavička Databázový soubor společnosti Growe DataBase

.INI soubor - soubor konfigurace

.LOG soubor - soubor, kam jsou ukládány běhové informace a události

Moduly - programové celky ve zdrojovém kódu, obsahující procedury a funkce

Program Skript Builder Pro - samostatný program pro tvorbu skriptu, je implementován do Skript Studio I.

Program DBF Reader Pro - samostatný program pro kontrolu a inicializaci skriptu, je implementován do Skript Studio I.

Validace - proces, při kterém je skript spuštěn v simulačním módu a kompletně testován

Řídící program - program, využívající skript pro hlavní řídicí proces

Řídící skript - soubor databáze .DBG, obsahující příkazy pro hlavní řídicí proces

Řídící příkazy - příkazy, kterými se řídí hlavní řídicí proces

Řídící reléová deska - hardwarová reléová jednotka, spínající jednotlivé spotřebiče v řídicím procesu

Tabulka databáze - popis proměnných, sekvenčně ve větách ukládaných do databáze

Zálohování - proces zdvojení důležitých souborů, jejichž kopie se uloží na bezpečné, zpravidla jiné zařízení, než na kterém je uložen originální soubor