

Vizualizace dat na vzdáleném PC v TheKing

Příručka uživatele a programátora



SofCon[®] spol. s r.o.
Střešovická 49
162 00 Praha 6
tel/fax: +420 220 180 454
E-mail: sofcon@sofcon.cz
www: <http://www.sofcon.cz>

Informace v tomto dokumentu byly pečlivě zkontrolovány a SofCon věří, že jsou spolehlivé, přesto SofCon nenese odpovědnost za případné nepřesnosti nebo nesprávnosti zde uvedených informací.

SofCon negarantuje bezchybnost tohoto dokumentu ani programového vybavení, které je v tomto dokumentu popsáno. Uživatel přebírá informace z tohoto dokumentu a odpovídající programové vybavení ve stavu, jak byly vytvořeny a sám je povinen provést validaci bezchybnosti produktu, který s použitím zde popsaného programového vybavení vytvořil.

SofCon si vyhrazuje právo změny obsahu tohoto dokumentu bez předchozího oznámení a nenese žádnou odpovědnost za důsledky, které z toho mohou vyplynout pro uživatele.

Datum vydání: 01.07.2004

Datum posledního uložení dokumentu: 01.07.2004

(Datum vydání a posledního uložení dokumentu musí být stejné)

Upozornění:

V dokumentu použité názvy výrobků, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah :

1.	O dokumentu	5
1.1.	Revize dokumentu	5
1.2.	Účel dokumentu	5
1.3.	Rozsah platnosti	5
1.4.	Související dokumenty	5
2.	Termíny a definice	5
3.	Úvod	6
4.	Adresářová struktura a kopírování souborů	9
5.	Používané direktivy	11
6.	Dodávané příklady	12
6.1.	Example1	13
6.2.	Example2	14
6.3.	Example3	14
7.	Postup návrhu aplikace	15
7.1.	Vytvoření CSV souboru	15
7.2.	Tvorba oken pro vizualizaci	16
7.3.	Aplikace pro řídicí systémy KIT	16
7.3.1.	Úpravy šablon a nastavení konstant popisující paměť	16
7.3.2.	Předefinování metod funkčních objektů	17
7.4.	Nahrání aplikace do řídicího systému KIT	17

1. O dokumentu

1.1. Revize dokumentu

Verze dokumentu	Verze SW balíku KitKing	Autor	Datum vydání	Popis změn
1.00	1.XX	Hv	18.12.2002	První vydání.
1.10	1.XX	Tu	20.5.2003	Úprava dokumentu dle ISO9000
1.20	2.XX	Hv	28.01.2004	Doplnění poznámek o práci s LOADER
1.30	2.XX	Wil	01.07.2004	

1.2. Účel dokumentu

Tento dokument popisuje postup při návrhu nové aplikace komunikující s programem TheKing, který vizualizuje data z řídicího systému KIT v nadřazeném počítači s operačním systémem Windows.

1.3. Rozsah platnosti

Určen pro programátory a uživatele programového vybavení SofCon.

1.4. Související dokumenty

Pro čtení tohoto dokumentu je potřeba seznámit se s manuály popisujícími balík knihoven „KitKing“, „Archivy“ a balík knihoven „LdrLib“. Dále je třeba se seznámit s programem TheKing.

2. Termíny a definice

Používané termíny a definice jsou popsány v samostatném dokumentu „Termíny a definice“.

ŘS = řídicí systém

3. Úvod

Účelem tohoto dokumentu je popsat postup při návrhu nové aplikace komunikující s vizualizačním programem TheKing, který vizualizuje data z řídicího systému KIT v nadřazeném počítači s operačním systémem Windows. Jelikož mohou existovat různé modifikace tohoto programu (např. TheQueen pro počítače s dotykovým LCD panelem - TOUCHxx), budeme takovéto programy nazývat obecně „vizualizační program“.

V několika následujících kapitolách budou popsány jednotlivé kroky při návrhu aplikace s vizualizací dat. Pořadí kroků vychází ze zkušeností programátorů aplikací s podporou vzdálené vizualizace dat, tzn. komunikující s programem TheKing. Dále v těchto kapitolách najdete popis dodávaných příkladů a stručný popis šablon případně odkazy na další manuály.

4. Struktura vizualizovaných dat

Komunikace mezi řídicím systémem a vizualizačním programem probíhá po sériovém rozhraní zpravidla RS232 s použitím vlastního firemního protokolu „SofCon DF2“. Jedná se o komunikaci typu Master-Slave, kde řídicí systém je Slave (podřízený) a vizualizační program je Master (nadřízený). To znamená, že komunikaci zahajuje vždy vizualizační program, řídicí systém pouze odpovídá na příchozí zprávy a vykonává požadované příkazy.

Komunikaci mezi řídicím systémem a vizualizačním programem lze podle struktury přenášených zpráv a dat rozdělit do následujících skupin:

- OnLine parametry
- Archivy
- Binární datové bloky
- Nahrávání nové řídicí aplikace pomocí pomocné aplikace Loader

Pro uživatele mají největší význam OnLine parametry, které jsou součástí každé vizualizace, a proto se o nich zmíníme podrobněji. To však neznamená, že ostatní přenášené datové struktury mají význam zanedbatelný. Uživatel se s nimi jen neseťkává tak často.

4.1. OnLine parametry

Pojmem OnLine parametry rozumíme komunikované parametry ŘS, které jsou předem definované jednoduchého typu. OnLine parametrem mohou být nastavované parametry řídicího systému, pracovní proměnné, nebo jím může být i příkaz pro provedení určité akce kontroléru.

OnLine parametry se dělí na systémové a aplikační.

Systémové parametry jsou předem „zakompilované“ do vizualizačního programu, mají pevnou (neměnnou) někdy i zcela specifickou strukturu a jejich vlastnosti se nemohou měnit. Jedná se o parametry, které jsou společné pro všechny aplikace. Každá aplikace například obsahuje Restart strukturu. Přidání nového

systémového parametru vyžaduje úpravu i vizualizačního programu a jeho novou kompilaci. Logické adresy systémových parametrů jsou od 0 do 99.

Aplikační parametry nejsou předem zakompilované do vizualizačního programu. Vizualizační program si je při spuštění načte z vygenerovaného seznamu parametrů. Jsou poplatné pro danou konkrétní aplikaci a přidání nebo změna vlastností aplikačních parametrů nemá vliv na vizualizační program. Aplikační parametry mají svou strukturu danou vlastností Type, která je rovněž popsána v dalších kapitolách. Logické adresy aplikačních parametrů jsou od hodnoty 100 výše.

Čtení a zápis OnLine parametrů může probíhat buď jednotlivě (jedna zpráva obsahuje jeden parametr) nebo blokově (jedna zpráva obsahuje více parametrů). Z hlediska rychlosti přenosu je pro čtení více parametrů výhodné použít blokové čtení. Definice blokového čtení a zápisu parametrů se provádí při tvorbě vizualizačních obrazovek.

Každý OnLine parametr je charakterizován určitými vlastnostmi. Seznam vlastností je uveden v následující tabulce. Některé vlastnosti jsou nutné pouze pro vizualizační program, jiné pouze pro řídicí aplikaci, ale většina jich je společných.

Tabulka: Seznam vlastností OnLine parametrů

Vlastnost	Popis
LA	Logická adresa parametru (pro vizualizační SW i řídicí systém) Je to jedna z nejdůležitějších vlastností. Má ho každý parametr a žádné dva parametry nesmí mít tuto adresu stejnou.
Level	Přístupová úroveň pro zápis hodnoty parametru (hlavně pro vizualizační SW) – viz kapitoly dále.
LowLimit	Dolní mez hodnoty parametru (pro vizualizační SW i řídicí systém)
HighLimit	Horní mez hodnoty parametru (pro vizualizační SW i řídicí systém)
Default Value	Implicitní hodnota (pro vizualizační SW i řídicí systém)
Type	Datový typ (pro vizualizační SW i řídicí systém)

Všechny vlastnosti v této tabulce jsou konstanty (převážně číselné), kromě vlastnosti Type, která určuje datový typ OnLine parametru.

Pro vizualizační program je třeba definovat ještě další důležité vlastnosti, zejména **jméno parametru**, **fyzikální jednotku**, periodu automatického čtení, formáty výpisu apod. Definice těchto vlastností je nad rámec tohoto dokumentu.

Vazba vlastností OnLine parametrů mezi vizualizačním programem a řídicím systémem je prostřednictvím souboru .APP v adresáři APP vizualizačního programu. O postupu vytvoření tohoto souboru bude pojednáno později.

4.1.1. Přístupová úroveň pro zápis hodnoty parametru

Tato vlastnost je nutná zejména pro vizualizační prostředí, kde každému uživateli je přiřazena určitá přístupová úroveň. Přístupová úroveň je číslo od 0 do 4. Úroveň 0 je nejnižší a úroveň 4 naopak nejvyšší. Uživatel s vyšší úrovní přístupu může editovat i parametry s nižší úrovní.

Existuje ještě speciální přístupová úroveň 255. Ta je určena pro parametry, které jsou pouze pro čtení (z vizualizačního programu do nich nelze zapsat hodnotu).

4.2. Archivy

Pojmem archiv rozumíme pole záznamů stejné předem definované struktury. Každý záznam se skládá z jednotlivých položek, přičemž první a nezbytnou položkou záznamu každého archivu je datum a čas záznamu. Ostatní položky závisí na účelu použití archivu (např. archiv alarmů může mít jako další položky kód chyby a příznak vzniku/zániku alarmu). Každý archiv má svou maximální velikost (maximální počet uchovatelných záznamů). Do archivů se data mohou ukládat periodicky nebo v důsledku vzniklých událostí. Při zaplnění archivu se staré záznamy mohou přemazávat novými, tj. takovýto archiv je potom typu kruhový buffer.

Každému archivu kromě vlastního pole záznamů jsou přidruženy další informace:

- jméno archivu
- indexy do pole záznamů (který záznam je poslední, který byl naposledy přečten apod.)
- popisovač struktury archivu (popis struktury položek záznamu pro vizualizační prostředí)
- popisovač grafického vyjádření (jakým způsobem se mají jednotlivé položky zobrazovat do grafů), který je nepovinný, definuje se pouze pokud si programátor přeje data zobrazovat do grafu

Vizualizační program předem nezná počet ani strukturu jednotlivých archivů řídicího systému. Tyto informace si přečte při prvním přístupu. Tento způsob má velkou výhodu. Jakákoliv změna archivů či jejich struktury v řídicím systému nemá téměř žádný vliv na vizualizační prostředí (slovem téměř se rozumí případná úprava šablon pro grafické vyjádření dat v archivech).

4.3. Binární datové bloky

Binární datový blok je blok dat (bytů), které nemají blíže specifikovanou strukturu. Jedná se o přenos dat nejnižší možné úrovně. Tímto způsobem se mohou komunikovat např. technologické parametry v binární podobě či jiné binární datové bloky specifické pro danou aplikaci.

Podobně jako u archivů tak i u binárních datových bloků vizualizační program předem nezná strukturu, velikost ani počet těchto bloků poskytovaných řídicím systémem. Tyto informace si přečte při prvním přístupu.

4.4. Nahrávání nové řídicí aplikace

Pokud si programátor přeje pomocí vizualizačního programu nahrávat i případné budoucí nové verze řídicí aplikace do ŘS, musí nejprve vytvořit pomocnou aplikaci Loader. Tuto aplikaci nejlépe vytvoří pomocí knihovního balíku LdrLib, který je součástí systémových knihoven LIB. Aplikaci poté nahraje pomocí programátoru nebo RTD do paměti EEPROM.

Dále do řídicí aplikace doplní správné reakce na následující zprávy:

- Loader_QueryApp - Dotaz běží-li řídicí aplikace či Loader.
- Loader_StartPrg - Přepnutí do nahrávacího režimu – provede restart systému a spustí Loader.

- Loader_RestartPrg - Provede samotný restart aplikace.

Loader používá pro nahrání nové řídicí aplikace ještě další zprávy, jejichž popis není součástí tohoto dokumentu.

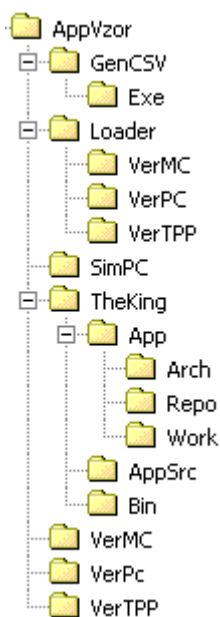
5. Adresářová struktura a kopírování souborů

Pro usnadnění vytvoření doporučené adresářové struktury a získání souborů vyžadovaných aplikací s podporou vizualizace, můžete použít dodávanou šablonu v adresáři **Examples\KitKing\AppVzor**. Tento adresář je podadresářem systémových knihoven LIB. Při používání této šablony doporučujeme provést kontrolu a případně aktualizaci tohoto souboru:

Turbo.tpl v adresáři AppVzor\VerMc a AppVzor\Loader\VerMc.

Název (označení) aplikace doporučujeme max. 7 písmen, protože první znak se v některých případech používá pro prefix **x (Exchange)** označující sdílené jednotky mezi více programy.

Pokud budete chtít vše provést sami, musíte nejprve vytvořit následující adresářovou strukturu.



Po vytvoření této adresářové struktury nakopírujte z příloženého příkladu následující soubory. Soubory, které v názvu obsahují **AppVzor** přejmenujte dle označení (názvu) Vaší aplikace.

Do adresáře Loader umístěte následující soubory:

uxDesc.pas, Loader.pas. Tyto soubory upravte podle požadavků Vaší aplikace.

Pozn. Vzhledem k vývoji LOADER je doporučováno, používat nebo alespoň kontrolovat změny v příkladech umístěných v adresáři LOADER.

Do adresáře GenCSV umístěte následující soubory:

AppVzor.pas, uAppVzor.pas a xAppVzor.pas. (Pozn. Tyto adresáře přejmenujete dle označení (názevu) Vaší aplikace.) Tyto soubory budete později upravovat dle požadavků Vaší aplikace a to pouze v případě, že budete používat online parametry.

Do adresářů SimPc, VerPc, VerTpp a VerMc umístěte soubor Sets.inc, ve kterém se musí správně nastavit direktivy překladače pro jednotlivé adresáře. Do adresáře VerMc je potřeba ještě nakopírovat Turbo.tpl pro řídicí systém KIT. Tento soubor získáte v dodávaných knihovnách v adresáři MCP7 při používání Borland Pascal verze 7.x. V těchto adresářích budete později provádět vlastní překlad aplikace.

Popis adresářů:

AppVzor	označení (název) Vaší aplikace. Doporučujeme používat max. 7 písmen.
Loader	verze programu LOADER respektující rozložení paměti.
GenCSV	v adresáři je umístěn program generující CSV soubor, který se používá v programu Cr_TheKing. Dále je v tomto adresáři soubor xAppVzor (označení-název Vaší aplikace), jenž je sdílen s Vaší aplikací.
VerMC	adresář pro překlad aplikace určené pro KIT. Ze začátku obsahuje upravený soubor Sets.inc a Turbo.tpl.
VerPC	adresář pro překlad aplikace určené pro PC s připojeným HW prostřednictvím PC Kit.
VerTPP	adresář pro překlad aplikace pro PC pracující v chráněném režimu s připojeným HW prostřednictvím PC Kit..
SimPC	adresář pro překlad aplikace určené pro PC bez připojeného HW. HW je programově simulován, zpravidla pomocí extra procesu nebo procesů.
TheKing	adresář obsahující INI soubory, soubory s navrženými okny pohledů online parametrů a další pracovní adresáře.

Pozn.: Jména souborů a adresářů odpovídající ukázkové aplikaci (šabloně) se mohou lišit dle Vašich požadavků. Popis těchto adresářů je popsán v samostatném manuálu programu TheKing.

V adresáři App je soubor s koncovkou VIE, který obsahuje navržená okna s pohledem na online parametry. Dále v tomto adresáři mohou být různé soubory s obrázky, které jsou použity při vykreslování navržených oken a soubory generované programem Cr_TheKing (viz Tvorba oken pro vizualizaci)

V adresáři AppSrc jsou uloženy soubory s výčtovými typy zobrazovanými v programu TheKing a soubor s koncovkou CSV generovaný programem v adresáři GenCSV. Soubor

s CSV a soubory s výčtovými typy tvoří vstup programu Cr_TheKing, který z těchto souborů vytvoří soubor s koncovkou APP.

6. Používané direktivy

Šablona a dodávané příklady obsahují soubory Sets.inc, pomocí kterých vybíráte, jak se příklad vytvoří. V tomto souboru mohou být následující direktivy.

- volba komunikačního kanálu
 - V40 komunikační port na procesoru V40
 - COM standardní komunikační port
 - UDP přenos dat probíhá pomocí Ethernet protokol UDP
- volba nastavení překladu
 - SimPC verze pro počítač PC s monitorem a klávesnicí a simulací HW
 - VerPC verze pro počítač PC s monitorem a klávesnicí a reálným HW
 - VerTPP verze pro počítač PC s monitorem a klávesnicí a reálným HW pracujícím v chráněném módu
 - VerMC verze pro řídicí systém Kit s HW
- volba překládaného příkladu
 - FILE1 příklad se dvěma jednoduchými souborovými bankami. Pomocí těchto bank se přenáší struktura tFixData (menší než 4KB, tzn. přenáší se v jednom komunikačním paketu). V tomto příkladu index dotazované banky souborů odpovídá indexu banky v řídicím systému KIT.
 - CONVERT_FILENO and FILE1 příklad se dvěma jednoduchými souborovými bankami. Pomocí těchto bank se přenáší struktura tFixData (menší než 4KB, tzn. přenáší se v jednom komunikačním paketu). V tomto příkladu index dotazované banky souborů se liší od indexu banky v řídicím systému KIT, proto se provádí jeho konverze.
 - FILE2 příklad se dvěma souborovými bankami implementovanými jako oblast paměti o velikosti 4KB, tzn. přenáší se v několika komunikačních paketech. V tomto příkladu index dotazované banky souborů odpovídá indexu banky v řídicím systému KIT.
 - LOADER příklad pro podporu příkazů programu LOADER, který umožňuje nahrát novou verzi firmware řídicího systému.
 - PARAM příklad s nastavováním jednotlivých i blokových online parametrů v řídicím systému, čtení parametrů a přenos příkazů. Při spuštění příkladu s online parametry můžete v programu TheKing tyto parametry nastavovat, číst a zobrazovat pomocí oken **Prenos jednotlivých parametru** a **Blokovy prenos parametru** v menu **Views\Controllor1\OnLine View**.

ARCHIVE_INIT

při nastavení této direktivy může dojít k poškození archivu. Proto by se měla používat pouze, pokud jste provedli změny struktury archivu případně jste je umístily do jiného paměťového prostoru. Při jejím použití se nastaví příznak `af_NotCheckAStruct`, což způsobí jejich nastavení podle předaných parametrů. Pokud tento příznak není nastaven, provádí se kontroly popisovačů archivu v paměti s požadovanými parametry. Pokud se zjistí rozdíl, vrací se chyba, viz `GetLastError`.

ARCHIVE1 příklad práce se dvěma archivy používající paměť s kruhovým bufferem s podporou zobrazení tabulky, tzn. nepodporuje zobrazení grafů. Při čtení dat z archivu se vrací data uložená v archivu. (Pozn. Při zvolení jiného typu archivu se může vykonávat jejich modifikace.) V tomto příkladu index dotazované banky archivů odpovídá indexu banky v řídicím systému KIT.

CONVERT_BANKNO and ARCHIVE1

příklad práce se dvěma archivy používající paměť s kruhovým bufferem s podporou zobrazení tabulky, tzn. nepodporuje zobrazení grafů. Při čtení dat z archivu se vrací data uložená v archivu. (Pozn. Při zvolení jiného typu archivu se může vykonávat jejich modifikace.) V tomto příkladu index dotazované banky archivů se liší od indexu banky v řídicím systému KIT, proto se provádí jeho konverze.

ARCHIVE2 příklad práce se dvěma archivy používající paměť s kruhovým bufferem s podporou zobrazení grafů. Při čtení dat z archivu se vrací data uložená v archivu. (Pozn. Při zvolení jiného typu archivu se může vykonávat jejich modifikace.) V tomto příkladu index dotazované banky archivů odpovídá indexu banky v řídicím systému KIT.

7. Dodávané příklady

Součástí vizualizace jsou následující příklady, které implementují jednotlivé funkce vizualizace.

Abychom usnadnili přidávání těchto funkcí, je každá funkce implementována v samostatném souboru. Proto je možné po editaci funkcí s prefixem **fn**, tyto soubory přidat přímo k Vaší aplikaci.

Pozn. Při spuštění příkladů na počítači PC se může vygenerovat chyba. Tato chyba se generuje, pokud soubory pro ukládání statických struktur (koncovka SAV) nejsou vytvořeny nebo jsou poškozeny. Pokud chyba při opakovaném spuštění nezmizí, smažte nejdříve soubory s koncovkou SAV a dvakrát zopakujte spuštění příkladu. Nedaří-li se příklad spustit, kontaktujte firmu SofCon s.r.o.

Pozn. Vzhledem k vývoji **LOADER** je doporučováno, používat nebo alespoň kontrolovat změny v příkladech umístěných v adresáři **LOADER**.

7.1. Example1

Pomocí výše uvedených direktiv můžete vytvořit příklad dle Vašich požadavků. Dále bude následovat popis souborů s implementovanou funkcí aplikace.

Pozn.: V rámci zjednodušení tohoto kódu se ve verzi pro PC neuvolňuje všechna alokovaná paměť a nekontrolují se ukazatele na hodnotu NIL. Tento příklad byl později doplněn o přenos dat pomocí UDP a rozšířen o dříve uvedené kontroly a uvolňování paměti a je dodáván pod označením Example3.

Param.pas

V tomto souboru je předvedeno, jak doplnit do Vaší aplikace přenos online parametrů ve formě jednoduchých nebo blokových přenosů a implementaci příkazů.

Při první editaci tohoto souboru upravujte pouze funkce `fnGetParamVal`, `fnPutParamVal`, `fnGetParamSize`, `Init`, `SetResultCode` a `ShowState`. Funkce `fnGetFirmwareParamVal` a `fnPutFirmwareParamVal` byste měnit neměli, protože Vám ve spolupráci s jednotkou `AppDesc` zajišťují čtení a inicializaci struktury pro zachytávání chyb aplikace, které můžete v programu `TheKing` zobrazit pomocí **Views\Controller 1\Restart structure**.

Podrobný popis metod objektu najdete v manuálu `KitKing` v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickParam*.

Loader.pas

V tomto souboru je předvedeno, jak doplnit do Vaší aplikace podporu pro nahrávání nové verze do řídicího systému KIT pomocí vizualizace.

Při první editaci tohoto souboru upravujte pouze funkci `ShowState`.

Podrobný popis metod objektu najdete v manuálu `KitKing` v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickLoader*.

File1.pas nebo File2.pas

V těchto souborech je předvedeno, jak doplnit do Vaší aplikace funkce vizualizace pro přenos souborů (viz Používané direktivy)

Při první editaci těchto souborů upravujte funkce `InitBankDescUserData`, `ShowState`, `fnConvertFileNo`, `fnGetFileCnt`, `fnGetBankListItem`, `fnGetBankDataOpen`, `fnGetBankDataBlock`, `fnPutBankDataOpen` a `fnPutBankDataBlock`.

Podrobný popis metod objektu najdete v manuálu `KitKing` v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickFile*.

Archive1.pas nebo Archive2.pas

V těchto souborech je předvedeno, jak doplnit do Vaší aplikace funkce vizualizace pracující s archivy (viz Používané direktivy)

Při první editaci těchto souborů upravujte funkce `TextModif`, `ShowState`, `WriteToArchive`, `Init`, `fnConvertArchiveNo`, `fnGetArchiveCnt`, `fnClose`, `fnGetListOpen`, `fnGetListItem`, `fnGetState`, `fnOpenArchive`, `fnReadArchive` a `fnClearArchive`.

Pokud zachováte použití pole při správě instancí objektů archivu stačí upravit pouze funkce `TextModif`, `ShowState`, `WriteToArchive` a `Init`.

Podrobný popis metod objektu najdete v manuálu KitKing v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickFile*.

7.2. Example2

Tento příklad implementuje pouze práci s online parametry a nahrávání nové verze aplikace. Dále bude následovat popis souborů s implementovanou funkcí aplikace.

Param.pas

V tomto souboru je předvedeno, jak doplnit do Vaší aplikace přenos online parametrů ve formě jednoduchých nebo blokových přenosů a implementaci příkazů.

Při první editaci tohoto souboru upravujte pouze funkce `fnGetParamVal`, `fnPutParamVal`, `fnGetParamSize`, `Init`, `SetResultCode` a `ShowState`. Funkce `fnGetFirmwareParamVal` a `fnPutFirmwareParamVal` byste měnit neměli, protože Vám ve spolupráci s jednotkou `AppDesc` zajišťují čtení a inicializaci struktury pro zachytávání chyb aplikace, které můžete v programu TheKing zobrazit pomocí **Views\Controller 1\Restart structure**.

Podrobný popis metod objektu najdete v manuálu KitKing v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickParam*.

Loader.pas

V tomto souboru je předvedeno, jak doplnit do Vaší aplikace podporu pro nahrávání nové verze do řídicího systému KIT pomocí vizualizace.

Při první editaci tohoto souboru upravujte pouze funkci `ShowState`.

Podrobný popis metod objektu najdete v manuálu KitKing v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickLoader*.

7.3. Example3

Pomocí výše uvedených direktiv můžete vytvořit příklad dle Vašich požadavků. Dále bude následovat popis souborů s implementovanou funkcí aplikace.

Pozn.: Tento příklad vychází z Example1 a je rozšířen o výpisy stavu HEAP, uvolňování alokované paměti, kontroly ukazatelů na NIL a možnost přenosu dat pomocí UDP.

Param.pas

V tomto souboru je předvedeno, jak doplnit do Vaší aplikace přenos online parametrů ve formě jednoduchých nebo blokových přenosů a implementaci příkazů.

Při první editaci tohoto souboru upravujte pouze funkce `fnGetParamVal`, `fnPutParamVal`, `fnGetParamSize`, `Init`, `SetResultCode` a `ShowState`. Funkce `fnGetFirmwareParamVal` a `fnPutFirmwareParamVal` byste měnit neměli, protože Vám ve spolupráci s jednotkou `AppDesc` zajišťují čtení a inicializaci struktury pro zachytávání chyb aplikace, které můžete v programu TheKing zobrazit pomocí **Views\Controller 1\Restart structure**.

Podrobný popis metod objektu najdete v manuálu KitKing v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickParam*.

Loader.pas

V tomto souboru je předvedeno, jak doplnit do Vaší aplikace podporu pro nahrávání nové verze do řídicího systému KIT pomocí vizualizace.

Při první editaci tohoto souboru upravujte pouze funkci ShowState.

Podrobný popis metod objektu najdete v manuálu KitKing v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickLoader*.

File1.pas nebo File2.pas

V těchto souborech je předvedeno, jak doplnit do Vaší aplikace funkce vizualizace pro přenos souborů (viz Používané direktivy)

Při první editaci těchto souborů upravujte funkce InitBankDescUserData, ShowState, fnConvertFileNo, fnGetFileCnt, fnGetBankListItem, fnGetBankDataOpen, fnGetBankDataBlock, fnPutBankDataOpen a fnPutBankDataBlock.

Podrobný popis metod objektu najdete v manuálu KitKing v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickFile*.

Archive1.pas nebo Archive2.pas

V těchto souborech je předvedeno, jak doplnit do Vaší aplikace funkce vizualizace pracující s archivy (viz Používané direktivy)

Při první editaci těchto souborů upravujte funkce TextModif, ShowState, WriteToArchive, Init, fnConvertArchiveNo, fnGetArchiveCnt, fnClose, fnGetListOpen, fnGetListItem, fnGetState, fnOpenArchive, fnReadArchive a fnClearArchive.

Pokud zachováte použití pole při správě instancí objektů archivu stačí upravit pouze funkce TextModif, ShowState, WriteToArchive a Init.

Podrobný popis metod objektu najdete v manuálu KitKing v kapitole *Funkční objekty* a podkapitole *Objekt TWinTickFile*.

8. Postup návrhu aplikace

8.1. Vytvoření CSV souboru

Upravte soubory v adresáři GenCSV podle následujících poznámek.

Pozn. Názvy popisovaných souborů jsou původními názvy před jejich přejmenováním dle předchozí kapitoly.

Pozn. Pokud přejmenujete jednotky dle Vašich požadavků, musíte upravit jména používaných jednotek v dále uvedených souborech.

AppVzor.pas	hlavní tělo programu generující CSV soubor. V těle tohoto programu se musí změnit cesty a jméno generovaného souboru.
-------------	---

xAppVzor.pas	soubor obsahující popis přenášených parametrů a je sdílen s Vaší aplikací pro řídicí systém KIT.
uAppVzor.pas	v souboru jsou příkazy, které generují vlastní CSV soubor. Popis těchto příkazů najdete v samostatném manuálu popisujícím Cr_TheKing.

Po spuštění vytvořeného programu odpovídající požadavkům aplikace na přenos online parametrů se vytvoří na zadané cestě CSV soubor. Tento soubor by se měl generovat do adresáře AppSrc, viz nastavení Cr_TheKing.ini.

8.2. Tvorba oken pro vizualizaci

Před vytvářením oken ve vizualizaci se musí spustit program Cr_TheKing, který vygeneruje soubory s koncovkou LST a APP. Soubor s koncovkou APP obsahuje jak systémové tak i aplikační popisovače datových struktur, nad kterými pracuje vizualizační program TheKing. Popis podrobného postupu pro vytváření těchto oken a těchto souborů najdete v samostatném manuálu popisujícím aplikaci TheKing a Cr_TheKing.

Pozn. Pro bezchybnou práci programu TheKing musí být vždy vytvořen soubor s koncovkou APP.

Nejjednodušší postup jak získat soubory pro vizualizaci je editací souboru Cr_TheKing.ini, kde změňte následující položky (upravíte cesty) dle požadavků Vaší aplikace.

```
AppSrcFile='o:\Vzor\AppVzor\TheKing\AppSrc\AppVzor.CSV'
```

```
AppDestFile='o:\Vzor\AppVzor\TheKing\App\AppVzor.app'
```

8.3. Aplikace pro řídicí systémy KIT

Pokud píšete první aplikaci s vizualizací, tak ji nejjednodušeji vytvoříte, pokud vycházíte z dodávaných příkladů případně můžete použít návod uvedený v samostatném manuálu popisujícím vizualizační knihovny případně archivy.

8.3.1. Úpravy šablon a nastavení konstant popisující paměť

Po změně názvů souborů obsažených v šabloně dle Vašich požadavků byste měli nastavit konstanty popisující rozložení paměti Vaší aplikace.

- **uxDesc.pas:**

Popis konstant popisující paměť v souboru **uxDesc.pas** najdete v samostatném manuálu popisujícím program LOADER:

Pozn.: Pokud měníte soubor uxDesc.pas, tak musíte přeložit program LOADER a ten poté zapsat do KIT. Nahrání nové verze LOADER můžete provést buď nahráním pomocí RTD, pomocí programátoru nebo pomocí programu LdrPatch.

- **AppDesc.pas:**

V tomto souboru se musí doplnit konstanty popisující struktury uložené v zachovávané oblasti statické paměti RAM. Tuto jednotku neměňte více než je nutné, protože v jejím těle jsou obsaženy funkce a definice pro zpracování chyb aplikace. Při jejich poškození byste přišli o užitečný nástroj při vývoji a sledování chodu aplikace, protože při použití objektu TWinTickLoader (jednotka LOADER) můžete chyby zobrazovat v programu TheKing pomocí okna **Restart structure** v menu **Views\Controller 1** jak při spuštění aplikace tak programu LOADER.

Pro účely ladění je vhodné doplnit inicializace oblastí těchto konstant do funkce fnAppInit, která slouží především na PC. Pokud jsou tyto konstanty nastavovány dle dodaných příkladů a je doplněna kontrola ve funkci fnAppInit, tak se už ve verzi pro počítač PC nechají zjistit chyby při nastavení adres nebo změnách těchto struktur. Především se jedná o přepisy jednotlivých struktur.

8.3.2. Předefinování metod funkčních objektů

Pokud nastavíte paměťové konstanty a vytvoříte okna ve vizualizaci, můžete přikročit k poslednímu kroku - doplnění vizualizace do hotové aplikace případně můžete začít tuto aplikaci vytvářet. V rámci tohoto kroku doporučujeme vycházet z dodaných příkladů, viz Dodávané příklady.

Podrobný popis přidání vizualizace najdete v samostatném manuálu KitKing.

8.4. Nahrání aplikace do řídicího systému KIT

Pro nahrání aplikace do řídicího systému můžete použít program RTD nebo program TheKing. Při použití posledně jmenovaného programu musíte do řídicího systému nejdříve nahrát program LOADER, který najdete v podadresáři LOADER. Podrobný popis tohoto programu najdete v manuálu LOADER.

Při použití tohoto programu se Vám podstatně zjednoduší ladění aplikace, protože zobrazování dat a nahrávání vlastní aplikace provádíte z jednoho programu. Překlad a rozmístění programu LOADER provádíte jak jste zvyklý, tj. programem BP a RTD (rozmístění modulů aplikace).