**ScriptSeparator**

Obsah

[Adresářová struktura 1](#_Toc7371163)

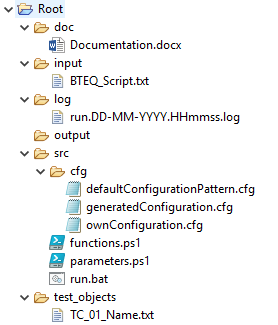
[Parametry nástroje 2](#_Toc7371164)

[Popis procesu nástroje 5](#_Toc7371165)

[Spuštění nástroje 11](#_Toc7371166)

# Adresářová struktura

Jednotlivé adresáře (složky) a soubory jsou vidět na následujícím obrázku a dále jsou popsány.



Hlavní adresář, který obsahuje všechny soubory a složky nástroje, je nazván Root. *Root* pod sebou má celkem šest podadresářů. Prvním z nich je *doc* obsahující dokumentaci nástroje, ve které uživatel může najít všechny potřebné informace včetně návodu, jak nástroj používat. Dalším podadresářem je defaultní vstupní adresář *input*, ze kterého nástroj bere vstupní BTEQ skripty, pokud není uživatelem zadaný jiný vstupní adresář.

Do složky *log* jsou vkládány logy z jednotlivých běhů nástroje, které obsahují informace o operacích provedených v rámci konkrétního běhu. Mimo jiné jsou v nich zaznamenány varování a chybové hlášky, pokud dojde v rámci procesu k nestandardní situaci. Každý log má ve svém názvu časové razítko, které odkazuje na čas jeho vytvoření. Podle toho se dá jednoduše identifikovat, o jakém běhu konkrétní log informuje.

Adresář *output* slouží k ukládání zpracovaných BTEQ skriptů. Nástroj ve svém procesu neupravuje BTEQ skripty přímo ve vstupním adresáři. Nejdříve skripty zkopíruje do adresáře *output* a až v něm probíhá jejich úprava. Díky tomu, a také díky automatickému promazávání složky *output* na začátku procesu, je proces nástroje opakovatelný a není nutné před každým procesem aktualizovat vstupní složku.

Všechny soubory obsahující zdrojový kód jsou uloženy v adresáři *src*. Přímo v tomto adresáři jsou uloženy dva PowerShellové skripty obstarávající logiku celého procesu. Prvním z nich je soubor *parameters.ps1*, který obsahuje parametry nástroje a pomocí úprav tohoto souboru může uživatel proces nástroje řídit. Druhým z nich je soubor *functions.ps1*, ve kterém jsou uloženy funkce, s jejichž pomocí jsou prováděné požadované akce. Tyto funkce se automaticky volají při spuštění skriptu parameters.ps1. Přímo ve složce *src* se nachází také dávkový soubor *run.bat*, který slouží pro spuštění skriptu *parameters.ps1* v rámci IFPC command tasku.

Adresář *src* obsahuje ještě jeden podadresář nazvaný *cfg*, ve kterém jsou přítomny konfigurační soubory využívané při konverzi BTEQ skriptů. Konfigurační soubory jsou brány vždy z tohoto adresáře, a to i v případě, že chce uživatel použít vlastní konfigurační soubor.

V poslední složce *test\_objects* jsou skripty, jejichž pomocí byl nástroj testován. Skripty se dají použít k ukázkovým běhům nástroje s účelem zjistit, jak skripty zpracovává.

# Parametry nástroje

Níže je vyobrazena tabulka se seznamem definovaných parametrů. V tabulce je u každého parametru, kromě jeho názvu, uvedeno, zda je parametr povinný, jakého je typu, jaké může obsahovat hodnoty a defaultní hodnota, kterou parametr má, pokud je nevyplněn nebo je vyplněn špatně.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parametr** | **Povinnost** | **Typ** | **Možné** **hodnoty** | **Defaultní** **hodnota** |
| **sourceSystem** | omezená | řetězec | "řetězec1;...;řetězecN" | - |
| **ignoreFileStrings** | nepovinný | řetězec | "řetězec1;...;řetězecN" nebo "" | "" |
| **copyFileStrings** | nepovinný | řetězec | "řetězec1;...;řetězecN" nebo "" | "" |
| **fromOwnInputFolder** | nepovinný | boolean | true nebo false | false |
| **ownInputFolder** | omezená | řetězec | validní cesta | - |
| **createFilesForIFPC** | nepovinný | boolean | true nebo false | true |
| **IFPCFolder** | omezená | řetězec | validní cesta | - |
| **isConversionWanted** | nepovinný | boolean | true nebo false | true |
| **doOnlyConversion** | nepovinný | boolean | true nebo false | false |
| **ownConfigurationFile** | nepovinný | boolean | true nebo false | false |
| **configurationFile** | omezená | řetězec | název konfiguračního souboru | - |
| **targetEnvironment** | omezená | řetězec | "dev" nebo "test" nebo "uat" | dev |
| **productionDataLoad** | nepovinný | boolean | true nebo false | false |

Pokud je povinnost parametru kvalifikována jako „omezená“, tak to znamená, že parametr je povinný jen při splnění konkrétních podmínek. Pokud jsou tyto podmínky splněny, parametr musí být uživatelem vyplněn. Pokud parametry vyplněny nejsou nebo jsou vyplněny špatně, tak by měl nástroj proces zastavit a chtít po uživateli, aby tyto parametry vyplnil správně. Výjimkou je parametr **targetEnvironment**, který má definovanou defaultní hodnotu. V případě špatného vyplnění tohoto parametru je použita jeho defaultní hodnota. Uživatel je v tomto případě jen informován, ale proces není zastaven.

Defaultní hodnoty definovaných parametrů pokrývají základní pracovní tok, který koresponduje s očekávaným nejčastějším případem užití nástroje. Parametry, které defaultní hodnotu nemají, tak se buď v základním pracovním toku nepoužívají nebo jsou naprosto nezbytné a uživatel je musí vždy vyplnit validní hodnotou (**sourceSystem** a **IFPCFolder).** Základní pracovní tok obsahuje tyto kroky:

1. Nástroj vezme všechny BTEQ skripty (**ignoreFileStrings** = "") z defaultního vstupního souboru root/input (**fromOwnInputFolder** = false).
2. Každý skript na vstupu bude zprocesován (**copyFileStrings** = "", **doOnlyConversion** = false) a budou v něm ponechány jen části týkající se zdrojových systémů uvedených v **sourceSystem.**
3. Zprocesované skripty budou automaticky (**ownConfigurationFile** = false) zkonvertovány (**isConversionWanted** = true) tak, aby byly spustitelné na defaultním prostředí "dev" (**targetEnvironment**). Skripty budou používat jen databáze tohoto prostředí (**productionDataLoad**  = false).
4. Překonvertované skripty budou přemístěny (**createFilesForIFPC** = true) do cílovém adresáře **IFPCFolder**.

Každý z definovaných parametrů je dále podrobněji popsán. Mimo jiné je u každého parametru uveden jeho vztah k ostatním parametrům.

#### Parametr – sourceSystem

Tento parametr uživateli slouží k zadání názvů zdrojových systémů, které jsou pro testovací nápočet důležité a jejich transformace mají ve zpracovávaných BTEQ skriptech zůstat. Uživatel může zadat libovolný počet zdrojových systémů, ale jejich názvy musí vždy oddělit středníkem. Parametr je povinný tehdy, pokud je požadováno, aby byly skripty procesovány. Skripty nejsou procesovány, pokud jsou parametry **isConversionWanted** a **doOnlyConversion** oba nastaveny na true.

#### Parametr – ignoreFileStrings

Tento parametr uživateli slouží k zadání řetězců, které má nástroj v BTEQ skriptech hledat. Pokud nějaký z definovaných řetězců nástroj ve skriptu najde, tak tento skript bude z procesování vyloučen. Parametr je nepovinný.

#### Parametr – copyFileStrings

Tento parametr uživateli slouží k zadání řetězců, které má nástroj v BTEQ skriptech hledat. Pokud nějaký z definovaných řetězců nástroj ve skriptu najde, tak tento skript bude s celým svým obsahem zkopírován do cílového adresáře. Parametr je nepovinný.

#### Parametr – fromOwnInputFolder

Nastavením tohoto parametru na true může uživatel předat nástroji informaci, že chce brát vstupní BTEQ skripty ze svého vlastního vstupního adresáře definovaného v parametru **ownInputFolder.** Parametr je nepovinný a jeho defaultní hodnota je false.

#### Parametr – ownInputFolder

Tento parametr uživateli slouží k určení složky, ze které se mají brát vstupní BTEQ skripty. Parametr je povinný jen v případě, že je **fromOwnInputFolder** nastaven na true. Pokud tomu tak není, tak je tento parametr ignorován a vstupní skripty se berou z defaultního vstupního adresáře, kterým je root/input. Root je základní adresář nástroje (více informací o adresářové struktuře je v kapitole 5.2.3).

#### Parametr – createFilesForIFPC

Nastavením tohoto parametru na true může uživatel předat nástroji informaci, že chce zprocesované BTEQ skripty vložit i do cílového IFPC adresáře definovaného v parametru **IFPCFolder.** Parametr je nepovinný, ale jeho defaultní hodnota je true.

#### Parametr – IFPCFolder

Tento parametr uživateli slouží k určení cílového adresáře na IFPC file systému, do kterého mají být uloženy zprocesované BTEQ skripty. Parametr je povinný jen v případě, že je **createFilesForIFPC** nastaven na true. Pokud tomu tak není, tak je tento parametr ignorován a vstupní skripty se do adresáře na IFPC neukládají. Jelikož je parametr **createFilesForIFPC** defaultně nastaven na true, tak je nezbytné, aby měl parametr **IFPCFolder** nastavenou správnou cestu i pro defaultní pracovní běh. Pro instance nástroje, které budou přiřazeny ke konkrétním workflow, bude zpravidla stačit nastavit cestu při prvním běhu nástroje a poté už tento parametr neměnit.

#### Parametr – isConversionWanted

Nastavením tohoto parametru na true může uživatel předat nástroji informaci, že chce zprocesované BTEQ skripty automaticky zkonvertovat do podoby, ve které budou skripty spustitelné v prostředí uvedeném v parametru **targetEnvironment,** nebo do podoby určené vlastním konfiguračním souborem uvedeném v parametru **configurationFile**. Parametr je nepovinný, ale jeho defaultní hodnota je true.

#### Parametr – doOnlyConversion

Nastavením tohoto parametru na true může uživatel předat nástroji informaci, že chce ve skriptech ponechat veškerý obsah a následně provést konverzi**.** Parametr je nepovinný a jeho defaultní hodnota je false.

#### Parametr – ownConfigurationFile

Nastavením tohoto parametru na true může uživatel předat nástroji informaci, že chce zprocesované BTEQ skripty zkonvertovat pomocí svého vlastního konfiguračního souboru uvedeného v parametru **configurationFile.** Parametr je nepovinný a jeho defaultní hodnota je false.

#### Parametr – configurationFile

Tento parametr uživateli slouží k určení názvu konfiguračního souboru, který se má při konverzi použít. Aby byl soubor správně nalezen, tak musí být uveden celý název souboru i s jeho typem (např. ownConfiguration.cfg) a soubor musí být uložen ve správné složce (root/src/cfg – viz adresářová struktura v kapitole 5.2.3). Parametr je povinný jen v případě, že jsou **ownConfigurationFile** i **isConversionWanted** oba nastaveny na true.

#### Parametr – targetEnvironment

Tento parametr uživateli slouží k určení cílového prostředí pro konverzi. Nástroj pak překonvertuje všechny BTEQ skripty tak, aby šly na tomto prostředí spustit a aby k uložení načtených dat používaly databáze tohoto prostředí. Aby mohla být taková konverze provedena, tak musí existovat soubor root\src\cfg\defaultConfigurationPattern.cfg, jehož pomocí je následně vygenerován konkrétní konfigurační soubor použitý ke konverzi do požadovaného prostředí. Pokud tento soubor nebude nalezen, tak bude uživatel upozorněn varováním a proces bude pokračovat bez konverze. Parametr je povinný v případě, že je **isConversionWanted** nastavený na true a **ownConfigurationFile** na false. Defaultní hodnota parametru je dev, což je označení vývojového prostředí datového skladu. Při špatné specifikaci parametru bude tedy provedena konverze do vývojového prostředí.

#### Parametr – productionDataLoad

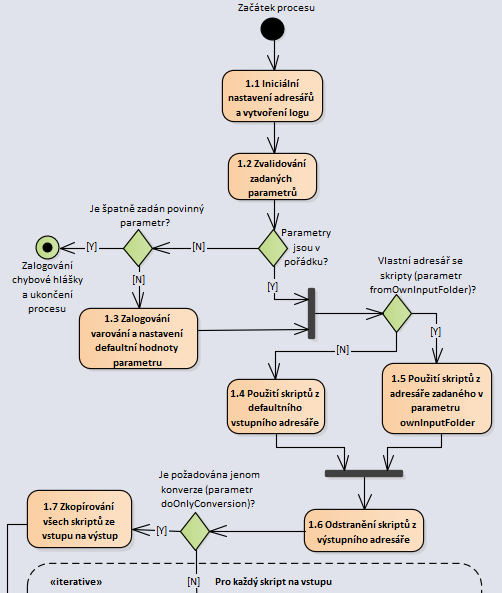
Nastavením tohoto parametru na true může uživatel předat nástroji informaci, že chce provést konverzi skriptů tak, aby byl ve skriptech zkonvertován jen obsah nezbytně nutný pro spuštění skriptu v prostředí uvedeném v **targetEnvironment.** Jedná se například o připojení k databázi, global temporary tabulky, error tabulky a cílové tabulky.Všechen ostatní obsah bude ponechán v produkčním stavu. V následném testovacím nápočtu se tedy budou načítat zdrojová data z produkčního prostředí. Parametr je nepovinný a jeho defaultní hodnota je false.

# Popis procesu nástroje

Pro znázornění procesu nástroje jsou použity celkem tři diagramy aktivit. První diagram znázorňuje hlavní řídící proces, který obsahuje kromě akcí také dvě dílčí aktivity. Ty jsou dále znázorněny pomocí vlastních diagramů aktivit.

#### Diagram 1 - Hlavní řídící proces

V rámci hlavního řídícího procesu probíhá hlavní rozhodování o prováděných akcích. Z důvodu velikosti je diagram rozdělen na dvě části. První část je vidět na následujícím obrázku.



První akcí je nastavení základních adresářů potřebných pro další akce nástroje a nedefinovaných uživatelem. Pokud některé adresáře neexistují, tak je nástroj založí (k takové situaci může dojít například při prvním spuštění nástroje). Mezi těmito adresáři je i adresář s logy, kde se zároveň vytvoří log, do kterého se budou vkládat informace o následujícím procesu nástroje. Pro každé spuštění nástroje bude vznikat nový log.

Následuje akce, v jejímž rámci dochází k validaci parametrů zadaných uživatelem. Zkontrolovány jsou jen parametry, které jsou důležité pro požadovaný proces. Pokud například není požadována konverze (**isConversionWanted** má hodnotu false), tak nejsou kontrolovány parametry týkající se konverze. Pokud je špatně zadán povinný parametr (například nesprávná cesta v parametru **ownInputFolder**, pokud je parametr **fromOwnInputFolder** roven true), tak je proces zastaven a v logu je vypsána chybová hláška. Pokud je špatně zadán nepovinný parametr, tak je v logu zaznamenáno jen varování a proces pokračuje s defaultní hodnotou parametru.

Pokračováním procesu je rozhodnutí, zda uživatel chce použít skripty z vlastního adresáře (parametr **fromOwnInputFolder**). Pokud ano, tak jsou v dalším procesu používány skripty z adresáře zadaného v parametru **ownInputFolder**. V opačném případě jsou používány skripty ze základního vstupního adresáře (Root/input). Dále jsou v rámci akce 1.6 odstraněny všechny soubory z výstupního adresáře (Root/output).

Následuje rozvětvení procesu podle toho, zda je požadována jen konverze (parametr **doOnlyConversion**). Pokud ano, tak je přeskočeno procesování jednotlivých skriptů a všechny skripty ze vstupního adresáře jsou zkopírovány do výstupního adresáře. Proces pokračuje konverzí skriptů. K tomuto může dojít jen v případě, že je požadována konverze (parametr **isConversionWanted** = true). Když konverze požadována není, tak se procesování skriptů nepřeskakuje.

Procesování jednotlivých skriptů a další akce v rámci hlavního řídícího procesu jsou vidět na obrázku vyobrazeném na další stránce.

Procesování skriptů je uzavřeno v rozšiřující oblasti. Akce v této oblasti se provádí pro každou položku v kolekci (v tomto případě pro každý skript ve vstupním adresáři). Je použita tzv. iterační rozšiřující oblast, v jejímž rámci se zpracovávají vstupní prvky popořadě (jeden po druhém).

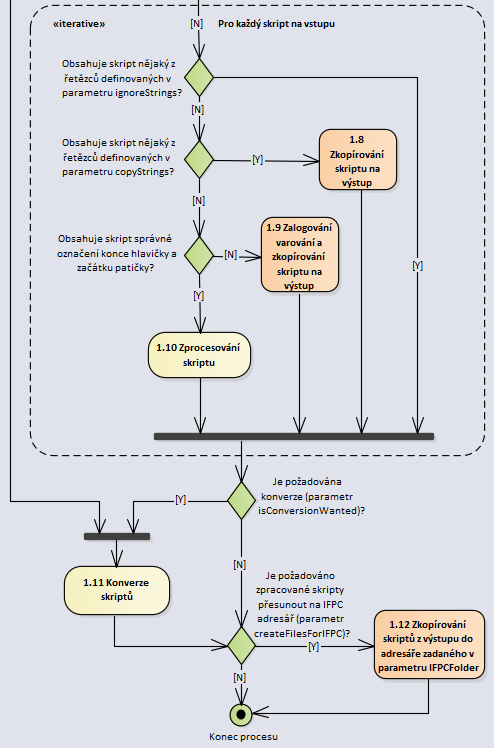
U každého skriptu je nejdříve rozhodnuto, zda neobsahuje některý z řetězců definovaných v parametrech **ignoreFileStrings** a **copyFileStrings**. První se vyhodnocují řetězce z **ignoreFileStrings** a pokud je některý z řetězců ve skriptu nalezen, tak se skript ponechá ve vstupním adresáři a nic se s ním dále neprovádí. Pokud je nalezen některý z řetězců v **copyFileStrings**, tak je skript zkopírován do výstupního adresáře s celým svým obsahem.

Dále se u každého skriptu kontroluje existence správného označení konce hlavičky a začátku patičky. Správnost obou označení je nezbytně nutná pro správné promazání skriptu a zajištění, že skript bude spustitelný. Pokud tedy označení nejsou správná, tak není možné skript promazávat a skript je zkopírován do výstupního adresáře s celým svým obsahem. Zároveň je zaznamenáno varování v logu.

Pokud skript neobsahuje žádný z hledaných řetězců a jsou v něm správně označeny konec hlavičky a začátek patičky, je skript připuštěn k samotnému zprocesování. Zprocesování je znázorněno pomocí vlastního aktivity diagramu (Diagram 2 – Zprocesování BTEQ skriptu).

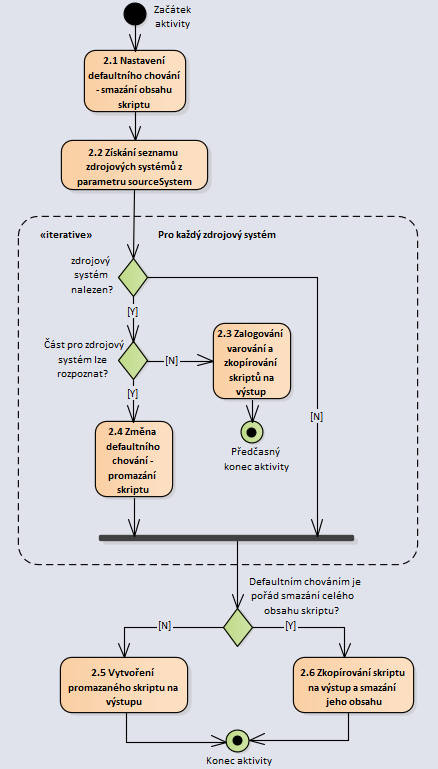
Po proběhnutí aktivit v rozšiřující oblasti pro každý skript ze vstupního adresáře je rozhodováno, zda mají být skripty zkonvertovány (parametr **isConversionWanted**). Pokud ano, tak proběhne konverze všech skriptů ve výstupním adresáři. Průběh konverze je znázorňen pomocí vlastního diagramu aktivity (Diagram 3 – Konverze skriptů).

Proces pokračuje rozhodnutím, zda mají být skripty z výstupního souboru zkopírovány do adresáře na IFPC file systému a stávající skripty nahrazeny (parametr **createFilesForIFPC**). Pokud ano, tak jsou skripty z výstupního adresáře zkopírovány do adresáře uvedeného v parametru **IFPCFolder.** Po této akci nebo v případě, že je **createFilesForIFPC** rovno false,následuje ukončení celého procesu.



#### Diagram 2 - Zprocesování BTEQ skriptu

Akce prováděné v rámci procesování každého BTEQ skriptu, který splňuje patřičné podmínky a je k tomu hlavním řídícím procesem přiveden, jsou vidět na samostatném diagramu aktivity na následujícím obrázku.



Procesování BTEQ skriptu začíná nastavením defaultního chování nástroje, kterým je smazání celého obsahu skriptu. To je předpokládané nejčastější chování, ke kterému dochází, pokud skript neobsahuje ani jednu transformaci pro kterýkoliv ze zdrojových systémů uvedených v parametru **sourceSystem**.

Další akcí je získání seznamu zdrojových systémů uvedených v parametru **sourceSystem**. Několik navazujících akcí je uzavřeno v iterační rozšiřující oblasti, v jejímž rámci akce probíhají pro každý zdrojový systém ze získaného seznamu.

Pokud část pro daný zdrojový systém není ve skriptu nalezena, tak nejsou provedeny žádné akce a pokračuje se s definovaným defaultním chováním. Pokud je část pro zdrojový systém nalezena, ale nejde rozpoznat její začátek a konec, tak je zaznamenáno varování v logu a skript je s celým svým obsahem zkopírován do výstupního adresáře. Proces aktivity je tím ukončen. K tomu může dojít, pokud:

* je nalezeno více začátků nebo konců,
* je nalezen pouze začátek nebo konec nebo
* je nalezen dříve konec než začátek.

K výše uvedeným nestandardním situacím může docházet, protože jednotlivé části jsou ve skriptech oddělovány řetězci, které píší sami vývojáři. Vývojáři při takovém psaní můžou udělat chybu. Nástroj by tedy měl umět na tyto omyly reagovat a nenarušit tím celý proces.

V případě, že je část pro zdrojový systém ve skriptu nalezena a je možné určit její konec a začátek, tak je změněno defaultní chování nástroje na promazání skriptu. To znamená, že následně bude tato část ve skriptu ponechána a ostatní nepotřebné části budou odmazány.

Poté, co výše uvedené akce proběhnou pro všechny zdrojové systémy, tak dojde k vyhodnocení další aktivity, která se bude provádět. Pokud defaultní aktivita zůstala nezměněna, tak dojde ke zkopírování skriptu do výstupního adresáře a následnému smazání celého jeho obsahu.

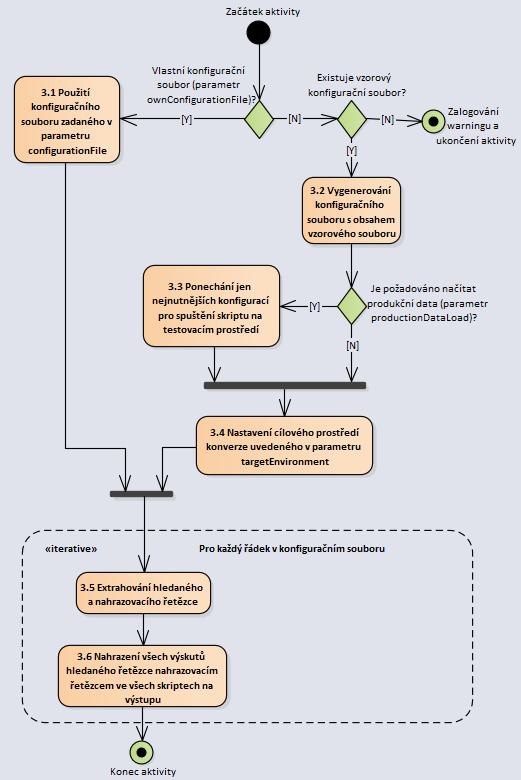
Pokud se defaultní chování změnilo, tak dojde k vytvoření nového skriptu ve výstupním adresáři se stejným názvem. Do tohoto skriptu budou z toho původního (ve vstupním adresáři) zkopírovány jen potřebné části, kterými jsou:

* hlavička skriptu,
* přihlášení technickým uživatelem,
* jednotlivé části nalezených zdrojových systémů a
* patička skriptu.

Následuje ukončení aktivity.

#### Diagram 3 - Konverze skriptů

Konverze skriptů, ke které v rámci hlavního řídícího procesu dochází jen v případě, že je vyplněn parametr **isConversionWanted** hodnotou true, je znázorněna svým vlastním diagramem aktivit zobrazeném na obrázku na následující stránce.



K určení, jakým způsobem proběhne konverze, slouží konfigurační soubor. Konfigurační soubor obsahuje řetězce, které mají být ve skriptech hledány, a přiřazuje k nim hodnoty, kterými mají být při nalezení nahrazeny. K provedení konverze je konfigurační soubor nezbytný.

Nejdříve je rozhodnuto, zda chce uživatel ke konverzi použít vlastní konfigurační soubor (parametr **ownConfigurationFile** je roven true). Pokud ano, tak je pro konverzi použit konfigurační soubor uvedený v parametru **configurationFile**. Proces dále pokračuje samotnou konverzí. V opačném případě se bude provádět automatizovaná konverze. K jejímu provedení je zapotřebí, aby existoval vzorový konfigurační soubor. Pokud neexistuje, tak je v logu zaznamenáno varování a aktivita končí. Skripty tedy nebudou zkonvertovány.

Při nalezení vzorového konfiguračního souboru je dále vygenerován nový konfigurační soubor se stejným obsahem jako ten vzorový. Při samotné konverzi bude použit právě tento vygenerovaný konfigurační soubor. Ještě předtím je však upraven. V rámci jeho úpravy je nejdříve rozhodnuto, zda je požadován nápočet s použitím produkčních dat. Pokud ano, tak jsou v konfiguračním souboru ponechány jen konfigurace konverzí, které jsou naprosto nezbytné k tomu, aby skript proběhl na testovacím prostředí. Ostatní konfigurace jsou z konfiguračního souboru odmazány a jimi konvertované objekty jsou ponechány v produkčním stavu. To zapříčiní použití produkčních objektů obsahujících produkční data při následném testovacím nápočtu. Mezi objekty (části skriptů – viz kapitola 3.5), které je k provedení testovacího nápočtu nutné zkonvertovat, patří:

* přihlášení do databáze,
* nastavení metadat,
* global temporary tabulky,
* error tabulky a
* cílové tabulky.

V případě, že nápočet s použitím produkčních dat není požadován, se z konfiguračního souboru nic neodmazává a do testovací podoby jsou následně překonvertovány všechny objekty. Proces pokračuje další úpravou konfiguračního souboru, kterou je přizpůsobení konfigurací požadovanému cílovému prostředí uvedeném v parametru **targetEnvironment**. U všech konfigurací jsou nahrazující hodnoty upraveny tak, aby odpovídaly potřebným objektům v cílovém prostředí.

Poté, co je konfigurační soubor připraven (ať už vlastní nebo automaticky vytvořený), se konečně provádí konerze. Akce prováděné v rámci konverze jsou uzavřeny v iterativní rozšiřující oblasti a probíhají pro každý řádek (konfiguraci) v konfiguračním souboru. Z každého řádku jsou nejdříve extrahovány hledaný řetězec a nahrazující hodnota. Poté jsou ve všech skriptech ve výstupním adresáři všechny výskyty hledaného řetězce nahrazeny nahrazující hodnotou. Po proběhnutí těchto akcí pro všechny řádky konfiguračního souboru jsou skripty ve výstupním adresáři zkonvertovány a aktivita je ukončena.

# Spuštění nástroje

Nástroj se spouští spuštěním skriptu parameters.ps1, který obsahuje parametry nástroje. Uživateli tedy stačí po úpravě parametrů rovnou spustit upravený skript. Obsah tohoto skriptu je vidět v následujícím výpise programového kódu.

1  <#### PARAMETERS - for additional info about each parameter check documentation ####>

2

3  <## File Processing Parameters ##>

4  #Following parameters can contain one or multiple strings divided by ';'

5  $sourceSystem = "SourceA;SourceB"

6  $ignoreFileStrings = ""

7  $copyFileStrings = ""

8

9  <## Folder Management Parameters ##>

10 $fromOwnInputFolder = $false

11 $ownInputFolder = "Path\_To\_Own\_Input\_Folder"

12 $createFilesForIFPC = $true

13 $IFPCFolder = "Path\_To\_IFPC" #this path has to be set correctly for default run

14

15 <## Conversion Parameters ##>

16 $isConversionWanted = $true

17 $doOnlyConversion = $false

18 $ownConfigurationFile = $false

19 $configurationFile = "ownConfiguration.cfg"

20 $targetEnvironment = "dev" #this parametr can contain only values: dev, test or testuat

21 $productionDataLoad = $false

22

23 . "$PSScriptRoot\functions.ps1"

24 Main

V souboru parameters.ps1 jsou uvedeny všechny definované parametry. Pro zvýšení přehlednosti jsou parametry rozděleny do tří sekcí, kterými jsou:

* parametry ovlivňující procesování skriptů,
* parametry ovlivňující organizaci adresářů a
* parametry ovlivňující konverzi skriptů.

Spustit soubor parameters.ps1 v prostředí PowerShell lze několika způsoby. Při všech je však nutné mít PowerShell nainstalovaný. V operačních systémech Windows je PowerShell předinstalovaný. Nejjednodušším způsobem je otevřít soubor parameters.ps1 v programu Windows PoweShell ISE. V něm je možné upravit parametry a následně skript také spustit. Pro demonstraci aktivit nástroje je potřeba použít nějaké vstupní skripty. K takovému účelu mohou sloužit testovací skripty ve složce test\_objects.