

ABRA Software a.s.
Jeremiášova 1422/7b
155 00 Praha 13

IČ 25097563
DIČ: CZ25097563
Zaps. v OR u Městského soudu
v Praze, odd. B, vložka 4475

TECHNICKÁ DOKUMENTACE SOUVISEJÍCÍ S PŘECHODEM NA NOVOU TECHNOLOGII ČÍSELNÍKŮ

Datum: 11.11.2019

Dokumentace odpovídá verzi ABRA Gen 20.0.1.

Pro aktuální informace o konkrétních aspektech implementace číselníků v novějších verzích systému kontaktujte oddělení zákaznické podpory výrobce nebo servisního konzultanta.

OBSAH

1.	PŘEHLED ZMĚN V DYNSQL A VE SKRIPTINGU	3
1.1	NOVÝ ČÍSELNÍK A DYNSQL	3
1.1.1	NOVÁ KLAUZULE {LIMIT}	3
1.1.2	PODMÍNKA HIDDEN	3
1.1.3	KLÍČ ID	3
1.1.4	POUŽITÍ FUNKCE COLLATE V KLÍČÍCH	3
1.1.5	PARAMETR ALWAYSUSED	3
1.2	ZMĚNY SKRIPTINGOVÝCH HÁČKŮ PRO DRUH SKRIPTU ČÍSELNÍK	3
1.2.1	HÁČKY V PACKAGE-ČÍSELNÍKU	3
1.2.2	HÁČKY V NOVÉM ČÍSELNÍKU	4
1.3	ZMĚNY SKRIPTINGOVÝCH OBÁLEK	4
1.3.1	ZMĚNĚNÉ VLASTNOSTI TŘÍD	4
1.3.2	NOVÉ VLASTNOSTI TŘÍD	5
1.3.3	NOVÉ METODY PROVÁDĚJÍCÍ JEDNOTLIVÉ VALIDAČNÍ SLUŽBY A JEJICH SPECIALIZOVANÉ VARIANTY	5
1.3.4	NOVÉ METODY ZVEŘEJNĚNÉ VÝHRADNĚ K TESTOVÁNÍ FUNKČNOSTI NAŠEPTÁVAČE VE SKRIPTINGOVÝCH TESTECH (V CLASSEXPLODERU BALÍČKŮ SKRIPTŮ MAJÍ V POPISU UPOZORNĚNÍ „NEPOUŽÍVAT“)	5
1.4	NEVIZUÁLNÍ SLUŽBY VOLANÉ VE SKRIPTINGU METODOU TNXCUSTOMROLL.VALIDATE NEBO NOVĚ VALIDATE2 – ZMĚNY V POUŽITÍ PARAMETRŮ (PARAMETR APARAMS: TNXPARAMETERS)	6
1.5	METODA TNXCUSTOMROLL.RELOAD VS. OBČERSTVENÍ DAT VE FORMULÁŘI, POUŽITÍ VE SKRIPTINGU	6
1.5.1	TNXCUSTOMROLL.RELOAD	6
1.5.2	OBČERSTVENÍ DAT VE FORMULÁŘI	6
1.6	PŘÍZNAK PRO ROZLIŠENÍ "PACKAGE-ČÍSELNÍK" X "NOVÝ ČÍSELNÍK" PRO SKRIPTING	7
1.7	ZMĚNA CHOVÁNÍ PŘI POUŽITÍ NEVIZUÁLNÍ SLUŽBY PREFILL	7
1.8	ZMĚNA CHOVÁNÍ PŘI POUŽITÍ NEVIZUÁLNÍ SLUŽBY CHECKONLY	7
1.9	NEVIZUÁLNÍ SLUŽBY A SKUPINY ZÁZNAMŮ	7
1.10	NOVÝ ČÍSELNÍK A SKUPINY ZÁZNAMŮ V OTEVŘENÉM ROZHRAŇÍ (OLE)	8
2.	PŘEHLED ODLIŠNOSTÍ PLYNOUCÍ Z POUŽÍVÁNÍ SQL PRO ZÍSKÁVÁNÍ DAT	8
2.1	FIREBIRD	8
2.2	ORACLE	8
3.	NEVIZUÁLNÍ SLUŽBY ČÍSELNÍKŮ	9
3.1	SLUŽBA CORRECTSELECTED	9
3.2	SLUŽBA FINDPART	10
3.3	SLUŽBA CHECKONLY	10
3.4	SLUŽBA LOOKUP	11
3.5	SLUŽBA PREFILL	12
3.6	SLUŽBA GETIDS	13

1. PŘEHLED ZMĚN V DYNSQL A VE SKRIPTINGU

1.1 Nový číselník a DynSQL

Čtení dat novým číselníkem je založeno na existujících definicích dynamických SQL.

SQL-fragmenty pro nevizuální služby, načítání dat do stránek, atp. se v novém číselníku vytváří prostřednictvím za běhu vytvářené speciální podmínky dynamického SQL.

1.1.1 Nová klauzule {LIMIT}

Nové číselníky vyžadují, aby jejich hlavní SQL select obsahoval klauzuli {LIMIT} – viz příklad hlavního SQL selectu číselníku Adresář firem:

```
SELECT {LIMIT} {FIELDS} FROM  
Firms A  
{JOIN}  
{WHERE}  
{ORDERBY}
```

1.1.2 Podmínka Hidden

Nové číselníky (pokud podporují skrývání) vyžadují, aby podmínka Hidden měla šablonu *HiddenInRollWithPaging*.

1.1.3 Klíč ID

Nové číselníky musí mít skrytý klíč *ID*.

1.1.4 Použití funkce Collate v klíčích

Řetězcové klíče nových číselníků nepoužívají v šabloně SQL funkci *Collate PXW_CSY*, protože zpomaluje jejich používání.

1.1.5 Parametr AlwaysUsed

Za určitých okolností mohla být z výsledného SQL selectu vyřazena podmínka s parametrem AlwaysUsed – to působilo chybné chování např. u podmínky Hidden vytvářené za běhu => pokud je nyní podmínka definována s parametrem AlwaysUsed=True, při její inicializaci (resp. vyčištění dynamického SQL metodou Clear) se tato hodnota vždy nastaví.

1.2 Změny skriptingových háček pro druh skriptu Číselník

Háčky pro druh skriptu „Číselník“ umožňují rozšířit filtrování/omezování dat číselníku. Stávající háčky v package-číselníku zůstávají beze změny, v novém číselníku je k dispozici pouze jeden (*_CreateDynSQLWhere_Hook*).

1.2.1 Háčky v package-číselníku

- **OnAddCustomKeys_Hook** : háček přidává uživatelský klíč, pro který se načítají data do paměti (do package), aby bylo podle něj možno záznamy řadit a vyhledávat
- **OnCanLoadData_Hook** : háčkem je možné zakázat výchozí načtení package číselníku
- **InFilter_Hook** : pomocí tohoto háčku je možné filtrovat data číselníku - vyvolává se pro každý záznam z daného číselníku. Skript pro každý záznam určuje, zda se daný záznam použije. Háček tedy umožňuje filtrovat číselník nad rámec podmínek systémového filtru/omezení.

1.2.2 Háčky v novém číselníku

- **OnAddCustomKeys_Hook** : háček není nadále použitý, má smysl pouze v package-číselníku, v novém číselníku se klíče přidávají do dynamického SQL-dotazu za běhu
- **OnCanLoadData_Hook** : háček není nadále použitý, má smysl pouze v package-číselníku
- **InFilter_Hook** : háček není nadále použitý, k filtrování nad rámec podmínek systémového filtru/omezení lze využít háček `_CreateDynSQLWhere_Hook`

Doporučení po přechodu na verzi s číselníkem (firem) s novým „motorem“: skripty na háčcích `InFilter_Hook`, `OnCanLoadData_Hook` a `OnAddCustomKeys_Hook` se v novém číselníku nespouští. Pokud tedy byly pro číselník definovány skripty na uvedených háčcích, doporučujeme je z balíčků skriptů odstranit.

Skripty, které obsahují uvedené háčky lze najít za pomoci omezení v agendě Balíčky skriptů, omezení za Řádky/Zdrojový kód skriptu. Identifikátor háčku je třeba zadat např. Takto: **InFilter_Hook**.

1.3 Změny skriptingových obálek

1.3.1 Změněné vlastnosti tříd

Vlastnosti označené **[malfunction]** nelze ve skriptingu používat, jejich použití vyvolá výjimku s hlášením „Vlastnost X třídy T není podporována.“. Skripty, které používají vlastnosti označené **[malfunction]**, je třeba opravit.

Vlastnosti označené **[deprecated]** jsou nadále funkční, v budoucnosti ale mohou být ze skriptingu odstraněny a doporučujeme místo nich používat vlastnosti nové.

1.3.1.1 TConditionLink

- property Children = [malfunction]
 - vlastnost je určena výhradně k internímu použití a ve skriptingu není nadále podporována (bez náhrady)

1.3.1.2 TNxDataItem

- property Collection = [malfunction]
 - vlastnost je určena výhradně k internímu použití a ve skriptingu není nadále podporována

1.3.1.3 TNxCustomRoll

- function Validate = [deprecated]
 - univerzální metoda, kde typ validační služby je dán vstupním parametrem
 - metoda je určena ke zrušení, plně ji nahrazují nové metody `LookUp`, `CheckOnly`, `FindPart`, `CorrectSelected` a `Prefill`

1.3.1.4 TNxBusinessRoll

Změny vlastností souvisejících s package:

- property Package
- property KeysDefinition
- procedure SetDefaultKeysDefinition
 - v package-číselníku zůstávají tyto vlastnosti beze změny
 - v novém číselníku nemají tyto vlastnosti smysl a jejich použití ve skriptingu vyvolá výjimku s hlášením: „Číselník XY nepodporuje vlastnost Package, ani vlastnosti od Package odvozené.“, pro nový číselník se tedy jedná o [malfunction]

1.3.2 Nové vlastnosti tříd

1.3.2.1 TNxCustomRoll

- function Validate2(AType: TNxValidate, AParams: TNxParameters, APrefilling: TNxPrefillingMethod): boolean;
 - volání validačních služeb číselníku – varianta metody Validate bez nutnosti předávat parametr AContext, který metoda nepotřebuje
 - metoda je [deprecated], doporučená náhrada: LookUp/CheckOnly/FindPart/CorrectSelected/Prefill

1.3.3 Nové metody provádějící jednotlivé validační služby a jejich specializované varianty

- function CheckOnly(AParams: TNxParameters): boolean
 - podle obsahu položky předané v AParams vyhledá záznam a vrátí jeho interní identifikátor (ID), v AParams může vrátit také hodnoty dalších položek podle nastavení AParams na vstupu
- function CheckOnly2(AFieldName: string, AFieldValue: string, AParams: TNxParameters): string
 - vrátí ID záznamu nalezeného podle řetězcové hodnoty položky zadané v AFieldName, v AParams na vstupu lze předat např. další podmínky pro vyhledání
- function CheckOnly3(AFieldName: string, AFieldValue: Extended, AParams: TNxParameters): string
 - vrátí ID záznamu nalezeného podle extended hodnoty položky zadané v AFieldName, v AParams na vstupu lze předat např. další podmínky pro vyhledání
- function CorrectSelected(AParams: TNxParameters): boolean
 - upraví seznam interních identifikátorů (ID) předaný v AParams, v AParams vrátí seznam bez neexistujících ID, výsledkem funkce je True, pokud se výstupní seznam liší od vstupního
- function FindPart(AParams: TNxParameters): boolean
 - vrátí True, pokud je podle části obsahu položky předané v AParams nalezen záznam, v AParams vrátí jeho interní identifikátor (ID) a celý obsah zadané položky
- function LookUp(AParams: TNxParameters): boolean
 - Vrábí True, pokud je nalezen záznam ID, předaného v AParams. V AParams pak vrátí hodnotu požadované položky, případně i dalších položek podle nastavení AParams na vstupu.
- function LookUp2(AID: string, AFieldName: string, AParams: TNxParameters): string
 - Vrábí hodnotu položky zadané v AFieldName ze záznamu nalezeného podle ID. V AParams na vstupu lze předat např. další podmínky pro vyhledání.
- function LookUp3(AID: string, AParams: TNxParameters): Boolean
 - Vrábí True, pokud je nalezen záznam zadaného ID. V AParams na vstupu lze předat např. další podmínky pro vyhledání.
- function Prefill(AParams: TNxParameters, APrefilling: TNxPrefillingMethod):
 - vyhledá hodnotu pro předvyplnění, pokud číselník obsahuje právě 1 záznam, hodnotu vrací v AParams, pro APrefilling = pmNone se bez hledání vrátí prázdný řetězec, pro pmOnlyIfOne se provede vyhledání, výsledkem funkce je vždy True

1.3.4 Nové metody zveřejněné výhradně k testování funkčnosti našeptávače ve skriptingových testech (v ClassExploreru balíčků skriptů mají v popisu upozornění „NEPOUŽÍVAT“)

- function NxInternal_GetWhispererItems(AValue: String): TNxInternal_WhispererItems

- volá metodu číselníku GetWhispererItems, která podle vstupních parametrů dohledá data pro našeptávač
- procedure NxInternal_ConfigWhisperer(ATextFieldToWhispererSource: string; AFilters: TNxParameters)
 - volá metodu číselníku ConfigWhisperer, která provádí základní nastavení našeptávače

1.3.4.1 TNxBusinessRoll

Obsahuje všechny změny a nové vlastnosti zděděné z TNxCustomRoll a navíc novou vlastnost definovanou v této třídě

- procedure TNxBusinessRoll.GetIDs2
 - specializovaná varianta metody GetIDs
 - vrátí seznam ID všech záznamů číselníku omezených filtrem za položku AFieldName

Nové třídy zveřejněné výhradně k testování funkčnosti našeptávače ve skriptingových testech (v ClassExploreru balíčků skriptů mají v popisu upozornění „NEPOUŽÍVAT“):

1.3.4.2 TNxInternal_WhispererItems

= řádky vrácené z funkce NxInternal_GetWhispererItems (viz Nové metody zveřejněné výhradně k testování funkčnosti našeptávače v TNxCustomRoll)

1.3.4.3 TNxInternal_WhispererItem

= řádek s výsledky funkce NxInternal_GetWhispererItems (viz Nové metody zveřejněné výhradně k testování funkčnosti našeptávače v TNxCustomRoll)

1.4 Nevizuální služby volané ve skriptingu metodou TNxCustomRoll.Validate nebo nově Validate2 – změny v použití parametrů (parametr AParams: TNxParameters)

Z kódu jednotlivých služeb byly v novém číselníku odstraněny parametry předávané v AParams:

- parametr _AsyncRequest – v novém číselníku se v kódu Abra Gen nevyužívá
- parametr _DataSetCookie – v novém číselníku se v kódu Abra Gen nevyužívá
- parametr _FoundInDB, služba LookUp – v novém číselníku nemá jeho použití smysl

1.5 Metoda TNxCustomRoll.Reload vs. občerstvení dat ve formuláři, použití ve skriptingu

1.5.1 TNxCustomRoll.Reload

- jedná se o metodu nevizuální části číselníku
- metoda je zveřejněná do skriptingu pro oba typy číselníků
- v package-číselníku metoda načte aktuální data z databáze do paměti (do Package)
- v novém číselníku metoda neprovádí nic (není to třeba, protože každý dotaz do číselníku získává data vždy z databáze)

1.5.2 Občerstvení dat ve formuláři

- ve vizuálu je občerstvení voláno funkcí Občerstvit
- lze je vyvolat také metodou RefreshData formuláře číselníku (metoda je zveřejněná do skriptingu pro oba typy číselníků)
- package-číselník při tom načte kompletní data z databáze do package voláním Reload
- nový číselník načte data aktuální stránky podle aktuálního zobrazení

=> v obou typech číselníku tedy občerstvení načte aktuální data z databáze, rozdíl je jen v množství načítaných dat.

1.6 Příznak pro rozlišení "package-číselník" x "nový číselník" pro skripting

Nový číselník je funkčně velmi podobný package-číselníku, rozlišovací příznak tedy zřejmě nebude nikdy ve skriptingu zapotřebí.

1.7 Změna chování při použití nevizuální služby Prefill

Služba Prefill vrací záznam pro předvyplnění, pokud číselník s podmínkami obsahuje právě 1 viditelný záznam (podrobný popis v kap. 3. Nevizuální služby číselníků).

V package-číselníku služba Prefill vyhledává záznam pro předvyplnění pouze v záznamech, které jsou aktuálně v package (tj. načtené v paměti) => pokud tedy např. skriptem přidáme záznam do tabulky číselníku v databázi (package z ní původně získala jen 1 záznam), pak služba Prefill v package-číselníku vrátí uvedený původní záznam, který má v paměti. Přidaný záznam se v tomto případě ignoruje.

V novém číselníku služba Prefill provádí dotaz do databáze, tj. pracuje s aktuálními daty z databáze => pokud např. skriptem přidáme záznam do tabulky nového číselníku v databázi, služba Prefill v tomto novém číselníku nevrátí nic, protože v databázi najde 2 záznamy.

Shodného výsledku na obou typech číselníků dosáhneme, pokud před voláním služby Prefill v package-číselníku provedeme jeho občerstvení.

1.8 Změna chování při použití nevizuální služby CheckOnly

Služba CheckOnly slouží k nevizuálnímu vyhledání záznamu v číselníku podle hodnoty libovolného údaje záznamu (podrobný popis v kap. 3. Nevizuální služby číselníků).

V package-číselníku služba CheckOnly se zadaným ID a dalším údajem záznamu a s parametry `_MultiChoice=False`, `_ExactLookup=False` a `_NewObject=False` hledá ve všech záznamech, tj. i mezi skrytými. Pokud tedy nalezneme záznam se zadaným ID a dalším údajem, vrací True bez ohledu na příznak Hidden.

V novém číselníku služba uvedené parametry nevyužívá a rovnou pracuje pouze s viditelnými záznamy. Za stejných podmínek tedy skrytý záznam v novém číselníku nenajde a vrátí False.

Vzhledem k primárnímu určení služby pro vyhledávání jen ve viditelných záznamech a tomu odpovídajícímu použití v kódu Abra Gen je chování v novém číselníku korektní.

1.9 Nevizuální služby a skupiny záznamů

Nový číselník skupinování záznamů nepodporuje. Dříve definované skupiny tedy záznamy nového číselníku nefiltrují.

- **Služba Prefill:** v novém číselníku se chová odlišně od package-číselníku. Pokud tedy tabulka číselníku obsahuje více než 1 záznam, pak platí:
 - **package-číselník:** pokud má definovanou výchozí skupinu záznamů tak, aby vracela pouze jeden záznam, služba Prefill vrátí právě tento záznam (package číselníku obsahuje pouze záznam, daný uvedenou skupinou)

- **nový číselník:** neumožňuje skupiny používat, pro službu Prefill je tedy viditelný víc než 1 záznam a služba Prefill v tomto případě nevrací nic.

Tato změna chování se projeví např. u předvyplnění firmy na hlavičce nového dokladu, pokud není ve firemních údajích nastavená firma pro předvyplnění.

- **Ostatní služby:** uvedená zásadní změna výsledku se neprojeví, protože u ostatních služeb není úspěch hledání daný nalezením právě jednoho záznamu. Obecně se ale bez podpory skupinování u všech služeb číselníku mění množina záznamů, ve kterých se vyhledává, a pokud nebude zavedena alternativní funkčnost, je třeba s touto změnou počítat.

1.10 Nový číselník a skupiny záznamů v Otevřeném rozhraní (OLE)

V konfiguraci Abra Gen (soubor *Nexus.cfg*) v sekci *[OleSection]* lze parametrem *RollGroups* zapnout podporu použití skupin záznamů v OLE.

Definování skupin záznamů umožňuje rozdělit záznamy v package-číselníku do skupin a pracovat pak pouze s vybranou skupinou – do paměti se tak načte pouze vybraná skupina.

Nový číselník skupinování nepodporuje a dříve definované skupiny tedy data nijak nefiltrují. Také nastavení parametru *RollGroups* v *Nexus.cfg* nemá žádný vliv ani na získávání dat novým číselníkem v OLE.

Data v číselníku nyní lze omezit pouze podmínkami filtru. Jako náhradu je tak možné použít např. uložený filtr (omezení).

2. PŘEHLED ODLIŠNOSTÍ PLYNOUCÍ Z POUŽÍVÁNÍ SQL PRO ZÍSKÁVÁNÍ DAT

2.1 Firebird

Dotazy je občas nutné upravit pomocí hintů pro kompilator. V případě Firebirdu se jedná o doplnění přičtení 0 nebo prázdného řetězce nebo použití jiné operace neměnicí výsledek ke sloupcům do where části podmínky. Doplnění této operace ke sloupci způsobí, že optimalizátor pro naplánování dotazu nepoužije index přes tento sloupec.

Příklad:

```
SELECT FIRST 100 Code, ID
FROM Firms
WHERE Firm_ID||'' is Null and Hidden||''='N' and
CODE >= '1900870'
AND NOT (CODE||'' = '1900870' AND ID <= 'P5WG100101')
ORDER BY Code ASC NULLS FIRST, ID ASC NULLS FIRST;
```

Zde jsou hinty použité na sloupcích *Firm_ID* a *Hidden* a způsobí, že se vykonání nenaplánuje přes index spojený s jedním z těchto sloupců. V tomto případě to totiž bylo pomalejší. Tyto optimalizace se dělají jen v nutných případech a vždy je lépe je odladit na konkrétních datech.

Pro řazení není možné použít *COLLATE*. Číselník firem nově načítá stránky dat z databáze. Aby toto načítání bylo rychlé, je třeba mít index, který bude stejný jako položky v *ORDER BY*. Při načítání se postupuje podle tohoto indexu a není třeba načítat všechna data, a pak je řadit. Index musí obsahovat navíc sloupec *ID*, který zajistí jeho unikátnost. Ve FB je ale možné vytvořit tento tzv. *expression index* (z důvodu přítomnosti *collate*) jen s jedním sloupcem. Ze stejného důvodu nepůjde v *order by* použít i jiné funkce jako *Upper* apod. Při vytváření databáze v systému *ABRA Gen* se ale pro sloupec nebo doménu nastaví *collate* podle lokalizace aktuální instalace, takže ve výsledku se nic nemění.

2.2 Oracle

Pro správné řazení na Oracle je použito linguistické řazení, které řadí SQL dotaz podle funkce NLS_SORT. Řazení se liší od původního řazení. Původní řazení se provedlo celé na klientovi a nemělo s řazením v databázi nic společného. Toto řazení třídí jen podle secondary level collation. Tzn. Že nebere do úvahy znaky jako ., #, ", + atd. Obzvláště nápadné to je, pokud těmito znaky tříděné položky začínají. Jeví se jako nesetříděné. Setříděné jsou, ale jen podle znaků abecedy.

V SQL dotazu nejsou v sekci Order By použity klauzule asc null first pro vzestupné třídění a asc null last pro sestupné třídění. S těmito klauzulemi se nepodařilo dosáhnout rychlého prováděcího plánu podle indexu dle řadících sloupců. Defaultní hodnoty jsou v Oraclu přesně opačné. Při vzestupném třídění jsou null hodnoty řazeny na konci, při sestupném třídění na začátku.

3. NEVIZUÁLNÍ SLUŽBY ČÍSELNÍKŮ

V této části se nejedná o popis změn, ale o informace, které jsou užitečné pro používání validačních služeb číselníků dostupných ze skriptingu a nejsou jinde podrobně zveřejněné.

3.1 Služba CorrectSelected

Slouží ke zkorigování seznamu interních identifikátorů (ID) daného číselníku. Korekce spočívá ve vypuštění ID, která nevyhovují generickým filtrům nebo neexistují v databázi. Pokud došlo ke zkorigování, nebo bylo požadováno vrácení všech ID, je výsledkem funkce True. Pokud se ve výstupním seznamu vrátí pouze 1 ID a je-li zadaný název fieldu v parametru `_TextField`, v parametru `_Text` se vrátí hodnota fieldu z `_TextField`.

Na službu mohou být aplikované i předané generické filtry. Dokumentace ke generickým filtrům je v helpu, kap. Administrace/Technická dokumentace/AbraOLE/Popis tříd a typů/Filtrování číselníků.

Položky `AParams` slouží k předání vstupních parametrů a k vrácení výstupních hodnot:

vstup

AParams

- `_All` (default = False): příznak požadavku vrácení všech ID záznamů číselníku, uplatní se jen pro `Mask<>0` (číselník je chráněný)
- `_SelectedList` (může chybět): seznam ID ke zkorigování (každé jednotlivé ID je předáno jako jeden `TNxParameter`)
- `_Text` (může chybět): parametr pro návratovou hodnotu definovaného fieldu, pokud je vráceno právě 1 ID
- `_TextField` (může chybět): field, jehož hodnotu chceme vrátit
- `_NewObject` (default = False): příznak dotazu na číselník z nově vytvářeného objektu, je-li (`_NewObject = True`) & (`_ExactLookup = True`) & číselník podporuje skrývání => z hledání jsou vynechány skryté záznamy
- `_ExactLookup` (default = False): příznak požadavku na zohlednění skrývání a oprávnění k záznamům, je-li `_ExactLookup = True` => z hledání jsou vynechány záznamy, ke kterým není oprávnění
- `|<field_name>`: parametry s předznamenáním | (svislá čára) definují požadavek na vrácení hodnoty fieldu `<field_name>`

Pokud je u parametru uvedeno „(může chybět)“, pak není definovaná jeho defaultní hodnota a služba neprovede akci s parametrem spojenou, např. pokud chybí `_TextField`, nevrací se hodnota žádného fieldu, i když byl nějaký záznam nalezen.

Pokud je u parametru uvedeno “(default = xy)” a parametr na vstupu chybí, pak je službou použita tato hodnota.

výstup

Result

- jako výsledek funkce vrátí True, pokud došlo ke zkorigování, nebo pokud bylo požadováno vrácení všech ID (`_All = True`), jinak False

AParams

- `_SelectedList` : výsledný seznam ID po korekci (každé jednotlivé ID je předáno jako jeden `TNxParameter`)
- `_Text`: obsahuje návratovou hodnotu definovaného fieldu, pokud je vráceno právě 1 ID

- *<field_name>* : obsahuje návratovou hodnotu fieldu zadaného na vstupu s předznamenáním | (svislá čára), pokud se vrátil jen jeden záznam
- [AParms mohou obsahovat další parametry, samotný CorrectSelected je ale nevyužívá.]

3.2 Služba FindPart

Slouží k nevizuálnímu vyhledání záznamu v číselníku podle části nějakého údaje. Respektive pomocí této funkce je možné dohledat první záznam, který začíná zadaným řetězcem. Pro údaje jiných než řetězcových typů se hledá podle úplně zadané hodnoty. Pokud byl záznam dohledán, je vráceno True.

Parametr *_TextField* definuje, podle jakého údaje se bude hledat (Code, Name ap.).

Parametr *_Text* obsahuje hledaný řetězec.

Parametr *_ID* obsahuje interní identifikátor nalezeného záznamu.

V parametru *_Text* je také v případě úspěšného nalezení vrácen celý údaj. Pokud číselník podporuje skrývání, vyhledává se pouze ve viditelných záznamech.

Na službu mohou být aplikované i předané generické filtry. Dokumentace ke generickým filtrům je v helpu, kap. Administrace/Technická dokumentace/AbraOLE/Popis tříd a typů/Filtrování číselníků.

Položky AParams slouží k předání vstupních parametrů a k vrácení výstupních hodnot:

vstup

AParams

- *_ID* (může chybět): je-li parametr zadaný, slouží k vrácení OID nalezeného záznamu
- *_Text* (může chybět): hledaná hodnota fieldu zadaného v *_TextField*
- *_TextField* (může chybět): field, jehož hodnota/hodnoty se mají vyhledat
- |*<field_name>*: parametry s předznamenáním | (svislá čára) definují požadavek na vrácení hodnoty fieldu *<field_name>*

Pokud je u parametru uvedeno „(může chybět)“, pak není definovaná jeho defaultní hodnota a služba neprovede akci s parametrem spojenou, např. pokud chybí *_TextField*, nevrací se hodnota žádného fieldu, i když byl nějaký záznam nalezen.

Pokud je u parametru uvedeno „(default = xy)“ a parametr na vstupu chybí, pak je službou použita tato hodnota.

Výstup

Result

- jako výsledek funkce vrátí True, pokud je záznam nalezen

AParams

- *_ID* : obsahuje OID nalezeného záznamu
- *_Text*: obsahuje (celou) návratovou hodnotu definovaného fieldu
- *<field_name>* : obsahuje návratovou hodnotu fieldu zadaného na vstupu s předznamenáním | (svislá čára), hodnota každého takto zadaného fieldu se vrací ve zvláštním parametru.

[AParms mohou obsahovat další parametry, samotný FindPart je ale nevyužívá.]

3.3 Služba CheckOnly

Slouží k nevizuálnímu vyhledání záznamu v číselníku. Pokud byl záznam nalezen, je vráceno True.

Parametr *_TextField* definuje, podle jakého údaje se bude hledat (Code, Name ap.). Parametr *_Text* obsahuje hledaný řetězec.

Parametr *_ID* obsahuje interní identifikátor nalezeného záznamu.

Parametr *_ID* je ale vstupně/výstupní, a číselník ho použije v případě, že pro předaný Text existuje více záznamů.

Na službu mohou být aplikované i předané generické filtry. Dokumentace ke generickým filtrům je v helpu, kap. Administrace/Technická dokumentace/AbraOLE/Popis tříd a typů/Filtrování číselníků.

Položky AParams slouží k předání vstupních parametrů a k vrácení výstupních hodnot:

vstup

AParams

- `_ID` (může chybět): OID hledaného záznamu/seznam OID oddělených středníkem pro `_MultiChoice = True`, pokud je parametr zadán (nemusí být), vyhledává se přednostně podle ID, hledání podle OID je úspěšné, pokud se obsah fieldu zadaném v `_TextField` shoduje s hodnotou v parametru `_Text`
- `_Text` (může chybět): hledaná hodnota (nebo hodnoty pro `MultiChoice`) fieldu zadaného v `_TextField`
- `_TextField` (může chybět): field, jehož hodnota/hodnoty se mají vyhledat
- `_ForcedField` (default = False):
- je-li = False & není zadán vstupní ID => nehledá se, vrátí se True a prázdná hodnota
- je-li = False & je zadán vstupní ID => provede se hledání
- je-li = True => provede se hledání (v tomto případě se při nezadaném ID vrátí False)
- `_MultiChoice` (default = False): ID a hodnota fieldu `_TextField` se hledá k více zadaným vstupním hodnotám
- `_NewObject` (default = False): příznak dotazu na číselník z nově vytvářeného objektu (např. dotaz na firmu z nové FV)
- je-li (`_NewObject = True`) & (`_ExactLookup = True`) & číselník podporuje skrývání -> z hledání jsou vynechány skryté záznamy
- je-li zadáno ID a (`_NewObject = False`) nebo (`_ExactLookup = False`), hledání zahrnuje i skryté záznamy
- `_ExactLookup` (default = False): příznak požadavku na zohlednění skrývání a oprávnění k záznamům
- je-li `_ExactLookup = True` => z hledání jsou vynechány záznamy, ke kterým není oprávnění
- `|<field_name>`: parametry s předznamenáním | (svislá čára) definují požadavek na vrácení hodnoty fieldu `<field_name>`

Pokud je u parametru uvedeno „(může chybět)“, pak není definovaná jeho defaultní hodnota a služba neprovede akci s parametrem spojenou, např. pokud chybí `_TextField`, nevrací se hodnota žádného fieldu, i když byl nějaký záznam nalezen.

Pokud je u parametru uvedeno „(default = xy)“ a parametr na vstupu chybí, pak je službou použita tato hodnota.

výstup

Result

- jako výsledek funkce vrátí True, pokud je záznam nalezen

AParams

- `_ID` : obsahuje OID nalezeného záznamu/seznam nalezených OID oddělených středníkem pro `_MultiChoice = True`
- `_Text`: obsahuje návratovou hodnotu (nebo hodnoty pro `MultiChoice`) definovaného fieldu
- `<field_name>` : obsahuje návratovou hodnotu fieldu zadaného na vstupu s předznamenáním | (svislá čára), hodnota každého takto zadaného fieldu se vrací ve zvláštním parametru.

[AParams mohou obsahovat další parametry, samotný `CheckOnly` je ale nevyužívá.]

3.4 Služba LookUp

Slouží k nevizuálnímu vyhledání záznamu v číselníku v případě, že je znám interní identifikátor záznamu (ID). Pokud byl záznam nalezen, je vráceno True.

Parametr `_ID` obsahuje interní identifikátor hledaného záznamu.

Parametr `_TextField` definuje, jaký údaj je požadován (např. Code, Name).

Parametr `_Text` obsahuje nalezený řetězec.

Na službu mohou být aplikované i předané generické filtry. Dokumentace ke generickým filtrům je v helpu, kap. Administrace/Technická dokumentace/AbraOLE/Popis tříd a typů/Filtrování číselníků.

Položky AParams slouží k předání vstupních parametrů a k vrácení výstupních hodnot:

vstup

AParms

- `_ID` (může chybět): OID hledaného záznamu/seznam OID oddělených středníkem pro `_MultiChoice = True`
- `_Text` (může chybět): hledaná hodnota (nebo hodnoty pro `MultiChoice`)
- `_TextField` (může chybět): field, jehož hodnotu chceme vrátit
- `_ForcedField` (default = `False`):
- je-li = `False` & není zadán vstupní ID => nehledá se, vrátí se `True` a prázdná hodnota
- je-li = `False` & je zadán vstupní ID => provede se hledání
- je-li = `True` => provede se hledání (v tomto případě se při nezadaném ID vrátí `False`)
- `_MultiChoice` (default = `False`): hodnota fieldu `_TextField` se hledá k více zadaným ID
- `_NewObject` (default = `False`): příznak dotazu na číselník z nově vytvářeného objektu (např. dotaz na firmu z nové FV)
- je-li (`_NewObject = True`) & (`_ExactLookup = True`) & číselník podporuje skrývání -> z hledání jsou vynechány skryté záznamy
- `_ExactLookup` (default = `False`): příznak požadavku na zohlednění skrývání a oprávnění k záznamům
- je-li `_ExactLookup = True` => z hledání jsou vynechány záznamy, ke kterým není oprávnění
- `|<field_name>`: parametry s předznamenáním `|` (svislá čára) definují požadavek na vrácení hodnoty fieldu `<field_name>`

Pokud je u parametru uvedeno "(může chybět)", pak není definovaná jeho defaultní hodnota a služba neprovede akci s parametrem spojenou, např. pokud chybí `_TextField`, nevrací se hodnota žádného fieldu, i když byl nějaký záznam nalezen. Pokud chybí `_ID`, služba nic nehledá a vrátí rovnou `False`.

Pokud je u parametru uvedeno "(default = xy)" a parametr na vstupu chybí, pak je službou použita tato hodnota.

výstup

Result

- jako výsledek funkce vrátí `True`, pokud je záznam nalezen, pro `MultiChoice = True` funkce vrátí `True`, pokud je nalezen aspoň 1 záznam

AParms

- `_Text`: obsahuje návratovou hodnotu (nebo hodnoty pro `MultiChoice`) definovaného fieldu

`<field_name>` : obsahuje návratovou hodnotu fieldu zadaného na vstupu s předznamenáním `|` (svislá čára), hodnota každého takto zadaného fieldu se vrací ve zvláštním parametru, pro `MultiChoice = True` se vrací hodnoty takto zadaných fieldů pouze pro první nalezený záznam.

[AParms mohou obsahovat další parametry, samotný `LookUp` je ale nevyužívá.]

3.5 Služba Prefill

Slouží k nevizuálnímu vyhledání hodnoty pro předvyplnění, pokud číselník obsahuje právě 1 záznam. Výsledkem volání služby je vždy `True`.

Parametr `_TextField` definuje, jaký údaj je požadován (např. `Code`, `Name`).

Parametr `_Text` obsahuje na výstupu nalezený řetězec.

Parametr `_ID` obsahuje na výstupu OID nalezeného záznamu.

Pokud číselník neobsahuje žádný záznam, nebo jich obsahuje více než jeden, vrací se prázdný řetězec.

Dalším vstupním parametrem `APrefilling` lze ovlivnit, zda se má provést vyhledání nebo ne.

Pokud číselník podporuje skrývání, vyhledává se pouze ve viditelných záznamech.

Na službu mohou být aplikované i předané generické filtry. Dokumentace ke generickým filtrům je v helpu, kap. Administrace/Technická dokumentace/AbraOLE/Popis tříd a typů/Filtrování číselníků.

Položky `AParms` slouží k předání vstupních parametrů a k vrácení výstupních hodnot:

vstup

APrefilling

- pro hodnotu `pmNone` se bez hledání vrátí prázdný řetězec, pro `pmOnlyIfOne` se provede vyhledání

AParms

- `_ID` (může chybět): slouží pro vrácení OID nalezeného záznamu
- `_Text` (může chybět): hledaná hodnota (nebo hodnoty pro MultiChoice)
- `_TextField` (může chybět): field, jehož hodnotu chceme vrátit
- `_ForcedField` (default = False):
- je-li = False & není zadán vstupní ID => nehledá se, vrátí se True a prázdná hodnota
- je-li = False & je zadán vstupní ID => provede se hledání
- je-li = True => provede se hledání (v tomto případě se při nezadaném ID vrátí False)
- `_MultiChoice` (default = False): parametr se uplatní pouze v případě, že není žádná hodnota pro předvyplnění vrácena (je-li True, pak se při nenalezení vrátí NullOID, jinak prázdný řetězec), pro vlastní vyhledání se parametr nevyužívá a zřejmě není pro Prefill nikde použitý
- `|<field_name>`: parametry s předznamenáním | (svislá čára) definují požadavek na vrácení hodnoty fieldu `<field_name>`

Pokud je u parametru uvedeno „(může chybět)“, pak není definovaná jeho defaultní hodnota a služba neprovede akci s parametrem spojenou, např. pokud chybí `_TextField`, nevrací se hodnota žádného fieldu, i když byl nějaký záznam nalezen. Pokud chybí `_ID`, služba nic nehledá a vrátí rovnou True (pozor, to je rozdíl oproti jiným službám – tato vrací vždy true).

Pokud je u parametru uvedeno „(default = xy)“ a parametr na vstupu chybí, pak je službou použita tato hodnota.

výstup

Result

- jako výsledek funkce vždy vrátí True

AParms

- `_ID`: obsahuje OID nalezeného záznamu
- `_Text`: obsahuje návratovou hodnotu definovaného fieldu
- `<field_name>` : obsahuje návratovou hodnotu fieldu zadaného na vstupu s předznamenáním | (svislá čára), hodnota každého takto zadaného fieldu se vrací ve zvláštním parametru

[AParms mohou obsahovat další parametry, samotný Prefill je ale nevyužívá.]

3.6 Služba GetIDs

Slouží ke získání interních identifikátorů (OID) všech záznamů číselníku, které vyhovují filtračním parametrům (generickým filtrům) předaným v parametru `AFilterParams`.

Parametr `AApostrophes` určuje, zda budou výstupní hodnoty uzavřeny do apostrofů.

Dokumentace ke generickým filtrům je v helpu, kap. Administrace/Technická dokumentace/AbraOLE/Popis tříd a typů/Filtrování číselníků.

Položky `AFilterParams` slouží k předání vstupních parametrů a k vrácení výstupních hodnot:

vstup

AFilterParams

- mohou obsahovat generické filtry

AApostrophes

- příznak určující, zda se mají výstupní hodnoty uzavřít do apostrofů

výstup

- varianta procedure `GetIDs`: OID se vrací v parametru `AList`: `TStrings`
- varianta function `GetIDs`: OID se vrací jako `Result` typu `INxList`