

Integrace systému správy dokumentace AXALANT a IS IFS

Radim Stuchlý



Univerzita Tomáše Bati ve Zlíně
Fakulta technologická
Institut řízení procesů a aplikované informatiky
akademický rok: 2004/2005

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Radim STUHLÝ**

Studijní program: **B3902 Inženýrská informatika**

Studijní obor: **Informační technologie**

Téma práce: **Integrace systému správy dokumentace AXALANT
a IS IFS.**

Zásady pro vypracování:

1. Analýza současného stavu.
2. Výběr způsobu výměny dat.
3. Návrh projektu pro automatickou výměnu dat.
4. Realizace projektu.
5. Implementace projektu.

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Ľudoslav Lacko: Oracle – Správa, programování a použití databázového systému,
Computer Press, Brno, 2002, ISBN: 80-7226-699-3.

Kolektiv autorů: Příručka systému správy dokumentace AXALANT.

Kolektiv autorů: Příručka vstupně výstupního modulu IS IFS.

Vedoucí bakalářské práce:

Ing. Zdenka Prokopová, CSc.

Institut řízení procesů a aplikované informatiky

Datum zadání bakalářské práce:

21. února 2005

Termín odevzdání bakalářské práce:

10. června 2005

Ve Zlíně dne 18. února 2005

prof. Ing. Josef Šimoník, CSc.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Souhlasím s tím, že s výsledky mé práce může být naloženo podle uvážení vedoucího bakalářské práce a ředitele ústavu. V případě publikace budu uveden jako spoluautor.

Prohlašuji, že jsem na celé bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.

Ve Zlíně, 09. 06. 2005

.....

podpis

RESUMÉ

Hlavním cílem této bakalářské práce bylo navrhnout a realizovat integraci systému správy technické dokumentace Axalant a informačního systému IFS ve firmě SECA, s.r.o. Úvodní část se zabývá analýzou problému, na základě které je vybrán způsob komunikace mezi systémy. V dalších částech je zdokumentována fáze realizace a implementace projektu z pohledu systému Axalant až po spuštění ostrého provozu.

The purpose of the thesis is to design and realize an integration of the technical documentation management system Axalant and the information system IFS in company SECA, ltd. The introductory part analyses the problem, on the basis of which the way of communication between the systems is chosen. The following parts of the thesis document the phase of realization and implementation of the project from the view of the system Axalant up to the initiation of the real operation.

OBSAH

ÚVOD.....	5
1 ANALÝZA SOUČASNÉHO STAVU	7
1.1 Současný stav systému pro správu dokumentace	7
1.2 Současný stav informačního systému.....	9
1.3 Způsob komunikace mezi systémy Axalant a Dialog.....	9
1.3.1 Přenos dat směrem Axalant - Dialog	9
1.3.2 Přenos dat směrem Dialog – Axalant	10
2 VÝBĚR ZPŮSOBU VÝMĚNY DAT	11
2.1 Analýza použití stávajícího řešení (textové soubory)	11
2.2 Analýza použití integrace využívající databázi ORACLE.....	12
2.3 Výběr způsobu integrace.....	13
3 NÁVRH PROJEKTU PRO AUTOMATICKOU VÝMĚNU DAT	14
3.1 Úprava datového modelu systému Axalant.....	14
3.2 Návrh exportu dat z hlediska systému Axalant	14
3.3 Návrh importu dat do systému Axalant	15
3.4 Popis modulu IESYST aplikace IFS	15
3.5 Struktura předávaných dat	17
3.5.1 Tabulka IeExecuteBatch.....	18
3.5.2 Tabulka IeDataLine	19
3.6 Struktura konverzních schémat	21
3.7 Přiřazení konverzních schémat objektům v systému Axalant.....	22
3.7.1 Uvolnění součásti, kusovníku a technologického postupu	23
3.7.2 Nová verze součásti a kusovníku.....	23
3.7.3 Oprava součásti a kusovníku	24
3.7.4 Oprava technologického postupu	24
4 REALIZACE PROJEKTU	25
4.1 Vytvoření duplicitní aplikace Axalantu.....	25
4.2 Úprava datového modelu systému Axalant.....	25
4.2.1 Změny v tabulce T_MASTER_DAT	25
4.2.2 Změny v tabulce T_POSTUPY_DAT	27
4.2.3 Změny v tabulce T_OPERACE_DAT	28
4.3 Úprava formuláře součástí v systému Axalant	28
4.4 Vytvoření přenosových tabulek.....	29
4.5 Programování přenosových funkcí v systému Axalant	32
4.5.1 Pravidla pro programování exportních přenosových funkcí v Axalantu.....	32
4.5.2 Pravidla pro programování importních přenosových funkcí v Axalantu	32
5 IMPLEMENTACE PROJEKTU	34
5.1 Zaškolení uživatelů	34
5.1.1 Zaškolení konstruktérů	34
5.1.2 Zaškolení technologů.....	34
5.1.3 Zaškolení správce systému	35
5.2 Testovací provoz	35
5.3 Odstavení původní aplikace.....	35
5.4 Spuštění nové aplikace do ostrého provozu.....	35
5.4.1 Převod dat DIALOG - IFS.....	35
5.4.2 Převod dat z původní aplikace Axalantu do nové aplikace	36
ZÁVĚR.....	37
SEZNAM POUŽITÉ LITERATURY	38

SEZNAM OBRÁZKŮ.....	39
SEZNAM TABULEK	40
SEZNAM PŘÍLOH.....	41

ÚVOD

Firma TECHNODAT – Informační systémy, s.r.o. (implementátor) začala s firmou SECA, s.r.o. (zákazník – předmětem činnosti je projekce a výroba pásových pil) jednat o zavedení systému správy technické dokumentace Axalant 2000 v době, kdy se zde používal systém pro řízení výroby DIALOG. Úkolem systému Axalant bylo usnadnit a zefektivnit správu technické dokumentace jak výkresové, tak i 3D modelů. Do této doby byla data z 3D CAD systémů ukládána pouze na společném diskovém prostoru na serveru. S narůstajícím množstvím dat docházelo k jejich obtížnější správě. Situace došla tak daleko, že konstruktéři konstruovali něco, co už bylo dávno uděláno a tím se začala snižovat produktivita práce.

Systém Axalant, který tato data spravuje a umožňuje jejich snadnou a rychlou dostupnost, nastolil určitý systém. Díky tomu je možno na základě struktury 3D modelů automaticky generovat kusovníky, na základě kterých (pořád ještě v systému Axalant), technologové vytvoří technologické postupy.

Tato data (kusovníky a postupy) bylo nutno nějakým způsobem předávat do systému pro řízení výroby. To se vždy provádělo ručně – zodpovědná osoba dostala v papírové podobě výstupy ze systému Axalant a přenesla je do systému DIALOG. Tento způsob byl časově velmi náročný a docházelo k poměrně častým chybám při „ručním přenosu dat“. Proto se začalo hledat řešení, které by tuto činnost zautomatizovalo.

Nabízelo se řešení propojení těchto systémů na úrovni databází. Jelikož systém DIALOG byl již však celkem zastaralý a neběžel na žádné „klasické“ databázi, toto řešení nebylo možné. Proto bylo zvoleno řešení přenosu dat pomocí textových souborů.

V systému Axalant byla po uvolnění kusovníku do výroby vytvořena dávka (textový soubor obsahující potřebné údaje), která pak byla naimportována do systému DIALOG. Systém DIALOG naopak předával informace o nakupovaných součástech (stav skladu, pojistná zásoba, skladová cena...) do systému Axalant. Na základě těchto informací konstruktéři okamžitě věděli, jaké součásti mohou použít.

Toto řešení posunulo komunikaci mezi systémy na vyšší kvalitativní úroveň – nebylo potřeba pracovní síly, která by toto předávání dat zajišťovala. Navíc se minimalizoval počet chyb při přenášení dat.

Postupem času začaly vznikat větší nároky na kvalitu řízení výroby. Systém DIALOG však již nebyl schopen tyto požadavky uspokojit, proto se vedení společnosti rozhodlo nahradit jej moderním systémem pro řízení výroby – IFS.

S příchodem nového systému se opět začala řešit otázka přenosu dat mezi konstrukcí a výrobou. Jelikož IFS (stejně jako AXALANT) běží na databázi ORACLE, bylo rozhodnuto, že při hledání možností integrace se tato společná vlastnost zohlední a vypracuje se systém, který umožní spolehlivě přenášet data mezi systémy Axalant a IFS.

Právě návrh a realizace integrace těchto systémů je předmětem této práce. Jelikož systém IFS je dodáván a implementován jinou firmou, bude pohlíženo na integraci mezi oběma systémy z pohledu systému pro správu dokumentace Axalant, jehož programové úpravy a změny datového modelu jsme schopni jako dodavatelé realizovat.

1 ANALÝZA SOUČASNÉHO STAVU

Analýzu současného stavu ve firmě SECA, s.r.o., je možno rozdělit do tří hlavních oblastí:

- Analýzu současného stavu systému pro správu dokumentace.
- Analýzu současného stavu informačního systému.
- Analýzu komunikačního rozhraní mezi těmito systémy.

Jednotlivé oblasti budou rozebrány v samostatných kapitolách.

1.1 Současný stav systému pro správu dokumentace

V současné době je ve firmě SECA, s.r.o. zaveden systém pro správu dokumentace Axalant – produkt německé firmy Eigner+Partner. Tento systém kompletně řeší správu technické dokumentace, generování kusovníků a vytváření technologických postupů. Dokumenty (3D modely a výkresy) jsou vytvářeny programem ProEngineer a přes integraci ukládány přímo do systému Axalant. Zde je každému modelu (3D dokumentu) přiřazeno jednoznačné číslo, konstruktérem vyplněny atributy a na základě údajů z ProEngineeru vygenerována struktura (pokud se jedná o sestavu). Ke každému modelu je přiřazen výkres (2D dokument) jednak v nativním formátu (výkres ProEngineeru *.drw) a taky ve formě pdf souboru. Pdf soubor je zde kvůli tisku a prohlížení výkresu i na pracovištích, kde není dostupný program ProEngineer.

Data v systému Axalant procházejí změnovým řízením – každý stav je reprezentován číslem. Díky změnovému řízení je možno ovlivňovat přístup k datům. Podle toho, v jakém stavu se dokument nachází, jsou uživatelům přidělována práva pro manipulaci s daty (viz tabulka stavů).

Tabulka I. Tabulka stavů dokumentů systému Axalant

název stavu	číslo	popis	práva
v práci	110	vytvoření dokumentu, konstrukčního kusovníku	skupina Projekce
Ve schvalování	120	kontrola vedoucího projekce	skupina Vedoucí projekce
technologie	222	vytvoření montážního kusovníku, technologického postupu	skupina Technologie
uvolněno	230	uvolnění do výroby, přenos do IS	práva pouze pro čtení
oprava	235	oprava technologie	skupina Technologie

název stavu	číslo	popis	práva
ve změně	240	vytvoření nové verze	práva pouze pro čtení
nahrazeno	250	nahrazeno při uvolnění nové verze	práva pouze pro čtení
zrušeno	260	zrušeno	práva pouze pro čtení

Jakmile jsou data založena (vytvořena), nacházejí se ve stavu „v práci“ – číslo stavu 110.

Po ukončení práce na modelu je k němu vygenerována „součást“ (vyplněn formulář s atributy pro výrobu). Součást je z pohledu integrace mezi oběma systémy nejdůležitější entitou – ze součásti jsou načítány pole při sestavování dávky přenosu. K součásti je přiřazen kusovník (BOM – Bill of Material) jak konstrukční, tak i montážní. Dále pak technologický postup a 3D dokument s modelem součásti.

Obrázek 1. Formulář součásti systému Axalant

Z programu ProEngineer je přes integraci do systému Axalant automaticky uložen u sestav konstrukční kusovník (ten je složen opět z jednotlivých součástí) a tato součást je pak i s dokumentem předána ke kontrole vedoucímu konstrukce.

Jakmile jsou data vedoucím konstrukce překontrolována, jsou předána technologovi, který na základě konstrukčního kusovníku sestaví montážní kusovník a technologický postup, k němuž přiřadí operace a jejich texty. Při uvolnění do stavu 230 jsou data (jak součásti s kusovníky, tak technologické postupy) automaticky přenášena do informačního systému. Systém pro správu dokumentace Axalant nemá žádný standardní modul, který by bylo možno k výměně dat využít, proto bylo třeba jej naprogramovat.

1.2 Současný stav informačního systému

V současné době je ve firmě SECA, s.r.o. používán informační systém DIALOG. Tento systém nemá žádný využitelný modul pro výměnu dat se systémem Axalant, proto zde muselo být naprogramováno rozhraní pro komunikaci.

1.3 Způsob komunikace mezi systémy Axalant a Dialog

Jelikož žádný z těchto systémů nemá řešeno standardní rozhraní pro komunikaci, byl vzhledem k velkým odlišnostem obou systémů zvolen způsob předávání dat pomocí textových souborů. Tyto se ukládají na vyhrazené místo na disku. Soubory mají přesně definovaný formát s pevnou délkou věty. Pro kontrolu správného exportu je na konci každé věty vložen znak „X“. Pokud tento znak na konci věty chybí, import nebude zahájen, dávka bude označena za chybnou a zkopírována do chybového adresáře. Po převzetí dávky ji systém zazálohuje do zvláštního adresáře, kde je možno ji po vzniku chyb při importu zkontrolovat a opravit. Importovaný soubor je poté smazán.

1.3.1 Přenos dat směrem Axalant - Dialog

Z Axalantu do Dialogu jsou přenášena data, která se týkají vlastní výroby. Jsou to:

- záhlaví součástí
- kusovníky nově uvolněných součástí
- technologické postupy
- operace technologických postupů
- texty operací

Data, která bylo importována ze systému Axalant, není možno v Dialogu spravovat, ani měnit pozice součástí v kusovníku. Veškeré změny týkající se vyráběných položek,

technologických postupů a operací se provádějí v Axalantu, poté jsou exportovány pro Dialog. Existuje-li již v Dialogu kusovník, který je v importní dávce, nejdříve se smaže a poté je znovu vytvořen na základě dat v importní dávce.

1.3.2 Přenos dat směrem Dialog – Axalant

Z Dialogu do Axalantu jsou přenášena data týkající se nakupovaných součástí a některých číselníků, které jsou spravovány v Dialogu. Jsou to tyto věty:

- nakupované součásti
- standardní názvy součástí
- standardní názvy operací
- měrné jednotky

Dávky vytvořené Dialogem jsou ukládány do adresáře, kde čekají na zpracování. Samotnou akci importu do Axalantu zabezpečuje oprávněná skupina uživatelů (funkce v menu systému Axalant). Po úspěšném importu je dávka zazálohována a poté smazána.

Zakládání nakupovaných součástí pro použití v systému Axalant je provedeno automaticky bez ohledu na předepsané změnové řízení. Ihned po založení jsou součásti převedeny do stavu „uvolněno (230)“ a připraveny k okamžitému použití.

Správa dat, která byla importována z Dialogu, není v Axalantu možná. Pokud je potřeba provést změnu, děje se toto v Dialogu a provádí se export do Axalantu.

2 VÝBĚR ZPŮSOBU VÝMĚNY DAT

Jelikož ve firmě SECA, s.r.o. dochází v této době k výměně informačního systému (Dialog je nahrazován systémem IFS), nabízí se možnost zamyšlení nad propojením se systémem pro správu dokumentace. Bylo by samozřejmě možno použít systém stávající (přes textové dávky), systém IFS má však kromě jiných výhod oproti Dialogu pro nás i jednu podstatnou přednost – běží na databázi ORACLE, tedy na stejné databázi, jakou využívá Axalant. Zde se přímo nabízí možnost využití komunikace na databázové úrovni.

Další velkou předností systému IFS je skutečnost, že již základní konfigurace obsahuje modul IESYST, který je připraven pro komunikaci s jinými systémy prostřednictvím speciálních databázových tabulek. Tím by odpadlo složité programování na straně IFS a problém komunikace by se víceméně zúžil pouze na úpravy v systému Axalant. To by ovšem znamenalo naprogramovat kompletně nové řešení a upustit od řešení již zavedeného (textové soubory).

2.1 Analýza použití stávajícího řešení (textové soubory)

Použití stávajícího systému integrace má své výhody i nevýhody. Mezi výhody lze započítat:

- Řešení je odzkoušeno.
- V systému Axalant je již naprogramováno, připraveno k použití s minimálními úpravami.
- Obsluha je již na používání systému vyškolená a zapracována.

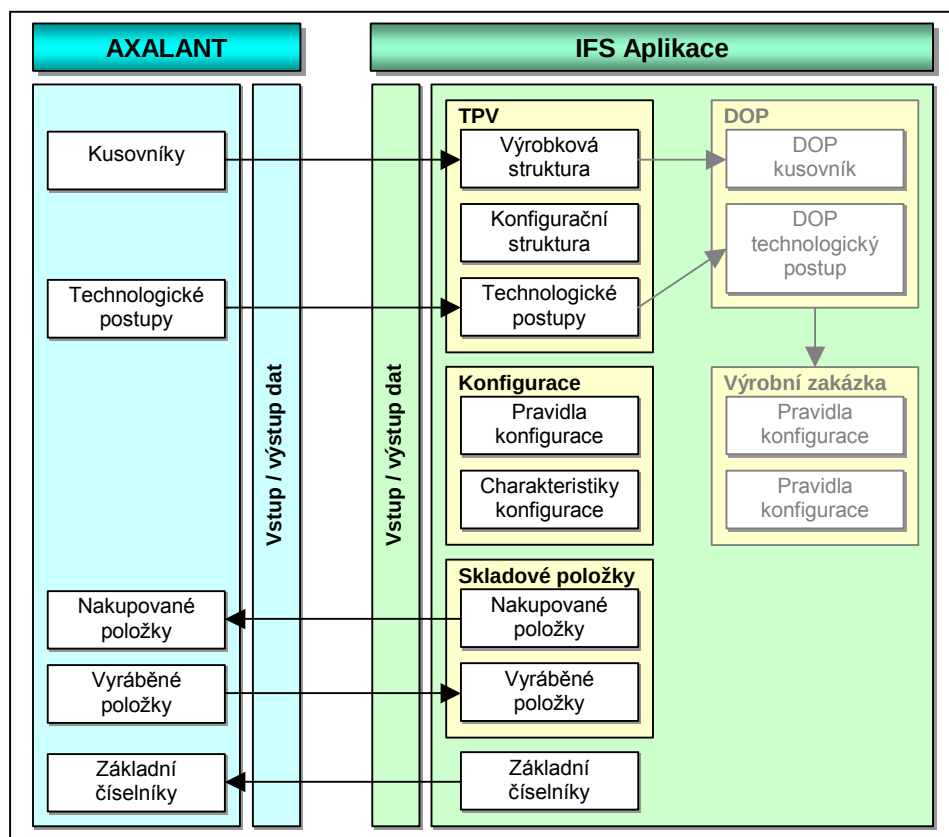
Nevýhody použití stávajícího systému:

- Nutnost kompletního naprogramování v systému IFS.
- Nižší spolehlivost – při výpadku sítě Ethernet dojde k chybě z důvodu umístění dávek na síťovém disku.
- Špatná operativnost při řešení chyb vzniklých při importu/exportu – je nutná práce s textovými soubory.
- Práce s textovými soubory je nepřehledná a pomalá.

2.2 Analýza použití integrace využívající databázi ORACLE

Použití integrace mezi oběma systémy, která využívá databázi ORACLE, je úzce svázáno se vstupně/výstupním modulem IESYST systému IFS. Zde je možno se opřít o osvědčené řešení tohoto systému. Rozhraní pro komunikaci je podrobně popsáno v příručce – systém IFS je připraven pro komunikaci s jiným systémem prakticky bez větších programových úprav.

K dalším přednostem tohoto řešení je možno zařadit skutečnost, že každý ze systémů může běžet samostatně – i při výpadku systému druhého. Databázové tabulky přenosu jsou totiž vytvořeny jak v systému IFS, tak v Axalantu. Na straně IFS je spuštěn serverový proces, který tyto tabulky synchronizuje. Při výpadku jednoho ze systémů k synchronizaci nedochází, avšak zápis do přenosových tabulek může pokračovat. Jakmile je síťové spojení opět navázáno, proces synchronizace je automaticky spuštěn a tabulky synchronizovány.



Obrázek 2. Datové schéma převodu Axalant - IFS

Z pohledu systému pro správu dokumentace je věc trochu složitější. Jelikož Axalant není pro tento způsob komunikace připraven, bylo by nutné rozhraní kompletně naprogramovat a vytvořit přenosové tabulky. Vzhledem k tomu, že Axalant disponuje vlastním pokročilým programovacím jazykem, nebyla by tvorba takového rozhraní příliš komplikovaná. Největší důraz by byl poté kladen na přesný popis struktury přenášených dat, popřípadě na úpravu datového modelu Axalantu – přidání nových polí do tabulky a formuláře součástí.

Výhody použití integrace na úrovni databáze ORACLE:

- Minimální úpravy v aplikaci IFS
- Možnost využití modulu IESYST aplikace IFS
- Možnost autonomního provozu systémů Axalant a IFS – po novém navázání spojení jsou přenosové tabulky synchronizovány
- Práce s přenosovými dávkami v databázi je přehlednější a je možno rychleji a lépe vyhledávat a opravovat chyby

Nevýhody:

- Nutnost naprogramování rozhraní v systému pro správu dokumentace Axalant
- Nutnost zaškolení obsluhy

2.3 Výběr způsobu integrace

Po zvážení všech výhod a nevýhod obou řešení bylo rozhodnuto, že komunikace mezi systémy Axalant a IFS bude realizována na úrovni databáze ORACLE. Hlavní roli v rozhodování sehrála skutečnost, že zvolený způsob je bezpečnější, méně náročný na správu a obsluhu, a zachovává možnost ukládání dávek i při selhání jednoho ze systémů.

3 NÁVRH PROJEKTU PRO AUTOMATICKOU VÝMĚNU DAT

V této kapitole bude navržen postup a hlavní principy pro systém výměny dat. Vycházet se bude z datového modelu systému IFS. Datový model Axalantu bude rozšířen tak, aby byla integrace co nejefektivnější. Je třeba najít takové společné prvky, aby byla odstraněna duplicita zadávání dat. Nemělo by se stát, že se stejná data budou zadávat do Axalantu i do IFS. Proto je potřeba důkladně analyzovat datový model IFS a pokud bude jednodušší zadávat některé hodnoty již při projektování nového výrobku, rozšířit datový model Axalantu a zadávat tyto hodnoty již zde. Tyto údaje by pak byly součástí komunikační dávky mezi oběma systémy.

3.1 Úprava datového modelu systému Axalant

Jak již bylo zmíněno, vzhledem k výměně informačního systému dojde i k úpravám datového modelu Axalantu tak, aby práce s oběma systémy byla díky integraci co nejefektivnější. Určitou úpravou projde i změnové řízení, bude nadefinován nový stav „235 – oprava“. V tomto stavu bude mít možnost oprávněný uživatel provést opravu dat, která již byla přes integraci zaslána do IFS. Ze stavu 235 bude možno přejít pouze zpět do stavu 230, čímž se opravená data automaticky znovu odešlou do IFS.

3.2 Návrh exportu dat z hlediska systému Axalant

Veškerá data, která budou exportována do informačního systému IFS, musí být schválena a uvolněna do výroby. Export je proto vhodné provést okamžitě při uvolnění dat (výrobku, kusovníku, technologického postupu, operací), to je při převodu do stavu 230 (viz tabulka 1 – změnové řízení). Akci provedení exportu je možno začlenit přímo do funkce změny stavu – po úspěšném uvolnění budou data okamžitě bez zásahu uživatele exportována.

Při sestavování přenosových dávek je nutno počítat s těmito třemi možnostmi:

- Exportovaná data ještě v IFS nejsou – na základě dávky přenosu jsou založena.
- Vytváří se nová verze – data se stejnou identifikací již v IFS jsou, provede se pouze změna.
- Oprava dat – data již v IFS jsou, v Axalantu však nedochází ke vzniku nové verze, ale k opravě údajů.

Pro každou z uvedených možností bude definována jiná struktura přenosové dávky.

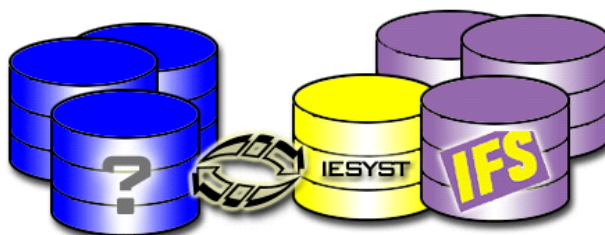
3.3 Návrh importu dat do systému Axalant

Dávky budou do systému Axalant importovány na vyžádání obsluhy. Po výběru funkce v menu Axalantu se spustí program, který bude jednotlivé dávky načítat z přenosových tabulek a aktualizovat či zakládat nová data. Je třeba zajistit, aby importovaná data nebylo možno v Axalantu upravovat – veškeré úpravy těchto dat se budou dít v IFS a poté budou přeneseny do Axalantu. Zamezení úprav se docílí tím, že ihned po importu budou data převedena do stavu 230 (uvolněno). V tomto stavu nejsou možné žádné úpravy. Po zpracování se importovaná dávka označí jako zpracovaná. Pokud dojde k chybě, dávka bude označena příznakem chyby, nebo příznakem nedokončeného importu.

3.4 Popis modulu IESYST aplikace IFS

Modul IESYST je plnohodnotným modulem systému IFS. Serverová část se instaluje společně s aplikací na server do databáze Oracle a klientská část má vytvořen vlastní spustitelný soubor.

Na následujícím obrázku je schématicky znázorněno, jakým způsobem probíhá předávání dat mezi systémem IFS (v pravé části obrázku) a libovolnou jinou aplikací (v levé části obrázku). Libovolnou aplikací se rozumí systém, který dokáže importovat/exportovat data v podobě zpracovatelné modulem IESYST.

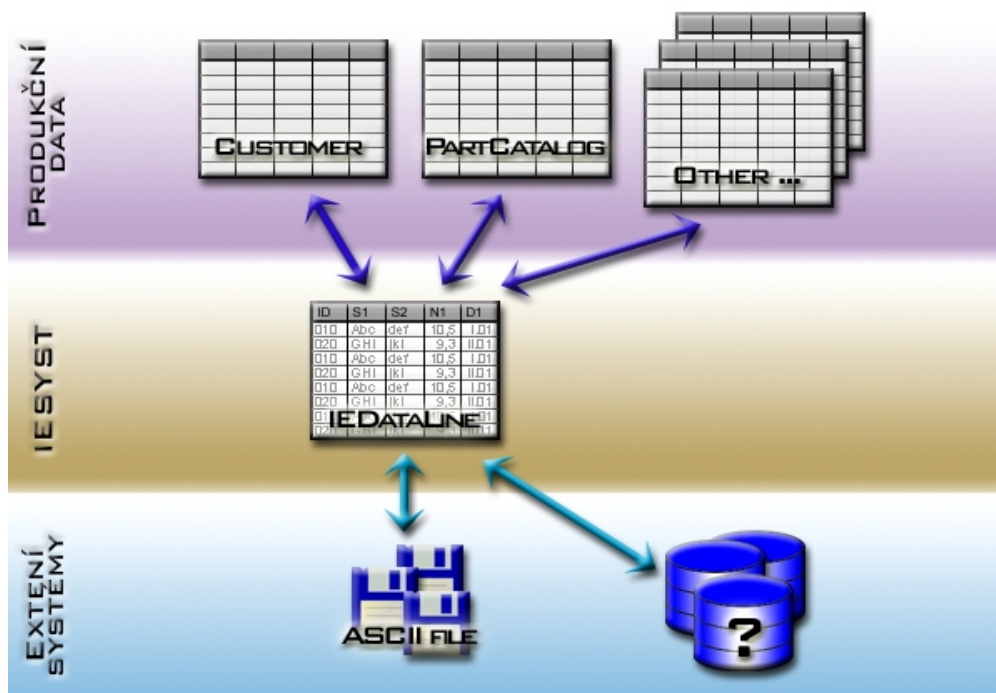


Obrázek 3. Vazba mezi IFS a ostatními systémy
(převzato z příručky modulu IESYST systému IFS)

Data ze systému IFS a do systému IFS jsou vždy předávána přes modul IESYST. Tím vzniká jakási vrstva, přístupná dalším aplikacím. Přístup do modulu IESYST z dalších aplikací je možno realizovat přes přesně definované rozhraní. Data do tohoto rozhraní mohou být nahrávána bez rizika, že se ovlivní produkční data systému IFS.

Data z jiných systémů se do modulu IESYST dostávají v podobě tzv. konverzní dávky. Data konverzních dávek mohou být v modulu IESYST před vlastním nahráním do produkčních dat systému IFS libovolně upravována. Konverzní dávku si lze představit jako sadu příkazů, které mají být v aplikaci vykonány. Každý jednotlivý řádek (příkaz) dávky obsahuje identifikaci konverzního schématu (instrukci), pomocí něhož má být zpracován a dále obsahuje vlastní data (parametry příkazu). Při spuštění dávky se postupně zpracovávají jednotlivé řádky dávky s tím, že konverzní schéma určí, jakým způsobem mají být data zpracována při přenosu do produkční části a datová část určí vlastní hodnotu jednotlivých dat.

Vnitřní logika modulu IESYST (znázorněná na následujícím obrázku) je postavena zejména na univerzální tabulce IE_DATA_LINE_TAB.



Obrázek 4. Vnitřní schéma modulu IESYST
(převzato z příručky modulu IESYST systému IFS)

Tabulka IE_DATA_LINE obsahuje (mimo jiné) 30 numerických polí označených n1..n30, 40 alfanumerických polí označených s1..s40 a 10 datumových polí označených d1..d10. Tato pole složí pro všechna konverzní schémata bez rozdílu a význam dat v nich vyplněných má význam právě jen ve spojení s konkrétním konverzním schématem. To znamená, že např. konverzní schéma pro nahrání odběratele může mít v poli s1 identifikaci

odběratele, ale konverzní schéma pro nahrání skladové položky může mít v poli s1 název položky. Význam jednotlivých polí pro konkrétní schéma je jednoznačně definováno textovým popisem konverzního schématu (viz. kapitola 3.6 Struktura konverzních schémat).

Tedy při importu dat do systému IFS se v univerzální tabulce IE_DATA_LINE vytvářejí jednotlivé záznamy obsahující identifikaci konverzního schématu a v příslušných datových polích obsahující patřičné údaje. V rámci zápisu dat do této univerzální tabulky nejsou prováděny žádné datové kontroly, s výjimkou kontroly datových typů (do datumových a numerických polí nelze vkládat text).

Data v univerzální tabulce jsou agregována číslem importní dávky. Po nahrání všech dat do dávky se tato dávka spustí pro zpracování do produkční části dat. Data do univerzální tabulky mohou být při dodržení zásad spojených s konkrétním konverzním schématem zadávána z různých míst v podstatě bez omezení a bez aplikačních kontrol. Při spuštění dávky se pak data zpracovávají a přenáší se na základě konverzního schématu do konkrétních logických jednotek.

Při tomto přenosu dat do produkční části jsou již prováděny všechny aplikační kontroly tak, jak by byly prováděny při ručním vstupu dat. Je tedy zaručeno, že data přenesená pomocí konverzního schématu prošla kontrolami systému a mají tak všechny náležitosti, které mají mít. Pokud při přenosu konkrétního řádku z univerzální tabulky do produkční části IFS dojde k nějaké chybě, je takovýto řádek označen za chybný a chyba se запиše do speciálního pole univerzální tabulky tak, aby si tuto chybu mohl uživatel prohlédnout a případně provést korekci dat spustit přenos znovu. Pokud je řádek zpracován úspěšně, změní se jeho stav tak, aby byl z dalších importů vyloučen.

3.5 Struktura předávaných dat

Data budou přenášena pomocí dvou tabulek: IeExecuteBatch a IeDataLine. Na straně IFS již tyto tabulky obsahuje modul IESYST, v Axalantu je bude nutno vytvořit.

3.5.1 Tabulka IeExecuteBatch

Tabulka IeExecuteBatch obsahuje informace o dávkách. Dávka sdružuje data, která se spustí pro import/export. Tabulka bude umístěna v databázi Oracle systému Axalant pro databázového uživatele SECA.

Následuje popis struktury tabulky s popisem významu jednotlivých polí.

Tabulka II. Popis tabulky IeExecuteBatch

Sloupec	Popis	Typ	Pov	Poznámka
<u>ExecuteBatchId</u>	Id dávky	num(22)	Ano	jednoznačné ID dávky; na straně IFS bráno ze sekvence; na straně Axalantu taktéž
IsImport	Import	vchr2(5)	Ano	příznak, zda se jedná o importní dávku (z pohledu IFS); TRUE-import / FALSE-export
CreatedDate	Datum vytvoření	date(7)		datum a čas vytvoření dávky
ExecutedDate	Datum spuštění	date(7)		datum a čas spuštění dávky v IFS (případně v Axalantu)
ClosedDate	Datum uzavření	date(7)		datum a čas ukončení spuštění dávky v IFS (případně v Axalantu)
Description	Popis	vchr2(2000)		textový popis dávky
State	Popis	vchr2(30)	Ano	stav dávky, jeden ze stavů: - <i>Ready</i> připraveno ke zpracování - <i>Open</i> právě se zpracovává - <i>Error</i> chyba při zpracování - <i>PartiallyImported</i> částečně importováno - <i>Imported</i> plně importováno - <i>Canceled</i> zrušeno
Result	Výsledek	vchr2(2000)		popis případné chyby
ErrorCount	Počet chyb	num(22)		počet chyb v dávce
ImportParams	Parametry importu	vchr2(2000)		parametry importu; závisí na typu předávaných dat
NextLineId	Id další řádky	num(22)	Ano	první volné číslo pro řádky dávky
ErrorFatal	Závažná chyba	vchr2(5)	Ano	příznak, zda dávka musí být zpracována v jediné transakci; TRUE-jakákoli chyba zabrání importu celé dávky / FALSE-částečný import povolen
ForceCommit	Okamžitý commit	vchr2(5)	Ano	příznak, zda se má pro každý importovaný záznam provádět commit; má význam pouze pokud <i>ErrorFatal=FALSE</i> ; slouží pro

Sloupec	Popis	Typ	Pov	Poznámka
				rychlejší zpracování dávek; TRUE-commit každého řádku / FALSE-commit celé dávky
GrantedUsers	Oprávnění uživatelé	vchr2(2000)		uživatelé oprávnění k ručnímu spuštění dávky
ExecutedBy	Provedl	vchr2(30)		jméno uživatele, který naposledy spustil dávku
SystemId	Id systému	vchr2(20)	Ano	Id systému z číselníku; konstanta AXALANT
ExtExecBatchId	Externí Id dávky	num(22)		Id dávky externího systému; slouží pro potřeby synchronizace dat (pouze pro potřeby IFS)
AutoExecuteTime	Čas automatického spuštění	date(7)		čas, kdy má být importní dávka automaticky spuštěna; <SYSDATE odpovídá „on-line“ spuštění (pouze pro potřeby IFS)

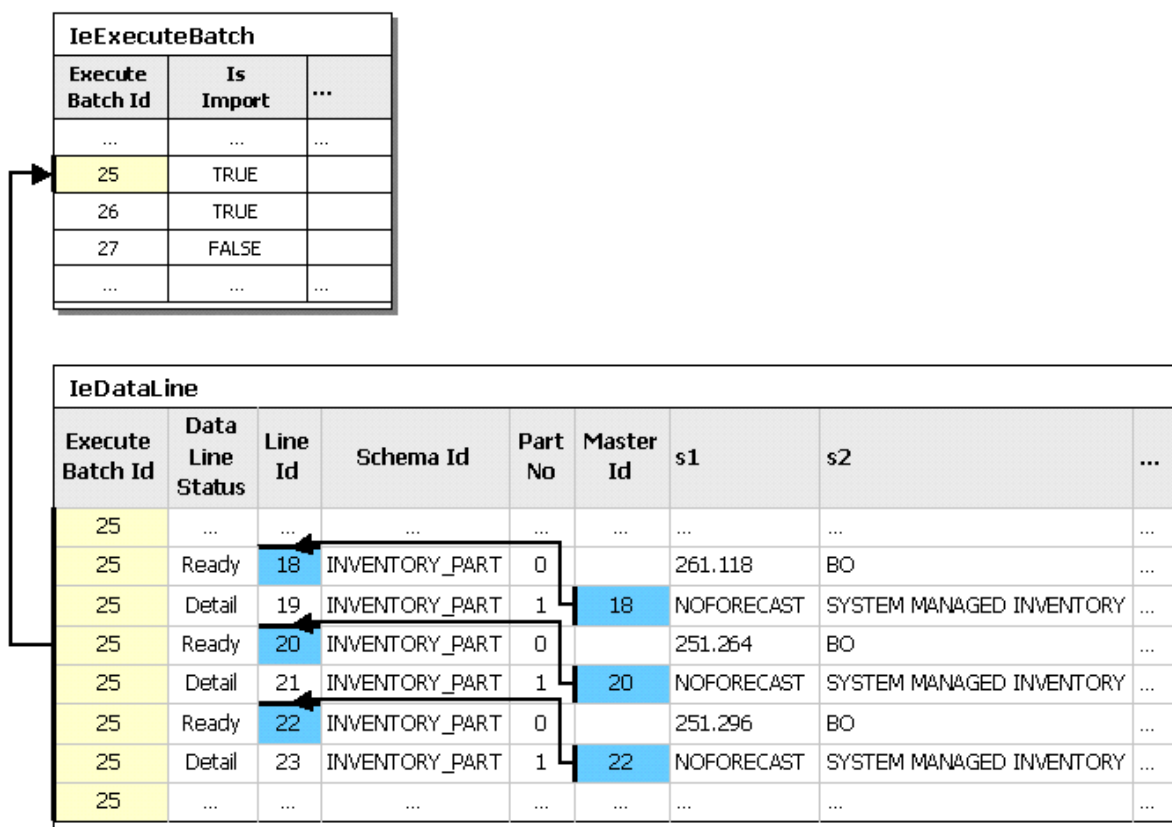
3.5.2 Tabulka IeDataLine

Tabulka IeDataLine obsahuje jednotlivé řádky importu/exportu. Pokud je struktura jednoho datového řádku delší než řádek této tabulky, nebo pokud je třeba realizovat vazbu master-detail, je možno rozložit tyto případy do více řádků tabulky IeDataLine. Nejprve je založen běžný řádek, který je označován „první datový řádek“. Ten má obvyklou syntaxi včetně platného stavu (viz. pole IeDataLineStatus). Pak je možno zakládat libovolné množství detailových řádků. Tyto řádky mají vyplněn atribut MasterId hodnotou LineId prvního datového řádku. Dále mají tyto řádky atribut IeDataLineStatus vyplněn konstantou DETAIL. Pro identifikaci jednotlivých typů detailových řádků se používá atribut PartNo. Za správnou interpretaci a následný import takto vzniklé struktury je odpovědná importní funkce. Tabulka je umístěna v databázi Oracle systému Axalant pro databázového uživatele SECA.

Následuje popis struktury tabulky s popisem významu jednotlivých polí.

Sloupec	Popis	Typ	Pov	Poznámka
<u>ExecuteBatchId</u>	Id dávky	num(22)	Ano	vazba do tabulky IeExecuteBatch
<u>LineId</u>	Číslo řádku	num(22)	Ano	jednoznačné Id řádku; zpracování záznamů probíhá vzestupně podle LineId
IeDataLineStatus	Stav řádku		Ano	jeden ze stavů: - Ready připraveno ke zpracování

Sloupec	Popis	Typ	Pov	Poznámka
				- <i>Error</i> chyba při zpracování - <i>Imported</i> importováno - <i>Exported</i> exportováno - <i>Cancelled</i> zrušeno - <i>Processing</i> právě zpracováváno - <i>Detail</i> speciální stav pro realizaci vazby master-detail
SchmeId	Id konverzního schématu	vchr2(80)	Ano	identifikace konverzního schématu (typu záznamu); dokumentace v příloze I dle této identifikace specifikuje pravidla pro vyplnění jednotlivých datových polí n1..n30, s1..s40, d1..d10
PartNo	Číslo části věty	num(22)		identifikace části záznamu; pokud je záznam delší než umožňují prvky n1..n30, s1..s40, d1..d10, je rozdělen na několik částí identifikovaných hodnotou <i>PartNo</i>
MasterId	Id master věty	num(22)		<i>LineId</i> nadřazeného záznamu; pokud je záznam delší než umožňují prvky n1..n30, s1..s40, d1..d10, nebo pokud je typu master-detail, je rozdělen na několik částí identifikovaných hodnotou <i>PartNo</i> ; jedna z těchto částí je hlavní a její <i>LineId</i> je pak vyplněno zde (viz. Obrázek 3 pod tabulkou)
Result n1..n30	Výsledek	vchr2(2000) num(22)		popis případné chyby obecné datové pole typu number; význam pole v konkrétním záznamu je určen hodnotou v poli <i>SchmeId</i> a dokumentací popisující dané konverzní schéma (viz. Příloha I)
s1..s40		vchr2(2000)		obecné datové pole typu varchar2; význam pole v konkrétním záznamu je určen hodnotou v poli <i>SchmeId</i> a dokumentací popisující dané konverzní schéma (viz. Příloha I)
d1..d10		date		obecné datové pole typu date; význam pole v konkrétním záznamu je určen hodnotou v poli <i>SchmeId</i> a dokumentací popisující dané konverzní schéma (viz. Příloha I)



Obrázek 5. Schéma struktury tabulek IeExecuteBatch a IeDataLine

(převzato z příručky modulu IESYST systému IFS)

3.6 Struktura konverzních schémat

V tabulce IeDataLine je pro nás z hlediska přenosů důležité pole SchmeId – zde je uložena informace o struktuře konverzního schématu. Jelikož celý systém přenosu dat vychází z řešení systému IFS, je třeba respektovat i definice konverzních schémat. Pro přenos budou využívána tato konverzní schémata:

a) pro přenos z IFS do Axalantu

- Inventory_part pro nakupované položky (součásti) – zakládány v IFS, konstrukce nesmí použít nic, co není založeno v IFS
- Work_center pro číselník pracovišť – zakládány v IFS, konstrukce nesmí použít nic, co není založeno v IFS

- Standard_names pro číselník standardních názvů součástí – zakládány v IFS, konstrukce nesmí použít nic, co není založeno v IFS
- Code_B pro číselník středisek – zakládány v IFS, konstrukce nesmí použít nic, co není založeno v IFS

b) pro přenos z Axalantu do IFS

- Part_catalog pro katalog položek
- Inventory_part pro vyráběné položky (součásti)
- Inv_part_discrete_char pro skladovou položku – diskrétní charakteristiku
- Inventory_part_planning pro skladovou položku – data plánování
- Part_revision pro skladovou položku – správu verzí
- Manuf_structure pro přenos kusovníků
- Routing_head pro přenos hlaviček technologických postupů
- Routing_operation pro přenos operací technologických postupů
- Routing_oper_work_guide pro přenos textů operací

Podrobný popis všech výše uvedených schémat lze nalézt v příloze I. Díky odlišnostem datových modelů obou systémů není možno z Axalantu plnit všechny hodnoty smysluplnými daty, proto jsou do polí, která jsou povinná, vkládány předem dohodnuté konstanty. Pokud je možno nepodstatná pole v přenosových schématech vynechat, pak se tato pole nepřenášejí.

3.7 Přiřazení konverzních schémat objektům v systému Axalant

Jelikož byla konverzní schémata modulu IESYST navržena tak, aby co nejvíce vyhovovala datovému modelu IFS, je třeba při exportu objektům Axalantu přiřadit správná konverzní schémata. Jak již bylo řečeno, k exportu dat z Axalantu dochází při schválení (uvolnění) položky do výroby přesunem do stavu 230. Podle toho, o jakou položku se jedná, je třeba

zvolit pro export správná konverzní schémata a to ve správném pořadí. Následuje popis jednotlivých objektů Axalantu a k nim přiřazených konverzních schémat:

3.7.1 Uvolnění součásti, kusovníku a technologického postupu

Uvolněním součásti je zároveň uvolnění její kusovník a technologický postup. Mechanismy uvnitř Axalantu zabrání automaticky uvolnění součásti, pokud technologický postup uvolněn není.

Tabulka III. Uvolnění součásti

Data Axalantu	Konverzní schémata
součást	Part_catalog Inventory_part Inv_part_discrete_char Inv_part_discrete_char Inventory_part_planning Part_revision
kusovník	Manuf_structure
technologický postup	Routing_head
operace postupu	Routing_operation
úkony operací	Routing_oper_work_guide

V tabulce je 2x uvedeno konverzní schéma Inv_part_discrete_char. Pomocí jednoho záznamu tohoto schématu se totiž přenáší jedna diskretní charakteristika, v našem případě je potřeba přenést charakteristiky dvě.

3.7.2 Nová verze součásti a kusovníku

Při uvolnění nové verze součásti v systému Axalant již vypadá přenosová dávka jinak. Jelikož součást již v IFS existuje, přenesou se pouze informace o nové verzi a nový kusovník. Technologické postupy se neverzují, proto k jejich přenosu dochází při opravě (převod do stavu 235) a následném uvolnění (převod do stavu 230) v Axalantu.

Tabulka IV. Uvolnění nové verze součásti a kusovníku

Data Axalantu	Konverzní schémata
součást	Part_revision
kusovník	Manuf_structure

3.7.3 Oprava součásti a kusovníku

Při opravě součásti a kusovníku (stav 230 → 235 → 230) dochází pouze k přenosu těch konverzních schémat, u nichž je možnost, že nastala nějaká změna. Kusovník se přenáší celý. Před importem do IFS se totiž v IFS smaže a pak znovu naimportuje.

Tabulka V. Oprava součásti kusovníku

Data Axalantu	Konverzní schémata
kusovník	Manuf_structure
součást	Part_catalog Inventory_part Inv_part_discrete_char Inv_part_discrete_char

3.7.4 Oprava technologického postupu

Při opravě technologického postupu (TP) se automaticky přenášejí pouze operace TP a jejich úkony. Exportují se všechna data, neboť před importem do IFS jsou data smazána a poté naimportována z přenosové dávky. Záznamy jsou exportovány při převodu technologického postupu ze stavu 235 (oprava) do stavu 230 (uvolněno).

Tabulka VI. Oprava technologického postupu

Data Axalantu	Konverzní schémata
operace postupu	Routing_operation
úkony operací	Routing_oper_work_guide

4 REALIZACE PROJEKTU

Při realizaci projektu je třeba dbát na to, aby přechod na nový systém ve firmě SECA, s.r.o. byl plynulý. Není možný výpadek systémů na dobu delší než 2 dny. Proto je nutné vytvořit duplicitní aplikaci Axalantu, která bude upravena pro komunikaci s IFS. Jakmile bude vše naprogramováno a otestováno ve zkušebním provozu, bude potřeba převést data z původní aplikace Axalantu do aplikace nové a zároveň bude do ostrého provozu spuštěn i systém IFS. Realizaci je možno rozdělit na jednotlivé fáze, které budou popsány v následujících kapitolách.

4.1 Vytvoření duplicitní aplikace Axalantu

Původní aplikace Axalantu se jmenuje SECA. Tato aplikace musí běžet do doby, než bude připravena aplikace nová – SECA2. Nová aplikace bude vytvořena kopií aplikace SECA, všechny úpravy od té chvíle budou probíhat v aplikaci SECA2. Pokud bude nutné provádět programové úpravy na aplikaci SECA, musí být provedeny i na aplikaci SECA2. Při přechodu ze staré aplikace na novou budou převedena pouze uživatelská data (3D modely, 2D_výkresy, součásti, technologické postupy, operace, úkony, historie...).

4.2 Úprava datového modelu systému Axalant

Úpravy datového modelu se týkají hlavně tabulek, které přímo souvisejí s daty, která jsou součástí přenosu Axalant – IFS. Pro zvýšení efektivity spojení mezi systémy bylo nutno přidat některá pole, která IFS vyžaduje a u kterých bude jednodušší, když budou vyplňována v Axalantu. Některá pole mají zase v IFS oproti DIALOGU jinou velikost, proto musí dojít k úpravě velikosti. Týká se to hlavně tabulek Axalantu:

- T_MASTER_DAT, která obsahuje pole týkající se popisu součástí
- T_POSTUPY_DAT, která popisuje záhlaví technologického postupu
- T_OPERACE_DAT, která popisuje operace technologického postupu

4.2.1 Změny v tabulce T_MASTER_DAT

V tabulce T_MASTER_DAT došlo k nejrozsáhlejší změnám oproti původnímu stavu. Změny jsou označeny tučným písmem. Pokud je tučně označena pouze hodnoty typu pole,

bude upravena pouze velikost pole. V případě, že nějaké pole nebude v nové aplikaci použito, zůstane v tabulce i s původními daty, neobjeví se však ve formuláři.

Tabulka VII. Změny v tabulce T_MASTER_DAT

Název pole	Typ	Unikátní	Popis
T_MASTER_DAT.TECHN_ID	S3	n	T.Id
T_MASTER_DAT.PART_ID	S32	y	ID
T_MASTER_DAT.PART_VERSION	S10	n	Ver.
T_MASTER_DAT.PT_ID	S15	n	TechnID+PartID
T_MASTER_DAT.PART_REVISION	S10	n	Rev.
T_MASTER_DAT.PART_NAME	S62	n	Název
T_MASTER_DAT.TEXT_ID	I5	n	Kód textu
T_MASTER_DAT.PART_NAME_NR	I4	n	Název - číslo
T_MASTER_DAT.ITEM_TYPE	S20	n	Typ součásti
T_MASTER_DAT.UNIT	S3	n	sMJ
T_MASTER_DAT.BOM_FLAG	I4	n	Pod
T_MASTER_DAT.AGG_FLAG	I4	n	Nad
T_MASTER_DAT.CHK_NAME	S20	n	Průběh
T_MASTER_DAT.LEV_IND	I10	n	Stav
T_MASTER_DAT.TYP	S20	n	Typ
T_MASTER_DAT.VAR_NUM	I5	n	V-číslo
T_MASTER_DAT.VAL_FROM	D	n	Platno od
T_MASTER_DAT.VAL_UNTIL	D	n	Platno do
T_MASTER_DAT.CUR_FLAG	L	n	A
T_MASTER_DAT.MATERIAL	S20	n	Materiál konečný
T_MASTER_DAT.DTS_REF	I6	n	RT-ID
T_MASTER_DAT.ZEICHNR	S20	n	Ref.výkr.
T_MASTER_DAT.VOLUMEN	S16	n	Objem
T_MASTER_DAT.FLAECHE	S16	n	Povrch
T_MASTER_DAT.GEWICHT	S16	n	Hmotnost
T_MASTER_DAT.RES_REF	I6	n	Rezerv.
T_MASTER_DAT.NTM_REF	I10	n	Poznámky
T_MASTER_DAT.STEP_SOURCE	S3	n	Zdroj
T_MASTER_DAT.STEP_DESCR	S240	n	Popis
T_MASTER_DAT.EXTERNAL_ITEM	L	n	Externí
T_MASTER_DAT.EDB_URL	S255	n	URL
T_MASTER_DAT.EDB_ICON	S40	n	Název ikony
T_MASTER_DAT.ART_BLOB	B	n	Náhled
T_MASTER_DAT.ART_BLOB_NAME	S255	n	Blob Name
T_MASTER_DAT.DRUHPOL	S1	y	Druh položky
T_MASTER_DAT.CENA	M11.2	n	Předp. cena
T_MASTER_DAT.PREDPOKLAD_C	M10.2	n	Předpokládané náklady (práce)
T_MASTER_DAT.PREDPOKLAD_CM	M10.2	n	Předp. materiál. náklady
T_MASTER_DAT.POPIS	S25	n	Popis
T_MASTER_DAT.C_HMOTNOST	F13.5	n	Čistá hmotnost
T_MASTER_DAT.H_HMOTNOST	F13.5	n	Hrubá hmotnost
T_MASTER_DAT.NAZEV_MAT	S40	n	Název a rozměr materiálu
T_MASTER_DAT.MAT_VYCHOZI	S30	n	Výchozí materiál
T_MASTER_DAT.MAT_KONECNY	S30	n	Konečný materiál
T_MASTER_DAT.SKLAD	S8	n	Sklad
T_MASTER_DAT.UCET	S7	n	Účet
T_MASTER_DAT.UMISTENI_SKLAD	S35	n	Umístění
T_MASTER_DAT.SKLAD_UM	S15	n	Sklad+umístění
T_MASTER_DAT.STAV_SKLADU	F11.3	n	Stav skladu
T_MASTER_DAT.POJISTNA_ZASOBA	I8	n	Pojistná zásoba

Název pole	Typ	Unikátní	Popis
T_MASTER_DAT.CAS_ZNOVUPOR	I3	n	CZP
T_MASTER_DAT.SKUPINA_VYROBKU	S20	n	Plánovač
T_MASTER_DAT.ROZMER	S40	n	Rozměr
T_MASTER_DAT.TJ	S3	n	TJ
T_MASTER_DAT.MJ_TJ	F11.5	n	sMJ->TJ
T_MASTER_DAT.NJ	S3	n	NJ
T_MASTER_DAT.MJ_NJ	F12.3	n	sMJ->NJ
T_MASTER_DAT.SCHVALIL	S40	n	Schválil
T_MASTER_DAT.MAT_DELKA	S40	n	Množství materiálu
T_MASTER_DAT.TVAR_C	S5	n	Tvar.č.
T_MASTER_DAT.MAT_POZNAMKA	S250	n	Mat.pozn.-Povrch.úpr.
T_MASTER_DAT.SPEC_KOD	S2	n	Speciální kód
T_MASTER_DAT.KOD_ROZPADU	I1	n	Kód rozpadu
T_MASTER_DAT.EDB_ID	I10	n	EDB_ID

Hodnoty ve sloupci Typ:

S – text, I – integer, F – float, D – datum. Číslo za znakem udává maximální velikost pole.

4.2.2 Změny v tabulce T_POSTUPY_DAT

Zde jsou popsány změny v tabulce T_POSTUPY_DAT. Změny jsou označeny tučným písmem. Pokud je tučně označena pouze hodnoty typu pole, bude upravena pouze velikost pole.

Tabulka VIII. Změny v tabulce T_POSTUPY_DAT

Název pole	Typ	Unikátní	Popis
T_POSTUPY_DAT.TD_ID	S40	y	Id. postupu
T_POSTUPY_DAT.TD_VERSION	S10	y	Verze
T_POSTUPY_DAT.TD_REVISION	S10	y	Revize
T_POSTUPY_DAT.TD_NAME	S80	n	Název
T_POSTUPY_DAT.TEXT_ID	I4	n	Číslo textu
T_POSTUPY_DAT.CHK_NAME	S20	n	Průběh
T_POSTUPY_DAT.LEV_IND	I10	n	Stav
T_POSTUPY_DAT.VAL_FROM	D	n	Platný od
T_POSTUPY_DAT.VAL_UNTIL	D	n	Platný do
T_POSTUPY_DAT.CUR_FLAG	L	n	A
T_POSTUPY_DAT.POZNAMKA	S240	n	Popis
T_POSTUPY_DAT.TEST	S20	n	Atest
T_POSTUPY_DAT.POC_OPER	I10	n	Počet operací
T_POSTUPY_DAT.TGDAV	I10	n	Opt.dávka
T_POSTUPY_DAT.KMENSTR	S7	n	Středisko
T_POSTUPY_DAT.INDEX_VYKR	S4	n	Index výkresu
T_POSTUPY_DAT.ART_CID	I10	n	C ID nadřazené součásti
T_POSTUPY_DAT.REZIJNI_NAKL	M10.2	n	Režijní náklady
T_POSTUPY_DAT.REZIJNI_MAT	M10.2	n	Režijní materiál (cena)
T_POSTUPY_DAT.PRACNOST	F10.3	n	Pracnost[Nh]

Hodnoty ve sloupci Typ:

S – text, I – integer, F – float, D – datum. Číslo za znakem udává maximální velikost pole.

4.2.3 Změny v tabulce T_OPERACE_DAT

Zde jsou popsány změny v tabulce T_OPERACE_DAT. Změny jsou označeny tučným písmem. Pokud je tučně označena pouze hodnota typu pole, bude upravena pouze velikost pole.

Tabulka IX. Změny v tabulce T_OPERACE_DAT

Název pole	Typ	Unikátní	Popis
T_OPERACE_DAT.TD_ID	S40	n	Operace
T_OPERACE_DAT.TD_VERSION	S10	n	Verze
T_OPERACE_DAT.TD_REVISION	S10	n	Rev.
T_OPERACE_DAT.TD_NAME	S80	n	Název
T_OPERACE_DAT.TEXT_ID	I4	n	Číslo textu
T_OPERACE_DAT.CHK_NAME	S20	n	Průběh
T_OPERACE_DAT.LEV_IND	I10	n	Stav
T_OPERACE_DAT.VAL_FROM	D	n	Platný od
T_OPERACE_DAT.VAL_UNTIL	D	n	Platný do
T_OPERACE_DAT.CUR_FLAG	L	n	A
T_OPERACE_DAT.TAR_TRIDA	S10	n	Tarifní třída
T_OPERACE_DAT.CAS_PRIP	F8.3	n	Přípravný čas
T_OPERACE_DAT.CAS_PRIP_MZ	F8.3	n	Přípravný čas mzdový
T_OPERACE_DAT.CAS_KUSOVY	F9.4	n	Kusový čas
T_OPERACE_DAT.CAS_KUS_MZ	F9.4	n	Čas kusový mzdový
T_OPERACE_DAT.NAZEV_PRAC	S40	n	Pracoviště
T_OPERACE_DAT.PRACOVISTE	S7	n	Pracoviště
T_OPERACE_DAT.POZNAMKA	S20	n	Poznámka
T_OPERACE_DAT.AKORD	S1	n	Akord
T_OPERACE_DAT.CAS_PRECHODU	I3	n	Čas přechodu
T_OPERACE_DAT.PS	S2	n	Splitting
T_OPERACE_DAT.POC_STROJU	F10.3	n	Poč.strojů
T_OPERACE_DAT.VICE_STROJU	I2	n	Strojů
T_OPERACE_DAT.DRUH_OPERACE	S1	n	D
T_OPERACE_DAT.KOOP_POL	S20	n	Kooperační položka
T_OPERACE_DAT.TEXT	S240	n	Text
T_OPERACE_DAT.CENA_KOOPERACE	M10.2	n	Cena kooperace
T_OPERACE_DAT.JMENO_KOOP	S20	n	Jméno kooperanta
T_OPERACE_DAT.KOEF_PRIP_MZ	I3	n	%
T_OPERACE_DAT.KOEF_KUS_MZ	I3	n	%

Hodnoty ve sloupci Typ:

S – text, I – integer, F – float, D – datum. Číslo za znakem udává maximální velikost pole.

4.3 Úprava formuláře součástí v systému Axalant

Změnami samozřejmě musel projít i formulář součástí. Ten obsahuje pole tabulky T_MASTER_DAT, došlo i ke grafické úpravě a logickému seskupení jednotlivých polí. Pole tabulky T_MASTER_DAT, která nejsou využívána pro zadávání údajů, jsou skryta a buď jsou nevyužita, nebo se využívají pro systémovou správu.

Součást - Formulář / KT20-001-4 / 0

Prohlížeč Verze-vše Sklad Založit TP

T.Id: 30 ID: KT20-001-4 Ver.: 0 Pod: 1 Stav: 230 Uvolněno Průběh: STD-PRT

Název: OSA Nad: 6 Platno od: 21.05.2004 Platno: 01.01.3000 A

Rozměr: []

Obecná data | BOM | Montážní BOM | Postupy | V součástech | V souč.(montáž) | Dokumenty | Náklady na součást | His

Popis: [] Druh položky: V

sMJ	Předp. cena	TJ	sMJ->TJ	NJ	sMJ->NJ	CZP	Plánovač
ks	32,63	ks	1.00000	ks	1.000	20	VD12000

Sklad: M06 Účet: 122100 Umístění: []

Název a rozměr materiálu: TYC KRUHOVA TAZENA 20

Výchozí materiál: 40.140.01.11 Konečný materiál: 11373

Mat.pozn.-Povrch.upr.: [] Množství materiálu: 0.3000 m

Tvar.č.: K0-02

Předp. nákl. (práce): 42,00 Předp. mat. náklady: 15,86

Čistá hmotnost: 0.67300 Speciální kód: []

Hrubá hmotnost: 0.00000 Kód rozpadu: 1

Schválil: []

4 100

Obrázek 6: Formulář součásti po úpravách aplikace Axalantu

4.4 Vytvoření přenosových tabulek

Přenosové tabulky IE_EXECUTE_BATCH a IE_DATA_LINE musí být vytvořeny přesně podle vzoru tabulek modulu IESYST. Stejná tabulka je vytvořena v databázi systému IFS. Pole si musí odpovídat typem, názvy polí se mohou lišit (akceptují se syntaktická pravidla a zvyklosti jednotlivých systémů). Použitý skript pro vytvoření tabulky IE_DATA_LINE:

```
CREATE TABLE ie_data_line
(
  executebatchid NUMBER NOT NULL,
  lineid NUMBER NULL,
  result VARCHAR2(2000) NULL,
  partno NUMBER NULL,
  masterid NUMBER NULL,
  n1 NUMBER NULL,
  n2 NUMBER NULL,
  n3 NUMBER NULL,
  n4 NUMBER NULL,
  n5 NUMBER NULL,
  n6 NUMBER NULL,
  n7 NUMBER NULL,
  n8 NUMBER NULL,
  n9 NUMBER NULL,
  n10 NUMBER NULL,
  n11 NUMBER NULL,
)
```

n12	NUMBER	NULL,
n13	NUMBER	NULL,
n14	NUMBER	NULL,
n15	NUMBER	NULL,
n16	NUMBER	NULL,
n17	NUMBER	NULL,
n18	NUMBER	NULL,
n19	NUMBER	NULL,
n20	NUMBER	NULL,
n21	NUMBER	NULL,
n22	NUMBER	NULL,
n23	NUMBER	NULL,
n24	NUMBER	NULL,
n25	NUMBER	NULL,
n26	NUMBER	NULL,
n27	NUMBER	NULL,
n28	NUMBER	NULL,
n29	NUMBER	NULL,
n30	NUMBER	NULL,
s1	VARCHAR2(2000)	NULL,
s2	VARCHAR2(2000)	NULL,
s3	VARCHAR2(2000)	NULL,
s4	VARCHAR2(2000)	NULL,
s5	VARCHAR2(2000)	NULL,
s6	VARCHAR2(2000)	NULL,
s7	VARCHAR2(2000)	NULL,
s8	VARCHAR2(2000)	NULL,
s9	VARCHAR2(2000)	NULL,
s10	VARCHAR2(2000)	NULL,
s11	VARCHAR2(2000)	NULL,
s12	VARCHAR2(2000)	NULL,
s13	VARCHAR2(2000)	NULL,
s14	VARCHAR2(2000)	NULL,
s15	VARCHAR2(2000)	NULL,
s16	VARCHAR2(2000)	NULL,
s17	VARCHAR2(2000)	NULL,
s18	VARCHAR2(2000)	NULL,
s19	VARCHAR2(2000)	NULL,
s20	VARCHAR2(2000)	NULL,
s21	VARCHAR2(2000)	NULL,
s22	VARCHAR2(2000)	NULL,
s23	VARCHAR2(2000)	NULL,
s24	VARCHAR2(2000)	NULL,
s25	VARCHAR2(2000)	NULL,
s26	VARCHAR2(2000)	NULL,
s27	VARCHAR2(2000)	NULL,
s28	VARCHAR2(2000)	NULL,
s29	VARCHAR2(2000)	NULL,
s30	VARCHAR2(2000)	NULL,
s31	VARCHAR2(2000)	NULL,
s32	VARCHAR2(2000)	NULL,
s33	VARCHAR2(2000)	NULL,
s34	VARCHAR2(2000)	NULL,
s35	VARCHAR2(2000)	NULL,
s36	VARCHAR2(2000)	NULL,
s37	VARCHAR2(2000)	NULL,
s38	VARCHAR2(2000)	NULL,
s39	VARCHAR2(2000)	NULL,
s40	VARCHAR2(2000)	NULL,
d1	DATE	NULL,
d2	DATE	NULL,
d3	DATE	NULL,
d4	DATE	NULL,
d5	DATE	NULL,
d6	DATE	NULL,

```

        d7                DATE                NULL,
        d8                DATE                NULL,
        d9                DATE                NULL,
        d10               DATE                NULL,
        iedatalinestatus  VARCHAR2(20)       NOT NULL,
        schmeid           VARCHAR2(80)       NOT NULL)
TABLESPACE &edb
STORAGE (&normal)
/
ALTER TABLE ie_data_line
  ADD ( CONSTRAINT ie_data_line_pk PRIMARY KEY (executebatchid, lineid)
        USING INDEX
          TABLESPACE &edb_idx
          STORAGE (&normal))
/

```

Použitý skript pro vytvoření tabulky IE_EXEC_BATCH (odpovídá tabulce IE_EXECUTE_BATCH modulu IESYST):

```

CREATE TABLE ie_data_line
( executebatchid          NUMBER                NOT NULL,
  isimport                VARCHAR2(2000)       NOT NULL,
  createdate              DATE                  NULL,
  executedate             DATE                  NULL,
  closedate               DATE                  NULL,
  description              VARCHAR2(2000)       NULL,
  state                   VARCHAR2(2000)       NOT NULL,
  result                  VARCHAR2(2000)       NULL,
  errorcount              NUMBER                NULL,
  importparams            VARCHAR2(2000)       NULL,
  nextlineid              NUMBER                NOT NULL,
  errorfatal              VARCHAR2(2000)       NOT NULL,
  forcecommit             VARCHAR2(2000)       NOT NULL,
  grantedusers            VARCHAR2(2000)       NULL,
  executedby              VARCHAR2(2000)       NULL,
  systemid                VARCHAR2(2000)       NOT NULL,
  extexecbatchid          NUMBER                NULL,
  autoexecutetime         DATE                  NULL)
TABLESPACE &edb
STORAGE (&normal)
/
ALTER TABLE ie_exec_batch
  ADD ( CONSTRAINT ie_exec_batch_pk PRIMARY KEY (executebatchid)
        USING INDEX
          TABLESPACE &edb_idx
          STORAGE (&normal))
/

```

4.5 Programování přenosových funkcí v systému Axalant

Programování přenosových funkcí je realizováno v jazyce LogiView, který je součástí Axalantu. Tento jazyk umožňuje práci s databází a objekty Axalantu, jako jsou formuláře, seznamy, řízení průběhů a mnoho dalšího...

Vytváření procedur jednotlivých přenosových funkcí se bude řídit určitými pravidly, která budou vycházet z požadavků přenosového modulu IESYST. Tím by se mělo předejít nesprávnému vytváření přenosových dávek a špatné interpretaci ze strany IFS. Pravidla budou rozdělena na 2 části pro funkce exportu a pro funkce importu. Vlastní přenosové funkce jsou přiloženy v Příloze II.

4.5.1 Pravidla pro programování exportních přenosových funkcí v Axalantu

- 1) Funkce bude spuštěna pouze po úspěšném schválení položky (převod do stavu 230).
- 2) Pro číslování dávek bude použit generátor číselných řad Axalantu. Exportní dávky z pohledu Axalantu budou mít vždy sudé číslo.
- 3) Nejdříve bude do tabulky dávek (IE_EXEC_BATCH) zapsána dávka, jejíž stav (pole IE_EXEC_BATCH.STATE) bude nastaven na „Open“.
- 4) Poté následuje export řádků do tabulky IE_DATA_LINE. V poli IE_DATA_LINE.EXECUTE_BATCHID bude uvedeno číslo dávky. Pole IE_DATA_LINE.STATUS bude nastaveno na „READY“.
- 5) Po ukončení exportu do tabulky IE_DATA_LINE bude záznam v poli STATE tabulky IE_EXEC_BATCH opraven na „Ready“.

4.5.2 Pravidla pro programování importních přenosových funkcí v Axalantu

Procedura pro import bude prakticky jediná. Podle typu konverzního schématu bude volat podprocedury, které provedou správný import dat na základě přenosových vět.

- 1) Import bude provádět pouze oprávněný uživatel (nebo skupina uživatelů).
- 2) Postupně se budou zpracovávat dávky od nejnižšího lichého čísla, kde ve sloupci IE_EXEC_BATCH.STATE bude hodnota „Ready“.

- 3) Po zjištění čísla dávky se budou zpracovávat odpovídající záznamy tabulky IE_DATA_LINE, které budou mít v poli STATE hodnotu „READY“ a to v pořadí od nejmenšího k největšímu číslu v poli LINEID.
- 4) Právě zpracovávaný záznam tabulky IE_DATA_LINE se označí hodnotou „PROCESSING“ ve sloupci „STATE“.
- 5) Po úspěšném zpracování dávky se hodnota ve sloupci IE_DATA_LINE.STATE upraví na „IMPORTED“.
- 6) Po zpracování celé dávky se dávka označí hodnotou „Imported“ ve sloupci IE_EXEC_BATCH.STATE.

5 IMPLEMENTACE PROJEKTU

V této kapitole uvažujeme pouze tu část implementace, která se týká integrace a systému Axalant. Implementaci a zaškolení systému IFS provádí dodavatel systému IFS. Zkušební provoz IFS byl spuštěn již mnohem dříve, neboť aplikace je ve firmě SECA, s.r.o zcela nová a vyžaduje daleko více času na „customizaci“ a uvedení do provozu.

Poté, co byla fáze realizace projektu ze strany Axalantu ukončena, je nutno připravit podmínky pro nasazení do ostrého provozu. Tomu ještě ale musí předcházet vyškolení uživatelů a testovací provoz.

5.1 Zaškolení uživatelů

Zaškolování uživatelů bude prováděno po skupinách podle toho, jakým způsobem s Axalantem pracují. Je důležité, aby uživatelé absolvovali školení každý u jednoho počítače, neboť přechod na nový systém bude rychlý a vedoucí každé skupinky zodpovídá za to, že uživatelé budou schopni se systémem okamžitě efektivně pracovat. Proto je třeba připravit předem typové příklady a na konci školení nechat uživatele je řešit.

5.1.1 Zaškolení konstruktérů

Za schopnost konstruktérů pracovat s novým systémem je zodpovědný vedoucí konstrukce. Ten proto bude přítomen u všech školení konstruktérů a na konci školení může prověřit jejich znalosti.

Při školení konstruktérů bude kladen největší důraz na vytváření a vyplňování formulářů součástí, neboť zde došlo z jejich pohledu k největším změnám.

5.1.2 Zaškolení technologů

Při práci technologů se změny aplikace projeví nejvíce, proto bude školení rozděleno na 2 části: práci se součástmi a práci s technologickými postupy. Technologové mají také právo spouštět funkci pro import dat z informačního systému IFS.

5.1.3 Zaškolení správce systému

Školení správce systému Axalant se bude týkat hlavně práce s přenosovými soubory. Správce bude mít přístup k jednotlivým záznamům přenosových dávek, bude mít možnost dávky mazat a opravovat v případě výskytu chyby.

5.2 Testovací provoz

Testovací provoz je nejdůležitější součástí realizace projektu. Doba testovacího provozu byla stanovena na 14 dní, během kterých mají uživatelé poslední šanci aplikaci řádně otestovat a případně objevit případné chyby. Je důležité tuto fázi projektu provádět zodpovědně a s maximální pečlivostí. Případné neodhalené chyby by se jinak projevíly v ostrém provozu, kde již mohou mít citelný finanční dopad.

Do testovacího provozu budou zařazeni zástupci konstrukce a technologie. Testovací provoz bude probíhat na zkušebních aplikacích Axalantu i IFS, data tedy nebudou mít žádný dopad na chod firmy, ani nebudou při přechodu do ostrého provozu převáděna do ostrých aplikací.

5.3 Odstavení původní aplikace

Po ukončení testovacího provozu dojde k odstavení původních aplikací. To znamená, že v této chvíli nebude možná práce v systému Axalant a Dialog. Doba rozběhnutí nových aplikací je odhadnuta na 2 dny, proto bylo rozhodnuto, že akce proběhne o víkendu, aby nedošlo k výpadku při práci konstruktérů.

5.4 Spuštění nové aplikace do ostrého provozu

Před spuštěním nových aplikací budou smazána veškerá testovací data. Poté proběhne převod dat z původních aplikací do nových. Než bude tato akce provedena, nesmí být do Axalantu (ani do IFS) vkládána žádná data. Po převodu dat do nových aplikací proběhne kontrola počtu záznamů v nových a starých tabulkách.

5.4.1 Převod dat DIALOG - IFS

Převod dat z Dialogu do IFS řeší dodavatelská firma systému IFS, proto není technické řešení převodu předmětem této práce.

5.4.2 Převod dat z původní aplikace Axalantu do nové aplikace

Než bude spuštěna nová aplikace systému Axalant, musí být zkopírována data všechna uživatelská data ze staré aplikace. To bude provedeno ručně v databázi Oracle. Seznam tabulek, které se budou kopírovat:

- | | | |
|----------------|-----------------|----------------|
| • t_master_dat | • t_store_dat | • t_oper_ukon |
| • t_master_str | • t_fil_store | • t_ukon_his |
| • t_master_his | • t_master_doc | • t_user_dat |
| • t_doc_dat | • t_postupy_dat | • t_grp_dat |
| • t_doc_drw | • t_master_post | • t_pac_dat |
| • t_doc_3dm | • t_postupy_his | • t_pac_master |
| • t_doc_his | • t_operace_dat | • t_pac_doc |
| • t_doc_str | • t_post_oper | • t_pac_his |
| • t_file_dat | • t_operace_his | • t_pac_post |
| • t_doc_fil | • t_ukony_dat | |

Poslední akcí před spuštěním do ostrého provozu je import těchto kompletních číselníků z IFS:

- standardní texty součástí
- pracoviště
- střediska

Tyto číselníky jsou z IFS exportovány kompletní, aby bylo možno při spuštění ostrého provozu ihned používat data.

ZÁVĚR

Úkolem práce bylo nalézt nejvhodnější řešení integrace systému správy technické dokumentace Axalant a systému pro řízení výroby IFS ve firmě SECA, s.r.o. Nabízela se 2 principiální řešení: buď zůstat u původního modelu propojení se systémem DIALOG přes sdílené textové soubory, nebo využít společných vlastností obou systémů – především databáze ORACLE. Ukázalo se, že systém IFS je (na rozdíl od systému Axalant a DIALOG) rozhraním pro komunikaci s jinými systémy vybaven. Jakousi komunikační vrstvu tvoří jeho modul IESYST.

Po analýze obou řešení bylo rozhodnuto, že vhodnější bude použití integrace využívající databázi ORACLE a to přímým využitím přenosového modulu IESYST systému IFS. Hlavními argumenty pro výběr zvoleného řešení byly: spolehlivost, snadná administrace a skutečnost, že řešení umožňuje oběma systémům běžet autonomně. To je umožněno tím, že každý ze systémů má ve své databázi své přenosové tabulky. Synchronizace databázových přenosových tabulek je realizována serverovým procesem na straně IFS, po obnovení výpadku spojení tedy dojde k jejich synchronizaci.

Po uvedení do ostrého provozu se plně projeví výhody zvoleného řešení. Zadávání některých parametrů výrobků bylo přeneseno do systému Axalant, kde je to efektivnější. Tyto hodnoty jsou pak přenášeny do IFS pomocí integrace. Správa nakupovaných součástí zase mohla být plně přesunuta do systému IFS.

Uživatelé nejsou zatěžováni správou přenosových souborů, vytváření přenosových dávek probíhá na pozadí a případné chyby se velice jednoduše vyhledávají a opravují. Samotný proces importu a exportu je podstatně rychlejší než řešení postavené na přenosu textových souborů, tím pádem i přenášená data jsou aktuálnější. V případě potřeby rozšíření přenosových vět o další údaje je možno toto provést jednoduchými programovými úpravami v obou systémech.

Vytvoření integrace, které bylo předmětem této práce, posunulo na kvalitativně vyšší úroveň propojení systémů nezbytných pro efektivní chod podniku. Konstrukce a výroba byly propojeny a díky tomu jsou v těchto odděleních k dispozici aktuální údaje z obou systémů. Ty je možno velice rychle vyhledávat a spravovat.

SEZNAM POUŽITÉ LITERATURY

- [1] Ľudoslav Lacko: Oracle – Správa, programování a použití databázového systému, Computer Press, Brno, 2002, ISBN: 80-7226-699-3.
- [2] Kolektiv autorů: Příručka systému správy dokumentace AXALANT.
- [3] Kolektiv autorů: Příručka vstupně výstupního modulu IESYST informačního systému IFS.

SEZNAM OBRÁZKŮ

Obrázek 1. Formulář součásti systému Axalant	8
Obrázek 2. Datové schéma převodu Axalant - IFS	12
Obrázek 3. Vazba mezi IFS a ostatními systémy	15
Obrázek 4. Vnitřní schéma modulu IESYST.....	16
Obrázek 5. Schéma struktury tabulek IeExecuteBatch a IeDataLine.....	21
Obrázek 6: Formulář součásti po úpravách aplikace Axalantu	29

SEZNAM TABULEK

Tabulka I. Tabulka stavů dokumentů systému Axalant.....	7
Tabulka II. Popis tabulky IeExecuteBatch	18
Tabulka III. Uvolnění součásti	23
Tabulka IV. Uvolnění nové verze součásti a kusovníku	23
Tabulka V. Oprava součásti kusovníku	24
Tabulka VI. Oprava technologického postupu	24
Tabulka VII. Změny v tabulce T_MASTER_DAT	26
Tabulka VIII. Změny v tabulce T_POSTUPY_DAT	27
Tabulka IX. Změny v tabulce T_OPERACE_DAT	28

SEZNAM PŘÍLOH

Příloha 1: Popis konverzních schémat

Příloha 2: Procedury přenosových funkcí systému Axalant

Integrace systému správy dokumentace AXALANT a IS IFS

Příloha I.

Radim Stuchlý

Obsah

Nakupované položky (IFS→Axalant) INVENTORY_PART	3
Katalog položek – Nastavení (Axalant → IFS) PART_CATALOG	5
Skladová položka – Obecné (Axalant → IFS) INVENTORY_PART	6
Skladová položka – Obecné – změna stavu (Axalant → IFS) INVENTORY_PART	11
Skladová položka – Diskrétní charakteristika INV_PART_DISCRETE_CHAR	12
Skladová položka – Data plánování INVENTORY_PART_PLANNING	13
Skladová položka – Verze (Axalant → IFS) PART_REVISION	15
Pracoviště – Základní data (IFS→Axalant) WORK_CENTER	17
Struktury (výrobní, recepturová, plánovací, konfigurační) (Axalant→IFS) MANUF_STRUCTURE	18
Technologické postupy – Hlavička (Axalant → IFS) ROUTING_HEAD	21
Technologické postupy (Axalant→IFS) ROUTING_OPERATION	22
Pracovní směrnice technologického postupu (Axalant → IFS) ROUTING_OPER_WORK_GUIDE	24
Standardní názvy (IFS→Axalant) STANDARD_NAMES	25

Tato příloha popisuje význam jednotlivých polí jednotlivých konverzních schémat (typů předávaného záznamů). Popis vychází z datového modelu systému IFS.

Nakupované položky (IFS→Axalant)					INVENTORY_PART
LU		PurchasePart, InventoryPart, InventoryPartPlanning	IE LU		
Tabulka			Importní funkce		
Parametry			CAT		
Vstupní soubor			APY/API		
Poznámky					
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
s1 contract	Místo	VCHR2(5)	Ano		InventoryPart.PartNo
s2 part_no	Č. položky	VCHR2(25)	Ano		InventoryPart.PartNo
s3 description	Popis	VCHR2(35)	Ano		InventoryPart.Description
s4 unit_meas	MJ	VCHR2(10)	Ano	IsoUnit	InventoryPart.UnitCode i jako TJ
n1 std_name_id	Id stand. názvu	NUMBER(22)	Ano	StandardNames	InventoryPart.StdNameId
s5 planner_buyer	Plánovač	VCHR2(20)	Ano	InventoryPartPlanner	InventoryPart.PlannerBuyer; Skupina výrobků
s6 part_status	Stav polož.	VCHR2(1)	Ano	InventoryPartStatusPar	InventoryPart.PartStatus Axalant: speciální kód; IFS A-aktivní,Z-zrušeno,F-fiktivní
s7 type_designation	Typové označení	VCHR2(25)			InventoryPart.TypeDesignation Rozměr
n2 weight_net	Čistá hmotn.	NUMBER(22)	Ano		InventoryPart.WeightNet
n3 price	Odhad. materiál. náklady	NUMBER(22)	Ano		Inventory_Part_Config_API.Get_Inventory_Value_By_Method(contract, part_no, '*') IFS: Skladová cena InventoryPartDefLoc pro typ Odběr
s8 location_no	Umístění	VARCHAR2(35)			
s9 account	Účet	VARCHAR2(??)			zatím nebude z IFS plněno
n4 qty_onhand	Stav skladu	NUMBER	Ano		InventoryPartInStock.GetInventoryQtyOnhand
n5 safety_stock	Pojistná zásoba	NUMBER			InventoryPartPlanning.SafetyStockAuto
s10 char1	Charakteristika 1	VARCHAR2(60)			prozatím nespecifikováno
s11 char2	Charakteristika 2	VARCHAR2(60)			prozatím nespecifikováno

Nakupované položky (IFS→Axalant)					INVENTORY_PART
LU		PurchasePart, InventoryPart, InventoryPartPlanning	IE LU		
Tabulka Parametry Vstupní soubor Poznámky			Importní funkce CAT APY/API		
Jméno pole		Popis	Datový typ	Pov. Reference	Poznámka
s12 char3		Charakteristika 3	VARCHAR2(60)		prozatím nespecifikováno
s13 char4		Charakteristika 4	VARCHAR2(60)		prozatím nespecifikováno
s14 char5		Charakteristika 5	VARCHAR2(60)		prozatím nespecifikováno
n6 list_price		Nákupní cena	NUMBER		PurchasePartSupplier.ListPrice ... pro primárního dodavatele
s15 default_buy_unit_meas		Implicitní nákupní MJ	VCHR2(10)	Ano IsoUnit	PurchasePart ... z implicitního dodavatele
n7 conv_factor		Převodní koeficient	NUMBER	Ano	PurchasePart ... z implicitního dodavatele

Katalog položek – Nastavení (Axalant → IFS)						PART_CATALOG
LU		PART_CATALOG		IE LU		IePartCatalog
Tabulka		PART_CATALOG_TAB		Importní funkce		ImPartCatalog
Parametry		UPDATE~TRUE/FALSE		CAT		ie_partcatalog.cat
Vstupní soubor				APY/API		IeUtilPartCatalog.ap_
Poznámky						
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka	
s1 part_no	Č. položky	VCHR2(25)	Ano			
s2 description	Popis	VCHR2(35)	Ano		V Axalantu - Název	
s6 serial_tracking_code	Obecné Sledování série ve skladu	VCHR2(30)	Ano	PartSerialTracking	konstanta NOT SERIAL TRACKING jedna z ‚SERIAL TRACKING‘, ‚NOT SERIAL TRACKING‘, default ‚NOT SERIAL TRACKING‘	
s7 eng_serial_tracking_code	Sledování série po dodání	VCHR2(20)	Ano	PartSerialTracking	konstanta NOT SERIAL TRACKING jedna z ‚SERIAL TRACKING‘, ‚NOT SERIAL TRACKING‘, default ‚NOT SERIAL TRACKING‘	
s10 configurable	Lze konfigurovat	VCHR2(20)	Ano	PartConfiguration	dle konfigurovatelnosti položky (pokud obsahuje varianttni položky=lze konfigurovat) CONFIGURED, nebo NOT CONFIGURED default NOT CONFIGURED	
s8 serial_rule	Způsob vytváření série	VCHR2(20)	Ano	PartSerialRule	konstanta MANUAL jedna z ‚MANUAL‘, ‚AUTOMATIC‘, default MANUAL	
s9 lot_tracking_code	Sledování dávky	VCHR2(30)	Ano	PartLotTracking	konstanta NOT LOT TRACKING jedna z LOT TRACKING/NOT LOT TRACKING/ORDER BASED, default NOT LOT TRACKING	
n2 cust_warranty_id	Záruka zákazníka	NUMBER		CustWarranty	nepoužito	
n3 sup_warranty_id	Záruka dodavatele	NUMBER		SupWarranty	nepoužito	
s5 unit_code	Kód MJ	VCHR2(30)	Ano	IsoUnit		
n1 std_name_id	Id stand. názvu	NUMBER(22)	Ano	StandardNames		
s3 part_main_group	Hlavní skup. polož.	VCHR2(1)		EngPartMainGroup	nepoužito	
s4 info_text	Info text	VCHR2(2000)			nepoužito	

Skladová položka – Obecné (Axalant → IFS)						INVENTORY_PART
LU		INVENTORY_PART		IE LU	IeUtilInventoryPart	
Tabulka		INVENTORY_PART_TAB		Importní funkce	ImInventoryPart	
Parametry		UPDATE~TRUE/FALSE, MERGE~TRUE/FALSE		CAT	ie_inventorypart	
Vstupní soubor				APY/API	IeUtilInventoryPart.ap_	
Poznámky		Musí být zadáno: PART_CATALOG Uživatel IFSAPP musí mít povolena všechna umístění obsažená v importu Parametr MERGE má význam pouze pokud parametr UPDATE je TRUE. Význam: TRUE data původního záznamu jsou bezezbytku přepsána novými hodnotami (i kdyby měly být NULL) FALSE data původního záznamu jsou přepsána pouze pokud je nová hodnota různá od NULL				
Jméno pole		Popis	Datový typ	Pov.	Reference	Poznámka
Obecné						
s1	part_no	Sklad. položka	VCHR2(25)	Ano	PartCatalog	
s14	description	Popis položky	VCHR2(35)	Ano		
s2	contract	Místo	VCHR2(5)	Ano	Site	konstanta BO
s25	type_code	Typ položky	VCHR2(2)	Ano	InventoryPartType	konstanta 1 jedna z 1–,Manufactured'/2–,Manufactured Recipe'/ 3–,Purchased (raw)'/4–,Purchased'/6–,Expense`, default ,4`
s8	planner_buyer	Plánovač	VCHR2(20)	Ano	InventoryPartPlanner	Axalant: Skupina výrobků
s13	unit_meas	MJ	VCHR2(10)	Ano	IsoUnit	Axalant: skladová MJ
s9	prime_commodity	Skup. komodity 1	VCHR2(5)		CommodityGroup	nepoužito
s12	second_commodity	Skup. komodity 2	VCHR2(5)		CommodityGroup	nepoužito
s3	asset_class	Majet. třída	VCHR2(2)	Ano	AssetClass	konstanta R
s7	part_status	Stav polož.	VCHR2(1)	Ano	InventoryPartStatusPar	Axalant: speciální kód; IFS A-aktivní,Z-zrušeno,F-fiktivní
s15	abc_class	Kód ABC	VCHR2(1)	Ano	AbcClass	konstanta A
s6	hazard_code	Bezpečn. kód	VCHR2(6)		SafetyInstruction	nepoužito
s4	accounting_group	Účetní skupina	VCHR2(5)		AccountingGroup	nepoužito
s10	part_product_code	Kód výrobku	VCHR2(5)		InventoryProductCode	nepoužito
s11	part_product_family	Skup. příb. výrob.	VCHR2(5)		InventoryProductFamily	nepoužito
s26	type_designation	Typové označení	VCHR2(25)			Axalant: Rozměr
s17	dim_quality	Dimenze/Jakost	VCHR2(25)			nepoužito
n1-1	weight_net	Čistá hmotn.	NUMBER(22)			
	create_date	Vytvořeno	DATE(7)	Ano		-
	last_activity_date	Poslední změna	DATE(7)	Ano		-
	note_id	Pozn. (není na obrazovce)	NUMBER(22)	Ano		plněno automaticky

Skladová položka – Obecné (Axalant → IFS)					INVENTORY_PART
LU		INVENTORY_PART	IE LU		IeUtilInventoryPart
Tabulka		INVENTORY_PART_TAB	Importní funkce		ImInventoryPart
Parametry		UPDATE~TRUE/FALSE, MERGE~TRUE/FALSE	CAT		ie_inventorypart
Vstupní soubor			APY/API		IeUtilInventoryPart.ap_
Poznámky		Musí být zadáno: PART_CATALOG Uživatel IFSAPP musí mít povolena všechna umístění obsažená v importu Parametr MERGE má význam pouze pokud parametr UPDATE je TRUE. Význam: TRUE data původního záznamu jsou beze zbytku přepsána novými hodnotami (i kdyby měly být NULL) FALSE data původního záznamu jsou přepsána pouze pokud je nová hodnota různá od NULL			
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
s22 note_text	Text dokumentu (není na obrazovce)	VCHR2(2000)			nepoužito
avail_activity_status	Platný stav činnosti (není na obrazovce)	VCHR2(9)		InventoryPartAvailStat	plněno automaticky, default ,CHANGED'
	Pořízení				
lead_time_code	Kód průb. doby	VCHR2(1)	Ano	InvPartLeadTimeCode	plněno automaticky, jedna z M – ,Manufactured' / P – ,Purchased' , default ,P'
n6 purch_leadtime	Průb. doba nákupu	NUMBER(22)	Ano		konstanta 0
n5 manuf_leadtime	Průběžná doba výroby	NUMBER(22)	Ano		konstanta 0
n4 expected_leadtime	Oček. průb. doba	NUMBER(22)	Ano		konstanta 20
n3 durability_day	Trvanlivost-dny	NUMBER(22)			nepoužito
s24 supersedes	Nahrazuje položku	VCHR2(25)			nepoužito
s23 superseded_by	Nahrazeno položkou	VCHR2(25)			nepoužito
n1-3 std_name_id	Standard. název	NUMBER(22)		StandardNames	!!! srovnat číslování IFS - Dialog
s1-7 ean_no	Č. EAN	VCHR2(35)			nepoužito
s28 supply_code	Implicitní pořízení MP	VCHR2(3)	Ano	MaterialRequisSupply	konstanta IO
s1-5 dop_connection	Připojení DOP	VCHR2(200)	Ano	DopConnection	jedna z IO ´-,Inventory Order'/PO-,Purchase Order', default IO konstanta MAN
s1-6 manufacturer_id	Id výrobce	VCHR2(20)		ManufacturerInfo	jedna z AUT-,Automatic DOP'/MAN-,Manual DOP', default ,MAN'
s1-8 manufacturer_part_id	Id polož. od výrobce	VCHR2(25)			nepoužito
s1-9 nurm_code	Kód NURM	VCHR2(5)			nepoužito
s5 country_of_origin	Země původu	VCHR2(3)		ApplicationCountry	nepoužito
s18 customs_stat_no	Celní stat. číslo	VCHR2(10)		CustomsStatisticsNumber	nepoužito

Skladová položka – Obecné (Axalant → IFS)					INVENTORY_PART
LU		INVENTORY_PART	IE LU		IeUtilInventoryPart
Tabulka		INVENTORY_PART_TAB	Importní funkce		ImInventoryPart
Parametry		UPDATE~TRUE/FALSE, MERGE~TRUE/FALSE	CAT		ie_inventorypart
Vstupní soubor			APY/API		IeUtilInventoryPart.ap_
Poznámky		Musí být zadáno: PART_CATALOG Uživatel IFSAPP musí mít povolena všechna umístění obsažená v importu Parametr MERGE má význam pouze pokud parametr UPDATE je TRUE. Význam: TRUE data původního záznamu jsou beze zbytku přepsána novými hodnotami (i kdyby měly být NULL) FALSE data původního záznamu jsou přepsána pouze pokud je nová hodnota různá od NULL			
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
n1-2 intrastat_conv_factor	Intrastat - převod. koef.	NUMBER(22)			nepoužito
s1-12 technical_coordinator_id	Tech. koordinátor	VCHR2(20)		TechnicalCoordinator	nepoužito
	Náklady				
s27 zero_cost_flag	Nulové nákl.	VCHR2(1)	Ano	InventoryPartZeroCost	konstanta N jedna z Y-,Zero Cost Allowed'/N-,Zero Cost Forbidden'/ O-,Zero Cost Only', default ,N'
s1-10 inventory_valuation_method	Metoda ocenění zásob	VCHR2(50)	Ano	InventoryValueMethod	konstanta ST jedna z AV - Weighted Average / ST - Standard Cost / FIFO - First In First Out / LIFO - Last In First Out
s1-14 configuration_cost_method	Metoda konfigurace nákladů	VCHR2(35)	Ano	ConfigurationCostMethod	konstanta COST PER CONFIGURATION jedna z COST PER BASE PART - Cost per Base Part / COST PER CONFIGURATION - Cost per Configuration
s1-13 actual_cost	Skutečné náklady	VCHR2(50)	Ano	ActualCost	konstanta NO ACTUAL COST jedna z ACTUAL COST - Actual Cost / NO ACTUAL COST – No Actual Cost
d1-1 actual_cost_activated	Skutečné náklady aktivovány (není na obrazovce)	DATE(7)			nepoužito
s1-n7 estimated_material_cost	Odhad. materiál. náklady	NUMBER(22)	Ano		konstanta 0
s1-3 part_cost_group_id	Nákl. skup. polož.	VCHR2(5)			nepoužito
	Konfigurace				
	Různé info o položce				
n2 cycle_period	Období cyklu	NUMBER(22)	Ano		konstanta 0

Skladová položka – Obecné (Axalant → IFS)					INVENTORY_PART	
LU	INVENTORY_PART			IE LU	IeUtilInventoryPart	
Tabulka	INVENTORY_PART_TAB			Importní funkce	ImInventoryPart	
Parametry	UPDATE~TRUE/FALSE, MERGE~TRUE/FALSE			CAT	ie_inventorypart	
Vstupní soubor				APY/API	IeUtilInventoryPart.ap_	
Poznámky	Musí být zadáno: PART_CATALOG Uživatel IFSAPP musí mít povolena všechna umístění obsažená v importu Parametr MERGE má význam pouze pokud parametr UPDATE je TRUE. Význam: TRUE data původního záznamu jsou bezesbýtku přepsána novými hodnotami (i kdyby měly být NULL) FALSE data původního záznamu jsou přepsána pouze pokud je nová hodnota různá od NULL					
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka	
s16	count_variance	Celk. inv. rozd.	NUMBER(22)	Ano		nepoužito
	cycle_code	Periodická inventura	VCHR2(1)	Ano	InventoryPartCountType	konstanta N jedna z Y-,Cyclic Counting`/N-,Not Cyclic Counting`, default ,N`
n1-5	cust_warranty_id	Záruka zákazníka	NUMBER(22)		CustWarranty	
n1-6	sup_warranty_id	Záruka dodavatele	NUMBER(22)		SupWarranty	
s20	oe_alloc_assign_flag	Rezervace při vložení obj.	VCHR2(1)	Ano	CustOrdReservationType	konstanta N jedna z Y-,Priority reservation`/N-,Normal reservation` default na základě asset_class
s1-11	negative_on_hand	Záp. mn. na skladě povoleno	VCHR2(50)	Ano	NegativeOnHand	konstanta NEG ONHAND NOT OK jedna z NEG ONHAND OK - Negative On Hand Allowed / NEG ONHAND NOT OK - Negative On Hand Not Allowed
s30	shortage_flag	Upozornění na nedostat.	VCHR2(1)	Ano	InventoryPartShortage	konstanta Y jedna z Y-,Shortage Notation`/N-,No Shortage Notation` default na základě asset_class
s1-2	stock_management	V.M.I.	VCHR2(50)	Ano	InventoryPartManageme nt	konstanta SYSTÉM MANAGED INVENTORY jedna z VENDOR MANAGED INVENTORY-,Vendor Managed Inventory`/ SYSTEM MANAGED INVENTORY-,System Managed Inventory` default ,SYSTEM MANAGED INVENTORY`
s1-1	forecast_consumption_fla g	Online spotřeba	VCHR2(30)	Ano	InvPartForecastConsum	konstanta NOFORECAST jedna z FORECAST-,Make Forecast Consumption`/ NOFORECAST -`No Forecast Consumption` default na základě asset_class
s21	onhand_analysis_flag	Kontrola dostupnosti	VCHR2(1)	Ano	InventoryPartOnhAnalys	konstanta Y jedna z Y-,Make Onhand Analysis`/N-,Not Onhand Analysis`
s29	eng_attribute	Konstr. atribut	VCHR2(5)		CharacteristicTemplate	konstanta TPV
n1-4	max_actual_cost_update	Max. aktualizace skutečných nákl.	NUMBER(22)			nepoužito

① pouze insert ② pokud type_code jeden z Purchased (raw)/Purchased/Expense ③ pokud type_code jeden z Manufactured/Manufactured Recipe

Skladová položka – Obecné – změna stavu (Axalant → IFS)						INVENTORY_PART
LU		INVENTORY_PART		IE LU	IeUtilInventoryPart	
Tabulka		INVENTORY_PART_TAB		Importní funkce	ImInventoryPart	
Parametry		UPDATE~TRUE MERGE~TRUE		CAT	ie_inventorypart	
Vstupní soubor				APY/API	IeUtilInventoryPart.ap_	
Poznámky		Musí být zadáno: PART_CATALOG Uživatel IFSAPP musí mít povolena všechna umístění obsažená v importu Parametr MERGE má význam pouze pokud parametr UPDATE je TRUE. Význam: TRUE data původního záznamu jsou bezesbýtku přepsána novými hodnotami (i kdyby měly být NULL) FALSE data původního záznamu jsou přepsána pouze pokud je nová hodnota různá od NULL				
Jméno pole		Popis	Datový typ	Pov.	Reference	Poznámka
Obecné						
s1	part_no	Sklad. položka	VCHR2(25)	Ano	PartCatalog	
s2	contract	Místo	VCHR2(5)	Ano	Site	konstanta BO
s7	part_status	Stav polož.	VCHR2(1)	Ano	InventoryPartStatusPar	Axalant: speciální kód; IFS A-aktivní,Z-zrušeno,F-fiktivní

Skladová položka – Diskrétní charakteristika						INV_PART_DISCRETE_CHAR
LU		INV_PART_DISCRETE_CHAR		IE LU	IeUtilInvPartDisCh	
Tabulka		INVENTORY_PART_CHAR_TAB		Importní funkce	Im_Inv_Part_Discrete_Char	
Parametry		UPDATE~TRUE/FALSE		CAT	ie_inventory_part_char	
Vstupní soubor				APY/API	IeUtilInvPartDisCh	
Poznámky						
Jméno pole		Popis	Datový typ	Pov.	Reference	Poznámka
s1	part_no	Č. položky	VCHR2(25)	Ano	InventoryPart	číslo vyráběné položky
s2	contract	Místo	VCHR2(5)	Ano		konstanta BO
s3	characteristic_code	Kód charakteristiky	VCHR2(5)	Ano	Characteristic	KR, resp. TC (KR=kód rozpadu, TC=tvarové číslo)
s4	unit_meas	MJ	VCHR2(10)		IsoUnit	
s5	attr_value	Hodnota atributu	VCHR2(60)		DiscreteCharacValue	hodnota charakteristiky

Skladová položka – Data plánování			INVENTORY_PART_PLANNING
LU	INVENTORY_PART_PLANNING	IE LU	IeUtilInventPartPlan
Tabulka	INVENTORY_PART_PLANNING_TAB	Importní funkce	ImInventPartPlanning
Parametry		CAT	ie_inventorypart.cat
Vstupní soubor		APY/API	IeUtilInventoryPartPlanning.ap_
Poznámky	Musí být zadáno: PART_CATALOG, INVENTORY_PART Vytvořeno automaticky při založení skladové položky do INVENTORY_PART		

Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
	Data plánování				
s1 part_no	Sklad. položka	VCHR2(25)	Ano	InventoryPart	
s2 contract	Místo	VCHR2(5)	Ano		konstanta BO
s7 mrp_order_code	Kód objedn. MRP	VCHR2(1)	Ano	MrpRule	dle kódu rozbalení z Axalant Pro varáběné položky (MIDNO.DD=0) Pro KR=1;3 → konstanta A Pro KR=0 → konstanta G
n5 min_order_qty	Min. dávka	NUMBER(22)	Ano		konstanta 0
n4 max_order_qty	Max. dávka	NUMBER(22)	Ano		konstanta 0
n6 mul_order_qty	Násobek dávky	NUMBER(22)	Ano		konstanta 0
n12 shrinkage_fac	Koeficient ztrát (%)	NUMBER(22)	Ano		konstanta 0
n17 shrinkage_rounding	Zaokrouhl. ztrát	NUMBER(22)	Ano		konstanta 12
n10 service_rate	Udržov. sazba (%)	NUMBER(22)	Ano	ServiceRate	konstanta 50
n13 std_order_size	Stand. dávka	NUMBER(22)	Ano		konstanta 0
n1 carry_rate	Skl. úrok. sazba	NUMBER(22)	Ano		konstnata 0
n9 safety_stock	Pojistná zásoba	NUMBER(22)	Ano		konstanta 0
n7 order_point_qty	Úrov. pro doobj.	NUMBER(22)	Ano		konstanta 0
n2 lot_size	Mn. pro doobjedn.	NUMBER(22)	Ano		konstanta 0
n3 maxweek_supply	Plánov. na pevné obd.	NUMBER(22)	Ano		konstanta 0
n11 setup_cost	Specif. nákl.	NUMBER(22)	Ano		konstanta 0
d1 last_activity_date	Poslední změna	DATE(7)	Ano		SYSDATE
n8 qty_predicted_consumption	Předpov. roční spotř.	NUMBER(22)			nepoužito
s5 order_requisition	Typ pořízení	VCHR2(1)		InventoryPartSupplyType	pokud číslo výrobku začíná 1, nebo 2 and nezačíná 20, pak konstanta D, jinak konstanta R jedna z O-,Order\R-,Requisition\S-,Schedule\D-,DOP, default R

Skladová položka – Data plánování			INVENTORY_PART_PLANNING
LU	INVENTORY_PART_PLANNING	IE LU	IeUtilInventPartPlan
Tabulka	INVENTORY_PART_PLANNING_TAB	Importní funkce	ImInventPartPlanning
Parametry		CAT	ie_inventorypart.cat
Vstupní soubor		APY/API	IeUtilInventoryPartPlanning.ap_
Poznámky	Musí být zadáno: PART_CATALOG, INVENTORY_PART Vytvořeno automaticky při založení skladové položky do INVENTORY_PART		

Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
s8 proposal_release	Uvolnění návrhu	VCHR2(20)	Ano	OrderProposalRelease	konstanta RELEASE jedna z 'RELEASE'-'Release'/'DO NOT RELEASE'-'Do Not Release^' Default - 'RELEASE'
s6 safety_stock_auto	Pojistná zásoba	VCHR2(1)	Ano	InventoryPartSafeStock	konstanta N jedna z Y-,Auto Safety Stock'/N-,Manual Safety Stock'
s3 lot_size_auto	Úrov. pro doobjedn.	VCHR2(1)	Ano	InventoryPartLotSize	konstanta N jedna z Y-,Auto Lot Size'/N-,Manual Lot Size'
s4 order_point_qty_auto	Mn. pro doobjedn.	VCHR2(1)	Ano	InventoryPartOrderPoint	konstanta N jedna z Y-,Auto Order Point'/N-,Manual Order Point'
d2 order_trip_date	Dat. odesl. objedn.	DATE(7)			nepoužito
① old_lot_size	Staré mn. pro doobjedn.	NUMBER(22)	Ano		-
① old_safety_stock	Stará pojist. Zásoba	NUMBER(22)	Ano		-
① old_order_point_qty	Stará úrov. pro doobjedn.	NUMBER(22)	Ano		-

① automaticky dle předchozích hodnot, nelze provádět update

Skladová položka – Verze (Axalant → IFS)				PART_REVISION
LU	PART_REVISION	IE LU	IeUtilPartRevision	
Tabulka	PART_REVISION_TAB	Importní funkce	Im_Part_Revision	
Parametry	UPDATE~TRUE/FALSE	CAT	ie_inventorypart	
Vstupní soubor		APY/API	IeUtilPartRevision	
Poznámky				

Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
s1 part_no	Číslo položky	VCHR2(25)	Ano	InventoryPart	
s2 contract	Místo	VCHR2(5)	Ano		konstanta BO
s3 eng_chg_level	Konstrukční změna	VCHR2(3)	Ano		číslo verze
d1 eff_phase_in_date	Platnost od	DATE(7)	Ano		
d2 eff_phase_out_date	Platnost do	DATE(7)			
s4 part_revision_lock	Blokování verze položky	VCHR2(2)	Ano	PartRevisionLock	konstanta N
s5 revision_text	Text verze	VCHR2(80)			
s6 part_revision_report	Sestava verze položky	VCHR2(2)	Ano	PartRevisionReport	konstanta N
s7 part_revision_status	Stav verze položky	VCHR2(2)	Ano	PartRevisionStatus	konstanta E
s8 part_revision_usage	Použití verze položky	VCHR2(6)	Ano	PartRevisionUsage	konstanta UNLIM
s9 engineer_code	Kód výrobního inženýra	VCHR2(3)			
s10 eng_chg_no	Č. konstrukční změny	VCHR2(10)			
s11 eng_draw_ref	Odkaz na výkres	VCHR2(20)			
s12 federal_stock_no		VCHR2(25)			
s13 mil_spec		VCHR2(35)			
s14 classification_code	Kód klasifikace	VCHR2(25)			
s15 document_category	Kategorie dokumentu	VCHR2(10)			
s16 document_chg_class	Třída změny dokumentu	VCHR2(3)			
s17 document_chg_ord_no	Č. objednávky změny dokumentu	VCHR2(10)			
s18 document_level	Úroveň dokumentu	VCHR2(4)			
s19 document_no	Č. dokumentu	VCHR2(25)			
s20 document_no_sheets	Počet listů dokumentu	VCHR2(4)			
d3 document_release_date	Datum uvolnění dokumentu	DATE(7)			
s21 document_revision	Verze dokumentu	VCHR2(5)			
s22 document_size	Velikost dokumentu	VCHR2(2)			
s23 document_title	Nadpis dokumentu	VCHR2(25)			
d4 create_date	Datum vytvoření	DATE(7)	Ano		SYSDATE
d5 last_activity_date	Datum poslední činnosti	DATE(7)	Ano		SYSDATE
s24 entered_by	Vložil	VCHR2(30)			

Skladová položka – Verze (Axalant → IFS)					PART_REVISION
LU		PART_REVISION		IE LU	IeUtilPartRevision
Tabulka		PART_REVISION_TAB		Importní funkce	Im_Part_Revision
Parametry		UPDATE~TRUE/FALSE		CAT	ie_inventorypart
Vstupní soubor				APY/API	IeUtilPartRevision
Poznámky					
Jméno pole		Popis	Datový typ	Pov. Reference	Poznámka
s25	eng_revision	Konstrukční verze	VCHR2(6)		
s26	eng_revision_desc	Popis konstrukční verze	VCHR2(35)		
s27	effective_status	Stav účinnosti	VCHR2(20)	Ano RevisionEffectiveStatus	konstanta NO_EFFECT
s28	effective_repair_status	Stav účinnosti opravy	VCHR2(20)	Ano RevisionEffectiveStatus	konstanta NO_EFFECT

Pracoviště – Základní data (IFS→Axalant)					WORK_CENTER
LU	WORK_CENTER	IE LU	IeUtilWorkCenter		
Tabulka	WORK_CENTER_TAB	Importní funkce	ImWorkCenter		
Parametry	UPDATE~TRUE/FALSE	CAT	ie_manufstandards.cat		
Vstupní soubor		APY/API	IeUtilWorkCenter.ap_		
Poznámky					
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
s1 work_center_no	Pracoviště	VCHR2(5)	Ano		
s3 description	Popis	VCHR2(35)	Ano		
s2 contract	Místo	VCHR2(5)	Ano	Site	
n1 queue_time	Čekací doba	NUMBER(22)	Ano		default 0
s7 work_center_code	Kód pracoviště	VCHR2(20)	Ano	WorkCenterCode	jedna z I-'Internal work center'/O-'Outside work center', default I
s4 calendar_id	Id kalendáře	VCHR2(10)	Ano	WorkTimeCalendar	
n2 utilization	Využití	NUMBER(22)	Ano		default 100
s5 uom	Měrné jednotky	VCHR2(10)	Ano	IsoUnit	default 'HOUR'
s9 sched_capacity	Plán. kapacita	VCHR2(20)	Ano	SchedCapacity	jedna z I-'Infinite capacity'/F-'Finite capacity' default 'Finite capacity', pro CBS Site, jinak 'Infinite Capacity'
s8 usage_code	Kód použití	VCHR2(20)	Ano	WorkCenterUsageCode	jedna z A-Active/P-Planned, default A
	Detail				
s13 production_line	Výrobní linka	VCHR2(12)		ProductionLine	
n10 average_capacity	Průměrná kapacita	NUMBER(22)	Ano		default 8
n11 demonstrated_capacity	Prokázaná kapacita	NUMBER(22)	Ano		
s14 department_no	Oddělení	VCHR2(5)		WorkCenterDepartment	
n6 max_hours_per_order	Max. čas na operaci	NUMBER(22)	Ano		default 8
s19 cost_center_id	Id nákl. střediska	VCHR2(20)			středisko
s12 time_reporting	Odepisování času	VCHR2(20)	Ano	TimeReporting	jedna z Y-'Shop Floor Reporting'/N-'Shop Order', default N
	Kapacita RRP				
n7 units_per_days	Jednotky/den	NUMBER(22)	Ano		default 1
n9 rrp_efficiency	Účinnost RRP	NUMBER(22)	Ano		default 100
	Nezařazeno :				
n3 cum_deviation	Celková odchylka	NUMBER(22)			
s6 input_output_code	Kód vstupu/výstupu	VCHR2(20)			
n4 input_hrs_week	Vstupní hodiny týdne	NUMBER(22)			
n5 output_hrs_week	Výstupní hodiny týdne	NUMBER(22)			
n8 resource_days	Denní zdroj	NUMBER(22)	Ano		default 1
s10 sched_finite_lock		VCHR2(20)	Ano	SchedFiniteLock	jedna z Y-Lock/N-Unlocked
s11 time_type	Typ času	VCHR2(20)	Ano	TimeType	jedna z M-Machines/P-Personnel, default M

Pracoviště – Základní data (IFS→Axalant)					WORK_CENTER
LU		WORK_CENTER		IE LU	IeUtilWorkCenter
Tabulka		WORK_CENTER_TAB		Importní funkce	ImWorkCenter
Parametry		UPDATE~TRUE/FALSE		CAT	ie_manufstandards.cat
Vstupní soubor				APY/API	IeUtilWorkCenter.ap_
Poznámky					
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
s15	source	Zdroj	VCHR2(25)		
	note_id	Id pozn.	NUMBER(22)	Ano	
s16	note_text	Text pozn.	VCHR2(2000)		
d1	create_date	Dat. vytvoření	DATE(7)	Ano	
d2	last_activity_date	Dat. posl. činnosti	DATE(7)	Ano	
s17	company	Společnost	VCHR2(20)	Ano	
s18	code_part		VCHR2(1)	AccountingCodePartValue	musí být 'B', příp. logická reprezentace segmentu typu B
Struktury (výrobní, recepturová, plánovací, konfigurační) (Axalant→IFS)					MANUF_STRUCTURE
LU		PROD_STRUCTURE, RECIPE_STRUCTURE, PLAN_STRUCTURE		IE LU	IeUtilManufStructure
Tabulka		MANUF_STRUCTURE_TAB		Importní funkce	ImManufacturingStructure
Parametry		UPDATE~TRUE/FALSE		CAT	ie_manufstandards
Vstupní soubor				APY/API	IeUtilManufStructure.ap_
Poznámky		Musí být zadáno: PART_CATALOG, INVENTORY_PART hlavičky struktur se zakládají automaticky při založení INVENTORY_PART ve verzi 2002 přibyl nový typ struktury – konfigurační – není řešeno, princip je však obdobný			
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
Menu : Struktury/ Výrobní struktura					
s2	part_no	Č. položky	VCHR2(25)	Ano	InventoryPart
s3	contract	Místo	VCHR2(5)	Ano	Site
s4	eng_chg_level	Č. verze	VCHR2(2)	Ano	
d1	eff_phase_in_date	Platnost od	DATE(7)	Ano	
d2	eff_phase_out_date	Platnost do	DATE(7)		pokud neomezeno, tak NULL
s6	bom_type	Typ struktury	VCHR2(20)	Ano	ShopOrdCode
					konstanta M jedna z M-Manufacturing/F-Repair/P-Purchase/ nepoužito
n11	cust_warranty_id	Záruka zákazníka	NUMBER(22)		
Výrobní struktura					
s5	alternative_no	Alternativa	VCHR2(20)	Ano	ManufStructAlternate
n1	line_item_no	Číslo položky řádky	NUMBER(22)	Ano	konstanta *

Struktury (výrobová, recepturová, plánovací, konfigurační) (Axalant→IFS)						MANUF_STRUCTURE
LU		PROD_STRUCTURE, RECIPE_STRUCTURE, PLAN_STRUCTURE		IE LU	IeUtilManufStructure	
Tabulka		MANUF_STRUCTURE_TAB		Importní funkce	ImManufacturingStructure	
Parametry		UPDATE~TRUE/FALSE		CAT	ie_manufstandards	
Vstupní soubor				APY/API	IeUtilManufStructure.ap_	
Poznámky		Musí být zadáno: PART_CATALOG, INVENTORY_PART hlavičky struktur se zakládají automaticky při založení INVENTORY_PART ve verzi 2002 přibyl nový typ struktury – konfigurační – není řešeno, princip je však obdobný				
Jméno pole		Popis	Datový typ	Pov.	Reference	Poznámka
s7	component_part	Komponenta	VCHR2(25)	Ano	InventoryPart	položka nižší
n2	qty_per_assembly	Množství ke kompletaci	NUMBER(22)	Ano		
n5	operation_no	Číslo operace	NUMBER(22)			nepoužito
n3	component_scrap	Vyřazení komponenty	NUMBER(22)	Ano		konstanta 0
n4	shrinkage_factor	Procento ztrát	NUMBER(22)	Ano		konstanta 0
s9	consumption_item	Spotřební položka	VCHR2(20)	Ano	ConsumptionItem	konstanta Consumed jedna z Not Consumed/Consumed, default Consumed
n6	leadtime_offset	Posun průběžné doby	NUMBER(22)	Ano		konstanta 0
s18	promise_planned	Plánovaný příslibený termín	VCHR2(20)	Ano	PromisePlanned	konstanta Promised jedna z ‚Promised‘/‘Not Promised‘, default ‚Promised ‘
s10	phantom_consume	Fiktivní spotřeba	VCHR2(20)	Ano	PhantomConsume	konstanta NOT PHANTOM CONSUME jedna z ‚NOT PHANTOM CONSUME‘/‘PHANTOM CONSUME‘, default ‚NOT PHANTOM CONSUME‘
s11	charged_item	Položka s poplatkem	VCHR2(20)	Ano	ChargedItem	konstanta ITEM NOT CHARGED jedna z ‚CHARGED ITEM‘/‘ITEM NOT CHARGED‘, default ‘ITEM NOT CHARGED‘
s8	issue_to_loc	Výdej z umístění	VCHR2(35)		InventoryLocation	nepoužito
s19	draw_pos_no	Číslo pozice výkresu	VCHR2(10)			nepoužito
s16	note_text	Text poznámky	VCHR2(2000)			Axalant: Popis
s12	print_unit	MJ tisku	VCHR2(10)	Ano	IsoUnit	Axalant: MJ položky nižší
s20	value_analysis_func_id	Id funkce analýzy hodnoty	VCHR2(10)		ValueAnalysisFunc	nepoužito
s21	value_analysis_func_type	Typ funkce analýzy hodnoty	VCHR2(20)		ValueAnalysisFuncType	nepoužito
Menu : Struktury/						
Recepturová						
struktura						
Struktura receptury						
n9	line_sequence	Sekvence řádky	NUMBER(22)	Ano		totéž co v line_item_no

Struktury (výrobní, recepturová, plánovací, konfigurační) (Axalant→IFS)					MANUF_STRUCTURE
LU		PROD_STRUCTURE, RECIPE_STRUCTURE, PLAN_STRUCTURE	IE LU		IeUtilManufStructure
Tabulka		MANUF_STRUCTURE_TAB	Importní funkce		ImManufacturingStructure
Parametry		UPDATE~TRUE/FALSE	CAT		ie_manufstandards
Vstupní soubor			APY/API		IeUtilManufStructure.ap_
Poznámky		Musí být zadáno: PART_CATALOG, INVENTORY_PART hlavičky struktur se zakládají automaticky při založení INVENTORY_PART ve verzi 2002 přibyl nový typ struktury – konfigurační – není řešeno, princip je však obdobný			
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
n7 ① ③ weight_share	Hmotnostní podíl	NUMBER(22)			nepoužito
s 13 ① ③ fixed_qty_flag	Příznak pevného množství	VCHR2(20)		FixedQtyFlag	nepoužito jedna z ‚Not fixed‘/‚Fixed‘, default ‚Not fixed‘
n8 ① ③ procedure_step	Krok procedury	NUMBER(22)			nepoužito
Menu : Struktury/ Plánovací struktura Plánovací struktura					
s15 ① ② component_contract	Místo komponenty	VCHR2(5)	Ano	Site	konstanta BO pro PROD_STRUCTURE a RECIPE_STRUCTURE se použije contract
n6 leadtime_offset	Posun průběžné doby	NUMBER(22)	Ano		konstanta 0
Menu : Struktury/ Konfigurační struktura Konfigurační struktura					
d4 last_activity_date	Poslední činnost	DATE(7)	Ano		SYSDATE
n10 component usage_factor	Koeficient využití komponenty	NUMBER(22)	Ano		konstanta 1
Nezařazeno :					
s1 manuf_structure_type	Typ výrobní struktury	VCHR2(20)	Ano	ManufStructureType	konstanta ProdStructure, pokud kusovník položky vyšší neobsahuje variantní součást, konstanta ConfigStructure, pokud variantní součást obsahuje jeden z PlanStructure/ProdStructure/RecipeStructure/ConfigStructure
④ s14 eng_chg_no	Č. konstr. změny	VCHR2(10)			nepoužito

Struktury (výrobová, recepturová, plánovací, konfigurační) (Axalant→IFS)					MANUF_STRUCTURE
LU	PROD_STRUCTURE, RECIPE_STRUCTURE, PLAN_STRUCTURE		IE LU	IeUtilManufStructure	
Tabulka	MANUF_STRUCTURE_TAB		Importní funkce	ImManufacturingStructure	
Parametry	UPDATE~TRUE/FALSE		CAT	ie_manufstandards	
Vstupní soubor			APY/API	IeUtilManufStructure.ap_	
Poznámky	Musí být zadáno: PART_CATALOG, INVENTORY_PART hlavičky struktur se zakládají automaticky při založení INVENTORY_PART ve verzi 2002 přibyl nový typ struktury – konfigurační – není řešeno, princip je však obdobný				
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
note_id	Id pozn.	NUMBER(22)	Ano		přiděleno automaticky
s17 source	Zdroj	VCHR2(25)			nepoužito
d3 create_date	Vytvořit	DATE(7)	Ano		SYSDATE
n12 qty_kg		NUMBER(22)	Ano		konstanta 0
s22 active_position	Platná pozice struktury	VCHR2(5)			konstanta TRUE pro aktivní pozice, konstanta FALSE pro neaktivní
① údaj nemá význam pro PROD_STRUCTURE		② údaj nemá význam pro RECIPE_STRUCTURE			
③ údaj nemá význam pro PLAN_STRUCTURE		④ údaj nemá význam ve verzi 2002			
Technologické postupy – Hlavička (Axalant → IFS)					ROUTING_HEAD
LU	ROUTING_HEAD		IE LU	IeUtilRoutingHead	
Tabulka	ROUTING_HEAD_TAB		Importní funkce	Im_Routing_Head	
Parametry	UPDATE~TRUE/FALSE		CAT	ie_manufstandards.cat	
Vstupní soubor			APY/API	IeUtilRoutingHead.ap_	
Poznámky	bude odvozeno z ROUTING_OPERATION				
Jméno pole	Popis	Datový typ	Pov.	Reference	Poznámka
s1 part_no	Č. položky	VCHR2(25)	Ano	InventoryPart	
s2 contract	Místo	VCHR2(5)	Ano		konstanta BO
s3 routing_revision	Verze postupu	VCHR2(4)	Ano		verze Axalant + 1
s4 bom_type	Typ technologického postupu	VCHR2(20)	Ano	ShopOrdCode	konstanta M jedna z M-Manufacturing/F-Repair/P-Purchase
d1 phase_in_date	Platnost od	DATE(7)	Ano		
d2 phase_out_date	Platnost do	DATE(7)			
- note_id	Id poznámky	NUMBER(22)	Ano		
s5 note_text	Text poznámky	VCHR2(2000)			
d3 create_date	Datum vytvoření	DATE(7)	Ano		SYSDATE

Technologické postupy (Axalant→IFS)					ROUTING_OPERATION	
LU	ROUTING_OPERATION		IE LU	IeUtilRoutingOperation		
Tabulka	ROUTING_OPERATION _TAB		Importní funkce	ImRoutingOperation		
Parametry			CAT	ie_manufstandards		
Vstupní soubor			APY/API	IeUtilRoutingOperation.ap_		
Poznámky	Musí být zadáno: PART_CATALOG, INVENTORY_PART, WORK_CENTER hlavičky struktur se zakládají automaticky při založení INVENTORY_PART					
Je možno dělat UPDATE						
Jméno pole		Popis	Datový typ	Pov.	Reference	Poznámka
s1	part_no	Č. položky	VCHR2(25)	Ano	InventoryPart	číslo položky vyšší
s2	contract	Místo	VCHR2(5)	Ano		konstanta BO
s3	routing_revision	Verze postupu	VCHR2(4)	Ano		verze Axalant + 1
s5	bom_type	Typ postupu	VCHR2(20)	Ano	ShopOrdCode	konstanta M jedna z M-,Manufacturing'/F-,Repair'/P-,Purchase', ,P' nepoužito
d1	phase_in_date	Platnost od	DATE(7)	Ano		
d2	phase_out_date	Platnost do	DATE(7)			
Technologický postup						
s4	alternative_no	Alternativa	VCHR2(20)	Ano	RoutingAlternate	konstanta *
n2	operation_no	Číslo operace	NUMBER(22)	Ano		
s6	operation_description	Popis operace	VCHR2(35)	Ano		
s16	std_operation_name	Název standardní operace	VCHR2(12)		StandardOperation	
s8	work_center_no	Číslo pracoviště	VCHR2(5)		WorkCenter	! přechíslování Dialog→IFS
n6	mach_setup_time	Přípravný čas strojní	NUMBER(22)	Ano		Axalant: Přípravný čas/60 zaokrouhleno na 3 místa
n5	mach_run_factor	Koeficient strojní	NUMBER(22)	Ano		Axalant: Kusový čas/60 zaokrouhleno na 3 místa
s11	labor_class_no	Mzdová třída	VCHR2(10)		LaborClass	
n9	labor_setup_time	Přípravný čas obsluhy	NUMBER(22)	Ano		Axalant: Přípravný čas mzdový/60 zaokrouhleno na 3 místa
n8	labor_run_factor	Koeficient obsluhy	NUMBER(22)	Ano		Axalant: Kusobvý čas mzdový/60 zaokrouhleno na 3 místa
s10	run_time_code	Jednotka koeficientu	VCHR2(20)	Ano	OperationTimeCode	konstanta 1 jedna z 1-,Hours/Unit'/2-,Units/Hour'/3-,Hours', default ,Hours/Unit
n10	crew_size	Počet členů obsluhy	NUMBER(22)			Axalant: Splitting
n7	move_time	Doba přepravy	NUMBER(22)	Ano		Axalant: Čas přechodu
s7	parallel_operation	Paralelní operace	VCHR2(20)	Ano	ParallelOperation	konstanta konstanta N jedna z Y-,Parallel'/N-,Not parallel', default ,Not parallel'
n4	efficiency_factor	Výkonnost	NUMBER(22)			konstanta 100
s13	machine_no	Číslo stroje	VCHR2(15)			nepoužito

Technologické postupy (Axalant→IFS)					ROUTING_OPERATION	
LU		ROUTING_OPERATION	IE LU	IeUtilRoutingOperation		
Tabulka		ROUTING_OPERATION_TAB	Importní funkce	ImRoutingOperation		
Parametry			CAT	ie_manufstandards		
Vstupní soubor			APY/API	IeUtilRoutingOperation.ap_		
Poznámky	Musí být zadáno: PART_CATALOG, INVENTORY_PART, WORK_CENTER hlavičky struktur se zakládají automaticky při založení INVENTORY_PART					
Je možno dělat UPDATE						
Jméno pole		Popis	Datový typ	Pov.	Reference	Poznámka
s9	alternate_work_center	Alternativní pracoviště	VCHR2(5)			nepoužito
s12	outside_op_item	Položka externí operace	VCHR2(25)		PurchasePart	Axalant: Kooperační položka
s14	note_text	Text poznámky	VCHR2(2000)			Poznámka
n11	operation_usage_factor	Koeficient využití operace	NUMBER	Ano		konstanta 1
Nezařazeno :						
n1	operation_id	Id operace	NUMBER(22)	Ano		totéž jako v operation_no
n3	lot_qty	Mn. dávky	NUMBER(22)			konstanta 1
	note_id	Id pozn.	NUMBER(22)	Ano		nepoužito
s15	source	Zdroj	VCHR2(25)			nepoužito
d3	create_date	Dat. vytvoření	DATE(7)			nepoužito
d4	last_activity_date	Dat. posl. činnosti	DATE(7)			nepoužito
s17	active_operation	Platná operace	VCHR2(5)			konstanta TRUE

Pracovní směrnice technologického postupu (Axalant → IFS)					ROUTING_OPER_WORK_GUIDE	
LU	ROUTING_OPER_WORK_GUIDE		IE LU	IeUtilRoutOpWoGuide		
Tabulka	ROUTING_OPER_WORK_GUIDE_TAB		Importní funkce	Im_Routing_Oper_Wo_Guide		
Parametry	UPDATE~TRUE/FALSE		CAT	ie_routingoperworkguide		
Vstupní soubor			APY/API	IeUtilRoutingOperWoGuide		
Poznámky						
Jméno pole		Popis	Datový typ	Pov.	Reference	Poznámka
s1	part_no	Č. položky	VCHR2(25)	Ano		TPV_IFS.VYR
s2	contract	Místo	VCHR2(5)	Ano		konstanta BO
s3	routing_revision	Verze postupu	VCHR2(4)	Ano		konstanta 1
s4	alternative_no	Č. alternativy	VCHR2(20)	Ano		konstanta *
s5	bom_type	Typ struktury	VCHR2(20)	Ano	ShopOrdCode	konstanta M
n1	operation_id	Id operace	NUMBER(22)	Ano	RoutingOperation	TPV_IFS.POZ
n2	guideline_no	Č. směrnice	NUMBER(22)	Ano		POUZE Pro TPV_IFS.TYP věty „A“
s6	guideline_desc	Popis směrnice	VCHR2(50)	Ano		mastex68.prub
s7	guideline_text	Text směrnice	VCHR2(2000)	Ano		konstanta *
n3	guideline_seq	Pořadí směrnice	NUMBER(22)			mastex68.text
n4	guideline_template	Šablona směrnice	NUMBER(22)		WorkGuidelineTemplate	mastex68.prub
						nepoužito

Standardní názvy (IFS→Axalant)					STANDARD_NAMES
LU		STANDARD_NAMES		IE LU	IeUtilStandardNames
Tabulka		STANDARD_NAMES_TAB		Importní funkce	-
Parametry				CAT	ie_standardnames.cat
Vstupní soubor				APY/API	IeUtilStandardNames.ap_
Poznámky					
Jméno pole		Popis	Datový typ	Pov. Reference	Poznámka
n1	std_name_id	Id standardního názvu	NUMBER(22)	Ano	
s2	std_name	Standardní název	VARCHAR2(35)	Ano	
s3	language_code	Kód jazyka	VARCHAR2(2)		prázdné = ,cs'

Integrace systému správy dokumentace AXALANT a IS IFS

Příloha II.

Radim Stuchlý

Obsah

1	Procedury přenosu směrem Axalant – IFS.....	3
	Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene	3
	Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upd.....	19
	Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upds	25
	Procedura TD_IE_AXA_IFS/axa_ifs_postup	30
2	Procedury přenosu směrem IFS – Axalant.....	37
	Procedura TD_IE_AXA_IFS/ifs_axa_import	37
	Procedura TD_IE_AXA_IFS/ifs_axa_nakup	39
	Procedura TD_IE_AXA_IFS/ifs_axa_stred	45
	Procedura TD_IE_AXA_IFS/ifs_axa_textsou.....	46
	Procedura TD_IE_AXA_IFS/ifs_axa_prac	47

Tato příloha obsahuje procedury pro import a export přenosových dávek psané v programovacím jazyku Axalantu – LogiView.

1 Procedury přenosu směrem Axalant – IFS

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene	Export součástí, kusovníku, technologického postupu, jeho operací a úkonů operací
---	---

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek	P. Text
2	TD_POMI=TD_MASTER_CID_ACT
4	select(TD_MASTER_DRUHPOL)
6	where("T_MASTER_DAT.C_ID"=TD_POMI)
8	RES=exec_select(1)
10	if (TD_MASTER_DRUHPOL=="V")
12	select(TD_MASTER_PART_ID,TD_MASTER_TECHN_ID,TD_MASTER_TEXT_ID,TD_MASTER_DAT_ROZMER,TD_MASTER_DAT_POPIS,TD_MASTER_D AT_TJ,TD_MASTER_DAT_NJ,TD_MASTER_DAT_UNIT,TD_MASTER_DAT_MJTJ,TD_MASTER_DAT_MJNJ,TD_MASTER_DAT_USKLAD,TD_MASTER_S PEC_KOD,TD_MASTER_KOD_ROZP)
14	where("T_MASTER_DAT.C_ID"=TD_POMI)
16	RES=exec_select(1)
18	if (strpart(TD_MASTER_TECHN_ID,1,1)!="K" & strpart(TD_MASTER_TECHN_ID,1,2)!="20")
20	select(TD_TEXTSOU_NAME_FRA,TD_TEXTSOU_TEXT1_FRA)
22	where("T_TEXTSOU_DAT.ID"=TD_MASTER_TEXT_ID)
24	RES=exec_select(1)
26	select(TD_MASTER_SKUP_VYR,TD_MASTER_DAT_SKLAD,TD_MASTER_DAT_UCET,TD_MASTER_C_HMOTNOST,TD_MASTER_PART_VERS,TD_MA STER_VAL_FROM,TD_MASTER_DAT_PREDPM,TD_MASTER_DAT_PREDP,TD_MASTER_TVAR_C)
28	where("T_MASTER_DAT.C_ID"=TD_POMI)
30	RES=exec_select(1)
32	TD_MASTER_DAT_PREDPM=TD_MASTER_DAT_PREDPM+TD_MASTER_DAT_PREDP
34	TD_POMI5=0
36	RES=scan(TD_MASTER_PART_VERS,"%i",TD_POMI5)
38	TD_POMI5=TD_POMI5+1
40	TD_IE_IFS_GEN="#TD-IE-IFS-GEN"
42	RES=#nsv_get_num(TD_IE_IFS_GEN,TD_IE_EX_BATCH_ID)

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

44     if (RES!=0)
46         put("Chyba při aktualizaci generátoru čísel! Informujte správce...")
48         exit()
50     endif
52     TD_POMS4=strprint("%i",TD_IE_EX_BATCH_ID)
54     TD_POMS4=strpart(TD_POMS4,strlen(TD_POMS4),1)
56     if (TD_POMS4=="0" | TD_POMS4=="2" | TD_POMS4=="4" | TD_POMS4=="6" | TD_POMS4=="8")
58         TD_IE_EX_BATCH_ID=TD_IE_EX_BATCH_ID+1
60     else
62         TD_IE_EX_BATCH_ID=TD_IE_EX_BATCH_ID+2
64     endif
66     TD_IE_EX_ISIMPORT="TRUE"
68     TD_IE_EX_STATE="Open"
70     TD_IE_DL_LINEID=0
72     TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
74     TD_IE_EX_NEXTLINEID=TD_IE_DL_LINEID
76     TD_IE_EX_ERRORFATAL="TRUE"
78     TD_IE_EX_FORCECOMMIT="FALSE"
80     TD_IE_EX_SYSTEMID="AXALANT"
82     TD_IE_EX_DESCRIPTION=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
84     TD_IE_EX_CREATEDATE=date()
86     TD_IE_EX_EXECUTEDATE=date()
88     insert_ent("TD-IE-EXECUTE-BATCH")
90     var(TD_IE_EX_BATCH_ID,TD_IE_EX_ISIMPORT,TD_IE_EX_STATE,TD_IE_EX_NEXTLINEID,TD_IE_EX_ERRORFATAL)
92     var(TD_IE_EX_FORCECOMMIT,TD_IE_EX_SYSTEMID,TD_IE_EX_CREATEDATE,TD_IE_EX_EXECUTEDATE,TD_IE_EX_DESCRIPTION)
94     RES=execute(1)
96     TD_IE_DL_BATCH_ID=TD_IE_EX_BATCH_ID
98     TD_IE_DL_STATUS="READY"
100    TD_IE_DL_SCHMEID="PART_CATALOG"
102    TD_IE_DL_S1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
104    TD_IE_DL_S2=TD_TEXTSOU_TEXT1_FRA
106    TD_IE_DL_S6="NOT SERIAL TRACKING"
108    TD_IE_DL_S7="NOT SERIAL TRACKING"

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

110 select(TD_ART_STR_CID2[1])
112 where("T_MASTER_STR.C_ID_1"=TD_POMI)
114 TD_POMI1=exec_select(0)
116 TD_IE_DL_S10="NOT CONFIGURED"
118 if (TD_POMI1>0)
120     TD_COUNT=1
122     while (TD_COUNT<=TD_POMI1)
124         select(TD_MASTER_DAT_TYP)
126         where("T_MASTER_DAT.C_ID"=TD_ART_STR_CID2[TD_COUNT])
128         RES=exec_select(1)
130         if (TD_MASTER_DAT_TYP=="VARATT")
132             TD_IE_DL_S10="CONFIGURED"
134         endif
136         TD_COUNT=TD_COUNT+1
138     done
140     TD_COUNT=0
142 endif
144 TD_IE_DL_S8="MANUAL"
146 TD_IE_DL_S9="NOT LOT TRACKING"
148 TD_IE_DL_N1=TD_MASTER_TEXT_ID
150 TD_IE_DL_S5=TD_MASTER_DAT_UNIT
152 if (TD_MASTER_PART_VERS=="0")
154     insert_ent("TD-IE-DATA-LINE")
156     var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
158     var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S6,TD_IE_DL_S7,TD_IE_DL_S10,TD_IE_DL_S8,TD_IE_DL_S9,TD_IE_DL_S5,TD_IE_DL_N1)
160     RES=execute(1)
162     TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
164     TD_IE_DL_SCHMEID="INVENTORY_PART"
166     TD_IE_DL_S1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
168     TD_IE_DL_S2="BO"
170     TD_IE_DL_S14=TD_TEXTSOU_TEXT1_FRA
172     TD_IE_DL_S13=TD_MASTER_DAT_UNIT
174     TD_IE_DL_S25="1"

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

176 TD_IE_DL_S26=TD_MASTER_DAT_ROZMER
178 TD_IE_DL_N6=0
180 TD_IE_DL_N5=0
182 TD_IE_DL_N4=20
184 TD_IE_DL_S28="IO"
186 TD_IE_DL_S27="N"
188 TD_IE_DL_N2=0
190 TD_IE_DL_S16="N"
192 TD_IE_DL_S17=TD_MASTER_DAT_POPIS
194 TD_IE_DL_S20="N"
196 TD_IE_DL_S7=TD_MASTER_SPEC_KOD
198 if (TD_IE_DL_S7=="ZR")
200     TD_IE_DL_S7="Z"
202 else
204     TD_IE_DL_S7="A"
206 endif
207 TD_IE_DL_S8=TD_MASTER_SKUP_VYR
208 TD_IE_DL_S30="Y"
209 TD_IE_DL_S15="A"
210 TD_IE_DL_S21="Y"
211 TD_IE_DL_S3="R"
216 if (TD_MASTER_TECHN_ID=="30")
217     TD_IE_DL_N7=TD_MASTER_DAT_PREDPM
218 else
219     TD_IE_DL_N7=0.01
220 endif
221 if (TD_IE_DL_N7==0)
222     TD_IE_DL_N7=0.01
224 endif
226 TD_IE_DL_S29="TPV"
228 insert_ent("TD-IE-DATA-LINE")
230 var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
232 var(TD_IE_DL_N2,TD_IE_DL_N4,TD_IE_DL_N5,TD_IE_DL_N6,TD_IE_DL_N7)

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

234  var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S14,TD_IE_DL_S13,TD_IE_DL_S7,TD_IE_DL_S8,TD_IE_DL_S15,TD_IE_DL_S25,TD_IE_DL_S
26,TD_IE_DL_S28,TD_IE_DL_S27,TD_IE_DL_S16,TD_IE_DL_S17,TD_IE_DL_S20,TD_IE_DL_S21,TD_IE_DL_S30,TD_IE_DL_S29)
236  RES=execute(1)
238  TD_IE_DL_MASTERID=TD_IE_DL_LINEID
240  TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
242  TD_IE_DL_STATUS="DETAIL"
244  if (TD_IE_DL_STATUS=="DETAIL")
246      TD_IE_DL_PARTNO=1
248  else
250      TD_IE_DL_PARTNO=0
252  endif
254  TD_IE_DL_N1=TD_MASTER_C_HMOTNOST
256  TD_IE_DL_S5="MAN"
258  TD_IE_DL_N3=TD_MASTER_TEXT_ID
260  TD_IE_DL_S10="ST"
262  TD_IE_DL_S14="COST PER CONFIGURATION"
264  TD_IE_DL_S13="NO ACTUAL COST"
266  TD_IE_DL_D1=date()
268  TD_IE_DL_S11="NEG ONHAND NOT OK"
270  TD_IE_DL_S2="SYSTEM MANAGED INVENTORY"
272  TD_IE_DL_S1="NOFORECAST"
274  insert_ent("TD-IE-DATA-LINE")
276  var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID,TD_IE_DL_MASTERID,TD_IE_DL_PARTNO)
278  var(TD_IE_DL_N1,TD_IE_DL_N3)
280  var(TD_IE_DL_S5,TD_IE_DL_S10,TD_IE_DL_S14,TD_IE_DL_S13,TD_IE_DL_S11,TD_IE_DL_S2,TD_IE_DL_S1)
282  RES=execute(1)
284  TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
286  TD_IE_DL_STATUS="READY"
288  TD_IE_DL_SCHMEID="INV_PART_DISCRETE_CHAR"
290  TD_IE_DL_S1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
292  TD_IE_DL_S2="BO"
294  TD_IE_DL_S3="KR"
296  TD_IE_DL_S5=strprint("%i",TD_MASTER_KOD_ROZP)

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

298      insert_ent("TD-IE-DATA-LINE")
300      var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
302      var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S5)
304      RES=execute(1)
306      TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
308      TD_IE_DL_S3="TC"
310      TD_IE_DL_S5=TD_MASTER_TVAR_C
312      insert_ent("TD-IE-DATA-LINE")
314      var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
316      var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S5)
318      RES=execute(1)
320      TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
322      TD_IE_DL_SCHMEID="INVENTORY_PART_PLANNING"
324      TD_IE_DL_S1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
326      TD_IE_DL_S2="BO"
328      if (TD_MASTER_KOD_ROZP==1)
330          TD_IE_DL_S7="A"
331          TD_IE_DL_N3=0
332      else
334          TD_IE_DL_S7="G"
335          TD_IE_DL_N3=10
336      endif
338      TD_IE_DL_N5=0
340      TD_IE_DL_N4=0
342      TD_IE_DL_N6=0
344      TD_IE_DL_N12=0
346      TD_IE_DL_N17=12
348      TD_IE_DL_N10=50
350      TD_IE_DL_N13=0
352      TD_IE_DL_N1=0
354      TD_IE_DL_N9=0
356      TD_IE_DL_N7=0
358      TD_IE_DL_N2=0

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

362 TD_IE_DL_N11=0
364 if (strpart(TD_MASTER_TECHN_ID,1,1)=="1" | (strpart(TD_MASTER_TECHN_ID,1,1)=="2" & strpart(TD_MASTER_TECHN_ID,1,2)!="20"))
366 TD_IE_DL_S5="D"
368 else
370 TD_IE_DL_S5="R"
372 endif
374 TD_IE_DL_S8="RELEASE"
376 TD_IE_DL_S6="N"
378 TD_IE_DL_S3="N"
380 TD_IE_DL_S4="N"
382 insert_ent("TD-IE-DATA-LINE")
384 var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
386 var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S7,TD_IE_DL_N5,TD_IE_DL_N4,TD_IE_DL_N6,TD_IE_DL_N12,TD_IE_DL_N17,TD_IE_DL_N10,TD_IE_DL_N
13,TD_IE_DL_N1,TD_IE_DL_N9,TD_IE_DL_N7,TD_IE_DL_N2,TD_IE_DL_N3)
388 var(TD_IE_DL_N11,TD_IE_DL_D1,TD_IE_DL_S5,TD_IE_DL_S8,TD_IE_DL_S6,TD_IE_DL_S3,TD_IE_DL_S4)
390 RES=execute(1)
392 endif
394 TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
396 TD_IE_DL_STATUS="READY"
398 TD_IE_DL_SCHMEID="PART_REVISION"
400 TD_IE_DL_S1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
402 TD_IE_DL_S2="BO"
404 TD_IE_DL_S3=strprint("%i",TD_POMI5)
406 TD_IE_DL_D1=TD_MASTER_VAL_FROM
408 TD_IE_DL_S4="N"
410 TD_IE_DL_S6="N"
412 TD_IE_DL_S7="E"
414 TD_IE_DL_S8="UNLIM"
416 TD_IE_DL_D4=date()
418 TD_IE_DL_D5=date()
420 TD_IE_DL_S27="NO_EFFECT"
422 TD_IE_DL_S28="NO_EFFECT"
424 insert_ent("TD-IE-DATA-LINE")

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

426   var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID,TD_IE_DL_MASTERID,TD_IE_DL_PARTNO)
428   var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_D1,TD_IE_DL_S4,TD_IE_DL_S6,TD_IE_DL_S7,TD_IE_DL_S8,TD_IE_DL_D4,TD_IE_DL_D5,TD
   _IE_DL_S27,TD_IE_DL_S28)
430   RES=execute(1)
432 C   ***** ALTERNATIVY *****
434   TD_IE_DL_SCHMEID="MANUF_STRUCT_ALTERNATE"
436   TD_IE_DL_STATUS="READY"
438   TD_IE_DL_S1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
440   TD_IE_DL_S2="BO"
442   TD_IE_DL_S3=strprint("%i",TD_POMI5)
444   TD_IE_DL_S4="*"
446   TD_IE_DL_S5="M"
448   TD_IE_DL_S8="ProdStructAlternate"
450 C   ***** BOM *****
452   TD_POMS2=TD_MASTER_TECHN_ID
454   TD_POMS3=TD_MASTER_PART_ID
456   if (TD_MASTER_TECHN_ID=="")
458     TD_POMS1=TD_MASTER_PART_ID
460   else
462     TD_POMS1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
464   endif
466   select(TD_ART_STR_CID2[1],TD_ART_STR_PN_ARR[1],TD_ART_STR_PQUANTITY[1],TD_ART_STR_CF_ARR[1],TD_ART_STR_DESC_ARR[1])
468   order_by("T_MASTER_STR.POS_NO"+)
470   where("T_MASTER_STR.C_ID_1"=TD_POMI & "T_MASTER_STR.IN_CONTEXT_FAT"="PRD" & "T_MASTER_STR.C_ID">0)
472   TD_POMI1=exec_select(0)
474   if (TD_POMI1>0)
476     TD_COUNT=1
478     while (TD_COUNT<=TD_POMI1)
480       TD_MASTER_PART_ID=""
482       TD_MASTER_TECHN_ID=""
484       select(TD_MASTER_TECHN_ID,TD_MASTER_PART_ID,TD_MASTER_CUR_FLAG,TD_MASTER_DAT_UNIT)
486       where("T_MASTER_DAT.C_ID"=TD_ART_STR_CID2[TD_COUNT])
488       RES=exec_select(1)

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

490     if (TD_MASTER_CUR_FLAG==TRUE)
492         TD_IE_DL_D1=date()
494         TD_IE_DL_D2=""
496         TD_IE_DL_S7=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
498         TD_IE_DL_S16=TD_ART_STR_DESC_ARR[TD_COUNT]
500         TD_IE_DL_S12=TD_MASTER_DAT_UNIT
502         TD_IE_DL_N9=TD_ART_STR_PN_ARR[TD_COUNT]
504         TD_IE_DL_S1="ProdStructure"
506         TD_IE_DL_D3=date()
508         TD_IE_DL_S4=strprint("%i",TD_POMI5)
510         TD_IE_DL_S3="BO"
512         TD_IE_DL_S6="M"
514         TD_IE_DL_S5="*"
516         TD_IE_DL_N3=0
518         TD_IE_DL_N4=0
520         TD_IE_DL_S9="Consumed"
522         TD_IE_DL_N6=0
524         TD_IE_DL_S18="Promised"
526         TD_IE_DL_S10="NOT PHANTOM CONSUME"
528         TD_IE_DL_S11="ITEM NOT CHARGED"
530         TD_IE_DL_S15="BO"
532         TD_IE_DL_N6=0
534         TD_IE_DL_N12=0
536         TD_IE_DL_N1=TD_ART_STR_PN_ARR[TD_COUNT]
538         TD_IE_DL_N2=TD_ART_STR_PQUANTITY[TD_COUNT]
540         if (TD_ART_STR_CF_ARR[TD_COUNT]==TRUE)
542             TD_IE_DL_S22="TRUE"
544         else
546             TD_IE_DL_S22="FALSE"
548             TD_IE_DL_N2=0
550             TD_IE_DL_S9="Not Consumed"
552         endif
554         TD_IE_DL_S2=TD_POMS1

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

556 TD_IE_DL_STATUS="READY"
558 TD_IE_DL_S4=strprint("%i",TD_POMI5)
560 TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
562 TD_IE_DL_D4=date()
564 TD_IE_DL_SCHMEID="MANUF_STRUCTURE"
566 insert_ent("TD-IE-DATA-LINE")
568 var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
570 var(TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_D1,TD_IE_DL_D2,TD_IE_DL_S6,TD_IE_DL_S5,TD_IE_DL_N1,TD_IE_DL_S7,TD_IE_DL_N2)
572 var(TD_IE_DL_N3,TD_IE_DL_N4,TD_IE_DL_S9,TD_IE_DL_N6,TD_IE_DL_S18,TD_IE_DL_S10,TD_IE_DL_S11,TD_IE_DL_S16,TD_IE_DL_S12)
574 var(TD_IE_DL_N9,TD_IE_DL_S15,TD_IE_DL_N6,TD_IE_DL_S1,TD_IE_DL_D3,TD_IE_DL_D4,TD_IE_DL_N12,TD_IE_DL_S22)
576 RES=execute(1)
578 endif
580 TD_COUNT=TD_COUNT+1
582 done
584 else
586 TD_ART_STR_PN_ARR[1]=0
588 TD_COUNT=2
590 endif
592 if (TD_MASTER_SKUP_VYR=="VD99999")
594 if (sel_value(count("T_MASTER_STR.C_ID_2"),("T_MASTER_STR.C_ID_1"=TD_POMI) & ("T_MASTER_STR.CUR_FLAG"=TRUE) &
("T_MASTER_STR.IN_CONTEXT_FAT"="PRD"))==0)
596 TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
598 TD_IE_DL_SCHMEID="MANUF_STRUCTURE"
600 TD_IE_DL_D1=date()
602 TD_IE_DL_D2=""
604 TD_IE_DL_S7="MAT-D"
606 TD_IE_DL_S16=""
608 TD_IE_DL_S12="Kč"
610 TD_IE_DL_N9=TD_ART_STR_PN_ARR[TD_COUNT-1]+10
612 TD_IE_DL_S1="ProdStructure"
614 TD_IE_DL_D3=date()
616 TD_IE_DL_S4=strprint("%i",TD_POMI5)
618 TD_IE_DL_S3="BO"

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

620 TD_IE_DL_S6="M"
622 TD_IE_DL_S5="*"
624 TD_IE_DL_N3=0
626 TD_IE_DL_N4=0
628 TD_IE_DL_S9="Not Consumed"
630 TD_IE_DL_N6=0
632 TD_IE_DL_S18="Promised"
634 TD_IE_DL_S10="NOT PHANTOM CONSUME"
636 TD_IE_DL_S11="ITEM NOT CHARGED"
638 TD_IE_DL_S15="BO"
640 TD_IE_DL_N6=0
642 TD_IE_DL_N12=0
644 TD_IE_DL_N1=TD_ART_STR_PN_ARR[TD_COUNT-1]+10
646 TD_IE_DL_N2=0
648 TD_IE_DL_S22="TRUE"
650 TD_IE_DL_S2=TD_POMS1
652 TD_IE_DL_STATUS="READY"
654 TD_IE_DL_D4=date()
656 TD_IE_DL_SCHMEID="MANUF_STRUCTURE"
658 insert_ent("TD-IE-DATA-LINE")
660 var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
662 var(TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_D1,TD_IE_DL_D2,TD_IE_DL_S6,TD_IE_DL_S5,TD_IE_DL_N1,TD_IE_DL_S7,TD_IE_DL_N2)
664 var(TD_IE_DL_N3,TD_IE_DL_N4,TD_IE_DL_S9,TD_IE_DL_N6,TD_IE_DL_S18,TD_IE_DL_S10,TD_IE_DL_S11,TD_IE_DL_S16,TD_IE_DL_S12)
666 var(TD_IE_DL_N9,TD_IE_DL_S15,TD_IE_DL_N6,TD_IE_DL_S1,TD_IE_DL_D3,TD_IE_DL_D4,TD_IE_DL_N12,TD_IE_DL_S22)
668 RES=execute(1)
670 endif
672 endif
674 C ***** POSTUP *****
676 if (TD_MASTER_PART_VERS=="0")
678 if (sel_value(count("T_MASTER_POST.C_ID_2"),"T_MASTER_POST.C_ID_1"=TD_POMI)==1)
680 TD_MASTER_POST_CID2=sel_value("T_MASTER_POST.C_ID_2","T_MASTER_POST.C_ID_1"=TD_POMI)
682 select(TD_POSTUPY_TD_ID,TD_POSTUPY_STREDISKO,TD_POSTUPY_TD_VER,TD_POSTUPY_DAVKA,TD_POSTUPY_VAL_FROM,TD_POSTUPY_VAL_UNTIL)

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

684 where("T_POSTUPY_DAT.C_ID"=TD_MASTER_POST_CID2)
686 RES=exec_select(1)
688 RES=scan(TD_POSTUPY_TD_VER,"%i",TD_POMI4)
690 TD_POMI4=TD_POMI4+1
692 TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
694 TD_IE_DL_SCHMEID="ROUTING_HEAD"
696 TD_IE_DL_S1=TD_POSTUPY_TD_ID
698 TD_IE_DL_S2="BO"
700 TD_IE_DL_S3=strprint("%i",TD_POMI4)
702 TD_IE_DL_S4="M"
704 TD_IE_DL_D1=TD_POSTUPY_VAL_FROM
706 TD_IE_DL_D3=date()
708 TD_IE_DL_SCHMEID="ROUTING_ALTERNATE"
710 TD_IE_DL_S1=TD_POSTUPY_TD_ID
712 TD_IE_DL_S2="BO"
714 TD_IE_DL_S3=strprint("%i",TD_POMI4)
716 TD_IE_DL_S4="*"
718 TD_IE_DL_S5="M"
720 TD_IE_DL_D2=TD_POSTUPY_VAL_FROM
722 TD_IE_DL_N2=0
724 TD_IE_DL_N3=0
726 TD_IE_DL_N4=0
728 TD_IE_DL_N5=0
730 select(TD_POST_OPER_PCID2[1],TD_POST_OPER_PN_ARR[1])
732 order_by("T_POST_OPER.POS_NO"+)
734 where("T_POST_OPER.C_ID_1"=TD_MASTER_POST_CID2)
736 TD_POMI2=exec_select(0)
738 if (TD_POMI2>0)
740     TD_COUNT=1
742     while (TD_COUNT<=TD_POMI2)
744 select(TD_OPERACE_TEXT_ID,TD_OPERACE_DRUH_OP,TD_OPERACE_PRAC,TD_OPERACE_CAS_KUS,TD_OPERACE_CAS_KUSMZ,TD_OPERACE
_CAS_PRI,TD_OPERACE_CAS_PRIMZ)
746     where("T_OPERACE_DAT.C_ID"=TD_POST_OPER_PCID2[TD_COUNT])

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

748 RES=exec_select(1)
750 if (TD_OPERACE_CAS_KUS!=0)
752     TD_OPER_KOEF_KUS=TD_OPERACE_CAS_KUSMZ/TD_OPERACE_CAS_KUS*100
754 else
756     TD_OPER_KOEF_KUS=0
758 endif
760 if (TD_OPERACE_CAS_PRI!=0)
762     TD_OPER_KOEF_PRI=TD_OPERACE_CAS_PRIMZ/TD_OPERACE_CAS_PRI*100
764 else
766     TD_OPER_KOEF_PRI=0
768 endif
770 select(TD_OPERACE_AKTIV,TD_OPERACE_KOOP,TD_OPERACE_PS,TD_OPERACE_VICESTR,TD_OPERACE_TAR_TR,TD_OPERACE_CAS_PRECH,
TD_OPERACE_LEV_IND,TD_OPERACE_POZNAMKA,TD_OPERACE_TEXT_ID,TD_OPER_KOOP_POL)
772 where("T_OPERACE_DAT.C_ID"=TD_POST_OPER_PCID2[TD_COUNT])
774 RES=exec_select(1)
776 if (TD_OPERACE_LEV_IND==230)
778     TD_POMS1="TRUE"
780 else
782     TD_POMS1="FALSE"
784 endif
786 TD_IE_DL_S1=TD_POSTUPY_TD_ID
788 TD_IE_DL_S2="BO"
790 TD_IE_DL_S3=strprint("%i",TD_POMI4)
792 TD_IE_DL_S5="M"
794 TD_IE_DL_D1=TD_POSTUPY_VAL_FROM
796 TD_IE_DL_S4="*"
798 TD_IE_DL_N2=TD_POST_OPER_PN_ARR[TD_COUNT]
800 TD_IE_DL_N1=TD_IE_DL_N2
802 select(TD_TEXTOPR_TEXT_FRA)
804 where("T_TEXTOPR_DAT.ID"=TD_OPERACE_TEXT_ID)
806 RES=exec_select(1)
808 TD_IE_DL_S6=TD_TEXTOPR_TEXT_FRA
810 if (TD_IE_DL_S6=="")

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

812     TD_IE_DL_S6=""
814     endif
816     TD_IE_DL_S8=TD_OPERACE_PRAC
818     TD_IE_DL_N6=rnd(TD_OPERACE_CAS_PRI/60,3)
820     TD_IE_DL_N5=rnd(TD_OPERACE_CAS_KUS/60,3)
822     TD_IE_DL_S11=""
824     TD_IE_DL_S11=TD_OPERACE_TAR_TR
826     TD_IE_DL_N9=rnd(TD_OPERACE_CAS_PRIMZ/60,3)
828     TD_IE_DL_N8=rnd(TD_OPERACE_CAS_KUSMZ/60,3)
830     TD_IE_DL_S10="1"
832     TD_IE_DL_N10=0
834     RES=scan(TD_OPERACE_PS,"%i",TD_IE_DL_N10)
836     if (TD_IE_DL_S11!="")
838         if (TD_IE_DL_N10<1)
840             TD_IE_DL_N10=1
842         endif
844     endif
846     TD_IE_DL_N7=TD_OPERACE_CAS_PRECH
848     TD_IE_DL_S7="N"
850     TD_IE_DL_N4=100
852     TD_IE_DL_N11=1
854     TD_IE_DL_S12=TD_OPER_KOOP_POL
856     TD_IE_DL_S14=TD_OPERACE_POZNAMKA
858     TD_IE_DL_N3=1
860     TD_IE_DL_S17=TD_POMS1
862     TD_IE_DL_SCHMEID="ROUTING_OPERATION"
864     TD_IE_DL_S16=""
866     if (TD_OPERACE_LEV_IND!=260)
868         TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
870         if ((TD_IE_DL_N6==0) & (TD_IE_DL_N5==0) & (TD_IE_DL_N9==0) & (TD_IE_DL_N8==0))
872             TD_IE_DL_N5=1/1000
874             insert_ent("TD-IE-DATA-LINE")
876             var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek	P.	Text
878		var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_S5,TD_IE_DL_D1)
880		var(TD_IE_DL_N1,TD_IE_DL_N2,TD_IE_DL_N4,TD_IE_DL_S6,TD_IE_DL_S16,TD_IE_DL_S8,TD_IE_DL_N6,TD_IE_DL_N5)
882		var(TD_IE_DL_N9,TD_IE_DL_N8,TD_IE_DL_S10,TD_IE_DL_N11,TD_IE_DL_N7,TD_IE_DL_S7,TD_IE_DL_S12,TD_IE_DL_S14,TD_IE_DL_S17)
884		RES=execute(1)
886		else
888		insert_ent("TD-IE-DATA-LINE")
890		var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
892		var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_S5,TD_IE_DL_D1)
894		var(TD_IE_DL_N1,TD_IE_DL_N2,TD_IE_DL_N4,TD_IE_DL_S6,TD_IE_DL_S16,TD_IE_DL_S8,TD_IE_DL_N6,TD_IE_DL_N5,TD_IE_DL_S11)
896		var(TD_IE_DL_N9,TD_IE_DL_N8,TD_IE_DL_S10,TD_IE_DL_N10,TD_IE_DL_N11,TD_IE_DL_N7,TD_IE_DL_S7,TD_IE_DL_S12,TD_IE_DL_S14,TD_IE_DL_S17)
898		RES=execute(1)
900		endif
902	C	***** UKONY *****
904		select(TD_OPER_UKON_PCID2[1])
906		order_by("T_OPER_UKON.POS_NO"+)
908		where("T_OPER_UKON.C_ID_1"=TD_POST_OPER_PCID2[TD_COUNT])
910		TD_POMI3=exec_select(0)
912		if (TD_POMI3>0)
914		TD_IE_DL_SCHMEID="ROUTING_OPER_WORK_GUIDE"
916		TD_IE_DL_S1=TD_POSTUPY_TD_ID
918		TD_IE_DL_S2="BO"
920		TD_IE_DL_S3="1"
922		TD_IE_DL_S4="**"
924		TD_IE_DL_S5="M"
926		TD_IE_DL_N1=TD_IE_DL_N2
928		TD_IE_DL_S6="**"
930		TD_COUNT2=1
932		while (TD_COUNT2<=TD_POMI3)
934		select(TD_UKON_TEXT)
936		where("T_UKONY_DAT.C_ID"=TD_OPER_UKON_PCID2[TD_COUNT2])
938		RES=exec_select(1)
940		TD_IE_DL_S7=TD_UKON_TEXT

Procedura TD_IE_AXA_IFS/axa_ifs_vyrabene

Export součásti, kusovníku, technologického postupu, jeho operací a úkonů operací

Použití: Při uvolnění součásti do výroby (převod ze stavu 222 do 230)

Řádek P. Text

```

942      TD_IE_DL_N2=TD_COUNT2
944      TD_IE_DL_N3=TD_COUNT2
946      TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
948      insert_ent("TD-IE-DATA-LINE")
950      var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
952      var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_S5,TD_IE_DL_S6,TD_IE_DL_N1,TD_IE_DL_N2,TD_IE_DL_N3,TD_IE_DL_S7)
954      RES=execute(1)
956      TD_COUNT2=TD_COUNT2+1
958      done
960      endif
962 C      ***** UKONY KONEC *****
964      endif
966      TD_COUNT=TD_COUNT+1
968      done
970      endif
972      endif
974      endif
976 C      ***** POSTUP - KONEC *****
978      TD_IE_EX_STATE="Ready"
980      update(TD_IE_EX_STATE)
982      where("IE_EXEC_BATCH.EXECUTE BATCHID"=TD_IE_EX_BATCH_ID)
984      exec_update()
986      endif
988      endif
990 C      poslední řádek

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upd

Export kusovníku

Použití: Při opravě kusovníku (převod ze stavu 235 do 230)

Řádek P. Text

```

5   TD_POMI=TD_MASTER_CID_ACT
15  select(TD_MASTER_DRUHPOL)
20  where("T_MASTER_DAT.C_ID"=TD_POMI)
25  RES=exec_select(1)
30  if (TD_MASTER_DRUHPOL=="V")
35  select(TD_MASTER_PART_ID,TD_MASTER_TECHN_ID,TD_MASTER_TEXT_ID,TD_MASTER_DAT_ROZMER,TD_MASTER_DAT_POPIS,TD_MASTER_D
   AT_TJ,TD_MASTER_DAT_NJ,TD_MASTER_DAT_UNIT,TD_MASTER_DAT_MJTJ,TD_MASTER_DAT_MJNJ,TD_MASTER_DAT_USKLAD,TD_MASTER_S
   PEC_KOD,TD_MASTER_KOD_ROZP)
40  where("T_MASTER_DAT.C_ID"=TD_POMI)
45  RES=exec_select(1)
50  if (strpart(TD_MASTER_TECHN_ID,1,1)!="K" & strpart(TD_MASTER_TECHN_ID,1,2)!="20")
55  select(TD_TEXTSOU_NAME_FRA,TD_TEXTSOU_TEXT1_FRA)
60  where("T_TEXTSOU_DAT.ID"=TD_MASTER_TEXT_ID)
65  RES=exec_select(1)
70  select(TD_MASTER_SKUP_VYR,TD_MASTER_DAT_SKLAD,TD_MASTER_DAT_UCET,TD_MASTER_C_HMOTNOST,TD_MASTER_PART_VERS,TD_MA
   STER_VAL_FROM,TD_MASTER_DAT_PREDPM)
75  where("T_MASTER_DAT.C_ID"=TD_POMI)
80  RES=exec_select(1)
85  TD_POMI5=0
90  RES=scan(TD_MASTER_PART_VERS,"%i",TD_POMI5)
91  TD_POMI5=TD_POMI5+1
92  TD_IE_IFS_GEN="#TD-IE-IFS-GEN"
93  RES=#nsv_get_num(TD_IE_IFS_GEN,TD_IE_EX_BATCH_ID)
94  if (RES!=0)
95  put("Chyba při aktualizaci generátoru čísel! Informujte správce...")
96  exit()
97  endif
175 TD_POMS4=strprint("%i",TD_IE_EX_BATCH_ID)
180 TD_POMS4=strpart(TD_POMS4,strlen(TD_POMS4),1)
190 if (TD_POMS4=="0" | TD_POMS4=="2" | TD_POMS4=="4" | TD_POMS4=="6" | TD_POMS4=="8")
195   TD_IE_EX_BATCH_ID=TD_IE_EX_BATCH_ID+1
200 else
205   TD_IE_EX_BATCH_ID=TD_IE_EX_BATCH_ID+2
210 endif

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upd**Export kusovníku**

Použití: Při opravě kusovníku (převod ze stavu 235 do 230)

Řádek P. Text

```

215 TD_IE_EX_ISIMPORT="TRUE"
220 TD_IE_EX_STATE="Open"
245 TD_IE_DL_LINEID=0
250 TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
255 TD_IE_EX_NEXTLINEID=TD_IE_DL_LINEID
260 TD_IE_EX_ERRORFATAL="TRUE"
265 TD_IE_EX_FORCECOMMIT="FALSE"
270 TD_IE_EX_SYSTEMID="AXALANT"
275 TD_IE_EX_DESCRIPTION=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
280 TD_IE_EX_CREATEDATE=date()
285 TD_IE_EX_EXECUTEDATE=date()
290 TD_IE_EX_IMPORTPARAM=strprint("UPDATEALL~TRUE%s%sUPDATE~TRUE",chr(13),chr(10))
295 insert_ent("TD-IE-EXECUTE-BATCH")
300 var(TD_IE_EX_BATCH_ID,TD_IE_EX_ISIMPORT,TD_IE_EX_STATE,TD_IE_EX_NEXTLINEID,TD_IE_EX_ERRORFATAL)
305 var(TD_IE_EX_FORCECOMMIT,TD_IE_EX_SYSTEMID,TD_IE_EX_CREATEDATE,TD_IE_EX_EXECUTEDATE,TD_IE_EX_DESCRIPTION,TD_IE_EX_IMPORTPARAM)
310 RES=execute(1)
315 TD_IE_DL_BATCH_ID=TD_IE_EX_BATCH_ID
320 C ***** BOM *****
325 TD_POMS2=TD_MASTER_TECHN_ID
330 TD_POMS3=TD_MASTER_PART_ID
335 if (TD_MASTER_TECHN_ID=="")
340 TD_POMS1=TD_MASTER_PART_ID
345 else
350 TD_POMS1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
355 endif
360 select(TD_ART_STR_CID2[1],TD_ART_STR_PN_ARR[1],TD_ART_STR_PQUANTITY[1],TD_ART_STR_CF_ARR[1],TD_ART_STR_DESC_ARR[1])
365 order_by("T_MASTER_STR.POS_NO"+)
370 where("T_MASTER_STR.C_ID_1"=TD_POMI & "T_MASTER_STR.IN_CONTEXT_FAT"="PRD" & "T_MASTER_STR.C_ID">0)
375 TD_POMI1=exec_select(0)
380 if (TD_POMI1>0)
385 TD_COUNT=1
390 while (TD_COUNT<=TD_POMI1)

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upd

Export kusovníku

Použití: Při opravě kusovníku (převod ze stavu 235 do 230)

Řádek P. Text

```
395 TD_MASTER_PART_ID=""
400 TD_MASTER_TECHN_ID=""
405 select(TD_MASTER_TECHN_ID,TD_MASTER_PART_ID,TD_MASTER_CUR_FLAG,TD_MASTER_DAT_UNIT)
410 where("T_MASTER_DAT.C_ID"=TD_ART_STR_CID2[TD_COUNT])
415 RES=exec_select(1)
420 if (TD_MASTER_CUR_FLAG==TRUE)
425 TD_IE_DL_D1=date()
430 TD_IE_DL_D2=""
435 TD_IE_DL_S7=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
440 TD_IE_DL_S16=TD_ART_STR_DESC_ARR[TD_COUNT]
445 TD_IE_DL_S12=TD_MASTER_DAT_UNIT
450 TD_IE_DL_N9=TD_ART_STR_PN_ARR[TD_COUNT]
455 TD_IE_DL_S1="ProdStructure"
460 TD_IE_DL_D3=date()
465 TD_IE_DL_S4=strprint("%i",TD_POMI5)
470 TD_IE_DL_S3="BO"
475 TD_IE_DL_S6="M"
480 TD_IE_DL_S5="*"
485 TD_IE_DL_N3=0
490 TD_IE_DL_N4=0
495 TD_IE_DL_S9="Consumed"
500 TD_IE_DL_N6=0
505 TD_IE_DL_S18="Promised"
510 TD_IE_DL_S10="NOT PHANTOM CONSUME"
515 TD_IE_DL_S11="ITEM NOT CHARGED"
520 TD_IE_DL_S15="BO"
525 TD_IE_DL_N6=0
530 TD_IE_DL_N12=0
535 TD_IE_DL_N1=TD_ART_STR_PN_ARR[TD_COUNT]
540 TD_IE_DL_N2=TD_ART_STR_PQUANTITY[TD_COUNT]
545 if (TD_ART_STR_CF_ARR[TD_COUNT]==TRUE)
550 TD_IE_DL_S22="TRUE"
555 else
```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upd

Export kusovníku

Použití: Při opravě kusovníku (převod ze stavu 235 do 230)

Řádek P. Text

```

560     TD_IE_DL_S22="FALSE"
565     TD_IE_DL_N2=0
570     TD_IE_DL_S9="Not Consumed"
575     endif
580     TD_IE_DL_S2=TD_POMS1
585     TD_IE_DL_STATUS="READY"
590     TD_IE_DL_S4=strprint("%i",TD_POMI5)
595     TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
600     TD_IE_DL_D4=date()
605     TD_IE_DL_SCHMEID="MANUF_STRUCTURE"
610     insert_ent("TD-IE-DATA-LINE")
615     var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
620     var(TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_D1,TD_IE_DL_D2,TD_IE_DL_S6,TD_IE_DL_S5,TD_IE_DL_N1,TD_IE_DL_S7,TD_IE_DL_N2)
625     var(TD_IE_DL_N3,TD_IE_DL_N4,TD_IE_DL_S9,TD_IE_DL_N6,TD_IE_DL_S18,TD_IE_DL_S10,TD_IE_DL_S11,TD_IE_DL_S16,TD_IE_DL_S12)
630     var(TD_IE_DL_N9,TD_IE_DL_S15,TD_IE_DL_N6,TD_IE_DL_S1,TD_IE_DL_D3,TD_IE_DL_D4,TD_IE_DL_N12,TD_IE_DL_S22)
635     RES=execute(1)
640     endif
645     TD_COUNT=TD_COUNT+1
650     done
655     else
660     TD_ART_STR_PN_ARR[1]=0
665     TD_COUNT=2
670     endif
675     if (TD_MASTER_SKUP_VYR=="VD99999")
680     if (sel_value(count("T_MASTER_STR.C_ID_2"),("T_MASTER_STR.C_ID_1=TD_POMI) & ("T_MASTER_STR.CUR_FLAG"=TRUE) &
("T_MASTER_STR.IN_CONTEXT_FAT"="PRD"))==0)
685     TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
690     TD_IE_DL_SCHMEID="MANUF_STRUCTURE"
695     TD_IE_DL_D1=date()
700     TD_IE_DL_D2=""
705     TD_IE_DL_S7="MAT-D"
710     TD_IE_DL_S16=""
715     TD_IE_DL_S12="Kč"

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upd

Export kusovníku

Použití: Při opravě kusovníku (převod ze stavu 235 do 230)

Řádek P. Text

```

720 TD_IE_DL_N9=TD_ART_STR_PN_ARR[TD_COUNT-1]+10
725 TD_IE_DL_S1="ProdStructure"
730 TD_IE_DL_D3=date()
735 TD_IE_DL_S4=strprint("%i",TD_POMI5)
740 TD_IE_DL_S3="BO"
745 TD_IE_DL_S6="M"
750 TD_IE_DL_S5="*"
755 TD_IE_DL_N3=0
760 TD_IE_DL_N4=0
765 TD_IE_DL_S9="Not Consumed"
770 TD_IE_DL_N6=0
775 TD_IE_DL_S18="Promised"
780 TD_IE_DL_S10="NOT PHANTOM CONSUME"
785 TD_IE_DL_S11="ITEM NOT CHARGED"
790 TD_IE_DL_S15="BO"
795 TD_IE_DL_N6=0
800 TD_IE_DL_N12=0
805 TD_IE_DL_N1=TD_ART_STR_PN_ARR[TD_COUNT-1]+10
810 TD_IE_DL_N2=0
815 TD_IE_DL_S22="TRUE"
820 TD_IE_DL_S2=TD_POMS1
825 TD_IE_DL_STATUS="READY"
830 TD_IE_DL_D4=date()
835 TD_IE_DL_SCHMEID="MANUF_STRUCTURE"
840 insert_ent("TD-IE-DATA-LINE")
845 var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
850 var(TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_D1,TD_IE_DL_D2,TD_IE_DL_S6,TD_IE_DL_S5,TD_IE_DL_N1,TD_IE_DL_S7,TD_IE_DL_N2)
855 var(TD_IE_DL_N3,TD_IE_DL_N4,TD_IE_DL_S9,TD_IE_DL_N6,TD_IE_DL_S18,TD_IE_DL_S10,TD_IE_DL_S11,TD_IE_DL_S16,TD_IE_DL_S12)
860 var(TD_IE_DL_N9,TD_IE_DL_S15,TD_IE_DL_N6,TD_IE_DL_S1,TD_IE_DL_D3,TD_IE_DL_D4,TD_IE_DL_N12,TD_IE_DL_S22)
865 RES=execute(1)
870 endif
875 endif
880 TD_IE_EX_STATE="Ready"

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upd

Export kusovníku

Použití: Při opravě kusovníku (převod ze stavu 235 do 230)

Řádek P. Text

```
885      update(TD_IE_EX_STATE)
890      where("IE_EXEC_BATCH.EXECUTEBATCHID"=TD_IE_EX_BATCH_ID)
895      exec_update()
900  endif
905  endif
910 C  poslední řádek
```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upds**Export součásti**

Použití: Při opravě součásti (převod ze stavu 235 do 230)

Řádek P. Text

```

5   TD_POMI=TD_MASTER_CID_ACT
15  select(TD_MASTER_DRUHPOL)
20  where("T_MASTER_DAT.C_ID"=TD_POMI)
25  RES=exec_select(1)
30  if (TD_MASTER_DRUHPOL=="V")
35  select(TD_MASTER_PART_ID,TD_MASTER_TECHN_ID,TD_MASTER_TEXT_ID,TD_MASTER_DAT_ROZMER,TD_MASTER_DAT_POPI,TD_MASTER_D
   AT_TJ,TD_MASTER_DAT_NJ,TD_MASTER_DAT_UNIT,TD_MASTER_DAT_MJTJ,TD_MASTER_DAT_MJNJ,TD_MASTER_DAT_USKLAD,TD_MASTER_S
   PEC_KOD,TD_MASTER_KOD_ROZP)
40  where("T_MASTER_DAT.C_ID"=TD_POMI)
45  RES=exec_select(1)
50  if (strpart(TD_MASTER_TECHN_ID,1,1)!="K" & strpart(TD_MASTER_TECHN_ID,1,2)!="20")
55  select(TD_TEXTSOU_NAME_FRA,TD_TEXTSOU_TEXT1_FRA)
60  where("T_TEXTSOU_DAT.ID"=TD_MASTER_TEXT_ID)
65  RES=exec_select(1)
70  select(TD_MASTER_SKUP_VYR,TD_MASTER_DAT_SKLAD,TD_MASTER_DAT_UCET,TD_MASTER_C_HMOTNOST,TD_MASTER_PART_VERS,TD_MA
   STER_VAL_FROM,TD_MASTER_DAT_PREDPM,TD_MASTER_TVAR_C)
75  where("T_MASTER_DAT.C_ID"=TD_POMI)
80  RES=exec_select(1)
85  TD_POMI5=0
90  RES=scan(TD_MASTER_PART_VERS,"%i",TD_POMI5)
91  TD_POMI5=TD_POMI5+1
92  TD_IE_IFS_GEN="#TD-IE-IFS-GEN"
93  RES=#nsv_get_num(TD_IE_IFS_GEN,TD_IE_EX_BATCH_ID)
94  if (RES!=0)
95  put("Chyba při aktualizaci generátoru čísel! Informujte správce...")
96  exit()
97  endif
175 TD_POMS4=strprint("%i",TD_IE_EX_BATCH_ID)
180 TD_POMS4=strpart(TD_POMS4,strlen(TD_POMS4),1)
190 if (TD_POMS4=="0" | TD_POMS4=="2" | TD_POMS4=="4" | TD_POMS4=="6" | TD_POMS4=="8")
195   TD_IE_EX_BATCH_ID=TD_IE_EX_BATCH_ID+1
200 else
205   TD_IE_EX_BATCH_ID=TD_IE_EX_BATCH_ID+2
210 endif

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upds**Export součásti**

Použití: Při opravě součásti (převod ze stavu 235 do 230)

Řádek P. Text

```

215 TD_IE_EX_ISIMPORT="TRUE"
220 TD_IE_EX_STATE="Open"
245 TD_IE_DL_LINEID=0
250 TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
255 TD_IE_EX_NEXTLINEID=TD_IE_DL_LINEID
260 TD_IE_EX_ERRORFATAL="TRUE"
265 TD_IE_EX_FORCECOMMIT="FALSE"
270 TD_IE_EX_SYSTEMID="AXALANT"
275 TD_IE_EX_DESCRIPTION=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
280 TD_IE_EX_CREATEDATE=date()
285 TD_IE_EX_EXECUTEDATE=date()
290 TD_IE_EX_IMPORTPARAM=strprint("UPDATE~TRUE%s%sMERGE~TRUE",chr(13),chr(10))
295 insert_ent("TD-IE-EXECUTE-BATCH")
300 var(TD_IE_EX_BATCH_ID,TD_IE_EX_ISIMPORT,TD_IE_EX_STATE,TD_IE_EX_NEXTLINEID,TD_IE_EX_ERRORFATAL)
305 var(TD_IE_EX_FORCECOMMIT,TD_IE_EX_SYSTEMID,TD_IE_EX_CREATEDATE,TD_IE_EX_EXECUTEDATE,TD_IE_EX_DESCRIPTION,TD_IE_EX_IMPORTPARAM)
310 RES=execute(1)
315 TD_IE_DL_BATCH_ID=TD_IE_EX_BATCH_ID
320 TD_IE_DL_STATUS="READY"
325 TD_IE_DL_SCHMEID="PART_CATALOG"
330 TD_IE_DL_S1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
335 TD_IE_DL_S2=TD_TEXTSOU_TEXT1_FRA
340 select(TD_ART_STR_CID2[1])
345 where("T_MASTER_STR.C_ID_1"=TD_POMI)
350 TD_POMI1=exec_select(0)
355 TD_IE_DL_S10="NOT CONFIGURED"
360 if (TD_POMI1>0)
365 TD_COUNT=1
370 while (TD_COUNT<=TD_POMI1)
375 select(TD_MASTER_DAT_TYP)
380 where("T_MASTER_DAT.C_ID"=TD_ART_STR_CID2[TD_COUNT])
385 RES=exec_select(1)
390 if (TD_MASTER_DAT_TYP=="VARATT")

```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upds

Export součásti

Použití: Při opravě součásti (převod ze stavu 235 do 230)

Řádek P. Text

```
395     TD_IE_DL_S10="CONFIGURED"
400     endif
405     TD_COUNT=TD_COUNT+1
410     done
415     TD_COUNT=0
420     endif
425     TD_IE_DL_N1=TD_MASTER_TEXT_ID
430     TD_IE_DL_S5=TD_MASTER_DAT_UNIT
440     insert_ent("TD-IE-DATA-LINE")
445     var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
450     var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S10,TD_IE_DL_S5,TD_IE_DL_N1)
455     RES=execute(1)
460     TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
465     TD_IE_DL_SCHMEID="INVENTORY_PART"
470     TD_IE_DL_S1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
475     TD_IE_DL_S2="BO"
480     TD_IE_DL_S14=TD_TEXTSOU_TEXT1_FRA
485     TD_IE_DL_S13=TD_MASTER_DAT_UNIT
486     TD_IE_DL_S17=TD_MASTER_DAT_POPIS
490     TD_IE_DL_S26=TD_MASTER_DAT_ROZMER
495     TD_IE_DL_S7=TD_MASTER_SPEC_KOD
500     if (TD_IE_DL_S7=="ZR")
505         TD_IE_DL_S7="Z"
510     else
515         TD_IE_DL_S7="A"
520     endif
525     TD_IE_DL_S8=TD_MASTER_SKUP_VYR
526     if (TD_MASTER_TECHN_ID=="30")
530         TD_IE_DL_N7=TD_MASTER_DAT_PREDPM
532     else
533         TD_IE_DL_N7=0.01
534     endif
535     if (TD_IE_DL_N7==0)
```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upds**Export součásti**

Použití: Při opravě součásti (převod ze stavu 235 do 230)

Řádek P. Text

```
540     TD_IE_DL_N7=0.01
545     endif
550     insert_ent("TD-IE-DATA-LINE")
555     var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
560     var(TD_IE_DL_N7)
565     var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S14,TD_IE_DL_S13,TD_IE_DL_S7,TD_IE_DL_S8,TD_IE_DL_S17,TD_IE_DL_S26)
570     RES=execute(1)
575     TD_IE_DL_MASTERID=TD_IE_DL_LINEID
580     TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
585     TD_IE_DL_STATUS="DETAIL"
590     if (TD_IE_DL_STATUS=="DETAIL")
595         TD_IE_DL_PARTNO=1
600     else
605         TD_IE_DL_PARTNO=0
610     endif
615     TD_IE_DL_N1=TD_MASTER_C_HMOTNOST
620     TD_IE_DL_N3=TD_MASTER_TEXT_ID
625     insert_ent("TD-IE-DATA-LINE")
630     var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID,TD_IE_DL_MASTERID,TD_IE_DL_PARTNO)
635     var(TD_IE_DL_N1,TD_IE_DL_N3)
645     RES=execute(1)
650     TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
655     TD_IE_DL_STATUS="READY"
660     TD_IE_DL_SCHMEID="INV_PART_DISCRETE_CHAR"
665     TD_IE_DL_S1=strprint("%s.%s",TD_MASTER_TECHN_ID,TD_MASTER_PART_ID)
670     TD_IE_DL_S2="BO"
675     TD_IE_DL_S3="KR"
680     TD_IE_DL_S5=strprint("%i",TD_MASTER_KOD_ROZP)
685     insert_ent("TD-IE-DATA-LINE")
690     var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
695     var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S5)
700     RES=execute(1)
705     TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
```

Procedura TD_IE_AXA_IFS/axa_ifs_vyr_upds

Export součásti

Použití: Při opravě součásti (převod ze stavu 235 do 230)

Řádek P. Text

```
710    TD_IE_DL_S3="TC"
715    TD_IE_DL_S5=TD_MASTER_TVAR_C
720    insert_ent("TD-IE-DATA-LINE")
725    var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
730    var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S5)
735    RES=execute(1)
745    TD_IE_EX_STATE="Ready"
750    update(TD_IE_EX_STATE)
755    where("IE_EXEC_BATCH.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID)
760    exec_update()
765    endif
770    endif
775 C  poslední řádek
```

Procedura TD_IE_AXA_IFS/axa_ifs_postup

Export technologického postupu, jeho operací a úkonů operací

Použití: Při opravě technologického postupu (převod ze stavu 235 do 230)

Řádek P. Text

```
5   TD_MASTER_POST_CID2=TD_POSTUP_CID_ACT
10  select(TD_POSTUPY_TD_ID)
15  where("T_POSTUPY_DAT.C_ID"=TD_MASTER_POST_CID2)
16  RES=exec_select(1)
17  TD_IE_IFS_GEN="#TD-IE-IFS-GEN"
18  RES=#nsv_get_num(TD_IE_IFS_GEN,TD_IE_EX_BATCH_ID)
19  if (RES!=0)
20      put("Chyba při aktualizaci generátoru čísel! Informujte správce...")
21      exit()
22  endif
100 TD_POMS4=strprint("%i",TD_IE_EX_BATCH_ID)
105 TD_POMS4=strpart(TD_POMS4,strlen(TD_POMS4),1)
110 if (TD_POMS4=="0" | TD_POMS4=="2" | TD_POMS4=="4" | TD_POMS4=="6" | TD_POMS4=="8")
115     TD_IE_EX_BATCH_ID=TD_IE_EX_BATCH_ID+1
125 else
130     TD_IE_EX_BATCH_ID=TD_IE_EX_BATCH_ID+2
135 endif
140 TD_IE_DL_BATCH_ID=TD_IE_EX_BATCH_ID
145 TD_IE_DL_STATUS="READY"
150 TD_IE_EX_ISIMPORT="TRUE"
155 TD_IE_EX_STATE="Open"
160 TD_IE_DL_LINEID=0
165 TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
170 TD_IE_EX_NEXTLINEID=TD_IE_DL_LINEID
175 TD_IE_EX_ERRORFATAL="TRUE"
180 TD_IE_EX_FORCECOMMIT="FALSE"
185 TD_IE_EX_SYSTEMID="AXALANT"
190 TD_IE_EX_CREATEDATE=date()
195 TD_IE_EX_EXECUTEDATE=date()
200 TD_IE_EX_IMPORTPARAM="UPDATEALL~TRUE"
```

Procedura TD_IE_AXA_IFS/axa_ifs_postup

Export technologického postupu, jeho operací a úkonů operací

Použití: Při opravě technologického postupu (převod ze stavu 235 do 230)

Řádek P. Text

```

205 TD_IE_EX_DESCRIPTION=TD_POSTUPY_TD_ID
210 insert_ent("TD-IE-EXECUTE-BATCH")
215 var(TD_IE_EX_BATCH_ID,TD_IE_EX_ISIMPORT,TD_IE_EX_STATE,TD_IE_EX_NEXTLINEID,TD_IE_EX_ERRORFATAL)
    var(TD_IE_EX_FORCECOMMIT,TD_IE_EX_SYSTEMID,TD_IE_EX_CREATEDATE,TD_IE_EX_EXECUTEDATE,TD_IE_EX_IMPORTPARAM,TD_IE_EX_DE
220 Scription)
225 RES=execute(1)
230 C ***** POSTUP *****
    select(TD_POSTUPY_TD_ID,TD_POSTUPY_STREDISKO,TD_POSTUPY_TD_VER,TD_POSTUPY_DAVKA,TD_POSTUPY_VAL_FROM,TD_POSTUPY_VAL_UNTIL)
235 L_UNTIL)
240 where("T_POSTUPY_DAT.C_ID=TD_MASTER_POST_CID2)
245 RES=exec_select(1)
250 RES=scan(TD_POSTUPY_TD_VER,"%i",TD_POMI4)
255 TD_POMI4=TD_POMI4+1
260 TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
265 TD_IE_DL_SCHMEID="ROUTING_HEAD"
270 TD_IE_DL_S1=TD_POSTUPY_TD_ID
275 TD_IE_DL_S2="BO"
280 TD_IE_DL_S3=strprint("%i",TD_POMI4)
285 TD_IE_DL_S4="M"
290 TD_IE_DL_D1=TD_POSTUPY_VAL_FROM
295 TD_IE_DL_D3=date()
325 TD_IE_DL_SCHMEID="ROUTING_ALTERNATE"
330 TD_IE_DL_S1=TD_POSTUPY_TD_ID
335 TD_IE_DL_S2="BO"
340 TD_IE_DL_S3=strprint("%i",TD_POMI4)
345 TD_IE_DL_S4="*"
350 TD_IE_DL_S5="M"
355 TD_IE_DL_D2=TD_POSTUPY_VAL_FROM
360 TD_IE_DL_N2=0
365 TD_IE_DL_N3=0
370 TD_IE_DL_N4=0

```

Procedura TD_IE_AXA_IFS/axa_ifs_postup

Export technologického postupu, jeho operací a úkonů operací

Použití: Při opravě technologického postupu (převod ze stavu 235 do 230)

Řádek P. Text

```
375 TD_IE_DL_N5=0
400 select(TD_POST_OPER_PCID2[1],TD_POST_OPER_PN_ARR[1])
405 where("T_POST_OPER.C_ID_1"=TD_MASTER_POST_CID2)
410 TD_POMI2=exec_select(0)
415 if (TD_POMI2>0)
420     TD_COUNT=1
425     while (TD_COUNT<=TD_POMI2)
430         select(TD_OPERACE_TEXT_ID,TD_OPERACE_DRUH_OP,TD_OPERACE_PRAC,TD_OPERACE_CAS_KUS,TD_OPERACE_CAS_KUSMZ,TD_OPERACE
_CAS_PRI,TD_OPERACE_CAS_PRIMZ)
435         where("T_OPERACE_DAT.C_ID"=TD_POST_OPER_PCID2[TD_COUNT])
440         RES=exec_select(1)
445         if (TD_OPERACE_CAS_KUS!=0)
450             TD_OPER_KOEF_KUS=TD_OPERACE_CAS_KUSMZ/TD_OPERACE_CAS_KUS*100
455         else
460             TD_OPER_KOEF_KUS=0
465         endif
470         if (TD_OPERACE_CAS_PRI!=0)
475             TD_OPER_KOEF_PRIP=TD_OPERACE_CAS_PRIMZ/TD_OPERACE_CAS_PRI*100
480         else
485             TD_OPER_KOEF_PRIP=0
490         endif
495         select(TD_OPERACE_AKTIV,TD_OPERACE_KOOP,TD_OPERACE_PS,TD_OPERACE_VICESTR,TD_OPERACE_TAR_TR,TD_OPERACE_CAS_PRECH,T
D_OPERACE_LEV_IND,TD_OPERACE_POZNAMKA,TD_OPERACE_TEXT_ID,TD_OPER_KOOP_POL)
500         where("T_OPERACE_DAT.C_ID"=TD_POST_OPER_PCID2[TD_COUNT])
505         RES=exec_select(1)
510         if (TD_OPERACE_LEV_IND==230)
515             TD_POMS1="TRUE"
520         else
525             TD_POMS1="FALSE"
530         endif
535         TD_IE_DL_S1=TD_POSTUPY_TD_ID
```

Procedura TD_IE_AXA_IFS/axa_ifs_postup

Export technologického postupu, jeho operací a úkonů operací

Použití: Při opravě technologického postupu (převod ze stavu 235 do 230)

Řádek P. Text

```
540 TD_IE_DL_S2="BO"
550 TD_IE_DL_S3=strprint("%i",TD_POMI4)
555 TD_IE_DL_S5="M"
560 TD_IE_DL_D1=TD_POSTUPY_VAL_FROM
565 TD_IE_DL_S4="*"
570 TD_IE_DL_N2=TD_POST_OPER_PN_ARR[TD_COUNT]
575 TD_IE_DL_N1=TD_IE_DL_N2
580 select(TD_TEXTOPR_TEXT_FRA)
585 where("T_TEXTOPR_DAT.ID"=TD_OPERACE_TEXT_ID)
590 RES=exec_select(1)
595 TD_IE_DL_S6=TD_TEXTOPR_TEXT_FRA
600 if (TD_IE_DL_S6=="")
605 TD_IE_DL_S6="*"
610 endif
615 TD_IE_DL_S8=TD_OPERACE_PRAC
620 TD_IE_DL_N6=rnd(TD_OPERACE_CAS_PRI/60,3)
625 TD_IE_DL_N5=rnd(TD_OPERACE_CAS_KUS/60,3)
630 TD_IE_DL_S11=TD_OPERACE_TAR_TR
635 TD_IE_DL_N9=rnd(TD_OPERACE_CAS_PRIMZ/60,3)
640 TD_IE_DL_N8=rnd(TD_OPERACE_CAS_KUSMZ/60,3)
645 TD_IE_DL_S10="1"
650 RES=scan(TD_OPERACE_PS,"%i",TD_IE_DL_N10)
655 TD_IE_DL_N7=TD_OPERACE_CAS_PRECH
660 TD_IE_DL_S7="N"
665 TD_IE_DL_N4=100
670 TD_IE_DL_N11=1
675 TD_IE_DL_S12=TD_OPER_KOOP_POL
680 TD_IE_DL_S14=TD_OPERACE_POZNAMKA
685 TD_IE_DL_N3=1
690 TD_IE_DL_S17=TD_POMS1
```

Procedura TD_IE_AXA_IFS/axa_ifs_postup

Export technologického postupu, jeho operací a úkonů operací

Použití: Při opravě technologického postupu (převod ze stavu 235 do 230)

Řádek P. Text

```

695   TD_IE_DL_SCHMEID="ROUTING_OPERATION"
697   TD_IE_DL_S18="*"
700   TD_IE_DL_S16=""
705   if (TD_OPERACE_LEV_IND!=260)
710       TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
715       if ((TD_IE_DL_N6==0) & (TD_IE_DL_N5==0) & (TD_IE_DL_N9==0) & (TD_IE_DL_N8==0))
720           TD_IE_DL_N5=1/1000
725           insert_ent("TD-IE-DATA-LINE")
730           var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
735           var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_S5,TD_IE_DL_D1)
740           var(TD_IE_DL_N1,TD_IE_DL_N2,TD_IE_DL_N4,TD_IE_DL_S6,TD_IE_DL_S16,TD_IE_DL_S8,TD_IE_DL_N6,TD_IE_DL_N5)
var(TD_IE_DL_N9,TD_IE_DL_N8,TD_IE_DL_S10,TD_IE_DL_N11,TD_IE_DL_N7,TD_IE_DL_S7,TD_IE_DL_S12,TD_IE_DL_S14,TD_IE_DL_S17,TD_IE_DL_
745   S18)
750       RES=execute(1)
755   else
760       insert_ent("TD-IE-DATA-LINE")
765       var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
770       var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_S5,TD_IE_DL_D1)
775       var(TD_IE_DL_N1,TD_IE_DL_N2,TD_IE_DL_N4,TD_IE_DL_S6,TD_IE_DL_S16,TD_IE_DL_S8,TD_IE_DL_N6,TD_IE_DL_N5,TD_IE_DL_S11)
var(TD_IE_DL_N9,TD_IE_DL_N8,TD_IE_DL_S10,TD_IE_DL_N10,TD_IE_DL_N11,TD_IE_DL_N7,TD_IE_DL_S7,TD_IE_DL_S12,TD_IE_DL_S14,TD_IE_DL_
780   S17,TD_IE_DL_S18)
785       RES=execute(1)
790   endif
795 C   ***** UKONY *****
800       select(TD_OPER_UKON_PCID2[1])
805       order_by("T_OPER_UKON.POS_NO"+)
810       where("T_OPER_UKON.C_ID_1"=TD_POST_OPER_PCID2[TD_COUNT])
815       TD_POMI3=exec_select(0)
820       if (TD_POMI3>0)
825           TD_IE_DL_SCHMEID="ROUTING_OPER_WORK_GUIDE"
830           TD_IE_DL_S1=TD_POSTUPY_TD_ID

```

Procedura TD_IE_AXA_IFS/axa_ifs_postup

Export technologického postupu, jeho operací a úkonů operací

Použití: Při opravě technologického postupu (převod ze stavu 235 do 230)

Řádek P. Text

```

835     TD_IE_DL_S2="BO"
840     TD_IE_DL_S3="1"
845     TD_IE_DL_S4="*"
850     TD_IE_DL_S5="M"
855     TD_IE_DL_N1=TD_IE_DL_N2
860     TD_IE_DL_S6="*"
865     TD_COUNT2=1
870     while (TD_COUNT2<=TD_POMI3)
875         select(TD_UKON_TEXT)
880         where("T_UKONY_DAT.C_ID"=TD_OPER_UKON_PCID2[TD_COUNT2])
885         RES=exec_select(1)
890         TD_IE_DL_S7=TD_UKON_TEXT
895         TD_IE_DL_N2=TD_COUNT2
900         TD_IE_DL_N3=TD_COUNT2
905         TD_IE_DL_LINEID=TD_IE_DL_LINEID+1
910         insert_ent("TD-IE-DATA-LINE")
915         var(TD_IE_DL_BATCH_ID,TD_IE_DL_LINEID,TD_IE_DL_STATUS,TD_IE_DL_SCHMEID)
920         var(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_S5,TD_IE_DL_S6,TD_IE_DL_N1,TD_IE_DL_N2,TD_IE_DL_N3,TD_IE_DL_S7)
925         RES=execute(1)
930         TD_COUNT2=TD_COUNT2+1
935     done
940     endif
945 C ***** UKONY KONEC *****
950     endif
955     TD_COUNT=TD_COUNT+1
960     done
965     endif
970 C ***** POSTUP - KONEC *****
975     TD_IE_EX_STATE="Ready"
980     update(TD_IE_EX_STATE)

```

Procedura TD_IE_AXA_IFS/axa_ifs_postup

Export technologického postupu, jeho operací a úkonů operací

Použití: Při opravě technologického postupu (převod ze stavu 235 do 230)

Řádek P. Text

985 where("IE_EXEC_BATCH.EXECUTEBATCHID"=TD_IE_EX_BATCH_ID)

990 exec_update()

995 C poslední řádek

2 Procedury přenosu směrem IFS – Axalant

Procedura TD_IE_AXA_IFS/ifs_axa_import	Import dat z IFS
--	------------------

Použití: Při folbě funkce „Import dat z IFS“ v menu Axalantu. Jedná se o hlavní importní funkci, všechny ostatní jsou z ní volány.

Řádek	P. Text
5	if (group()=="TECHNOLOGIE" user()=="TDCUSTO")
10	put("Zahájen přenos dat z IFS do Axalantu...")
15	TD_ERROR=0
20	while (sel_value(count("IE_EXEC_BATCH.EXECUTE_BATCHID"),("IE_EXEC_BATCH.ISIMPORT"=="FALSE") & ("IE_EXEC_BATCH.STATE"="Ready"))>0)
30	select(TD_IE_EX_BATCH_ID)
40	order_by("IE_EXEC_BATCH.EXECUTE_BATCHID"-)
50	where(("IE_EXEC_BATCH.ISIMPORT"=="FALSE") & ("IE_EXEC_BATCH.STATE"="Ready"))
70	RES=exec_select(0)
90	TD_IE_DL_LINEID=0
110	while ((sel_value(count("IE_DATA_LINE.EXECUTE_BATCHID"),("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID) & ("IE_DATA_LINE.IEDATALINESTATUS"="READY"))>0))
150	select(TD_IE_DL_LINEID,TD_IE_DL_SCHMEID)
160	order_by("IE_DATA_LINE.LINEID"-)
170	where(("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID) & ("IE_DATA_LINE.IEDATALINESTATUS"="READY"))
180	RES=exec_select(0)
190	if (TD_IE_DL_SCHMEID=="INVENTORY_PART")
200	TD_IE_AXA_IFS/ifs_axa_nakup(TD_IE_EX_BATCH_ID,TD_IE_DL_LINEID,TD_ERROR)
210	endif
220	if (TD_IE_DL_SCHMEID=="CODE_B")
230	TD_IE_AXA_IFS/ifs_axa_stred(TD_IE_EX_BATCH_ID,TD_IE_DL_LINEID,TD_ERROR)
240	endif
250	if (TD_IE_DL_SCHMEID=="STANDARD_NAMES")
260	TD_IE_AXA_IFS/ifs_axa_textsou(TD_IE_EX_BATCH_ID,TD_IE_DL_LINEID,TD_ERROR)
270	endif
310	if (TD_IE_DL_SCHMEID=="WORK_CENTER")
320	TD_IE_AXA_IFS/ifs_axa_prac(TD_IE_EX_BATCH_ID,TD_IE_DL_LINEID,TD_ERROR)
330	endif
340	done

Procedura TD_IE_AXA_IFS/ifs_axa_import

Import dat z IFS

Použití: Při volbě funkce „Import dat z IFS“ v menu Axalantu. Jedná se o hlavní importní funkci, všechny ostatní jsou z ní volány.

Řádek P. Text

```
350 TD_IE_EX_STATE="Imported"
360 update(TD_IE_EX_STATE)
370 where("IE_EXEC_BATCH.EXECUTEBATCHID"=TD_IE_EX_BATCH_ID)
380 exec_update()
390 done
400 put("Přenos dat z IFS do Axalantu byl ukončen...")
402 if (TD_ERROR>0)
403     put("!!! V průběhu importu byly zjištěny chyby !!!")
404     put(sprintf("Počet chyb: %i",TD_ERROR))
405 else
406     put("Import proběhl bez chyb.")
409 endif
500 else
510     put("Import smí provádět pouze skupina TECHNOLOGIE")
512 endif
999 C poslední řádek
```

Procedura TD_IE_AXA_IFS/ifs_axa_nakup

Import nakupovaných součástí z IFS

Použití: Při folbě funkce „Import dat z IFS“ v menu Axalantu. Procedura je volána z ifs_axa_import.

Řádek P. Text

```
5   TD_IE_DL_STATUS="PROCESSING"
10  update(TD_IE_DL_STATUS)
15  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
20  exec_update()
30  select(TD_IE_DL_S1,TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S4,TD_IE_DL_S5,TD_IE_DL_S6)
35  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
40  RES=exec_select(1)
45  select(TD_IE_DL_S7,TD_IE_DL_N1,TD_IE_DL_N2,TD_IE_DL_N3,TD_IE_DL_N5,TD_IE_DL_S8,TD_IE_DL_S9,TD_IE_DL_S10)
50  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
55  RES=exec_select(1)
60  select(TD_IE_DL_S11,TD_IE_DL_S12,TD_IE_DL_S13,TD_IE_DL_N4,TD_IE_DL_N6,TD_IE_DL_N7,TD_IE_DL_S14,TD_IE_DL_S15,TD_IE_DL_S17)
65  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
70  RES=exec_select(1)
75  if (strchr(TD_IE_DL_S2,".")>0)
80    TD_MASTER_TECHN_ID=strpart(TD_IE_DL_S2,1,strchr(TD_IE_DL_S2,".")-1)
85    TD_MASTER_PART_ID=strpart(TD_IE_DL_S2,strchr(TD_IE_DL_S2,".")+1,strlen(TD_IE_DL_S2))
90  else
95    TD_MASTER_TECHN_ID=""
100   TD_MASTER_PART_ID=TD_IE_DL_S2
105  endif
110  TD_MASTER_TEXT_ID=TD_IE_DL_N1
115  TD_MASTER_UNIT=TD_IE_DL_S4
120  TD_MASTER_SKUP_VYR=TD_IE_DL_S5
125  if (TD_IE_DL_S6=="A")
130    TD_MASTER_DAT_LEVIND=230
135    TD_MASTER_SPEC_KOD=""
140  else
145    if (TD_IE_DL_S6=="Z")
150      TD_MASTER_DAT_LEVIND=260
155      TD_MASTER_SPEC_KOD="ZR"
160    else
165      TD_MASTER_DAT_LEVIND=230
170      TD_MASTER_SPEC_KOD=""
```

Procedura TD_IE_AXA_IFS/ifs_axa_nakup

Import nakupovaných součástí z IFS

Použití: Při volbě funkce „Import dat z IFS“ v menu Axalantu. Procedura je volána z ifs_axa_import.

Řádek P. Text

```

175   endif
180   endif
185   TD_MASTER_DAT_ROZMER=TD_IE_DL_S7
190   TD_MASTER_C_HMOTNOST=TD_IE_DL_N2
195   TD_MASTER_DAT_CENA=TD_IE_DL_N6
200   TD_MASTER_DAT_USKLAD=TD_IE_DL_S8
205   TD_MASTER_STAV_SKLAD=TD_IE_DL_N4
210   TD_MASTER_POJZAS=TD_IE_DL_N5
215   TD_MASTER_DAT_PREDPM=TD_IE_DL_N3
220   TD_MASTER_DAT_NJ=TD_IE_DL_S15
225   if (TD_MASTER_DAT_NJ=="L")
230       TD_MASTER_DAT_NJ="I"
235   endif
240   TD_MASTER_DAT_TJ=TD_MASTER_UNIT
245   if (TD_MASTER_DAT_TJ=="L")
250       TD_MASTER_DAT_TJ="I"
255   endif
260   TD_MASTER_DAT_MJNJ=TD_IE_DL_N7
265   TD_MASTER_DAT_POPI=TD_IE_DL_S17
270   TD_MASTER_DAT_NAZ_M=TD_MASTER_DAT_ROZMER
275   TD_MASTER_DAT_MAT_V=TD_MASTER_DAT_POPI
280   while (strchr(TD_MASTER_DAT_NAZ_M," ")>0)
285       TD_MASTER_DAT_NAZ_M=strsub(TD_MASTER_DAT_NAZ_M," ","")
290   done
295   while (strchr(TD_MASTER_DAT_NAZ_M,".")>0)
300       TD_MASTER_DAT_NAZ_M=strsub(TD_MASTER_DAT_NAZ_M,".",",")
305   done
310   while (strchr(TD_MASTER_DAT_NAZ_M,";")>0)
315       TD_MASTER_DAT_NAZ_M=strsub(TD_MASTER_DAT_NAZ_M,";",";")
320   done
325   while (strchr(TD_MASTER_DAT_NAZ_M,"-")>0)
330       TD_MASTER_DAT_NAZ_M=strsub(TD_MASTER_DAT_NAZ_M,"-","-")
335   done

```

Procedura TD_IE_AXA_IFS/ifs_axa_nakup

Import nakupovaných součástí z IFS

Použití: Při folbě funkce „Import dat z IFS“ v menu Axalantu. Procedura je volána z ifs_axa_import.

Řádek P. Text

```

340 while (strchr(TD_MASTER_DAT_NAZ_M,"-")>0)
345   TD_MASTER_DAT_NAZ_M=substr(TD_MASTER_DAT_NAZ_M,"-","-")
350 done
355 while (strchr(TD_MASTER_DAT_NAZ_M,".")>0)
360   TD_MASTER_DAT_NAZ_M=substr(TD_MASTER_DAT_NAZ_M,".",".")
365 done
370 TD_MASTER_DAT_NAZ_M=substr(TD_MASTER_DAT_NAZ_M,1,15)
375 while (strchr(TD_MASTER_DAT_MAT_V," ")>0)
380   TD_MASTER_DAT_MAT_V=substr(TD_MASTER_DAT_MAT_V," ","")
385 done
390 while (strchr(TD_MASTER_DAT_MAT_V,";")>0)
395   TD_MASTER_DAT_MAT_V=substr(TD_MASTER_DAT_MAT_V,";",";")
400 done
405 while (strchr(TD_MASTER_DAT_MAT_V,";")>0)
410   TD_MASTER_DAT_MAT_V=substr(TD_MASTER_DAT_MAT_V,";",";")
415 done
420 while (strchr(TD_MASTER_DAT_MAT_V,"-")>0)
425   TD_MASTER_DAT_MAT_V=substr(TD_MASTER_DAT_MAT_V,"-","-")
430 done
435 while (strchr(TD_MASTER_DAT_MAT_V,"-")>0)
440   TD_MASTER_DAT_MAT_V=substr(TD_MASTER_DAT_MAT_V,"-","-")
445 done
450 while (strchr(TD_MASTER_DAT_MAT_V,".")>0)
455   TD_MASTER_DAT_MAT_V=substr(TD_MASTER_DAT_MAT_V,".",".")
460 done
465 TD_MASTER_DAT_MAT_V=substr(TD_MASTER_DAT_MAT_V,1,15)
470 TD_MASTER_DRUHPOL="N"
475 TD_MASTER_VAL_UNTIL="3000-01-01 00:00:000"
480 if (TD_MASTER_UNIT!="Kč")
485   if (sel_value(count("T_MASTER_DAT.C_ID"),"T_MASTER_DAT.TECHN_ID"=TD_MASTER_TECHN_ID &
"T_MASTER_DAT.PART_ID"=TD_MASTER_PART_ID)==0)
490 C *****založení nové nakupované součásti
500   insert_ent("EDB-ARTICLE EDB-ART-SLI")

```

Procedura TD_IE_AXA_IFS/ifs_axa_nakup

Import nakupovaných součástí z IFS

Použití: Při folbě funkce „Import dat z IFS“ v menu Axalantu. Procedura je volána z ifs_axa_import.

Řádek P. Text

```

505      var(TD_MASTER_TECHN_ID,TD_MASTER_PART_ID,TD_MASTER_TEXT_ID,TD_MASTER_DRUHPOL,TD_MASTER_DAT_USKLAD,TD_MASTER_STAV
      _SKLAD,TD_MASTER_C_HMOTNOST,TD_MASTER_DAT_PREDPM)
510      var(TD_MASTER_POJZAS,TD_MASTER_DAT_TJ,TD_MASTER_DAT_NJ,TD_MASTER_UNIT,TD_MASTER_DAT_MJNJ,TD_MASTER_SKUP_VYR,TD_M
      ASTER_DAT_CENA,TD_MASTER_SPEC_KOD,TD_MASTER_DAT_LEVIND,TD_MASTER_DAT_ROZMER,TD_MASTER_VAL_UNTIL,TD_MASTER_DAT_
      POPIS)
515      RES=execute(1)
520      if (RES==1)
525          if (strpart(TD_MASTER_TECHN_ID,1,1)=="9")
530              if (sel_value(count("T_MASTER_DAT.C_ID"),"T_MASTER_DAT.TECHN_ID"=TD_MASTER_TECHN_ID &
      "T_MASTER_DAT.PART_ID"=TD_MASTER_PART_ID)==1)
535                  TD_MASTER_DAT_CID=sel_value("T_MASTER_DAT.C_ID","T_MASTER_DAT.TECHN_ID"=TD_MASTER_TECHN_ID &
      "T_MASTER_DAT.PART_ID"=TD_MASTER_PART_ID)
540                  update(TD_MASTER_DAT_MAT_V,TD_MASTER_DAT_NAZ_M)
545                  where("T_MASTER_DAT.C_ID"=TD_MASTER_DAT_CID)
550                  exec_update()
555                  endif
560              endif
565              TD_IE_DL_STATUS="IMPORTED"
570              update(TD_IE_DL_STATUS)
575              where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
580              exec_update()
585          else
590              TD_IE_DL_STATUS="ERROR"
595              update(TD_IE_DL_STATUS)
600              where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
605              exec_update()
610              TD_ERROR=TD_ERROR+1
615          endif
620      else
625          if (sel_value(count("T_MASTER_DAT.C_ID"),"T_MASTER_DAT.TECHN_ID"=TD_MASTER_TECHN_ID &
      "T_MASTER_DAT.PART_ID"=TD_MASTER_PART_ID)==1)
630              TD_MASTER_DAT_CID=sel_value("T_MASTER_DAT.C_ID","T_MASTER_DAT.TECHN_ID"=TD_MASTER_TECHN_ID &

```

Procedura TD_IE_AXA_IFS/ifs_axa_nakup

Import nakupovaných součástí z IFS

Použití: Při folbě funkce „Import dat z IFS“ v menu Axalantu. Procedura je volána z ifs_axa_import.

Řádek P. Text

```

        "T_MASTER_DAT.PART_ID"=TD_MASTER_PART_ID)
635
        update(TD_MASTER_TEXT_ID,TD_MASTER_DRUHPOL,TD_MASTER_DAT_USKLAD,TD_MASTER_STAV_SKLAD,TD_MASTER_DAT_PREDPM,TD_MA
        STER_C_HMOTNOST,TD_MASTER_DAT_POPIIS)
640         where("T_MASTER_DAT.C_ID"=TD_MASTER_DAT_CID)
645         exec_update()
650
        update(TD_MASTER_POJZAS,TD_MASTER_DAT_TJ,TD_MASTER_DAT_NJ,TD_MASTER_UNIT,TD_MASTER_DAT_MJNJ,TD_MASTER_SKUP_VYR,TD
        _MASTER_DAT_CENA,TD_MASTER_SPEC_KOD,TD_MASTER_DAT_RÖZMER,TD_MASTER_DAT_LEVIND)
655         where("T_MASTER_DAT.C_ID"=TD_MASTER_DAT_CID)
660         exec_update()
665         if (strpart(TD_MASTER_TECHN_ID,1,1)=="9")
670             update(TD_MASTER_DAT_MAT_V,TD_MASTER_DAT_NAZ_M)
675             where("T_MASTER_DAT.C_ID"=TD_MASTER_DAT_CID)
680             exec_update()
685         endif
687         TD_IE_AXA_IFS/NakupUpdDocNames(TD_MASTER_DAT_CID)
690         TD_IE_DL_STATUS="IMPORTED"
695         update(TD_IE_DL_STATUS)
700         where("IE_DATA_LINE.EXECUTEbatchID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
705         exec_update()
710     else
715         TD_IE_DL_STATUS="ERROR"
720         update(TD_IE_DL_STATUS)
725         where("IE_DATA_LINE.EXECUTEbatchID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
730         exec_update()
735         TD_ERROR=TD_ERROR+1
740     endif
745 endif
750 else
755     TD_IE_DL_STATUS="IMPORTED"
760     update(TD_IE_DL_STATUS)
765     where("IE_DATA_LINE.EXECUTEbatchID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
770     exec_update()

```

Procedura TD_IE_AXA_IFS/ifs_axa_nakup

Import nakupovaných součástí z IFS

Použití: Při folbě funkce „Import dat z IFS“ v menu Axalantu. Procedura je volána z ifs_axa_import.

Řádek P. Text

775 endif

780 C poslední řádek

Procedura TD_IE_AXA_IFS/ifs_axa_stred

Import středisek z IFS

Použití: Při folbě funkce „Import dat z IFS“ v menu Axalantu. Procedura je volána z ifs_axa_import.

Řádek P. Text

```
10  TD_IE_DL_STATUS="PROCESSING"
20  update(TD_IE_DL_STATUS)
30  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
40  exec_update()
50  select(TD_IE_DL_S3,TD_IE_DL_S4)
52  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
53  RES=exec_select(1)
60  TD_STREDISKO_KOD=TD_IE_DL_S3
70  TD_STREDISKO_NAZEV=TD_IE_DL_S4
80  select(TD_STREDISKO_CID)
90  where("T_STRED_DAT.KOD"=TD_IE_DL_S3)
100 RES=exec_select(1)
105 if (RES==1)
110  update(TD_STREDISKO_NAZEV)
120  where("T_STRED_DAT.C_ID"=TD_STREDISKO_CID)
130  exec_update()
140 else
150  insert_ent("TD-STREDISKA")
160  var(TD_STREDISKO_NAZEV,TD_STREDISKO_KOD)
170  RES=execute(1)
180 endif
190 TD_IE_DL_STATUS="IMPORTED"
220 update(TD_IE_DL_STATUS)
230 where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
240 exec_update()
999 C poseltní řádek
```

Procedura TD_IE_AXA_IFS/ifs_axa_textsou

Import názvů součástí z IFS

Použití: Při folbě funkce „Import dat z IFS“ v menu Axalantu. Procedura je volána z ifs_axa_import.

Řádek P. Text

```
10  TD_IE_DL_STATUS="PROCESSING"
20  update(TD_IE_DL_STATUS)
30  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
40  exec_update()
50  select(TD_IE_DL_S2,TD_IE_DL_S3,TD_IE_DL_S1,TD_IE_DL_N1)
52  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
53  RES=exec_select(1)
60  TD_TEXTSOU_ID=TD_IE_DL_N1
70  TD_TEXTSOU_TEXT_FRA=TD_IE_DL_S2
71  TD_PRENOS/cesky(TD_TEXTSOU_TEXT_FRA)
75  TD_TEXTSOU_TEXT1_FRA=TD_IE_DL_S2
80  select(TD_TEXTSOU_CID)
90  where("T_TEXTSOU_DAT.ID"=TD_TEXTSOU_ID)
100 RES=exec_select(0)
110 if (RES==0)
120   insert_ent("TD-TEXTY-SOUC")
130   var(TD_TEXTSOU_ID,TD_TEXTSOU_TEXT_FRA,TD_TEXTSOU_TEXT1_FRA)
140   RES=execute(1)
150 else
160   update(TD_TEXTSOU_TEXT_FRA,TD_TEXTSOU_TEXT1_FRA)
170   where("T_TEXTSOU_DAT.C_ID"=TD_TEXTSOU_CID)
180   exec_update()
190 endif
192 TD_IE_DL_STATUS="IMPORTED"
220 update(TD_IE_DL_STATUS)
230 where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
240 exec_update()
999 C poslední řádek
```

Procedura TD_IE_AXA_IFS/ifs_axa_prac**Import pracovišť z IFS**

Použití: Při folbě funkce „Import dat z IFS“ v menu Axalantu. Procedura je volána z ifs_axa_import.

Řádek P. Text

```
10  TD_IE_DL_STATUS="PROCESSING"
20  update(TD_IE_DL_STATUS)
30  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
40  exec_update()
50  select(TD_IE_DL_S3,TD_IE_DL_S1)
52  where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
53  RES=exec_select(1)
60  TD_PRAC_KOD=TD_IE_DL_S1
70  TD_PRAC_KOMENTAR=TD_IE_DL_S3
80  select(TD_PRAC_CID)
90  where("T_PRAC_DAT.KOD"=TD_IE_DL_S1)
100 RES=exec_select(0)
110 if (RES==0)
120   insert_ent("TD-PRACOVISTE")
130   var(TD_PRAC_KOD,TD_PRAC_KOMENTAR)
140   RES=execute(1)
150 else
160   update(TD_PRAC_KOMENTAR)
170   where("T_PRAC_DAT.C_ID"=TD_PRAC_CID)
180   exec_update()
190 endif
192 TD_IE_DL_STATUS="IMPORTED"
220 update(TD_IE_DL_STATUS)
230 where("IE_DATA_LINE.EXECUTE_BATCHID"=TD_IE_EX_BATCH_ID & "IE_DATA_LINE.LINEID"=TD_IE_DL_LINEID)
240 exec_update()
999 C  oslední řádek
```