

IBM

@server

iSeries

Globalizace (vývoj globálních aplikací)





@server

iSeries

Globalizace (vývoj globálních aplikací)

Obsah

Globalizace operačního systému OS/400	1
Vývoj globálních aplikací	2
Cíle a procesy	2
Návrh globálních aplikací	7
Doporučení pro programátory při návrhu globálních aplikací	45
Dodání globalizovaných aplikací	57

Globalizace operačního systému OS/400

Vzhledem k tomu, že firmy zařazují elektronický obchod mezi základní podnikové procesy v globálním měřítku, mohou jejich budoucí i zavedení zákazníci a aktivní partneři využít skutečnosti, že díky globalizaci softwaru se zvyšují jejich příjmy a snižují jejich výdaje. Mohou také zlepšit komunikaci se zákazníky a zvýšit úspory. Globalizovaný software vám poskytuje tyto výhody:

- Zvýšenou spokojenost zákazníků, která může podpořit prodej.
- Zdokonalení komunikace zákaznické podpory.
- Zdokonalené šíření globálních informací.
- Lepší návratnost investic do informačních technologií (IT).

Tyto informace ukazují, jak:

- Vytvořit aplikaci efektivně a za minimální náklady.
- Provést doplňkovou úpravu stávajících aplikací pro globalizaci a vytvořit nové aplikace navrhované pro globalizaci. Návrh aplikace pro globalizaci je však obvykle méně nákladný než dodatečná úprava stávající aplikace.
- Zajistit, aby návrh aplikace nekolidoval se současným nebo plánovaným návrhem jiných internacionalizovaných aplikací.

Tyto stránky poskytují jediný zdroj informací, které potřebujete k vypracování aplikací pro národní a mezinárodní uživatele. Můžete si také vyhledat informace o tom, co je nového v tomto vydání a jak si můžete vytisknout toto téma.

Přehled globalizace

Toto téma popisuje, jak byla globalizace implementována v operačním systému OS/400^(R), a zahrnuje témata popisující pro globalizaci typické hodnoty v systému a další témata, která popisují způsob podpory globalizace prostřednictvím služeb a funkcí v operačním systému OS/400.

Nastavení verze národního jazyka v operačním systému OS/400

Toto téma popisuje kroky, které musíte podniknout, abyste řádně nainstalovali a konfigurovali verzi národního jazyka v operačním systému OS/400, od výběru a instalace hardwaru, instalace softwaru a konfigurování vašeho prostředí, aby fungovalo v globalizovaném nastavení. Tyto informace můžete použít při instalaci vlastních serverů a zásady můžete uplatnit při vývoji aplikací pro zákazníky, kteří si v operačním systému OS/400 instalují verzi vlastního národního jazyka.

Vývoj globálních aplikací

Toto téma poskytuje návody pro návrh, vývoj a dodávku globalizovaných aplikací:

- Návrh funkcí, které jsou citlivé na národní jazyky.
- Podporování různých typů hardwarové podpory.
- Překlad textových dat ve vaší aplikaci.
- Zpřístupnění vaší aplikace v celosvětovém měřítku.

Práce s daty v globalizovaných aplikacích

Toto téma popisuje, jakým způsobem umožňuje operační systém OS/400 pracovat s daty v globalizovaném prostředí. Tyto stránky zahrnují témata popisující data Unicode a UCS-2, čínský standard GB18030, jak používat CCSID při důsledné integraci vícejazyčného prostředí a jak používat obousměrná data, data DBCS a lokality.

Referenční informace o globalizaci

Toto téma poskytuje podrobné podpůrné informace k pojmům a úkolům probíraným v kategorii globalizace.

Kontrolní seznamy globalizace

Toto téma uvádí přehled všech kontrolních seznamů, které jsou uvedeny u těchto stránek. Kontrolní seznamy jsou užitečné připomínky toho, co musíte brát v úvahu při tvorbě globálních aplikací a práci s nimi.

Vývoj globálních aplikací

Globální aplikace jsou aplikace, které mají podporu národního jazyka. Podpora národního jazyka umožňuje uživateli zadat, uložit, načíst, vytisknout a zobrazit data v jazyce dle vlastního výběru. Podpora národního jazyka také umožňuje, aby uživatelé viděli a zadávali data, příkazy, názny, zprávy a dokumentaci v požadovaném jazyce a ve formátech, které odpovídají jejich kulturním zvyklostem.

Ačkoli vaše důvody mohou být jiné, vzniká většina internacionalizovaných aplikací proto, že:

- Trh požaduje globální softwarové produkty, které jsou vnímány jako místní.
- Aplikace se používá ve společenství, které reprezentuje různé kultury.
- Rozšiřují se finanční možnosti.

Následující odkazy poskytují cenné informace, které byste měli znát na začátku procesu vývoje:

- Vývoj globálních aplikací: cíle a procesy
- Návrh globálních aplikací
- Doporučení pro programátory při návrhu globálních aplikací
- Dodání globalizovaných aplikací

Další informace

Informace o tom, jak pracovat s různými typy dat v globálním prostředí, uvádí téma [Práce s daty v globalizovaných aplikacích](#).

Cíle a procesy

Dříve než začnete investovat svůj čas a peníze do vývoje globálních aplikací, bude pro vás prospěšný proces plánování, při němž budete mít příležitost zvážit, jak účelně a efektivně posloužit globálním uživatelům. Následující témata vám pomohou takový plán vyvinout:

- Cíle vývoje
- Proces průzkumu trhu
- Proces vývoje
- Proces dokumentace
- Proces překladu
- Proces testování
- Proces balení a instalace
- Proces údržby aplikace

Cíle vývoje globalizace

Toto téma vám může pomoci při plánování a vytváření aplikací pro mezinárodní použití. Doporučení v tomto tématu předpokládají, že sledujete tyto základní cíle:

- Vytvořit aplikaci efektivně.
- Vytvořit aplikaci s minimálními náklady. Můžete provést dodatečnou úpravu stávajících aplikací za účelem globalizace nebo vytvořit nové aplikace navržené pro globalizaci. Návrh aplikace pro globalizaci je však obvykle méně nákladný než dodatečná úprava stávající aplikace.
- Zajistit, aby návrh aplikace nekolidoval se současným nebo plánovaným návrhem jiných internacionalizovaných aplikací.
- Při vytváření aplikací s podporou národního jazyka musíte naplánovat nebo realizovat tyto úlohy:

- Navrhnout funkce, které jsou citlivé na národní jazyky.
- Zajistit podporu různých typů hardwarové podpory.
- Přeložit textová data ve vaší aplikaci.
- Zpřístupnit vaši aplikaci v celosvětovém měřítku.

Proces plánování vývoje globalizace

Globální aplikace by měla být dobře naplánována již v nejrannějších fázích, má-li se ušetřit čas, práce i peníze. Nemělo by být nutné překompilovat programy ani znovu balit datové objekty. Od vašeho produktu se však může vyžadovat, aby používal různé datové objekty v závislosti na vámi používané verzi jazyka. Měli byste mít jednu sadu programového kódu a podle potřeby různé sady kódu závislého na textu a kulturních zvyklostech.

Při plánování globální aplikace zvažte následující procesy.

- Průzkum trhu.
- Vývoj.
- Dokumentace.
- Překlad.
- Testování.
- Balení a instalace.
- Údržba aplikace.

Proces průzkumu trhu

Nejdůležitějším faktorem při každém rozhodování je to, že víte, pro koho své aplikace navrhujete a vyvíjíte. Abyste dokázali odpovědět na tuto otázku, položte sobě i svým potenciálním zákazníkům následující typy otázek.

Jaké jsou mé dnešní a zítřejší cíle na trhu?

Odpověď na tuto otázku se výrazně liší podle toho, zda svůj tržní prostor definujete v různých zemích, nebo pouze v oblasti, kde se mluví vaším jazykem, nebo zda chcete zahrnout země, v nichž se mluví jinými jazyky. Pokud například vytváříte aplikaci v jazyce založeném na latině, zvýší se složitost aplikace, když se rozhodnete zahrnout země používající jazyky, které na latině založené nejsou, např. hebrejštinu, čínštinu nebo japonštinu. Složitost aplikace vzrůstá, neboť se musíte nějakým způsobem vypořádat s nekompatibilními znakovými sadami a komplikovanějšími metodami zadávání dat.

Kromě jazykových problémů je třeba zvážit ještě další oblasti. Musíte chápat kulturu, zvyklosti, způsoby podnikání a právní aspekty cílového trhu. Chcete-li se pro své zákazníky stát přijatelným obchodním partnerem, proniknout na jejich trh a podporovat je v jejich zemích, je nezbytné, abyste porozuměli jejich životnímu stylu.

Tyto faktory mohou ovlivnit:

- Schopnosti, které potřebujete (technické a kulturní znalosti, jazyk, právo).
- Prostředí, která je třeba brát v úvahu.
- Struktura vaší společnosti a organizace podpory.
- Váš vztah k jiným společnostem.
- Prostředky, které potřebujete (lidi, čas a peníze).

Kdo jsou uživatelé mé aplikace?

Musíte porozumět požadavkům, které budou mít budoucí uživatelé vaší aplikace. Například, chtějí:

- Pracovat s oddělenými databázemi pro různé jazyky?

- Pracovat se sdílenou databází pro všechny jazyky?
- Vyměňovat si nebo konsolidovat data?
- Pracovat s různými jazyky v závislosti na koncovém uživateli, společnosti nebo zákaznických společnostech?
- Používat databázové nástroje pro koncové uživatele za účelem jejich vlastních dotazů do aplikační databáze?

Všechny tyto faktory mohou ovlivnit návrh, pro který se rozhodnete, způsob, jakým bude vaše aplikace schopná přepínat z prostředí jednoho jazyka na jiný, a jak se bude provádět prezentace a konverze dat.

Jak velká podpora globalizace je nezbytná?

Když pochopíte požadavky vašich zákazníků a jejich koncových uživatelů, můžete přistoupit k rozhodování, jaký druh informací citlivých na národní zvyklosti musíte uchovávat a udržovat, jaký budete používat typ prezentace dat, které části musíte přeložit a do jaké míry má být vaše aplikace schopná integrace do jiných prostředí.

Jaké jsou náklady na tuto práci?

Chcete-li odhadnout očekávaný výnos, proveďte analýzu míst, která jste si zvolili jako cílový trh. Když už znáte požadavky, měli byste být schopni určit nezbytnou práci a náklady. Na základě této částky budete moci porovnat náklady a očekávaný výnos.

Co je nákladnější, přímé umožnění podpory v aplikaci nebo dodatečné upravování aplikace?

Počáteční náklady na návržení aplikace tak, aby umožňovala podporu národního jazyka, mohou být vyšší. Uvědomte si však, že kroky takového návrhu jsou více založeny na normálních stavebnicových metodách návrhu odvíjejících se od aktuálních dat, které zvyšují kvalitu vaší aplikace dokonce i bez aktivace podpory národního jazyka. Jelikož dobrý návrh pomáhá lidem pochopit a popsat aplikační systém, bude mít zajištěnou jistou návratnost vaší investice. Dobrý návrh umožňuje zvýšit produktivitu vývoje i údržby. Další výhoda spočívá v tom, že aplikaci navrhnete a implementujete pouze jednou, dokonce i pro mnoho různých jazykových verzí. Ve srovnání s dodatečným upravováním existující aplikace je výrazně levnější plánovat a navrhovat podporu od samého začátku.

Proces vývoje

Dříve než budete připraveni na vývoj aplikací podporujících národní prostředí, přemýšlejte o těchto otázkách, chcete-li, aby vaše práce byla úspěšná.

Školení pro vývoj internacionalizovaných aplikací

Máte-li v úmyslu vyvíjet aplikace podporující národní prostředí, měli byste uvažovat o dalším počátečním školení. Dále jsou uvedena důležitá témata ke studiu:

- Všeobecné pojmy globalizace.
- Podpora globalizace dostupná v operačním systému OS/400.
- Podpora dostupná v jiných systémech a aplikacích, s nimiž vaše aplikace spolupracuje.
- Izolace různých částí aplikace.
- Prezentace dat odpovídající konvencím dané kultury.
- Návrh a kódování částí s textovými daty.
- Proces překladu.
- Integrace produktu a systému.
- Balení, instalace a nastavení.
- Podpora a údržba produktu.

Na základě návodů pro podporu globalizace připravte nejprve prototyp aplikace a otestujte zvolený způsob implementace aplikace ve vašem specifickém prostředí. Potom zahrňte návody pro podporu globalizace do obecných procesů, zásad a standardů vývoje svých aplikací.

Implementace internacionalizovaných aplikací

Při implementaci internacionalizované aplikace je nejdůležitějším cílem vytvořit pouze jednu sadu prováděného kódu. Musíte důsledně rozlišovat mezi prováděným kódem a textovými daty. Je nanejvýš důležité, abyste zvolený přístup standardizovali v celé aplikaci. Pracujte s konvencemi pojmenování, které jsou jedinečné a jasně definované. Chcete-li pochopit a udržet tyto informace v aplikaci, zacházejte s parametry volanými z programu důsledným způsobem.

Proces dokumentace

Dokumentace by měla poskytovat informace koncovým uživatelům aplikačního systému v jejich vlastním jazyce. Dokumentace by také měla zahrnovat informace o instalaci, nastavení a o úpravách pro koncového uživatele, systémového operátora a správce systému aplikací.

Uživatelská dokumentace by měla mít formu textových dat, která lze snadno překládat. Kdykoli je to možné, kombinujte informace online nápovědy a uživatelské dokumentace, abyste omezili množství textu k překladu. Aplikace by měla vytvořit případné příklady zobrazení nebo rozvržení tisku a zahrnout je do dokumentace.

Proces překladu

Překlad textových dat je časově velmi náročný proces. Textová data by měli mít překladatelé k dispozici v rané fázi vývoje, i když kód ještě není stabilní. Při plánování překladu uvažte následující oblasti:

Fyzické vybavení

Každý překladatel by měl mít vybavení, které je kompatibilní s překládaným jazykem. Obrazovky i klávesnice by měly mít veškeré charakteristiky potřebné pro překlad a na klávesnicích by mělo být možné psát překládaný text.

Překladatelské nástroje

Poskytněte překladatelům nástroje, které zvyšují produktivitu a zabraňují překladu netextových dat aplikace. Když se rozhodujete o nákupu nebo vývoji překladatelského nástroje, měli byste zajistit tyto vlastnosti.

- Editor, jenž poskytuje schopnost zobrazit obrazovky, které se budou ukazovat koncovému uživateli, a schopnost překládat textová data v systému, přičemž jsou chráněny ty části aplikace, které nejsou textovými daty. Editor by měl také zahrnovat funkce snímání, nahrazení, vyhledání, kopírování, přesouvání a výmazu.
- Funkce slovníku, která zajišťuje konzistenci používaných slov a frází v rámci celého produktu.
- Proces ověření, jímž se kontrolují chyby v překladu, které by mohly způsobit nesprávné fungování aplikace.
- Funkce začlenění, která umožňuje začlenit přeložený text do nové verze původního textu. Díky této funkci je možné překládat pouze nový text, což vede k úsporám času i práce.
- Funkce tisku pro účely ověření.

Školení překladatelů

Důležitým faktorem je seznámení překladatelů s produktem, který překládají, a také s nástroji, které k překladu používají. Proces překladu neznámá nahrazovat jedno slovo druhým, ale je to vytváření pojmů v jiném jazyce. Pokud překladatel zná překládaný produkt, bude v závěru tento produkt srozumitelnější pro koncového uživatele. Z toho vyplývá, že by měly být předem pečlivě naplánovány časové i finanční prostředky pro školení překladatelů.

Návody a pokyny týkající se překladů

Chcete-li zajistit správný překlad, měli byste překladatelům poskytnout návody a pokyny pro překlad. Pokud se má například správně přeložit chybová zpráva, je důležité vědět, v jakém kontextu se tato zpráva zobrazuje. V tomto směru může překladatelům pomoci také poznámka uvádějící, jaká chyba zprávu způsobila.

Překladatelský slovník zvláštních významů

Chcete-li zajistit přesný překlad, použijte terminologii založenou na definicích ve standardních, běžně dostupných slovnících. Pokud aplikace používá výrazy, které se v běžných slovnících nevyskytují, nebo výrazy, jejichž použití se liší od standardních definic, dejte překladatelům k dispozici slovník těchto nestandardních výrazů. Ve vašich aplikacích se vyhněte používání zkratk a akronymů. Pokud je musíte používat, definujte je ve slovníku zvláštních významů. Uvědomte si, že zkratky a akronymy, které jsou zřejmé ve vašem jazyku, nemusí být jasné v jiném jazyku.

Proces testování

Testování produktu podporujícího globalizaci by se mělo provádět ve třech fázích:

1. Testování prováděného kódu.
Prováděný kód by se měl testovat v prostředí podporujícím globalizaci, aby bylo možné zkontrolovat všechny možné kombinace závislé na jazyku. Překladatelé by neměli testovat funkčnost produktu.
2. Kontrola textových dat.
Textová data by se měla testovat z hlediska správnosti překladu a konzistence v rámci celého produktu.
3. Integrace prováděného kódu a textových dat.
Po odděleném otestování textových dat i prováděného kódu by se měl provést integrační test, na jehož základě se zjistí, zda aplikace bere v úvahu všechna zpracování závislá na globalizaci a zda překlad textových dat nezpůsobil nesprávné fungování produktu.
Pokud se vaše aplikace bude provozovat v mezinárodních nebo vícejazyčných systémech, měli byste naplánovat zvláštní test, který by zahrnoval více sad textových dat.

Proces balení a instalace

Při balení aplikací berte v úvahu prováděný kód, přeložená textová data a instalační dokumenty. Dále vám předkládáme návrhy pro usnadnění balení a instalace vaší aplikace:

- Prováděný kód a textová data uchovávejte odděleně.
- Textová data zabalte tak, aby zákazníci obdrželi pouze data v těch jazycích, které si objednali. (Pokud by byla všem zákazníkům posílána textová data ve všech jazycích, vedlo by to k plýtvání systémovými prostředky a k problémům s údržbou.)
- Poskytněte dokumentaci s vyčerpávajícím popisem instalace (přeloženou do jazyka známého osobě, která bude produkt instalovat), abyste předešli zbytečným problémům ze strany operátora a abyste hned na začátku nezbudili mylný dojem, že aplikace není spolehlivá.

Dokumentace k instalaci by měla pokrývat tyto oblasti:

- Co je potřeba k instalaci a spuštění aplikace, např. hardwarové a softwarové požadavky.
- Jak instalovat aplikaci a jak postupovat, když se vyskytnou chyby.
- Jaké změny je třeba provést s ohledem na:
 - Definice podsystémů.
 - Popisy zařízení.
 - Uživatelské profily.
 - Systémové hodnoty.
 - Seznamy knihoven.
- Jaká jsou omezení aplikace?

Proces údržby aplikace

Při plánování údržby vícejazyčné aplikace mějte na paměti tyto body:

- Prováděný kód musí být udržován odděleně od textových dat. Tyto oddělené komponenty musí být plně synchronizovány. Přeprogramování návrhu jedné komponenty si může vyžádat přeprogramování druhé komponenty.
- Při každé změně textových dat zajistěte, aby byla provedena ve všech jazycích, do kterých byla tato data přeložena. Tímto způsobem můžete zajistit jedinou úroveň údržby pro celý produkt.
- Zajistěte testování prováděného kódu u každé změny textových dat, kterou distribuujete.

Návrh globálních aplikací

Cílem návrhu součástí mezinárodních aplikací je vytvořit komponenty, které nezávisle podporují národní jazyky. Podpora jednoho jazyka by neměla být v konfliktu s podporou jiného jazyka. Podpora jednoho jazyka by neměla znamenat žádné omezení funkcí produktu v jiném jazyce.

Vaše aplikace by měla být schopna podporovat několik jazyků současně. Například, podpora jazyka s dvoubajtovou znakovou sadou (DBCS) by neměla vylučovat podporu jazyků s jednobajtovou znakovou sadou (SBCS). Při instalaci knihoven zvažte použití několika knihoven s textovými daty, které je možné dynamicky alokovat pro ladění, balení a doručení.

Při vývoji globální aplikace pro server iSeries musíte brát v úvahu tyto a další otázky jedinečného návrhu, protože budou mít vliv na to, jak vypracujete a zakódujete aplikaci pro globálního uživatele. Následující témata vymezují rozsah těchto otázek a poskytují užitečná vodítka pro to, jak byste měli postupovat:

- Kontrolní seznam: návrh aplikace
- Globalizace a lokalizace
- Uspořádání a architektura aplikace
- Uživatelská rozhraní

Kontrolní seznam: návrh aplikace

Následující tabulka poskytuje pokyny, kterými se můžete řídit při tvorbě aplikace podporující národní prostředí.

Vyhovuje	Nepoužitelné	Pravidlo
		Existence určité znakové sady v rámci systému nebo jeho komponent se nesmí předpokládat.
		Konverze velkých a malých znaků musí být definovatelná u každého jazyka a kódové stránky.
		Překrývání (folding) musí být definovatelné u každého jazyka a kódové stránky. Překrývání je proces, ve kterém se znaky, které nemohou být vtištěny nebo zobrazeny na určitém zařízení, nahrazují znaky, které jsou tisknutelné nebo zobrazitelné.
		Použití grafického znaku k účelům softwarového řízení nesmí předem zabránit použití stejného znaku v textu zpráv, menu, náznaků, vstupních nebo výstupních polí.
		Sada znaků povolených při zápisu dat musí být definovatelná pro systémového operátora, uživatele nebo pro aplikaci.
		Grafické symboly a ikony musí být přeložitelné.
		Všechny znaky na aktivní kódové stránce musí být přístupné.
		Části produktu závislé na jazyce musí být kvůli snadné modifikaci izolovány od částí, které na jazyce závislé nejsou.

Vyhovuje	Nepoužitelné	Pravidlo
		Návrh produktu musí umožňovat, aby podpora národního jazyka byla u různých komponent produktu vzájemně nezávislá.
		Ve strategických bodech musí být zajištěny výstupní body pro národní jazyky.
		Musí být povolena diagnostika.
		Uživatelům musí být k dispozici logická rozvržení odlišná od rozvržení dané fyzické klávesnice.
		Všechny texty uživatelského rozhraní a informace pro ovládání prezentace musí být izolovány od prováděného kódu.
		Funkce závislé na délce pole zobrazení a na jeho pozici nebo pouze na pozici tohoto pole, nesmí být navrženy tak, aby je ovlivnilo rozšíření textu uživatelského rozhraní.
		Je nutné zabezpečit metodu, která umožní identifikaci a sledování panelů a zpráv v procesu překladu.
		Musí být povoleno, aby se proměnné v poli zobrazení objevily na libovolném místě a v libovolném pořadí.
		Zprávy a jiná zobrazená slova nebo fráze musí tvořit úplné entity a nesmí být tvořeny jednotlivými slovy nebo frázemi.
		Zápis příkazů, klíčových slov nebo odpovědí koncového uživatele musí být možný bez ohledu na to, zda jde o velká nebo malá písmena.
		Produkt s funkcemi závislými na národním jazyku musí být navržen tak, aby umožňoval přidání jiných zemí nebo národních jazyků.
		Nemělo by se předpokládat, že abeceda s malými písmeny je invariantní.
		Znakové sady by měl být schopen definovat operátor, uživatel nebo aplikace.
		Zvláštní znaky, včetně interpunkce, by měly být definovatelné a nezávislé na programu.
		Textové moduly uživatelského rozhraní by měly být v samostatném balíku odděleně od prováděného kódu.

Globalizace a lokalizace

Operační systém OS/400 řídí činnost programů a poskytuje služby, jako je řízení zdrojů, plánování úloh, řízení vstupů a výstupů a správa dat. Je navržen tak, aby doplňoval a rozšiřoval schopnosti serverů iSeries a byla tak zajištěna plně integrovaná podpora pro interaktivní i dávkové aplikace.

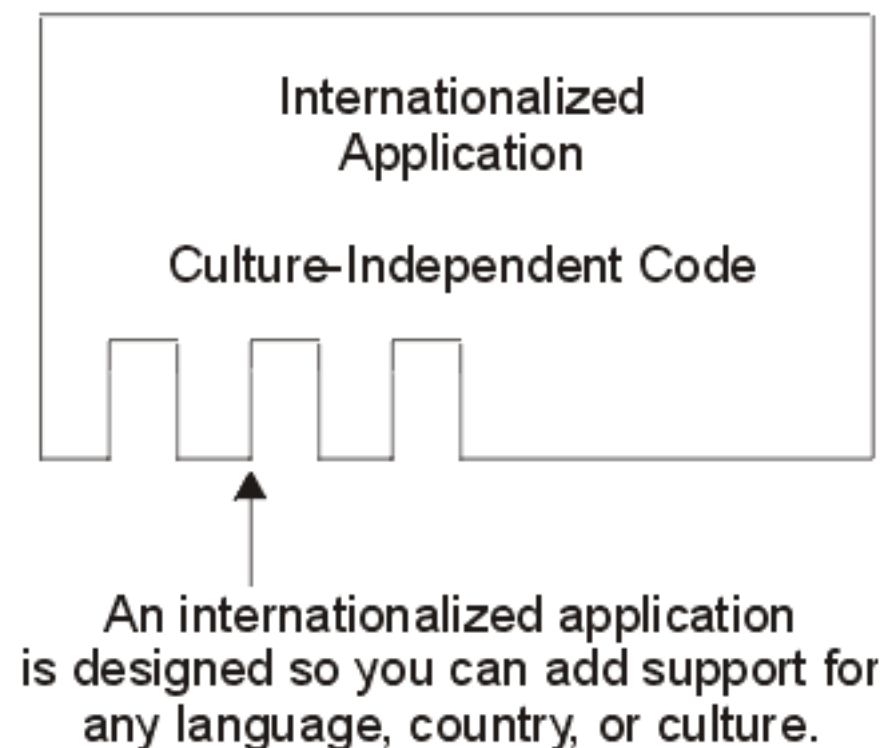
Řada funkcí operačního systému OS/400 se vztahuje přímo na interaktivní zpracování dat. Mezi tyto funkce patří:

- Databázová podpora, která umožňuje rychlé zpřístupnění aktuálních firemních dat z libovolné pracovní stanice.
- Řízení práce (Work management), která podporuje časové plánování zpracování požadavků od všech uživatelů pracovních stanic.
- Podpora vývoje aplikací, která umožňuje online vývoj a testování nových aplikačních programů souběžně s normálními provozními (produktivními) činnostmi.
- Podpora systémových operací, která umožňuje uživateli odpovědnému za provoz systému provádět úlohy z obrazovkové stanice s využitím jednoho jazyka CL, který je plně vybaven náznaky a nápovědou pro všechny příkazy.

- Podpora nápovědy a indexového vyhledávání, která umožňuje uživatelům vyžádat si online informace k široké škále témat.
- Podpora zpracování zpráv, která umožňuje komunikaci mezi systémem, uživatelem odpovědným za provoz systému, uživateli pracovních stanic a programy spouštěnými v systému.
- Podpora zabezpečení, která slouží k ochraně dat a jiných systémových prostředků před neoprávněným přístupem.

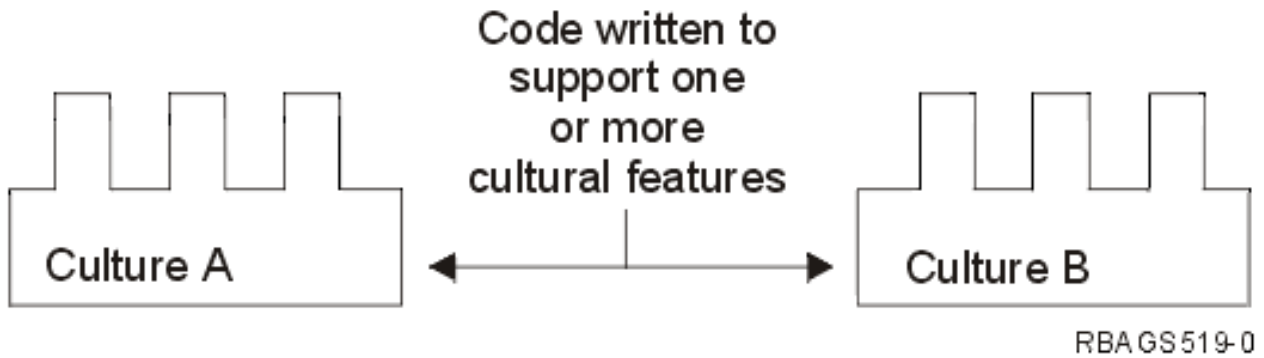
Kromě těchto funkcí operační systém OS/400 poskytuje podporu národního jazyka. Podpora národního jazyka umožňuje uživatelům vzájemně komunikovat se systémem v jazyce dle jejich výběru, přičemž jsou zajištěny výsledky, které jsou přijatelné z hlediska národních zvyklostí. Podpora národního jazyka je tvořena dvěma částmi: globalizací a lokalizací.

Globalizace je podpora, která umožňuje aplikaci fungovat ve všech jazykových prostředích, aniž by bylo nutné aplikaci jakkoliv měnit. Tento typ návrhu je také znám pod označením umožnění podpory národního jazyka u aplikace. Globalizovaná aplikace, jak je znázorněna na následujícím obrázku, je kulturně neutrální.

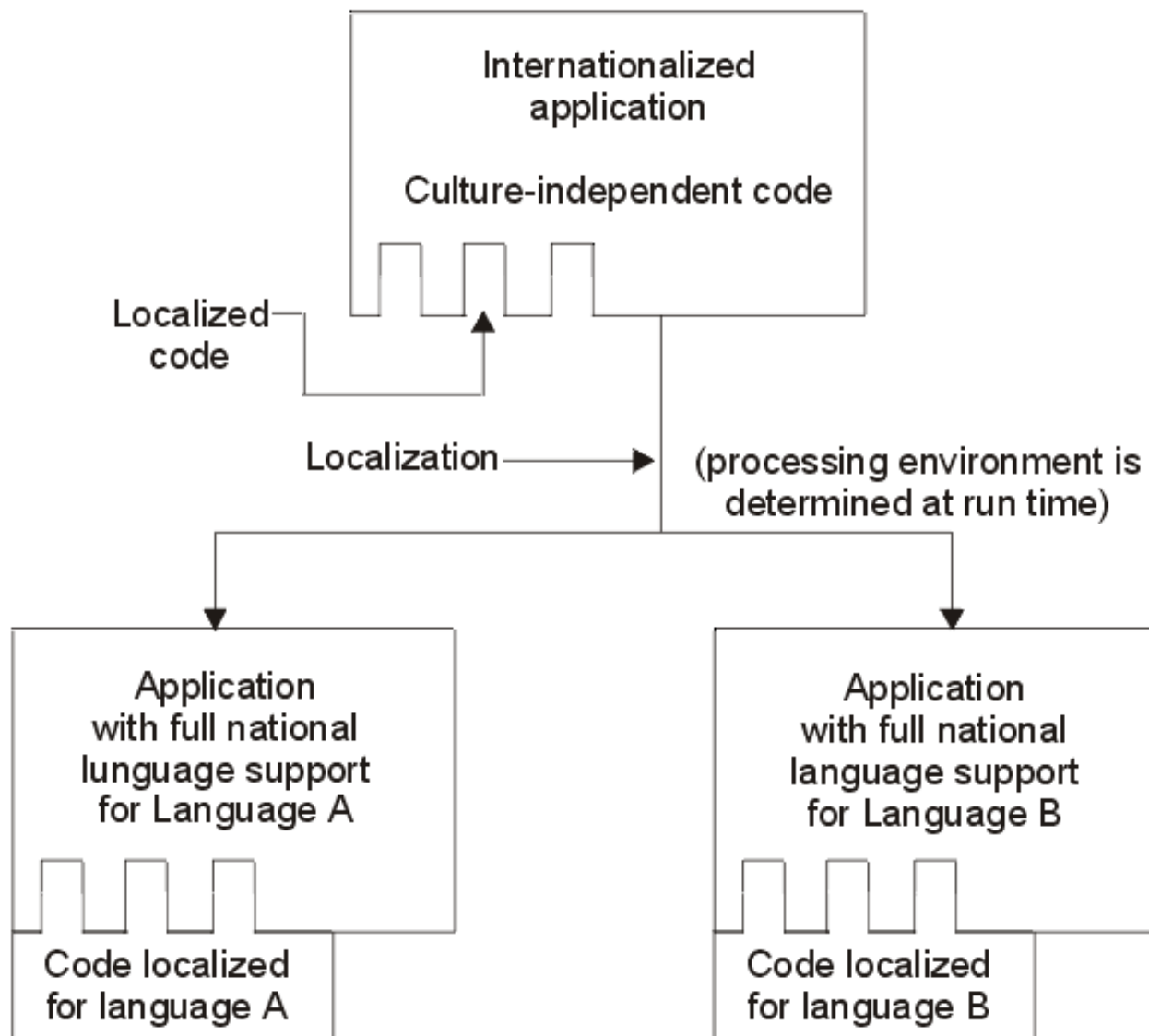


RBAGS519-0

Na druhou stranu **lokalizace** umožňuje, aby aplikace fungovala v konkrétním jazyku, zemi nebo kultuře. Lokalizace aplikace jde o krok pozadu za globalizací aplikace, jak je to znázorněno na tomto obrázku.



Když je lokalizovaný kód v době běhu programu integrován s globalizovaným kódem, jeví se výsledná aplikace uživateli jako plně podporovaná z hlediska národního jazyka. Prostředí zpracování definuje, který lokalizační kód se v době provádění programu zkombinuje s globalizovaným kódem, jak je to uvedeno na následujícím obrázku.



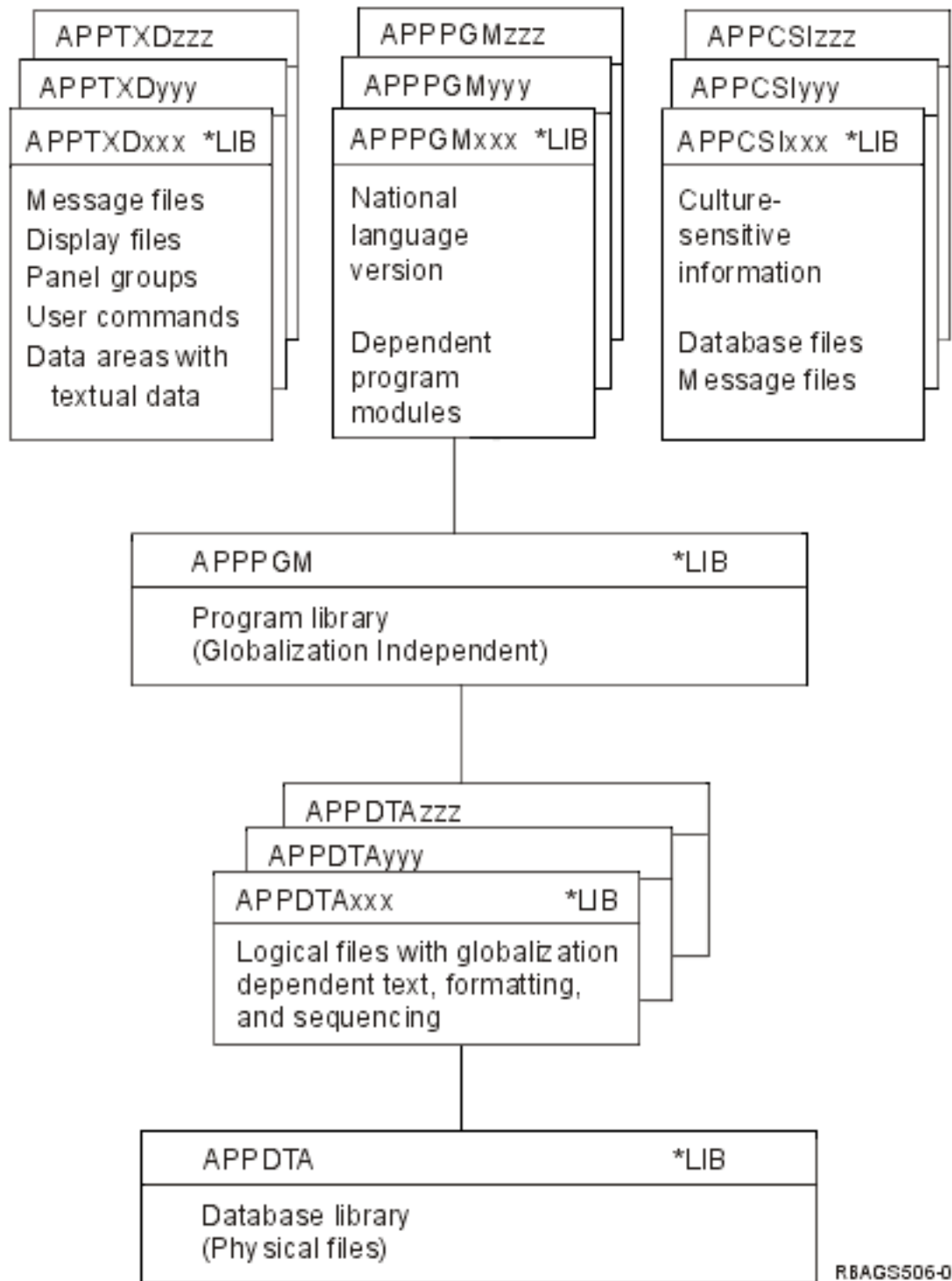
RBAGS521-0

Uspořádání a architektura aplikace

Při návrhu mezinárodní aplikace uvažujte o tom, jak organizovat a členit aplikaci tak, aby mohla být použita v mezinárodním prostředí. Především uvažte následující strategie:

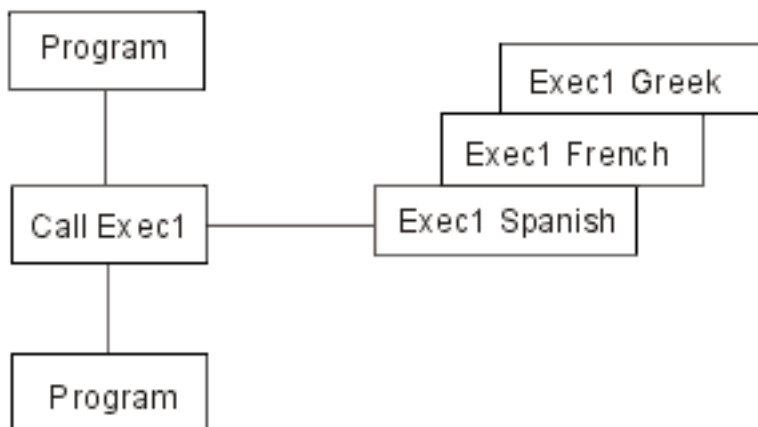
- Na odpovídajících místech oddělte programové moduly.
- Přidělte částem aplikace jména vhodná pro vícejazyčné prostředí.
- Kdykoli je to možné, informujte se v tématu specifikace.
- Při práci s definicemi databáze zajistěte, aby vícenásobné sady logických souborů byly v samostatných knihovnách.

Následující obrázek ukazuje doporučený způsob, jak organizovat části vaší aplikace.



Oddělení modulů programu: Z vašeho spuštěného kódu můžete oddělit části závislé na kulturních zvyklostech a můžete nastavit kulturně závislé prostředí. K tomu účelu můžete využít systémové hodnoty, uživatelské profily, atributy úloh a atributy objektů.

V případě, že není možné oddělit části s národním jazykem a části závislé na kulturních zvyklostech od prováděného kódu, musíte umožnit výstupy nebo volání národního jazyka na všech místech, kde jsou požadovány funkce závislé na podpoře národního jazyka. Na následujícím obrázku je znázorněn výstup národního jazyka.



RBAGS504-0

Jména částí aplikace: Chcete-li aplikace aktivovat pro různé jazyky a země, měly by vaše konvence pojmenování brát v úvahu prostředí cílových systémů. Používejte znaky, které jsou ve všech cílových prostředích dostupné, dají se zobrazit a vytisknout. Používejte pouze znaky invariantní znakové sady, kdykoli budete uvádět jména:

- knihoven
- databázových souborů
- souborů zařízení (obrazovky nebo tiskárny)
- panelů nápovědy
- souborů zpráv
- uživatelských příkazů
- programů
- formátů záznamu
- polí

Všechny ostatní znaky buď mají jiný význam, nebo nemusí být na klávesnici k dispozici.

Chcete-li vytvořit mezinárodně použitelnou aplikaci, musíte objekty aplikace rozdělit na související části, tj. textová a netextová data. Vaše konvence pojmenování by měly být schopny obě tyto části rozlišit. Také byste měli být schopni rozeznávat textová data v různých jazycích. Toho můžete dosáhnout tak, že rozdělíte objekty do samostatných knihoven.

Scénář: konvence pojmenování knihoven

Konvence pojmenování vašich knihoven by mohla vypadat takto:

AAATTTLLL

kde: **AAA** je identifikace aplikace, **TTT** je typ objektů a **LLL** je kód jazyka.

Konvence pojmenování umožňuje, abyste seskupili všechny knihovny náležící k aplikaci, protože mají na začátku jedinečný identifikátor (AAA).

Druhá část (TTT) vám umožňuje rozeznávat různé druhy objektů:

Textová data

- zobrazovací soubory
- tiskové soubory

- soubory zpráv
- panely nápovědy
- uživatelský příkaz
- hodnoty závislé na kulturních zvyklostech
- databázové soubory s informacemi a specifikacemi citlivými na podporu národního jazyka
- programové moduly závislé na podpoře národního jazyka

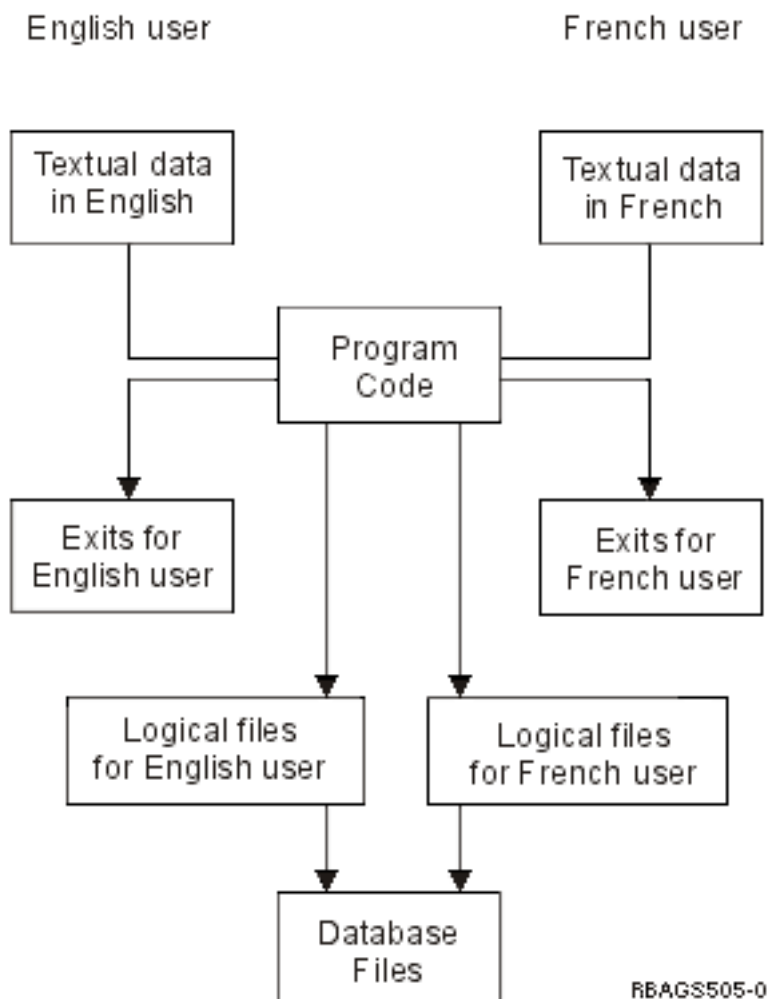
Netextová data

programy

Data databázové soubory

Třetí část (LLL) vám umožňuje zadat verzi národního jazyka pro všechny části textových dat. Tak můžete v různých knihovnách používat stejná jména pro objekty různých verzí národního jazyka. Váš program je schopen při zpracování úlohy používat různé objekty pouhým odpovídajícím přerovnáním seznamu knihoven.

Výchozí seznam knihoven může být převzat z popisu úlohy. Nový seznam knihoven můžete vypracovat uvedením seznamu knihoven v parametru INLLIBL příkazu CRTJOB (Vytvoření popisu úlohy) nebo v příkazu CHGJOB (Změna popisu úlohy) u existujícího popisu úlohy. Následující obrázek ukazuje takový příklad.



RBAGS505-0

Specifikace odkazů: Všechna vaše pole nejdříve definujte v souboru odkazů na pole v rámci vaší aplikace a dle potřeby na ně vytvořte odkazy: ve specifikacích databází, ve specifikacích souborů zařízení a v programech napsaných ve vyšších programovacích jazycích. Tato metoda vám umožní jednou definovat specifikace polí a pak je znovu používat. Pokud potřebujete rozlišovat mezi stejným polem v různých zdrojích, můžete je přejmenovat nebo kvalifikovat. Kdykoliv potřebujete změnit definici specifického pole, stačí pouze změnit atributy daného pole v souboru odkazů na pole a znovu vytvořit objekty. Změny se pak uplatní na všech místech, kde se dané pole používá.

Například:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
  A REF(jméno-ref-souboru-polí)
  A R record
  A field R line pos
nebo
  A field R line pos REFFLD(jméno-ref-pole)
```

Definice databáze: Definujete soubor k zadání určitých skutečností a tyto specifikace se pak použijí u databázových souborů. Následuje několik příkladů takových specifikací:

- Text popisu objektu souboru.
- Text komentáře (klíčové slovo TEXT) u popisu formátů záznamů a polí.
- Záhlaví sloupce (klíčové slovo COLHDG) u popisu polí.
- Formát data a času a příslušné oddělovače.
- Třídící posloupnost.
- Identifikátor jazyka.

Text popisu objektu ukazují všechny databázové nástroje, jako například produkt DB2^(R) UDB for iSeries SQL, iSeries Access a obslužný program datových souborů (DFU), na obrazovce výběr souboru.

Záhlaví sloupce ukazují databázové nástroje na obrazovce definice výstupního pole. Záhlaví sloupce používá také pomůcka pro návrh obrazovky (SDA) a obslužný program pro rozvržení sestav (RLU) jako navrhovaný text náznaku pole nebo záhlaví.

Řízení dat pracuje s poli typu datum a čas ve formátu zadaném v době vytvoření souboru, pokud vaše aplikace nebo databázový nástroj neprovádí konverzi, která by je převáděla podle vašeho požadavku nebo potřeb úlohy.

Když chcete všechny tyto informace předložit v souladu s jazykem a kulturními zvyklostmi uživatele, musíte vytvořit několik sad logických souborů v samostatných knihovnách. Spolu s přeloženým textem můžete specifikovat odlišný formát data a času nebo jinou třídící posloupnost a nechat řízení dat, aby provedlo konverze. Podobná technika se může použít také u numerických polí data (pokud nejsou pakovaná), pomocí funkce podřetězce (SST). Uživatel může k datům přistupovat jen prostřednictvím označených logických pohledů. Když definujete logické soubory s různou třídící posloupností, vyvarujte se použití jedinečného indexu u tabulky se sdílenou vahou. Ačkoli je to možné, jedinečný index brání použití kláves, které se liší jenom znaky se stejnou vahou.

Scénář v tématu Jména částí aplikace ukazuje příklad použití různých sad logických souborů u různých uživatelů.

Uživatelská rozhraní

Uživatelské rozhraní je součástí softwarového produktu, kterou vlastně vidí váš zákazník. Uživatelské rozhraní může zahrnovat grafickou úpravu obrazovek nebo tiskových výstupů, zobrazené nebo tištěné texty, příkazy, online nápovědu a zprávy. Je to také ta část softwarového produktu, kterou je třeba přeložit nebo změnit dle kulturních odlišností, pokud mají produkt používat uživatelé z jiných zemí.

Operační systém OS/400 poskytuje specifické softwarové funkce, které vám za účelem usnadnění překladu pomohou uspořádat text z vašeho uživatelského rozhraní a uložit ho do knihovny. Operační systém také nabízí správce uživatelského rozhraní (UIM), který zajišťuje konzistentní uživatelské rozhraní. Správce uživatelského rozhraní poskytuje komplexní podporu pro definici a spuštění panelů, jako jsou obrazovky a online nápověda.

Tato část vám poskytuje návod, podle kterého můžete postupovat při navrhování uživatelského rozhraní pro mezinárodní aplikace. Tyto postupy byste měli provést v počáteční fázi procesu navrhování. K dispozici jsou vám návody v tématech:

- Kontrolní seznam: návrh uživatelského rozhraní
- Návrh překladu textů
- Návrh kódu textových dat
- Správce uživatelského rozhraní
- Návrh zpráv programů
- Návrh menu
- Návrh příkazů
- Návrh závislý na kulturních zvyklostech
- Návrh zobrazovacích souborů
- Návrh a překlad tiskových souborů
- Návrh zdrojových souborů
- Návrh architektury CDRA (Character data representation architecture)
- Zpracování jazyků, které nemají podporu NLV

Kontrolní seznam: návrh uživatelského rozhraní: Při vytváření uživatelského rozhraní s globální podporou byste měli dodržovat určitá pravidla a postupy, jak je to znázorněno v následující tabulce:

Vyhovuje	Není použitelné	Pravidlo
		Použití grafického znaku pro účely řízení softwaru nesmí vylučovat použití stejného znaku v textu zpráv, menu, náznacích a ve vstupních či výstupních polích.
		Grafické symboly a ikony musí být přeložitelné.
		Za účelem snadné modifikace musí být části produktu závislé na jazyku izolovány od částí nezávislých na jazyku.
		Veškeré textové informace a informace pro řízení prezentace v rámci uživatelského rozhraní musí být izolovány od prováděného kódu.
		Musí být k dispozici dostatečný prostor pro rozšíření textu uživatelského rozhraní, ke kterému může dojít v důsledku překladu.
		Funkce závislé na délce pole zobrazení, případně na samotném umístění pole zobrazení, nesmí být navrženy takovým způsobem, aby na ně mělo vliv rozšíření textu uživatelského rozhraní.
		Musí existovat metoda, která by umožňovala identifikaci a sledování panelů a zpráv v průběhu zpracování překladu.
		Musí být povoleno, aby proměnné mohly přebírat libovolné umístění a pořadí v rámci pole zobrazení.
		Zprávy a další zobrazená slova nebo fráze musí být ucelené entity, které nesmí být tvořeny jednotlivými slovy nebo frázemi.
		Koncový uživatel musí mít možnost zadávat příkazy, klíčová slova a odpovědi bez ohledu na velká a malá písmena.
		Musí být možné zvolit formát data a času.

Vyhovuje	Není použitelné	Pravidlo
		Musí být možné zvolit nastavení numerické interpunkce.
		Musí být možné zvolit zaokrouhlování čísel a matematické formáty.
		Musí být možné definovat peněžní formáty.
		Musí být možné zvolit předvolený znak měny a její zkratku.
		Musí být možné zvolit umístění znaku měny.
		Musí být možné zvolit velikosti polí pro peněžní hodnoty.
		Musí být možné zvolit systém měrných jednotek.
		Nemělo by se předpokládat, že abeceda tvořená malými písmeny je neměnná (invariantní).
		Mělo by být možné definovat speciální znaky, včetně interpunkčních znamének. Tyto znaky by neměly být závislé na programech.
		Textové moduly uživatelského rozhraní by měly být sbaleny odděleně od prováděného kódu.
		Textové moduly uživatelského rozhraní pro systémy SBCS by měly být zaváděny odděleně od prováděného kódu.
		K označování proměnných a vstupních polí by se měly v rámci celého produktu používat konzistentní konvence.
		Místo čísel by neměla být používána slova.
		V rámci celého produktu by se měla používat jednotná terminologie pro texty uživatelského rozhraní.
		Měli byste se vyhnout zkratkám.
		Neměl by se používat odborný slang ani humor.
		Ochranné známky by měly být označeny a vysvětleny.
		Neměla by se používat nejednoznačná slova.
		V textech uživatelského rozhraní by se měl používat správný sloh a struktura vět.
		Neměly by se používat negativní otázky.

Návrh překladu textů: Následující informace obsahují některé obecné rady, které vám pomohou zjednodušit překlad vašich textových materiálů.

Oddělení textových dat od prováděného kódu

Abyste usnadnili překlad a zabránili jste překladu prováděného kódu, měli byste veškerá textová data oddělit od prováděného kódu. Je potřeba pouze jedna sada prováděného kódu, ale může se uskutečňovat mnoho překladů textových dat.

Poskytnutí prostoru pro rozšíření

Místo potřebné pro překlad textu z jednoho jazyka do druhého se liší podle jazyka. Chcete-li zajistit, aby překládaná verze zachovávala smysl a byla nadále použitelná, poskytněte dostatečný prezentační prostor pro rozšíření textových dat. Následující tabulka uvádí doporučený prostor pro rozšíření pro uživatelská rozhraní navržená s využitím americké angličtiny.

Počet znaků v textu	Požadovaný dodatečný prostor
Do 10	100 až 200%
11 až 20	80 až 100%

Počet znaků v textu	Požadovaný dodatečný prostor
21 až 30	60 až 80%
31 až 50	40 až 60%
51 až 70	31 až 40%
Nad 70	30%

Proměnné umístění objektu na obrazovce

Jelikož umístění jednoho prvku obrazovky často závisí na umístění a velikosti ostatních prvků, mohou některé z prvků v překládané verzi obrazovky vyžadovat přemístění. Bez ohledu na toto přemístění musí program nadále správně reagovat.

Pružné pořadí proměnných

Za účelem poskytování dynamických informací se ve zprávách obvykle využívají substituční proměnné. Každý mluvený jazyk však má svou syntaxi (pořadí uspořádání jednotlivých částí řeči). Když se zpráva překládá do jiného jazyka, může dojít ke změně místa a pořadí těchto substitučních proměnných v souladu s požadavky jazyka, do nějž se text překládá.

Úplné entity textových dat

Pokud konečná podoba neměnného textu závisí na sestavení různých částí, může být nepřeložitelná. Je to způsobeno tím, že překladatel nemusí vědět, který tvar slova použít, nebo tím, že v jiném jazyku neexistuje žádná vhodná kombinace jednotlivých částí.

Například byste měli záhlaví sloupců pro obrazovky definovat jako úplné entity. Záhlaví sloupců byste neměli definovat jako kombinaci slov nebo částí slov. Předpokládejme, že vytváříte aplikaci pro rozvržení úloh od pondělí do pátku. Aplikaci píšete ve francouzštině. Rozhodli jste se, že záhlaví sloupců pro sestavy a obrazovky vytvoříte zkombinováním první části jména dne a konstanty DI. V rámci celé aplikace jsou záhlaví sloupců a sestav sestavovány tímto způsobem:

První část jména dne:	V kombinaci s:	Výsledek:
LUN	DI	LUNDI
MAR	DI	MARDI
MERCRE	DI	MERCREDI
JEU	DI	JEUDI
VENDRE	DI	VENDREDI

Když pak vaši aplikaci překládáte z francouzštiny do němčiny, nelze kombinací dvou částí vytvořit názvy dnů: MONTAG, DIENSTAG, MITTWOCH, DONNERSTAG a FREITAG.

Práce s příkazy, odpověďmi a klíčovými slovy jako s textovými daty

Příkazy, odpovědi a klíčová slova by měla být přeložena do jazyka, kterým uživatel běžně hovoří. Například anglická aplikace byla přeložena do němčiny. Pokud odpovědi nadále zůstaly v angličtině jako Yes a No, mohou se němečtí uživatelé cítit zmateně a nepříjemně, neboť oni znají odpovědi Ja a Nein.

Vyjádření veškerého textu pokud možno jednoduše a jasně

- Používejte jednoduché fráze a věty a vyhýbejte se složitým souvětím. Jednoduchá slova umožňují snadný překlad.
- V rámci celého produktu udržujte jednotnou terminologii.

Pokud nedodržíte konzistentní terminologii, stráví překladatelé spoustu času snahou určit odpovídající slovo, které mají použít pro překlad.

- Do vašich informací vkládejte poznámky pro překladatele, které přibližují správné použití slova, abyste tak předešli nedorozumění.
- Nepoužívejte zkratky.
Pravidla pro použití zkratk se v jednotlivých jazycích liší. Zkratky slov mohou vést k tomu, že překladatel ani koncový uživatel text nepochopí.
- Vyhněte se odbornému slangu a humoru.
Odborný slang i humor jsou specifické pro konkrétní jazyk a těžko se překládají do jiného jazyka.
- Nepoužívejte záporné otázky.
Záporné otázky uživatel často chápe nesprávně. Když pokládáte otázky, formulujte je pozitivně.

Návrh kódu textových dat: Aplikační obrazovky, specifikace tiskových souborů a uživatelem vytvořené příkazy obvykle obsahují velké množství konstantního textu. Aplikační obrazovky, specifikace tiskových souborů a uživatelem vytvořené příkazy také obsahují vstupní a výstupní pole, jako například záhlaví, náznaky, řádky instrukcí a popisy funkčních kláves.

K zadání, uložení a použití konstantního textu můžete použít různé techniky. Každou techniku můžete použít pro specifický typ komponent textových dat. Každá technika má své výhody a nevýhody. Následující témata ukazují, jak jednotlivé techniky fungují, a popisují, které techniky můžete u různých komponent použít:

- Okamžité přiřazení zprávy
- Pozdější přiřazení zprávy
- Přímé kódování jako nepojmenované výstupní pole
- Text uložený v databázových souborech

Okamžité připojení zprávy: Text může být uložen mimo zdrojový kód v samostatném souboru zpráv, ale je svázán do objektu při jeho vytvoření. Tuto techniku je možné použít pro:

Zobrazovací soubory

Konstanty, jako například titulky, řádky instrukcí, definice voleb, záhlaví, náznaky polí, popisy příkazových kláves.

Tiskové soubory

Konstanty, jako například titulky, záhlaví, popisy součtových řádek.

Uživatelské příkazy

Popisy náznaků u definic příkazů.

U souborů zařízení (obrazovka a tiskárna) se na zprávu odvolává klíčové slovo MSGCON (konstanta zprávy) ve specifikaci zdroje v DDS.

Například:

```
A line pos  MSGCON(length message-ID [*libl/]jméno-souboru-zpráv)
                ^
                zahrnuje prostor pro rozšíření
```

U uživatelských příkazů se u klíčového slova PROMPT zadává místo literálu identifikátor zprávy xxxnnnn. Odkaz na soubor zpráv je v příkazu CRTCMD (Vytvoření příkazu).

Například:

```
CMD          PROMPT( xxxnnnn)
```

Na *jméno-souboru-zpráv* se odkazuje ve zdrojovém souboru následující příkaz:

```
CRTCMD CMD(jméno-příkazu) PGM(jméno-knihovny/jméno-programu) +
      PMTFILE([*libl/]jméno-souboru-zpráv)
```

Dříve než je možné objekt vytvořit, musíte zadat popis zprávy do uvedeného souboru zpráv. Popis zprávy zadejte příkazem ADDMSGD (Přidání popisu zprávy).

Například:

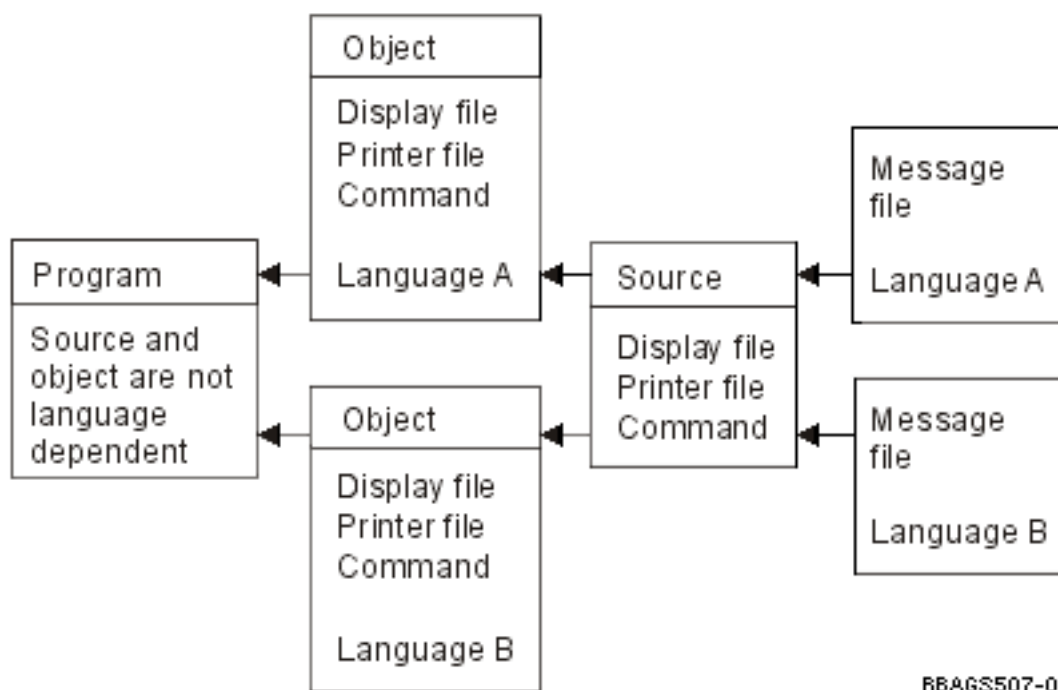
```
ADDMSGD MSGID(xxxxxxxx) MSGF(jméno-knihovny/jméno-souboru-zprāv) +  
MSG('Text')
```

kde xxxnnnn je identifikátor zprávy.

Tato technika vám umožňuje vytvořit libovolný počet objektů v různých jazycích a uložit je do různých knihoven, které budou používat stejný zdrojový kód a kterým se prostě jen v době vytváření objektu přiřadí jiný soubor zpráv.

Soubor zpráv je potřebný pouze při vytvoření objektu. Z klíčového slova MSGCON zvažte uvedení délky odpovídající různým jazykům. Pak poskytněte informace o délce překladateli.

Následující obrázek ukazuje, jak probíhá okamžité připojení zprávy:



Při vytváření souboru si můžete vybrat odpovídající textová data té jazykové verze, se kterou chcete pracovat tak, že sestavíte seznam knihoven se specifickou knihovnou obsahující textová data a s knihovnou programů.

Pozdější připojení zprávy: Text lze uložit externě mimo zdrojový kód DDS do popisu zprávy. Ten se pak připojí pouze k formátu zobrazení v době běhu programu.

Tuto techniku můžete použít pro:

Pouze zobrazovací soubory

Konstanty, jako například titulky, řádky instrukcí, definice voleb, záhlaví, náznaky polí, popisy příkazových kláves (klíčové slovo MSGID).

Předvolené hodnoty vstupních polí (klíčové slovo MSGID).

Specifikace ověření platnosti pole (klíčové slovo CHKMSGID).

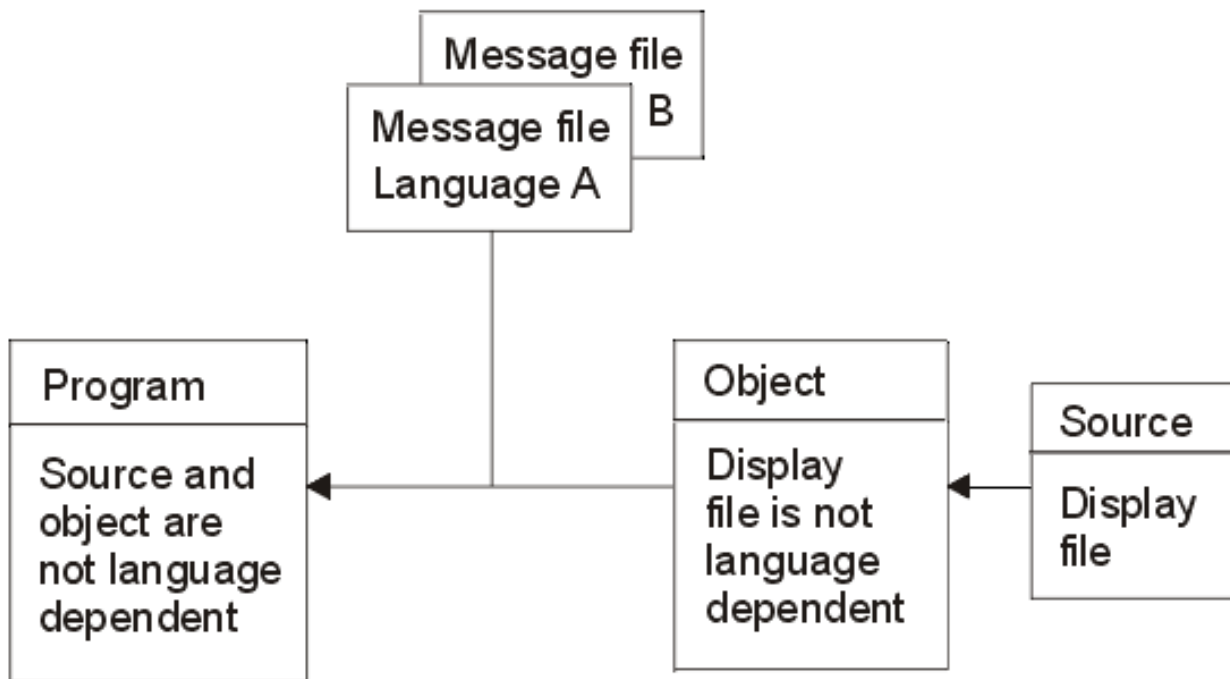
Chybové zprávy (klíčová slova ERRMSGID a SFLMSGID).

V DDS zobrazovacího souboru je zpráva zadána prostřednictvím klíčového slova MSGID (Identifikátor zprávy). Zpráva musí být zadána do uvedeného souboru zpráv pomocí příkazu ADDMSGD (Přidání popisu zprávy).

Například:

```
A FLD-name length line pos MSGID(message-ID [*libl/]message-filename)
      ^
      zahrnuje prostor pro rozšíření
ADDMSGD MSGID(xxxxxxx) MSGF(library-name/message-file-name) +
MSG('Text')
```

Tato technika umožňuje vytvářet libovolné množství souborů zpráv v různých jazycích a v různých knihovnách s jedním zdrojovým kódem DDS a jedním objektem typu zobrazovací soubor. V době provádění přiřadíte jiný soubor zpráv tím, že adekvátně nastavíte seznam knihoven. Následující obrázek ukazuje takový příklad.



RBAGS508-0

Poznámka: Tato metoda vyžaduje, aby aplikace prováděla veškerou editaci na základě kulturních konvencí.

Přímé kódování jako nepojmenované výstupní pole: Nejběžnějším způsobem, jak definovat konstantní text, je zadat text přímo do zdrojového kódu jako literál. I když jde o nejběžnější metodu definování konstantního textu, je nejobtížnější pro překlad. Vyvarujte se použití této metody, i kdybyste kódovali aplikaci, u které se překlad neplánuje.

Jestliže kódujete aplikaci, která se nebude překládat, mohli byste tuto techniku použít pro:

Zobrazovací soubory

Konstanty, jako například titulky, řádky instrukcí, definice voleb, záhlaví, náznaky polí, popisy příkazových kláves.

Předvolené hodnoty vstupních polí (klíčové slovo DFT).

Chybové zprávy (klíčové slovo ERRMSG/SFLMSG).

Tiskové soubory

Konstanty, jako například titulky, záhlaví, popisy součtových řádek.

Uživatelské příkazy

Popisy náznaků u povelů příkazů.

U souborů zařízení uvádějte text jako nepojmenované pole a označte počáteční řádku a sloupec vlastního konstantního textu.

Například:

```
A line pos      'Text . . . . . : '
```

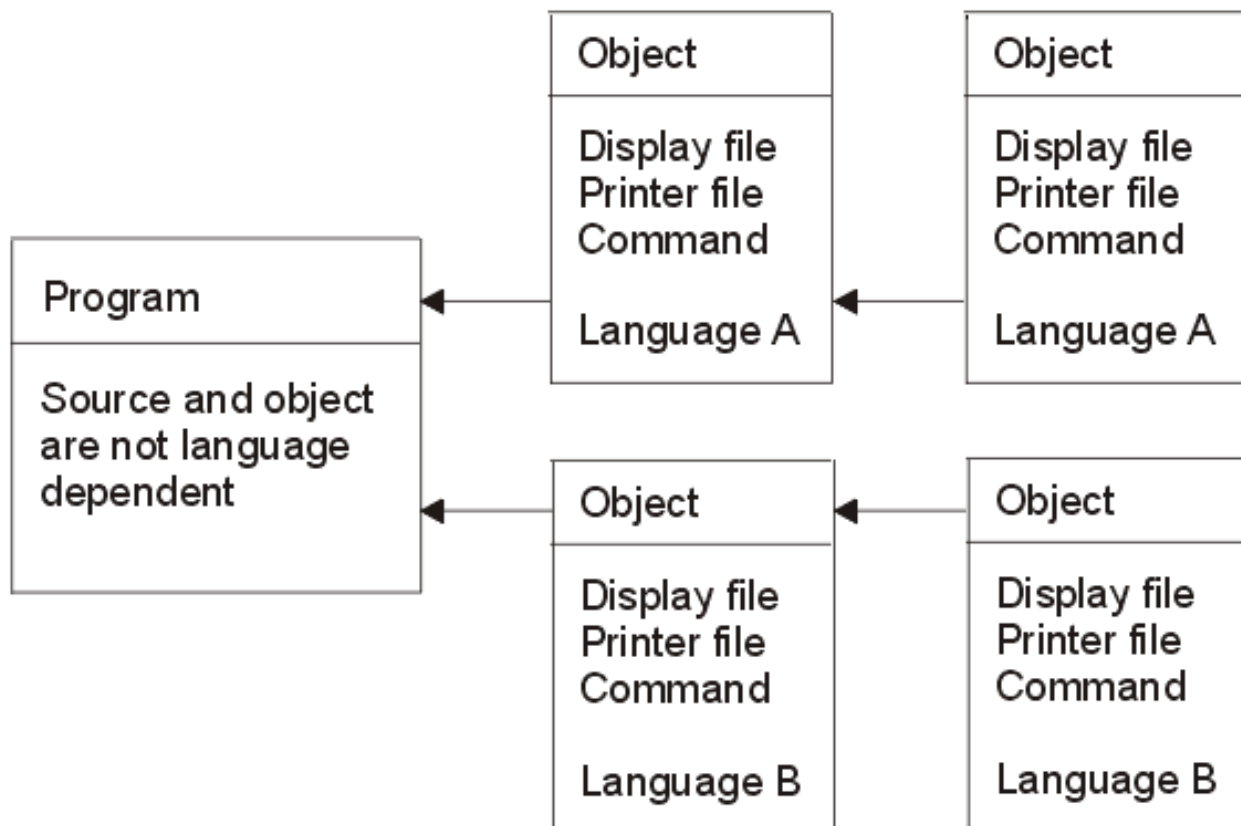
Podobné pravidlo platí u příkazů vytvořených uživatelem. Definujte text přímo u klíčových slov zdrojových příkazů.

Například:

```
CMD             PROMPT('   Popis příkazu      ')
```

Když definujete text přímo v klíčových slovech, standardizujte velikost různých prvků ve velkém literálu a neuvádějte tyto jednotlivé malé prvky jako samostatná slova. Zdrojový kód je tak čitelnější a je možné jej pružněji překládat.

Uvažte, že prostor potřebný pro vysvětlující text se může jazyk od jazyka lišit. Chcete-li mít dostatek prostoru pro překlad, nezapomeňte jej hned na začátku rezervovat. Je nutné překládat zdrojové členy a vytvářet objekty pro různé jazyky, jak to ukazuje následující obrázek:



RBAGS512-0

Každá verze národního jazyka má jednu sadu programů, ale může mít několik sad zdrojových členů a datových objektů. Když se aplikace spustí, můžete si vybrat odpovídající textová data té jazykové verze, se kterou chcete pracovat. To je možné provést, když v systémové části seznamu knihoven nastavíte specifickou knihovnu, která obsahuje jak textová data, tak knihovnu programů.

Text uložený v databázových souborech: Text lze uložit externě mimo zdrojový kód do databázového souboru, načíst aplikačním programem a pak v době běhu programu přesunout do tiskového formátu nebo do formátu zobrazení. Namísto kódování konstant v DDS můžete specifikovat výstupní pole, které může vyplňovat program. Uvažte zadání vhodné délky výstupních polí pro různé jazyky a jejich zpřístupnění překladatelům.

Tuto techniku můžete použít pro:

Zobrazovací soubory

- Všechny konstantní texty.
- Předvolené hodnoty ve vstupních polích.
- Chybové zprávy.

Tiskové soubory

- Všechny konstantní texty.

Programy.

- Veškeré konstanty, např. porovnávací hodnoty, snímané znaky a tabulky.

Tato technika umožňuje vytvářet libovolné množství databázových souborů v různých jazycích a v různých knihovnách, ale pouze s jedním zdrojovým kódem DDS a jedním objektem typu zobrazovací soubor. V době provádění přiřadíte odpovídající databázový soubor tím, že adekvátně nastavíte seznam knihoven.

Poznámka: Tato metoda vyžaduje, aby aplikace prováděla veškerou editaci na základě kulturních konvencí.

Správce uživatelského rozhraní: Správce uživatelského rozhraní OS/400, neboli UIM (User Interface Manager), je částí systému, která umožňuje definovat panely a dialogová okna pro vaše aplikace. UIM poskytuje následující podporu:

- Jazyk na bázi příznaků pro popis dat a panelů.
- Kompilátor pro tvorbu objektů typu skupina panelů a objektů typu menu s využitím jazyka na bázi příznaků.
- Sada rozhraní API, která slouží jako objekty typu skupina panelů k zobrazení a tisku panelů.

UIM také poskytuje následující funkce:

- Dialogové příkazy pro správu obrazovek.
- Kontextovou online nápovědu.
- Rozevírací okna.
- Pruhy nabídky.
- Příkazovou řádku pro zadávání CL příkazů.
- Přizpůsobení obsahu panelu různým uživatelům a různým prostředím.
- Rychlé cesty přes sítě menu.
- Jazyky DBCS (dvoubajtová znaková sada).
- Podporu jazyků BIDI (obousměrné).

UIM podporuje běžné typy panelů, jako jsou např. menu, informativní obrazovky, obrazovky seznamů a vstupní obrazovky. Pokud jsou všechny typy obrazovek a rozhraní konzistentní, zvyknou si uživatelé rychleji na práci s novými aplikacemi.

Aplikace UIM mohou společně existovat s dalšími otevřenými zobrazovacími soubory, které nejsou řízeny UIM, a mohou s nimi sdílet obrazovku žadatele. Panel UIM se však nemůže na obrazovce objevit současně s formátem záznamu definovaného DDS. Když panel UIM nahradí panel DDS či naopak, pozastaví systém operace s jedním souborem nebo skupinou panelů a dle potřeby obnoví obrazovku.

Další informace o správci uživatelského rozhraní poskytují tato témata:

- Návrh online nápovědy
- Příznaky indexového vyhledávání
- Indexové vyhledávání a DBCS

Návrh online nápovědy: Při definici online nápovědy můžete využít:

Skupiny panelů.

Objekty, do nichž je vkládán zdroj UIM.

Záznamy.

Sadu klíčových slov DDS obsaženou v členu zdrojového souboru.

Pokud k definici online nápovědy použijete správce uživatelského rozhraní, neboli UIM (User interface manager), jsou skupiny panelů definovány buď místo DDS, nebo v zobrazovacím souboru. V obou případech musí být kódování dat, která se mají zobrazovat, indikováno hodnotou CHRID v zobrazovacím souboru nebo skupině panelů.

Skupina panelů je objekt, který slouží k uchování informací nápovědy. Operační systém OS/400 používá *PNLGRP jako identifikátor pro typ objektu, který obsahuje množinu informací nápovědy.

Návod: online nápověda

Při definování informací online nápovědy, které se mají překládat do verzí národních jazyků, berte v úvahu tyto skutečnosti týkající se skupin panelů a záznamů:

- Pro záznamy není k dispozici textové zpracování (ačkoliv existují funkce pro kontrolu a zalamování textu v rámci systémových rozhraní API).
- Různé zprávy a skupiny panelů operačního systému OS/400 určují jazykové zvyklosti a překlady. Verze národního jazyka pro operační systém OS/400 nemají k dispozici všechny země. Ne všechny verze národního jazyka jsou zcela přeloženy a mnoho částí v nich zůstává nadále v angličtině. Zprávy a skupiny panelů, které nejsou přeloženy, neodrážejí národní jazykové a kulturní zvyklosti. Příklad přeloženého panelu, v němž určitá část zůstala v angličtině, neboť nebyly přeloženy všechny části verze národního jazyka, uvádí téma *Návrh příkazů*.
- Umožněte rozšíření překladů.

Návod: návrh online nápovědy DDS

Když je v jednom systému instalováno více jazyků, jsou dokumenty nápovědy uloženy v různých pořadačích. Pro každý jazyk v systému je nutné zkopírovat, změnit a znovu zkompileovat zdrojový soubor DDS.

Příznaky indexového vyhledávání: Skupiny panelů nápovědy mohou obsahovat moduly pro indexové vyhledávání. Indexové vyhledávání je doplňkem informací nápovědy, které jsou k dispozici pro každou obrazovku. Chcete-li informace ve skupinách panelů nápovědy použít pro funkci indexového vyhledávání, musíte přidat příslušné příznaky UIM do vašich modulů nápovědy.

Uživatelé mohou funkci indexového vyhledávání aktivovat z libovolné nápovědy k obrazovce, v níž je uvedeno, že je funkce indexového vyhledávání k dispozici.

Příznak ISCH

Příznak ISCH definuje název tématu v indexu (rejstříku) a specifikuje kořenová slova, která slouží jako odkazy mezi tématem a vyhledávanými slovy (synonyma) zadanými uživatelem. Příznak je hned vedle příznaku HELP, na který se odkazuje. V rámci jednoho modulu nápovědy může být pouze jeden příznak ISCH.

Pro každý příznak ISCH může existovat několik řádek kořenových slov za předpokladu, že jejich počet nepřesáhne 50. Jestliže použijete více než jednu řádku kořenových slov, musí být na začátku každé další řádky zopakováno ROOTS=.

```
:PNLGRP.  
:HELP name=entry1.  
:ISCH ROOTS='root1 root2 root3 root4 root5'  
  ROOTS='root6 root7 root8 root9 root10'  
  ROOTS='root11 root12 root13 ... root50'.  
Název prvního tématu
```

```
Toto je první modul indexového vyhledávání v této skupině panelů.  
:EHELP.  
:EPNLGRP.
```

Kořenová slova na všech řádkách musí být ohraničena apostrofy a tečka musí být umístěna pouze na konci poslední řádky kořenových slov. V příznaku ISCH následuje za tečkou název tématu, který může být umístěn na řádce následující po tečce.

Příznak ISCHSYN

Příznak ISCHSYN definuje slova (synonyma), která, pokud je zadá uživatel, odpovídají konkrétnímu kořenovému slovu. Jestliže slovo zadané uživatelem je synonymum kořenového slova, vyhledají se jako odpovídající všechna témata, jejichž příznak ISCH obsahuje tento kořen.

Pokud chcete, aby se slovo používané jako kořenové slovo použilo také jako synonymum, musíte toto slovo zahrnout jako synonymum do příznaku ISCHSYN. Například:

```
:ISCHSYN ROOT='oceán'.oceán voda moře
```

Synonyma v příznaku ISCHSYN musí být zadána na jednu řádku a pro každé kořenové slovo musí existovat alespoň jeden příznak ISCHSYN. Jestliže je potřeba více řádek, je možné pro stejné kořenové slovo zadat více příznaků ISCHSYN.

UIM nerozlišuje mezi synonymy zadanými velkými nebo malými písmeny, či jejich kombinací. Z toho důvodu není nutné opakovat synonyma, aby se pokryly všechny možné velikosti písmen.

Pro synonyma můžete použít alfabetické nebo numerické znaky. Použití následujících znaků (včetně jejich hexadecimálních ekvivalentů) však není v synonymech ani v jejich částech dovoleno:

- . (tečka)
- ((levá kulatá závorka)
-) (pravá kulatá závorka)
- ; (středník)
- , (čárka)
- ? (otazník)
- : (dvojtečka)

Příznaky ISCHSYN lze umístit kamkoliv ve skupině panelů, ale v zájmu snazší údržby a překladu je raději všechny umístit na jedno místo (např. na začátek vaší skupiny panelů nebo do objektu typu skupina panelů, který obsahuje pouze příznaky ISCHSYN).

Příklad: použití příznaků ISCH a ISCHSYN

Následující příklad ukazuje příznaky ISCHSYN a příznaky ISCH, které je používají:

```
:PNLGRP.  
:ISCHSYN ROOT='oceán'.oceán voda moře  
:ISCHSYN ROOT='jezero'.jezero voda rybník  
:ISCHSYN ROOT='definice'.definice definovat popis co  
:ISCHSYN ROOT='definice'.přehled koncepce informace vysvětlení  
:HELP name='defocean'.  
:ISCH ROOTS='definice oceán'.  
Definice oceánu
```

Oceán je pět obrovských těles slané vody, které společně pokrývají přibližně tři čtvrtiny světa.

```
:EHELP.  
:HELP name='defjezero'.  
:ISCH ROOTS='definice jezero'.  
Definice jezera
```

Jezero je těleso stojaté vody, která je ohraničena břehy.

```
:EHELP.  
:EPNLGRP.
```

Indexové vyhledávání a DBCS: Funkci indexového vyhledávání lze použít buď s daty DBCS (podpora dvoubajtových znaků), nebo s daty SBCS (podpora jednobajtových znaků). Když použijete data DBCS, musí být zařízení, z kterého požadavek pochází, schopné zadávání a prezentace dat v DBCS. Objekt, který obsahuje data indexového vyhledávání, má označení, že obsahuje data DBCS. Systém určí, zda je zařízení schopné zpracovávat data DBCS.

Když jsou data připravována pro formát DBCS a s těmito daty se používá funkce indexového vyhledávání, zvažte tyto skutečnosti:

- Když se data indexového vyhledávání připravují pro systém DBCS, musí být synonyma zadaná v příznaku ISCHSYN v režimu dvoubajtových znaků. To znamená, že první bajt po příznaku musí být znak Shift-out a poslední bajt dat musí být znak Shift-in. Systém nekonvertuje data v příznaku ISCHSYN na data DBCS.
- Slova musí být oddělena jednobajtovou mezerou. Za účelem vytvoření slova může být zkombinováno od 1 do 19 dvoubajtových znaků. Mezi nimi mohou být umístěny znaky Shift-out a Shift-in, ale ty jsou při indexovém vyhledávání ignorovány.
- Slova, která se používají ke spojení příznaků ISCH a ISCHSYN (atribut ROOTS příznaku ISCH a atribut ROOT příznaku ISCHSYN) musí být totožné a neměla by být zadávána v DBCS.
- Vyhledávaná slova mohou být zadána jak v režimu SBCS, tak v DBCS. Za účelem oddělení slov je možné zadat jednobajtové mezery.

Když se vyhledávaná slova zobrazí na obrazovce, zobrazí se dvoubajtová reprezentace znaků (tedy znaky, které byly ve skutečnosti použity ve vyhledávání). Provádí se speciální zpracování, aby indexové vyhledávání nerozlišovalo velká a malá písmena. Vyhledávaná slova z příznaku ISCHSYN se zkonvertují na velká písmena s využitím tabulky pro kódovou stránku, která je uvedena v atributu TXTCHRID příznaku PNLGRP. Jestliže jsou vyhledávaná slova DBCS, nejsou konvertována na velká písmena. Se znaky Shift-out a Shift-in se během analýzy zachází jako s mezerami, úvodní a koncové nuly se odstraní. Všechna slova SBCS se zkonvertují na velká písmena s využitím převodní tabulky pro kódovou stránku v popisu zařízení.

Návrh zpráv programů: V operačním systému OS/400 existují dva druhy zpráv: předdefinované nebo okamžité. Při návrhu a kódování vašich zpráv zvažte tyto skutečnosti:

- Nepoužívejte okamžité zprávy. Vytváří je odesílatel nebo program v okamžiku odesílání a nejsou uloženy v souboru zpráv. Z toho důvodu nemohou být přeloženy překladatelem.
- Používejte předdefinované popisy zpráv, které mohou:
 - Existovat mimo program, který je používá.
 - Být uloženy v souboru zpráv.
- Neudávejte maximální velikost pro soubor zpráv. Když se soubor zpráv naplní, není možné změnit jeho velikost. Je pak nutné vytvořit jiný soubor zpráv a znovu přidat popis zprávy.

K vytvoření souboru zpráv pro uchování předdefinovaného popisu zprávy slouží příkaz CRTMSGF (Vytvoření souboru zpráv). Obsah předdefinovaného popisu zprávy se do souboru zpráv uloží pomocí příkazu ADDMSGD (Přidání popisu zprávy). Další informace uvádí téma Jazyk CL.

- Náhradní proměnné používejte s opatrností. V různých jazycích existuje odlišné pořadí náhradních proměnných. Například v anglické zprávě:

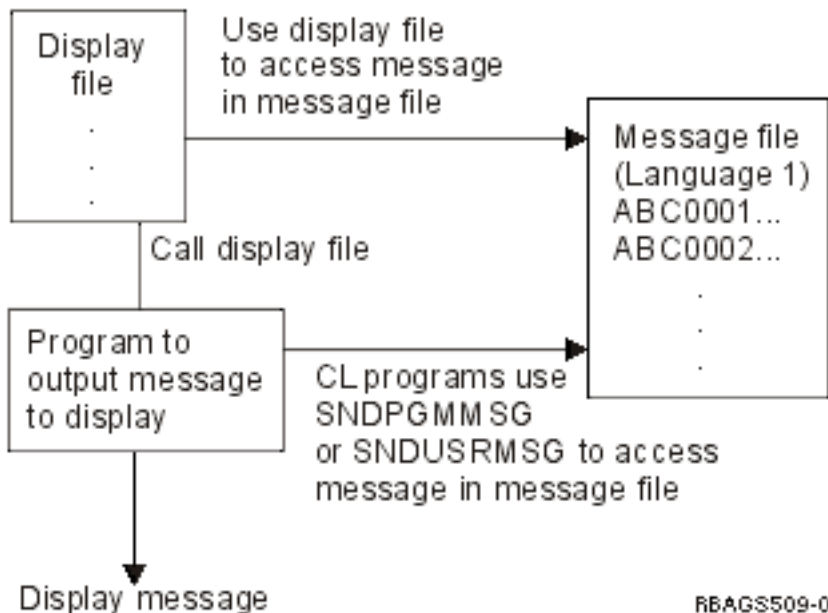
```
File &1 in Library &2 not found.
```

jsou &1; a &2; náhradní proměnné. Tyto proměnné se mohou v různých jazycích objevit na rozdílných pozicích.

- Při vašem návrhu a kódování se snažte, aby byly kódy odpovědí pochopitelné i v jiných jazycích. Například,

```
Angličtina   Y = Yes (ano)
Dánština    J = Ja (ano)
```

Následující obrázek představuje vytvoření zpráv v různých verzích národních jazyků z tiskových souborů.



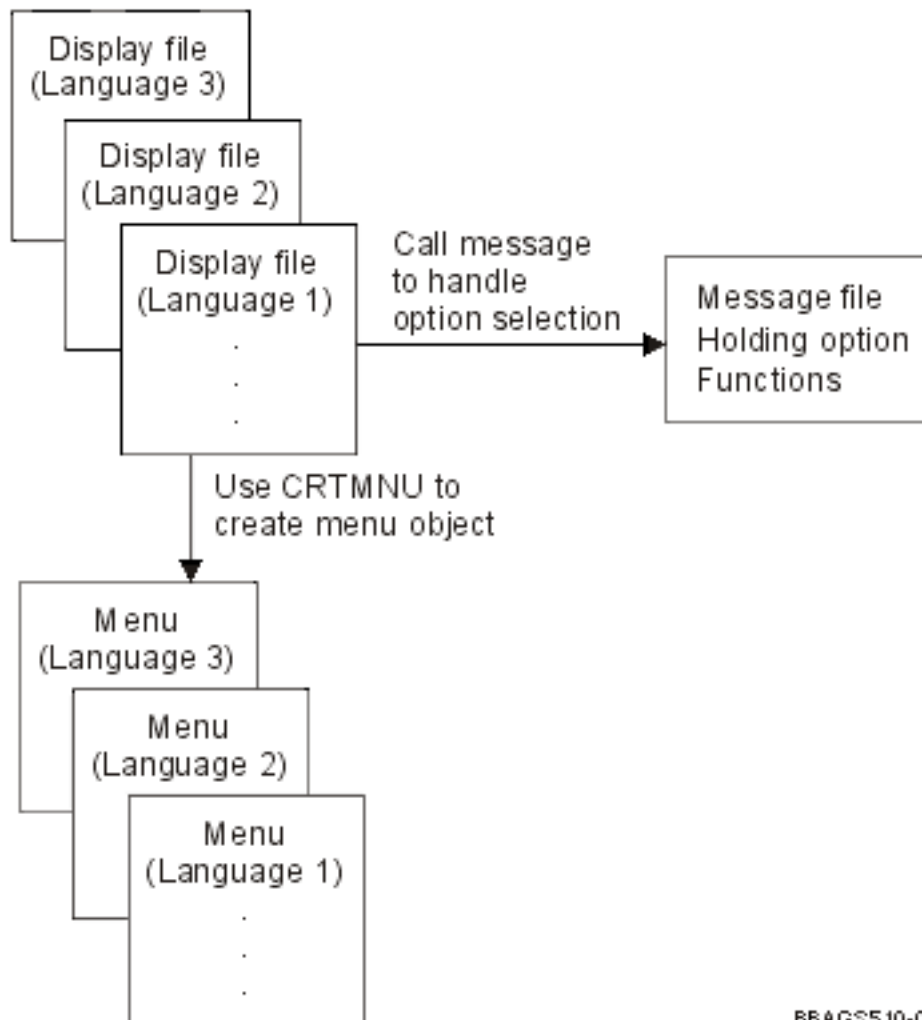
Program může mít přímý přístup k souboru zpráv za účelem získání zpráv programu nebo k němu může za stejným účelem přistupovat nepřímo prostřednictvím zobrazovacích souborů. Další informace o souborech zpráv uvádí téma Podpora CCSID u zpráv.

Návrh menu: V operačním systému OS/400 můžete navrhnout svá vlastní menu. Existují tři typy uživatelem definovaných menu: menu typu zobrazovací soubor, menu (reference) UIM a programová menu.

Při práci s aplikačním systémem se uživatelé musí zabývat řadou menu a obrazovek. Když se aplikace překládá z jednoho jazyka do druhého, pochází značná část textových dat určených k překladu právě z menu.

Menu typu zobrazovací soubor

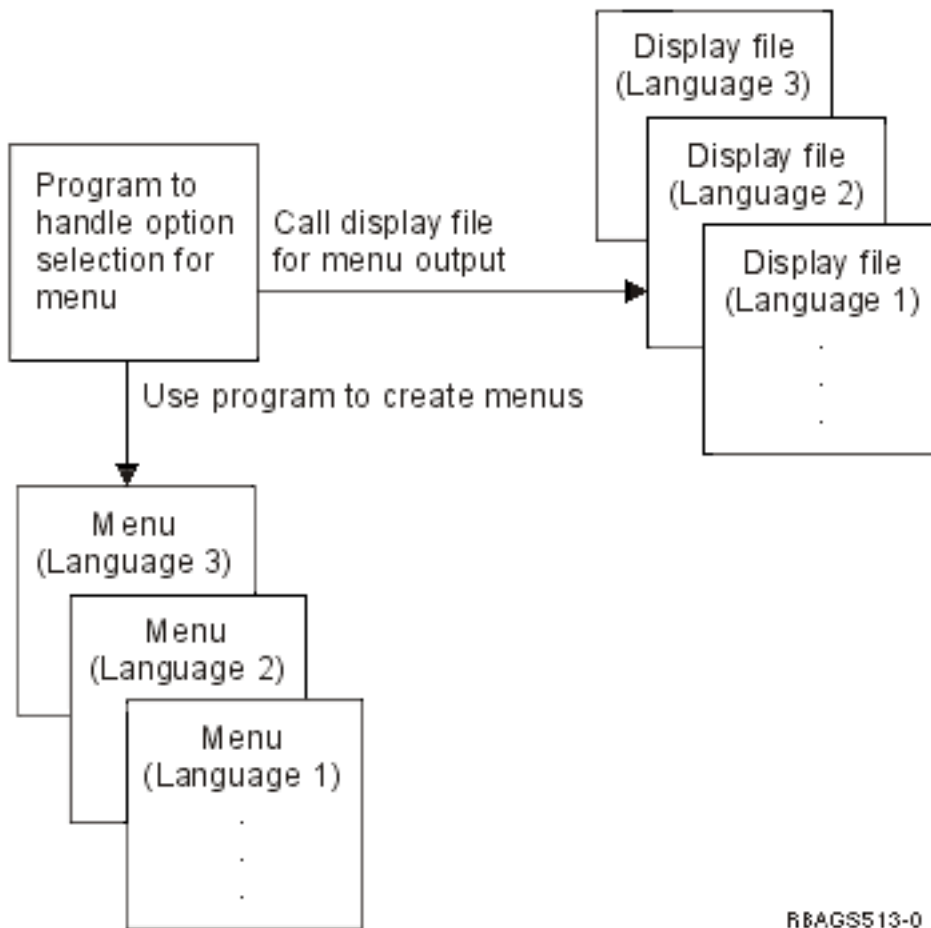
Menu typu zobrazovací soubor používá k zobrazení formátu menu obrazovku definovanou pomocí DDS. Funkce menu jsou řízeny objektem typu menu, který obsahuje příkazy pro spuštění jednotlivých voleb menu. Následující obrázek znázorňuje, jakým způsobem se vytvářejí menu typu zobrazovací soubor pro různé verze národního jazyka.



RBAGS510-0

Programové menu

Programové menu používá programy k prezentaci formátu menu (definované pomocí DDS) a k zajištění funkcí nezbytných pro spuštění jednotlivých voleb menu. Následující obrázek znázorňuje, jakým způsobem se vytvářejí programová menu v různých verzích národních jazyků.



R6AGS513-0

Překlad menu

Za účelem zajištění snadného překladu vašich menu do verzí národních jazyků:

- Uchovávejte textové informace obsažené v menu externě a to tak, že jsou konstantní texty uloženy jako externě definované popisy zpráv v souboru zpráv a tento text se začleňuje do souboru menu až v okamžiku spuštění programu.
- Nezapomeňte ponechat nějaký prostor pro rozšíření, který je nezbytný při překladu menu z jednoho jazyka do druhého. Prostor pro rozšíření překladu zakomponujte už při návrhu vašich menu.
- Zohledněte národní zvyklosti, pokud se v menu zobrazuje datum, čas nebo editovaná pole.
- Jako pole voleb určená pro výběr používejte číslice 0 až 9 místo velkých a malých anglických písmen (A až Z). Číselné znaky jsou v jiných jazycích běžnější.

Návrh příkazů: Operační systém OS/400 umožňuje, aby uživatelé vytvářeli a definovali vlastní příkazy. Chcete-li příkaz vytvořit, musíte jej nejprve definovat pomocí příkazů určených k této definici. Potom zpracujte příkazy definice příkazem CRTCMD (Vytvoření příkazu) a vytvořte objekt definice příkazu.

Při definování a vytváření příkazu berte v úvahu následující skutečnosti:

- Použijte skupiny panelů online nápovědy, které podávají informace o příkazu. Informace o panelech nápovědy a verzi národního jazyka najdete v tématu "Uživatelská rozhraní" na stránce 15.
- U klíčového slova PROMPT v CL příkazech CMD, PARM, ELEM a QUAL použijte místo textu v literálech identifikátory zprávy.

- Přeložte text, který je zobrazen vpravo od řádky náznaku každého parametru na náznakové obrazovce. Tento text je uveden v parametru CHOICE příkazů definice PARM, aby si vzhled zobrazené příkazové řádky uchoval svou souvislost.
- Zkompilujte text příkazové řádky do samostatných verzí objektu definice příkazu pro každý národní jazyk. Před vytvořením příkazu použijte příkaz CHGSYSLIBL (Změna systémového seznamu knihoven), abyste získali text náznaku ve verzi národního jazyka ze správné knihovny verzí národního jazyka.
- Funkční klávesy náznakové obrazovky poskytuje operační systém OS/400. Je-li verze národního jazyka (NLV) operačního systému OS/400 odlišná od verze příkazu, objevily by se na náznakové obrazovce dva různé jazyky. Například při překladu anglické obrazovky do němčiny by se na náznakové obrazovce objevila jak angličtina, tak němčina.

Další informace o vytváření a definování příkazů najdete v tématu Jazyk CL.

Návrh závislý na kulturních zvyklostech: Různé země mají různé standardy, které je nutné při vývoji aplikace umožňující podporu národního jazyka, brát v úvahu. Tyto na kulturní zvyklosti citlivé informace musí být umístěny mimo program stejným způsobem, jako je tomu u textových dat.

Mnoho jazyků má znaky (jako například běžně používané samohlásky podstatné pro správný pravopis slova) mimo abecedy A-Z, o nichž je nutné pro účely porovnání uvažovat.

Prostřednictvím systémových hodnot zajišťuje systém jazykovou podporu, podporu kulturních zvyklostí a také uspořádání dat. Seznam předvolených systémových hodnot pro jednotlivé verze národního jazyka uvádí téma Předvolené systémové hodnoty.

Následující témata se zabývají různými atributy, které je nutné uvážit při návrhu aplikací závislých na kulturních zvyklostech:

- Atributy databázových souborů
- Atributy úloh
- Atributy programů
- Informace ve zprávě CPX8416
- Formáty data
- Oddělovače data
- Editování prezentace data
- Formáty času
- Časové oddělovače
- Editování prezentace času
- Formáty zápisu desetinných míst
- Třídící posloupnosti

Atributy databázových souborů: Atributy databázových souborů závislé na kulturních zvyklostech jsou následující:

- Identifikátor kódované znakové sady (CCSID).
- Třídící posloupnost (SRTSEQ).
- Identifikátor jazyka (LANGID).

Atribut CCSID se týká pouze fyzických souborů. Atributy SRTSEQ a LANGID mohou být použity jak u fyzických, tak u logických souborů. Logický soubor může mít hodnotu CCSID jen tehdy, když převezme CCSID z fyzického souboru. Databázové atributy se ukládají s daty. Jsou statické v tom smyslu, že nemohou být dynamicky změněny procesem přístupu k datům.

Atributy úloh: Atributy úloh závislé na kulturních zvyklostech jsou následující:

- Identifikátor kódované znakové sady (CCSID).

- Třídící posloupnost (SRTSEQ).
- Identifikátor jazyka (LANGID).
- Identifikátor země nebo regionu (CNTRYID)
- Formát data (DATFMT).
- Oddělovač data (DATSEP).
- Formát zápisu desetinných míst (DECFMT).
- Časový oddělovač (TIMSEP).

Předvolené hodnoty atributů CCSID, SRTSEQ, LANGID a CNTRYID se nastavují z uživatelského profilu při spuštění úlohy. Hodnoty pro CCSID, DATFMT, DATSEP, DECFMT, SRTSEQ a TIMESEP mohou být nastaveny z atributů LOCALE a SETJOBATR asociovaných s uživatelským profilem. Příkazem CHGJOB (Změna úlohy) můžete dočasně přepsat hodnoty zadané u kteréhokoliv atributu úlohy uvedeného v seznamu.

Atributy programů: Parametry SRTSEQ a LANGID mohou být zadány také jako atributy programu, který patří k typu objektu *PGM. Parametr LANGID se používá společně s parametrem SRTSEQ pouze tehdy, je-li hodnota SRTSEQ nastavena na *LANIDUNQ nebo *LANGIDSHR. Jinak se parametr LANGID nepoužívá.

Jestliže se program explicitně odvolává na třídící posloupnost nebo identifikátor jazyka, vstupují v platnost atributy uložené v objektu typu program. Hodnota *JOBRUN pro tyto parametry se používá k odkazu na atributy úlohy, která program spouští. *JOBRUN umožňuje použití sady samostatných programů zpracovávajících data podle různých třídících posloupností. Hodnota *JOBRUN však ovlivňuje pouze zpracování dat, nikoli pořadí jejich vyvolání. Pořadí vyvolání dat je určeno atributy databáze. Načtení dat v jiné třídící posloupnosti, než jaká je definována v databázi, je možné pomocí logických souborů, které se vytvářejí samostatně.

Informace ve zprávě CPX8416: Bude-li vaše aplikace překládána do jiných jazyků, použijte zprávu CPX8416 ze souboru zpráv QCPFMSG, abyste získali správná nastavení některých hodnot závislých na kulturních zvyklostech pro jiné jazyky. Zpráva existuje pro váš primární jazyk a všechny instalované knihovny sekundárních jazyků. Systémová zpráva obsahuje tyto hodnoty:

- Kódovou stránku a znakovou sadu.
- Znak měny.
- Formát data.
- Oddělovač data.
- Formát zápisu desetinných míst.
- Přizpůsobení přestupného roku.
- Identifikátor kódované znakové sady.
- Časový oddělovač.
- Identifikátor jazyka.
- Identifikátor země nebo regionu

Pole závislá na kulturních zvyklostech v panelu nebo na obrazovce by neměla obsahovat pevně naprogramované hodnoty. Tato pole by měla být definována s maximální délkou povolenou pro pole na obrazovce.

Má-li vaše aplikace podporovat uživatele jiných jazyků, než je váš primární jazyk, měly by volatelné rutiny používat hodnoty zpráv CPX8416. Volatelná rutina používá hodnoty závislé na kulturních zvyklostech pro primární jazyk, aby stanovila obsah pole (například formát data), a umísťuje tyto hodnoty na obrazovku. Systémové hodnoty podpory národního jazyka (NLS) uchovávané ve zprávě CPX8416 určují formát hodnot v polích závislých na kulturních zvyklostech.

Vaše aplikace může použít podrobnosti ze systémové zprávy.

Následující tabulka ukazuje uspořádání zprávy CPX8416. Tento příklad ukazuje hodnoty v textovém sloupci za použití verze národního jazyka angličtina - velká a malá písmena (kód označení 2924).

	Pole	Začátek	Délka	Zarovnání
Popis Hodnota	QCHRID 697 37	0001 0012	10 21	L L
Popis Hodnota	QCURSYM \$	0034 0045	10 01	L L
Popis Hodnota	QDATFMT MDR	0047 0058	10 03	L L
Popis Hodnota	QDATSEP /	0062 0073	10 01	L L
Popis Hodnota	QDECFMT	0075 0086	10 01	L L
Popis Hodnota	QLEAPADJ 0	0088 0099	10 01	L L
Popis Hodnota	QCCSID 37	0101 0112	10 05	L L
Popis Hodnota	QTIMSEP :	0118 0129	10 01	L L
Popis Hodnota	QLANGID ENU	0131 0142	10 03	L L
Popis Hodnota	QCNTYID US	0146 0157	10 02	L L
Popis Hodnota	QIGCCDEFNT *NONE	0160 0171	10 21	L L

Formáty data: Pro prezentaci datumů neexistuje žádný celosvětový standard. Proto by měl být formát data vždy ukládán externě jako součást textových dat. V operačním systému OS/400 jsou platné tyto formáty data:

- *MDR (měsíc, den, rok).
- *DMR (den, měsíc, rok).
- *RMD (rok, měsíc, den).
- *JUL (rr/ddd).
- *ISO (RRRR-MM-DD).
- *USA (MM/DD/RRRR).
- *EUR (DD.MM.RRRR).
- *JIS (RRRR-MM-DD).

Poznámka: Některé funkce operačního systému OS/400 nepodporují všechny výše uvedené formáty data.

V databázových souborech je datum možné uložit jako:

- normální numerická datová pole
- datumová data typu SAA^(R)

Když ukládáte datum jako numerická data, musí vaše aplikace zadat formát, ve kterém se ukládají a prezentují.

Když uložíte datum jako typ dat DATE (L), můžete u databázového souboru specifikovat formát klíčovým slovem DDS DATFMT. Datum je v tomto předdefinovaném formátu zobrazeno jako znaková data, včetně oddělovačů data.

Je-li nutné třídění a další zpracování data, mělo by být uloženo ve formátu *ISO (RRRR-MM-DD) a konvertováno během vstupních a výstupních operací. Napište si rutinu pro tuto konverzi ve vyšším programovacím jazyce.

Oddělovače data: Platné oddělovače data jsou:

- / (lomítko)
- - (pomlčka)
- . (tečka)
- , (čárka)
- (mezera)

Oddělovač data pro prezentaci by měl být vždy ukládán externě jako součást textových dat.

Když pro datum používáte dekadická pole, musí vaše aplikace nejen zadat formát, ale musí také v průběhu vstupní operace a prezentace pracovat s oddělovači data.

Když používáte pole typu datum, jsou v datu vždy zahrnuty i oddělovače. Chcete-li změnit oddělovač data, můžete napsat rutinu ve vyšším programovacím jazyce, která bude datum konvertovat.

Editování prezentace data: S prezentací data na obrazovce a na tiskárně musíte pracovat různě podle toho, jak je datum uloženo:

- Jako běžné pole s dekadickými daty:

Váš aplikační program odpovídá za to, jakým způsobem se data zadávají, ukládají a prezentují. Aplikace musí kontrolovat, zda je datum uvedeno ve správném formátu, odstranit případné oddělovače data a, je-li to nutné, převést datum do jiného formátu a editovat je v zobrazovacím nebo tiskovém souboru.

Klíčové slovo DDS DATE se používá pouze jako výstupní pole. DATE používá atributy úlohy DATE, DATFMT a DATSEP. DATE můžete u 6 a 8číselných polí data upravit pomocí klíčového slova EDTCDE (editační kód).

Editování pomocí klíčového slova EDTCDE zahrnuje následující změny vzhledu zobrazených polí podle toho, který editační kód je zadán:

- Vedoucí nuly jsou potlačeny.
- Nulové hodnoty mohou být zobrazeny jako nuly nebo mezery.
- Pole může být dále editováno pomocí editačního kódu definovaného uživatelem.

U všech ostatních typů polí používajících klíčové slovo EDTCDE Y musí program zadat formát a systém použije oddělovač data úlohy, která vytvořila soubor zařízení. Oddělovač data je zařazen do objektu a v době běhu programu jej nemůžete dynamicky změnit.

- Jako pole SAA typu dat DATE (L)

Klíčové slovo DDS DATFMT (formát data) vám umožňuje zadávat na úrovni databázového pole různé formáty data, včetně předvolených oddělovačů data. U parametrů *MDY, *DMY, *YMD a *JUL je možné změnit předvolený oddělovač data klíčovým slovem DATSEP (oddělovač data). Hodnoty *ISO, *USA, *EUR a *JIS mají pevný oddělovač a klíčové slovo DATSEP není u těchto hodnot povoleno. Klíčová slova DATFMT a DATSEP vám umožňují zadat formát a editační znaky pro uložení polí data. Datum je zobrazeno jako znakový řetězec včetně oddělovačů.

Libovolnou konverzi formátu mezi vstupním datem a formátem, který požaduje databáze, je možné provést pomocí:

- Rutin aplikačních programů.
- Mapování polí pomocí logických souborů, které definují formáty a oddělovače data.

Například pomocí následujících CL programů můžete zajistit konverzi data, která je závislá na skutečných atributech úlohy:

```
PGM      PARM(&fromfmt &fromfld &tofld );
DCL      VAR(&fromfmt); TYPE(*CHAR) LEN(4)
DCL      VAR(&fromfld); TYPE(*CHAR) LEN(10)
DCL      VAR(&tofld); TYPE(*CHAR) LEN(10)
CVTDAT   DATE(&fromfld); TOVAR(&tofld);
FROMFMT  (&fromfmt); TOFMT(*JOB) TOSEP(*JOB)
ENDPGM
```

Váš aplikační program musí předat formát data, který chcete konvertovat a vlastní datum CL programu. CL program předpokládá, že atributy úlohy představují to, jak mají být pole data podle představ uživatele upravena. Načte tyto hodnoty a provede konverzi odpovídající těmto hodnotám a předá takto upravené pole zpět. Hodnoty *ISO, *USA, *EUR a *JIS mají pevný oddělovač, který nelze změnit. Jestliže parametr TOFMT obsahuje jednu z těchto hodnot, hodnota TOSEP se ignoruje.

Formáty času: Operační systém OS/400 podporuje tyto formáty času:

- *HMS (hh:mm:ss)
- *ISO (hh.mm.ss)
- *USA (hh:mm AM nebo hh:mm PM)
- *EUR (hh.mm.ss)
- *JIS (hh:mm:ss)

Systémová hodnota QTIME má jeden formát (hhmmss). Časový oddělovač je určen systémovou hodnotou QTIMSEP.

Formát času pro prezentaci by měl být vždy uložen externě jako součást textových dat.

V databázových souborech může být čas uložen jako:

- Normální numerické datové pole.
- Typ časových dat SAA.

Pokud čas uložíte jako numerická data, musí vaše aplikace specifikovat formát, v němž se má čas uložit a prezentovat.

Jestliže čas uložíte jako typ dat TIME (T), můžete určit jeho formát pomocí klíčového slova DDS TIMFMT v databázovém souboru. Čas je tříděn v tomto předdefinovaném formátu jako znaková data, včetně časových oddělovačů. Chcete-li převést pole času z jednoho formátu na jiný, napište CL program nebo rutinu ve vyšším programovacím jazyku, která by provedla danou konverzi.

Časové oddělovače: Platné znaky pro časové oddělovače v operačním systému OS/400 jsou tyto:

- : (dvojtečka)
- . (tečka)
- (mezera)
- , (čárka)

Časový oddělovač pro prezentaci by měl být vždy uložen externě jako součást textových dat.

Když používáte pole dekadických dat pro pole času, musí vaše aplikace specifikovat formát času a časové oddělovače při vstupních a prezentačních operacích.

Pokud používáte pole typu čas, jsou časové oddělovače vždy zahrnuty v poli času. Chcete-li změnit časové oddělovače, napište CL program nebo rutinu ve vyšším programovacím jazyku, která by provedla danou konverzi.

Editování prezentace času: S prezentací času v zobrazovacích a tiskových souborech potřebujete pracovat rozdílným způsobem v závislosti na tom, jak jsou uloženy:

- Jako dekadické datové pole.

Váš aplikační program odpovídá za způsob, jakým se hodnota zadává, ukládá a prezentuje. Program musí kontrolovat správný formát, odstraňovat časové oddělovače, v případě nutnosti převést čas na jiný formát a editovat ho v zobrazovacím nebo tiskovém souboru.

Editaci lze provést, když pro dané pole zadáte editační slovo (EDTWRD). Klíčové slovo TIME je pole určené pouze pro výstup. Jak editační slovo, tak klíčové slovo TIME používají informace, které byly dostupné v čase vytvoření. Časové oddělovače jsou integrovány do objektu souboru zařízení.

Oba způsoby vyžadují, abyste měli různé kopie zdroje a objektů pro různé editační požadavky.

- Jako pole TIME (T) s daty typu SAA.

Operační systém OS/400 vám umožňuje specifikovat na úrovni databázových souborů různé formáty času a časové oddělovače. Klíčová slova TIME umožňují specifikovat formát a editační znaky pro ukládání polí času. Pole typu čas se zobrazuje jako znakový řetězec, který obsahuje oddělovače.

Při typu dat SAA můžete v zobrazovacím nebo tiskovém souboru zadat pole času jako znaková pole. Během operace vstupu musí váš program zkontrolovat vložené hodnoty, zda mají správný formát a oddělovače, a pak je uvolní do pole databázového souboru. Při operaci výstupu pouze přesunete znakový řetězec z pole databázového souboru do pole souboru zařízení, a to včetně oddělovačů.

Jakékoliv konverze formátu mezi vstupním a výstupním formátem, které databáze požaduje, mohou být provedeny:

- Rutinami aplikačního programu.
- Mapováním polí prostřednictvím logických souborů, které definují různé formáty času a časové oddělovače.

Formáty zápisu desetinných míst:



Formát zápisu desetinných míst můžete změnit pomocí systémové hodnoty QDECFMT tak, aby odpovídal formátování desetinných míst ve vaší zemi nebo oblasti.



Třídící posloupnosti: Operační systém OS/400 podporuje třídící posloupnosti. Svá data můžete seřadit podle požadavků konkrétních aplikací v závislosti na kulturních zvyklostech, a to pomocí jedné z následujících možností:

- Hexadecimální třídění (nepoužívají se tabulky třídících posloupností). Toto je předvolené nastavení.
- Tabulka třídící posloupnosti se sdílenými nebo jedinečnými váhami dodaná uživatelem nebo systémem, určená parametrem SRTSEQ.

Následující příklad ukazuje, jak použít zdrojový soubor DDS k vytvoření databázových souborů s různými třídícími posloupnostmi. Je možné provést tyto kroky:

```
CRTxFL FILE(*CURLIB/NAME)
        SRTSEQ(*JOB)
        LANGID(*JOB)
```

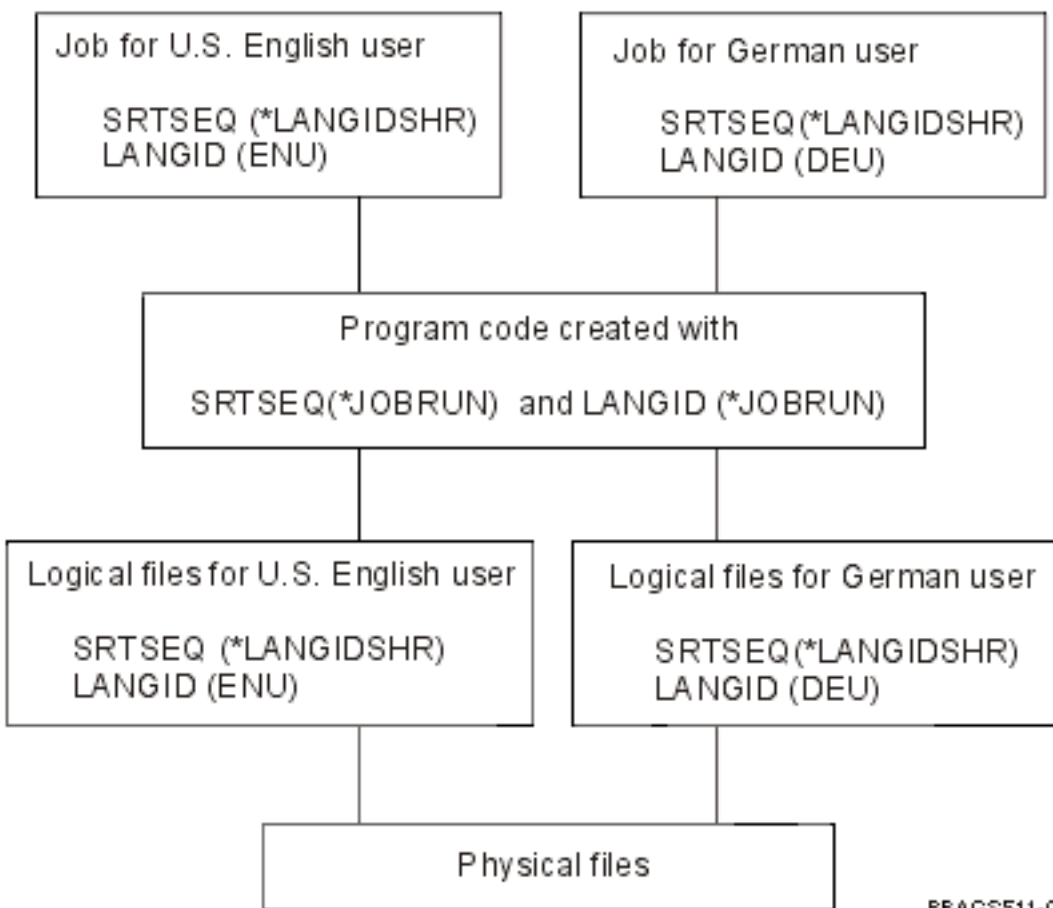
Potom můžete změnit atributy úlohy a vytvořit tak soubory s rozdílnými třídícími posloupnostmi.

Při vytváření CL programu a programů ve vyšších programovacích jazycích lze specifikovat buď okamžité, nebo pozdější připojení třídící posloupnosti. Při okamžitém připojení třídící posloupnosti se tabulka třídící posloupnosti, která se má použít, určuje v době kompilace. Při pozdějším připojení třídící posloupnosti se tabulka třídící posloupnosti, která se má použít, určuje v době běhu programu.

Pozdější připojení umožňuje použít jednu sadu programů v různých prostředích národních jazyků.

Následující obrázek znázorňuje použití různých třídících posloupností pro různé úlohy s jednou sadou

fyzických souborů a programového kódu. Tabulka třídící posloupnosti definovaná pro úlohu a používaná programem by se měla shodovat (nebo by měla být kompatibilní) s tabulkou třídící posloupnosti přiřazenou logickým souborům, k nimž se přistupuje prostřednictvím seznamu knihoven.



Návrh prostředí s rozdílnými třídícími posloupnostmi:

Pokud se očekává, že váš program bude fungovat s různými třídícími posloupnostmi, zvažte tyto aspekty:

- Presentace dat v odlišném pořadí.
- Zpracování odlišným vět.

Na základě kritérií výběru, jako je např. menší než nebo větší než, mohou být vybrány rozdílné vět.

Kritérium výběru rovná se může mít za následek vybrání odlišného počtu vět, pokud se použije tabulka třídící posloupnosti se sdílenými váhami.

- Zpracování podmíněného větvení může být rozdílné.

Poznámka: Podpora třídících posloupností nemá vliv na systémové seznamy (např. na výstup z příkazu WRKOBJ).

K zadání třídící tabulky a knihovny, v níž je obsažena, můžete použít alternativní posloupnost klíčového slova na úrovni souboru DDS (ALTSEQ). Systémem dodávané tabulky třídících posloupností se sdílenými a jedinečnými váhami lze použít pro definici alternativní porovnávací posloupnosti.

Tabulka alternativní porovnávací posloupnosti se vkládá do souboru v době kompilace a není potřeba v době provádění programu. Můžete mít různé soubory obsahující různé porovnávací posloupnosti, které používají jednu sadu DDS.

Poznámka: Alternativní porovnávací posloupnost definovaná ve vašich databázových souborech musí být definována také ve vašich aplikačních programech. Jinak byste mohli obdržet neočekávané výsledky.

Klíčové slovo DDS ALTSEQ poskytuje omezenou podporu řazení a nemá žádný vliv na logiku výběru nebo vynechání. Klíčové slovo ALTSEQ lze použít pouze s parametrem SRTSEQ(*SRC) v příkazech CRTPF a CRTLF.

Další informace

Další informace o třídících posloupnostech uvádí téma Implementace třídění grafických znaků (dat).

Návrh zobrazovacích souborů: Panely aplikace se obvykle skládají z těchto hlavních prvků:

- Řetězce konstantního textu.
- Vstupní a výstupní pole.
- Specifikace editování polí.
- Specifikace pozice kurzoru.
- Předvolené hodnoty vstupního pole.
- Specifikace ověření platnosti pole.
- Chybové zprávy.

Můžete s nimi pracovat jako s prvky popsány v programu nebo jako se souborem popsáným externě pomocí DDS. Informace v tomto tématu jsou založeny na technice externího popisu pomocí DDS.

Řetězce konstantního textu: Protože různé jazyky mají u stejných výrazů odlišné nároky na prostor, mějte to při návrhu panelů na paměti. Nedávejte mnoho polí na stejnou řádku, s výjimkou seznamů, které mají místo náznaku polí záhlaví sloupce. Nepřetěžujte panely informacemi. Pro jejich tvorbu si vyberte některou z technik popsanych v tématu "Návrh kódu textových dat" na stránce 19.

Vstupní a výstupní pole: Při definici polí je vhodné brát v úvahu potřeby různých jazyků, zemí, kultur, měn a právních aspektů, pro které má být vaše aplikace určena. Například předpokládejme, že chcete ukládat italskou liru a japonský jen do stejného pole jako americký dolar. Velikost pole tedy musíte uzpůsobit tak, aby se do něj vešel vysoký počet číslic potřebný pro italskou liru.

Specifikace editování polí: Při zadávání editačních úprav číselných, datumových a časových polí musíte brát v úvahu různé kulturní konvence uživatelů, které chcete oslovit. Instrukce pro formátování a editování nekódujte do aplikačních programů tak, abyste při nutnosti zavedení jiné konvence museli modifikovat program. Další informace uvádí téma Návrh závislý na kulturních zvyklostech.

Specifikace pozice kurzoru: Nezadávejte pevné umístění kurzoru na obrazovce, protože různé jazyky mají různé nároky na prostor. Při práci s různými zobrazovacími soubory můžete tyto soubory upravit v procesu překladu. Když potřebujete pracovat s umístěním kurzoru nezávislým na poli, uložte informace o pozici kurzoru mimo vlastní kód a proměnné hodnoty pro klíčová slova načítejte z programu.

Například:

```
A record-name      CSRL0C(field-name-1 field-name-2)
```

Pozice kurzoru na úrovni pole je užitečnější v prostředí NLS. U běžných vět se to provádí zadáním klíčového slova DSPATR(PC) u specifického pole. U podsouborů je možné kurzor umístit pomocí klíčového slova SFLRCDNBR(CURS0R) u speciálního pozičního pole. Mimoto musí být do tohoto pole uloženo číslo věty podsouboru předtím, než je formát zapsán.

Například:

```
A field-name      4S 0B line pos SFLRCDNBR(CURS0R)
```

Poznámka: Jméno záznamu a pole, kde je kurzor umístěn, relativní číslo záznamu a indikátor přetažení/zkrácení podsouboru může být vrácen do vašeho aplikačního programu. Tato funkce je obstarávána skrytými poli u klíčových slov DDS RTNCSRLOC, SFLCSRRRN a SFLMODE.

Předvolené hodnoty vstupního pole: Existují tři různé způsoby, jak do vstupních polí na vaší obrazovce umístit předvolené hodnoty, které by uživatel mohl přepsat svými vlastními daty:

- Získání informací z programu

Hodnoty nikdy neprogramujte napevno jako literál, pokud jsou to hodnoty závislé na kulturních zvyklostech. Používejte hodnoty, které můžete získat ze systémem poskytovaných informací (např. systémové datum nebo datum úlohy), nebo použijte hodnoty z datových objektů (např. databázový soubor nebo datová oblast) mimo daný program.

- Použití klíčových slov DDS DFT (Předvolba) nebo DFTVAL (Předvolená hodnota)

Předvolenou vstupní hodnotu zadejte přímo za klíčové slovo v DDS. Klíčové slovo DDS DFT je určeno pouze pro vstupní pole (I). Pro pouze výstupní (O) nebo vstupně výstupní (B) pole použijte klíčové slovo DFTVAL.

Například:

```
A field-name length type I line pos DFT('default ')
nebo A field-name length type O/B line pos DFTVAL('default value ')
```

- Použití klíčového slova DDS MSGID (Identifikace zprávy)

Použití klíčového slova MSGID (Identifikace zprávy) vám umožňuje načíst obsah zadaného popisu zprávy, když je program spuštěný, a vložit tuto hodnotu jako předvolbu do pole vašeho zobrazovacího souboru. Podmínkou použití této metody je, že se jedná o vstupně výstupní (B) pole.

Například:

```
A field-name length type B line pos MSGID(message-id [*libl/message-file])
```

Díky tomu můžete používat různé soubory zpráv pro každou verzi národního jazyka tak, že při spuštění programu příslušným způsobem nastavíte seznam knihoven.

Specifikace ověření platnosti pole: Následující klíčová slova DDS umožňují ověření platnosti polí, do kterých je možné zadávat vstupní data na obrazovce:

- RANGE (kontrola rozsahu).
- VALUES (kontrola hodnot).
- CMP a COMP (porovnání).
- CHECK (kontrola platnosti, ovládání klávesnice a kurzoru).

Použití klíčových slov DDS u všech pevně naprogramovaných změn, které jsou závislé na jazyku, zemi a kultuře způsobuje, že je nezbytné duplikování a modifikace DDS a aplikačního programu.

Příklad: ověření platnosti

Následuje příklad ověření platnosti polí schopných přijímat vstupy na obrazovce pomocí klíčových slov DDS VALUES, COMP a CHECK:

```
A field-name length type usage line pos VALUES('Y' 'N')
nebo A field-name length type usage line pos COMP(EQ 'US$')
nebo A field-name length type usage line pos CHECK(M10 or M11)
(kontrola modulo)
nebo A field-name length type usage line pos CHECK(RL)
(pravo-levá podpora)
```

Kontroly ověření platnosti se poskytují v souladu s třídící posloupností definovanou pro zobrazovací soubor při jeho vytvoření. Stejný zdrojový soubor DDS můžete použít k vytvoření objektů pro různé jazyky. Například, následující příkaz vytvoří zobrazovací objekt s příznakem tabulky třídící posloupnosti Latin 1: CRTDSPF FILE(jméno) SRTSEQ(*LANGIDSHR) LANGID(DEU)

Následující specifikace:

```
A field-name length type usage line pos COMP(EQ 'a')
```

přijímá všechna malá a velká písmena a písmena s diakritikou, jak jsou definována sdílenou vahou v třídící posloupnosti Latin 1.

Kromě toho si všimněte, že všechny kontroly zadané klíčovými slovy DDS jsou prováděny funkcí řízení dat operačního systému OS/400. Jakákoliv chybová zpráva způsobená chybným vstupem nebo chybným postupem uživatele se objeví v jazyce operačního systému OS/400. Může to být primární nebo sekundární jazyk, podle toho, jak je nastaven seznam knihoven úlohy.

Můžete to potlačit použitím dalšího klíčového slova DDS CHKMSGID (kontrola identifikátoru zprávy). Toto klíčové slovo vám umožňuje určit vlastní přizpůsobené zprávy a soubor zpráv, které mají používat kontrolní rutiny operačního systému OS/400.

Například:

```
A field-name length type usage RANGE(1 999)
A CHKMSGID(USR1234 [*lib1/]APPMSGF [&MSGFLD1])
A MSGFLD1 length type P TEXT('Pole pro data zprávy')

a
ADDMSGD MSGID(USR1234) MSGF(APPTXDENU/APPMSGF)
MSG('Hodnota &1; je mimo rozsah 1 až 999')

a
ADDMSGD MSGID(USR1234) MSGF(APPTXDDEU/APPMSGF)
MSG('Wert &1; ist ausserhalb des gltigen Bereichs 1 bis 999')
```

Chcete-li použít různé soubory zpráv s různými jmény knihoven, nezadávejte pevné jméno knihovny. Při spuštění programu můžete použít soubor zpráv pro různé jazyky nastavením seznamu knihoven.

Chybové zprávy: Existují dva způsoby, jak podat chybové zprávy v zobrazovacím souboru:

- Zadání textu jako konstanty u klíčových slov ERRMSG nebo SLFMSG.
Text zadejte přímo jako konstantu u klíčového slova DDS. Chcete-li mít více než jeden jazyk, musíte duplikovat zdrojový kód DDS a přeložit konstanty ve specifikacích DDS. Pak můžete vytvořit samostatný objekt zobrazovacího souboru pro každý jazyk.
- Použití předdefinovaných zpráv u klíčového slova ERRMSGID nebo SLFMSGID.
Při použití předdefinovaných zpráv místo konstanty musíte mít několik zobrazovacích souborů. Místo použití různých zobrazovacích souborů vyměňte jenom používaný soubor zpráv tak, že nastavíte knihovnu podle jazyka, který chcete použít.

Například:

```
A field-name length type usage EDTCDE(x)
A 61 ERRMSGID(USR3456 [*lib1/]APPMSGF [&MSGFLD2])
A MSGFLD2 length type P TEXT('Pole pro data zprávy')

a
ADDMSGD MSGID(USR3456) MSGF(APPTXDENU/APPMSGF)
MSG('Datum dodávky &1; je nižší než datum ukončení výroby &2;')

a
ADDMSGD MSGID(USR3456) MSGF(APPTXDDEU/APPMSGF)
```

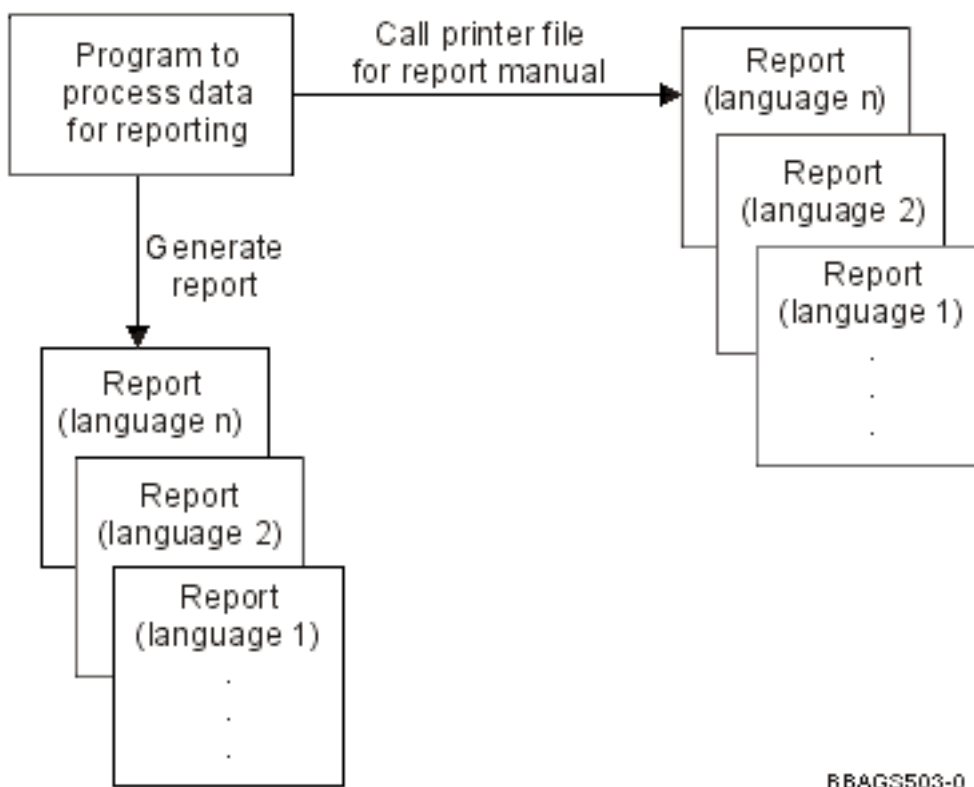
```
MSG('Lieferdatum &1; ist . . .')
```

```
.  
. .  
.
```

Návrh a překlad tiskových souborů: Existují tyto typy tiskových souborů:

- Tiskové soubory popsané programem
Soubory popsané programem závisí na programu ve vyšším programovacím jazyku, jenž definuje záznamy a pole, které se mají vytisknout.
- Tiskové soubory popsané externě
Tiskové soubory popsané externě používají k definici záznamů a polí, které se mají vytisknout, DDS namísto programu ve vyšším programovacím jazyku.

Následující obrázek znázorňuje způsob použití externě popsanych tiskových souborů při vytváření sestav pro různé verze národních jazyků.



Překlad tiskového souboru:

Při návrhu tiskových souborů, které se mají překládat do verze národního jazyka:

- Používejte externě popsané tiskové soubory k definici záznamů a polí, které se mají vytisknout. Nepoužívejte tiskové soubory popsané programem. Tiskové soubory popsané programem jsou popsány uvnitř programu napsaného ve vyšším programovacím jazyku. Překladatelé, kteří by se snažili přeložit text vložený v programu, mohou chybně přeložit literály obsažené uvnitř programu.
- Tiskněte data v jedné národní grafické znakové sadě na zařízeních, která podporují odpovídající znakové sady a kódové stránky. Ne všechny tiskárny podporují všechny parametry CHRID.
- Používejte klíčové slovo MSGCON k přístupu ke konstantnímu textu popsánému v souboru zpráv. Tiskový soubor nemá klíčové slovo MSGID. K zadání konstantního textu v tiskovém souboru však

můžete použít metodu přímého zakódování jako nepojmenovaného výstupního pole (literálu) a uložení textu do databázového souboru. Další informace uvádí téma “Návrh kódu textových dat” na stránce 19.

- Berte na zřetel kulturní zvyklosti, když jsou v tiskovém souboru popisovány čárové kódy. Různé země mají rozdílné normy pro čárové kódy.
- Při zadávání dat uvažte tyto parametry v příkazu CRTPRTF (Vytvoření tiskového souboru).
 - PAGESIZE (velikost stránky)
Různé země mají rozdílné normy pro velikost stránky.
 - OVRFLW (číslo řádky pro přetečení)
Číslo řádky pro přetečení musí být menší nebo rovné délce stránky.
 - CHRID (znaková sada a kódová stránka)
Je-li parametr CHRID tiskového souboru nastaven na *DEVD, použije tiskárna identifikátor znaků, který byl nastaven na ovládacím panelu nebo byl zadán v popisu zařízení.
Jestliže je parametr CHRID tiskového souboru nastaven na konkrétní hodnotu, označuje tato hodnota kódovou stránku a znakovou sadu používanou k tisku dat. V případě externě popsaných tiskových souborů se parametr CHRID používá pouze pro pole, která mají specifikováno také klíčové slovo DDS CHRID. Pro všechna ostatní pole je použita kódová stránka a znaková sada stejná, jako když je zadána hodnota *DEVD.
Pokud je parametr CHRID tiskového souboru nastaven na hodnotu *JOBCCSID, je konstantní text z externě popsaného tiskového souboru zkonvertován na CCSID úlohy. Tok tiskových dat je označen CHRID, který pochází z CCSID úlohy, přičemž se tato hodnota CHRID použije pro tisk dat. Když se v parametru CHRID použije hodnota *JOBCCSID, je klíčové slovo DDS CHRID ignorováno.
Poznámka: Všechny kódové stránky a znakové sady nelze zpracovávat na všech tiskárnách.

Návrh zdrojových souborů: Když se vytváří databázové zdrojové soubory, je jim implicitně přiřazen CCSID úlohy, pokud jim ovšem nebyla hodnota CCSID přiřazena explicitně pomocí parametru CCSID v příkazu CRTPF (Vytvoření fyzického souboru) nebo CRTSRCPF (Vytvoření zdrojového fyzického souboru). Má-li CCSID úlohy hodnotu 65535, použije se jako implicitně přiřazený CCSID předvolený CCSID (DFTCCSID) úlohy. Předvolený CCSID úlohy je určen systémovým identifikátorem jazyka a indikátorem úlohy podporujícím DBCS.

Návrh architektury CDRA (Character data representation architecture): Chcete-li aktivovat aplikaci pro vícejazyčné prostředí, zvažte následující body:

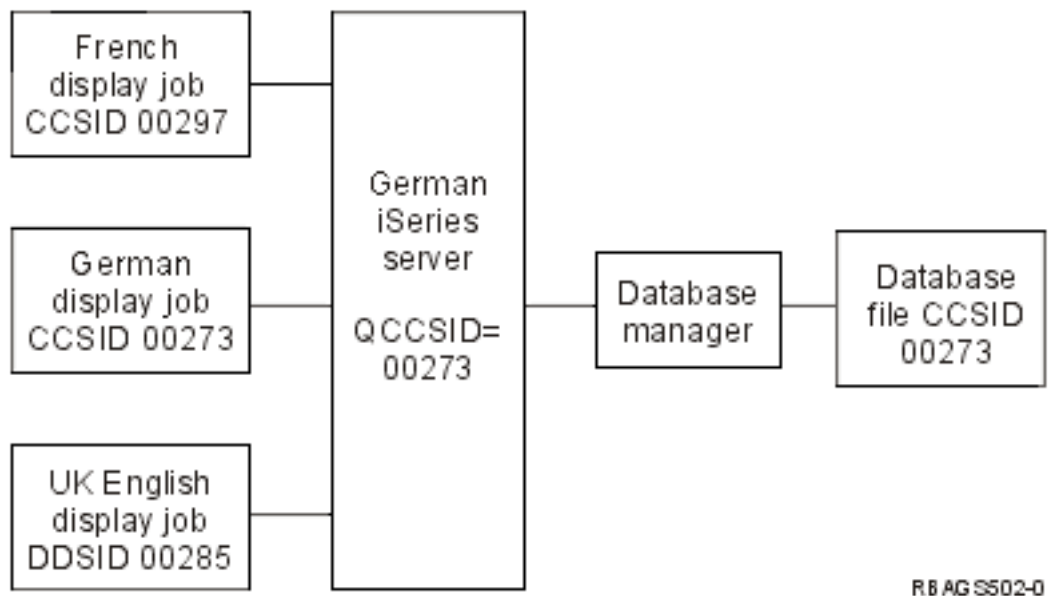
- U fyzických souborů se vyvarujte kódování hodnot CCSID přímo v DDS. Při vytváření různých fyzických souborů pro různé jazyky, změňte CCSID vaší úlohy (příkazem CHGJOB). Je nutné udržovat jen jednu sadu kódu DDS.
Konverze mezi všemi CCSID nemusí mít ve všech případech smysl. Jestliže například přistupujete k řecké databázi s CCSID 00875 z německé obrazovkové stanice s CCSID úlohy 00273, uvidíte na obrazovce zkomolená data.
Země mimo oblast znakové sady Latin-1 užívají znaky nepoužívající latinku. Mezi kódovými pozicemi používajícími a nepoužívajícími latinku není možná žádná smysluplná konverze. Arabština, hebrejšтина a turečtina jsou jazyky SBCS nepoužívající latinku.
- Když dochází ke sdílení databáze, definujte u souborů CCSID primárně používaného jazyka. Zajistěte, aby všichni uživatelé měli ve svých profilech definován CCSID toho jazyka, který používají.

Další informace o architektuře CDRA najdete v těchto tématech:

- Práce s CCSID
- Použití příkazu pro odeslání souboru v síti
- Scénář: vícejazyčný samostatný systém
- Scénář: vícejazyčná síť

Použití příkazu pro odeslání souboru v síti: Když použijete příkaz SNDNETF (Odeslání souboru v síti), předpokládá se, že data (pokud posíláte pouze člen) mají CCSID úlohy, která spustila tento příkaz. Proto se neprovádí žádná konverze. Při přijetí dat je pak důležité věnovat pozornost tomu, aby byl člen uložen do souboru se stejným CCSID, jako měl původní soubor. Jestliže příjemce nezná CCSID došlého členu souboru, lze jej přijmout do souboru s CCSID 65535, který indikuje, že se neprovádí žádná konverze.

Scénář: vícejazyčný samostatný systém: Na následujícím obrázku je znázorněn vícejazyčný samostatný systém, v němž je primárním jazykem němčina a sekundárními jazyky jsou angličtina a francouzština. Všichni uživatelé zadávají data do stejného databázového souboru.

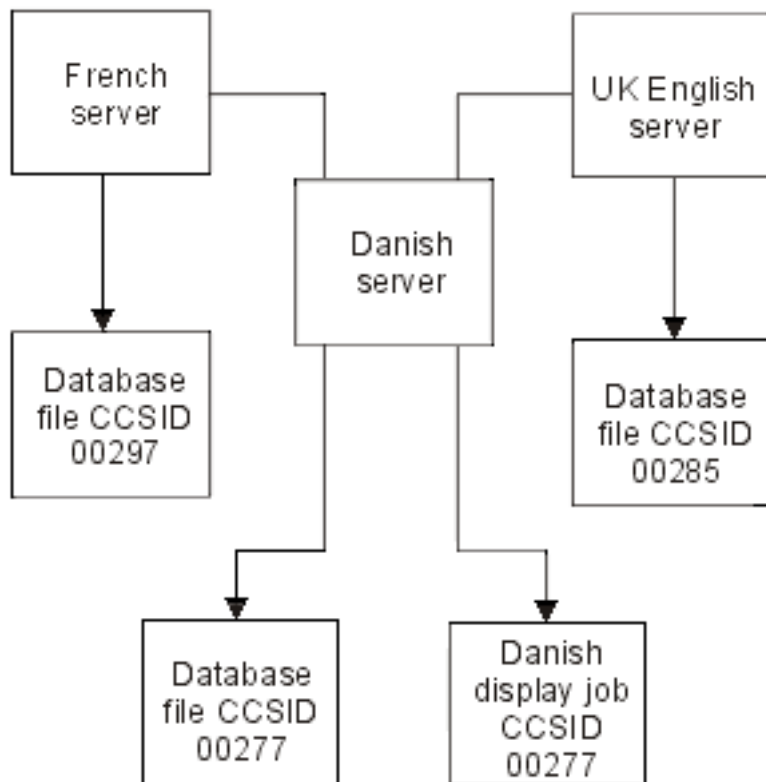


V tomto vícejazyčném systému zadávají všichni uživatelé znaková data do jediného souboru s CCSID 00273 (němčina), přičemž jsou znaková data zadaná z anglických a francouzských obrazovkových stanic mapována do německého souboru.

Má-li být zajištěno správné mapování, měla by pole definovaná jako znaková být skutečně znakovými poli. Jestliže pole obsahují hodnoty pro vývoj aplikací (např. řídicí znaky nebo pole, která se nepoužívají jako skutečná znaková pole), měla by být tato pole zadána buď jako hexadecimální, nebo by jim měla být přiřazena hodnota CCSID 65535.

Při použití CCSID jsou znaky, které nelze konvertovat mezi různými kódovými stránkami, nahrazeny náhradním kódem. Pokud k naformátování a rozvržení vaší obrazovky používáte uživatelem definovaný tok dat (UDDS) namísto DDS, mohou se vám vrátit náhradní kódy poté, co systém přečte a vloží uvedená data do vašeho uživatelem definovaného toku dat. Náhradní kódy mohou na obrazovce vést k nepředvídatelným výsledkům.

Scénář: vícejazyčná síť: Následující obrázek představuje příklad vícejazyčné sítě se třemi servery iSeries umístěnými ve třech různých zemích, přičemž se na každém z nich používá jiný jazyk. V tomto příkladu používá aplikace v dánském systému distribuovanou relační databázi. Všechny národní znaky (bez ohledu na jazyk, v němž jsou data uložena) se na dánské obrazovce zobrazují správně. Když databáze použije CCSID jazyka, je integrita databáze zachována. Konverze dat mezi různými kódovými stránkami je plně automatická a je součástí správy databáze zajišťované operačním systémem OS/400.



RBAGS501-0

Zpracování jazyků, které nemají podporu NLV: Jestliže potřebujete podporovat jazyk, který nemá podporu verze národního jazyka (NLV), postupujte podle těchto obecných kroků:

1. Projděte dostupné verze národních jazyků. Zjistěte, která z verzí národních jazyků se nejvíce podobá vašemu jazyku z hlediska reprezentace znaků.
2. Verzi národního jazyka, která nejvíce odpovídá, nainstalujte jako svůj primární jazyk.
3. Upravte systémové hodnoty tak, aby odpovídaly vašim kulturním zvyklostem. Například nastavte formát data a času tak, aby se shodoval s formátem používaným v kultuře, kterou chcete podporovat. Další informace o nastavení systémových hodnot tak, aby odpovídaly kulturním zvyklostem, obsahuje téma *Systémové hodnoty pro jiné jazyky*.
4. Nakonfigurujte vaše pracovní stanice a tiskárny, aby odpovídaly vašemu primárnímu jazyku. Potom odstraňte nesrovnalosti mezi podporou instalované verze národního jazyka a vašim vlastním jazykem.

Poznámka: Funkce přizpůsobení pracovní stanice může podporovat pouze ty schopnosti, které jsou zabudovány ve vašem hardwaru. Prostřednictvím přizpůsobení pracovní stanice nemůžete podporovat funkce, které není schopen podporovat váš hardware.

5. Pomocí příkazu CRTTBL (Vytvoření tabulky) vytvořte tabulku třídící posloupnosti založenou na existující tabulce, která nejvíce odpovídá příslušné třídící posloupnosti pro váš jazyk.
6. Je-li váš jazyk jazykem DBCS, vytvořte své vlastní znaky (UDC) reprezentující chybějící znaky v kódové stránce asociované s instalovanou verzí národního jazyka. UDC je akronym pro uživatelem definovaný znak (User-Defined Character), který byl vytvořen prostřednictvím obslužného programu pro generování znaků (CGU). CGU je rozšířením kódové stránky se speciálními uživatelem definovanými ideografickými znaky, symboly nebo logy.

Doporučení pro programátory při návrhu globálních aplikací

Při vývoji globálních aplikací vyžaduje prostředí verze národního jazyka často dodatečnou pozornost přípravě a kompilaci kódu. Následující témata popisují některé z těchto požadavků a nabízejí návody, jimiž se můžete řídit, chcete-li problémy minimalizovat:

- Jak psát globalizované aplikace ve vyšších programovacích jazycích
- Jak psát globalizované aplikace používající obousměrná data
- Použití katalogů zpráv

Jak psát globalizované aplikace ve vyšších programovacích jazycích

Vaším hlavním cílem musí být, abyste měli jen jednu obecnou množinu prováděného kódu, která je společná pro všechny jazykové verze, a aby se vaše programy co možná nejvíce řídily tabulkami. Měli byste:

- Zakládat kontroly platnosti na přístupu k databázovým souborům a souborům zpráv, spíše než k literálům a tabulkám.
- Zakládat výpočty na faktorech proměnných načtených ze souboru, spíše než je programovat inline.
- Umístit funkce závislé na kulturních zvyklostech do samostatných modulů aplikací a volat je, když je nemůžete pružně programovat.

Nepoužívejte pevně naprogramované hodnoty, pokud nejsou úplně nezávislé na jazyku a kulturních zvyklostech, u operací porovnání, snímání nebo volání. Kromě toho nepoužívejte hodnoty, které jsou citlivé na velká nebo malá písmena. Například, neprogramujte nikdy napevno odpovědi Ano a Ne (A nebo N), protože tyto hodnoty jsou v každém jazyce odlišné a měly by být součástí textových dat.

U literálů a konstant ve zdrojovém kódu používejte pouze znaky z invariantní znakové sady. Jestliže se v programu kontroluje platnost vstupních dat, ověřte si, že kontrolované znaky patří do invariantní znakové sady. Jinak může nastat situace, kdy je uživatel vyzván, aby zadal znak, který ani nemá na klávesnici. Například na arabských klávesnicích není levá ({) a pravá složená závorka (}). Výpis invariantní znakové sady uvádí téma Invariantní znaková sada.

Při kompilaci nepoužívejte pole, která by obsahovala zprávy nebo data citlivá na kulturní zvyklosti.

Chcete-li dosáhnout lepšího výkonu, volejte v případě potřeby moduly závislé na podpoře národního jazyka pevným jménem jako literál (ale na základě seznamu knihoven), a nikoliv jako proměnné pole, obsahující jméno programu. To umožňuje, aby vaše aplikace volala moduly z různých knihoven na základě přiřazeného seznamu knihoven.

Chcete-li umožnit uživatelům, aby pracovali s aplikací v jazyce a podle zvyklostí jejich kultury, zadejte editační hodnoty (např. datum, čas a příslušné oddělovače) jako závislé na jazyku a zemi či regionu. Potom je můžete načíst podle informací v uživatelském profilu. Parametry jsou LANGID (identifikátor jazyka) a COUNTRYID (identifikátor země nebo regionu). Informace citlivé na kulturní zvyklosti je potřeba načíst pouze jednou při spuštění programu. Můžete to provést pomocí počátečního programu CL nebo programu ve vyšším programovacím jazyku a připravit je jako:

- Parametry operace volání.
- Parametry lokální datové oblasti (LDA).
- Zaváděcí tabulky programu.

Použití počátečního programu vám umožňuje nastavit atributy úlohy uživatele tak, aby zajišťovaly konzistentní aplikaci, např. operační systém OS/400 nebo jiný licencovaný program.

Další informace o vyšších programovacích jazycích uvádějí témata:

- CCSID jazykových kompilátorů
- Správce relace
- Pokyny pro produkt ILE C/400^(R)

- Třídící posloupnost ILE RPG
- Třídící posloupnost ILE COBOL
- Třídící posloupnost DB2 a SQL
- Třídící posloupnost iSeries Access

CCSID jazykových kompilátorů: Některé jazykové kompilátory očekávají, že syntaktické operátory a konvence pojmenování pro zdrojový kód budou v CCSID 00037. (Viz dokumentace k jazykovému kompilátoru, který používáte.) V případě těchto kompilátorů může dojít k nesprávnému mapování, pokud je zdroj kompilován s jiným CCSID než 00037 nebo 65535. Je nutné se ujistit, že tyto kompilátory přijímají jakékoliv proměnné znaky používané v syntaxi jazyka v CCSID 00037.

Kompilátory jazyků ILE

Při kompilaci programů v jazycích ILE C/400, ILE RPG for iSeries nebo ILE COBOL for iSeries je zdroj z databázových zdrojových souborů konvertován na CCSID primárního zdrojového souboru.

Kompilátory těchto jazyků mohou pracovat se syntaktickými operátory ve většině CCSID. Rovněž mohou ovládat konvence pojmenování pro zdrojový kód ve většině CCSID.

Kompilátory jazyků jiných než ILE

Při kompilaci programů v jazycích CL, RPG nebo COBOL, které nejsou ILE, je zdroj z databázových zdrojových souborů konvertován na CCSID úlohy.

Jestliže nechcete používat jména, konstanty nebo literály konvertované na CCSID úlohy, můžete CCSID vaší úlohy změnit na 65535. Vaše konstanty, jména a literály zůstanou nedotčeny.

Poznámka: Procedury REXX/400 a v nich kódovaná data literálů nejsou konvertovány na CCSID úlohy.

1. příklad

Následující příklad je ukázkou programu RPG, který není ILE. Jedná se o anglický zdroj v systému ve Spojených státech.

```
* Zdroj RPG      (Zdrojový soubor vytvořený s CCSID 00037, ale
*                s příznakem CCSID 65535)
FFILE1 IF E          DISK          80
C                READ FILE1
C* Test char
C*
C          FLD1      IFEQ '$'
C                ...
C* Move char
C*
C                MOVE FLD1          FLD$
C                ...
C*
C                SETON              LR
```

2. příklad

Ve Finsku se program z prvního příkladu nezkompiluje, neboť jméno pole FLD\$ obsahuje proměnný znak (symbol dolaru). Proměnný znak představuje v kódové stránce jiné než 00037 odlišnou kódovou pozici. Následující obrázek je ukázkou stejného programu RPG, který není ILE, jako byl zdroj v americké angličtině, ale v systému ve Finsku (CCSID 278).

```
* Zdroj RPG      (Zdrojový soubor vytvořený s CCSID 00037, ale
*                s příznakem CCSID 65535)
FFILE1 IF E          DISK          80
C                READ FILE1
```

```

C* Test char
C*
C          FLD1      IFEQ ''
C          ...
C* Move char
C*
C          MOVE FLD1      FLD
C          ...
C*
C          SETON                      LR

```

3. příklad

Tuto chybu můžete opravit tak, že změníte CCSID souboru na 00037 a CCSID úlohy nastavíte na 00278 (pro Finsko). Následující příklad ukazuje změněný soubor, jak vypadá anglický zdroj ve Finsku.

```

* Zdroj RPG      (Zdrojový soubor vytvořený s CCSID 00037 a s příznakem
*                CCSID 00037)
FFILE1 IF E          DISK          80
C          READ FILE1
C* Test char
C*
C          FLD1      IFEQ '$'
C          ...
C* Move char
C*
C          MOVE FLD1      FLD$
C          ...
C*
C          SETON                      LR

```

Správce relace: Pro všechny aplikace, které používají správce relace, musíte zajistit, aby v toku výstupních dat nebyly žádné hodnoty X'3F'. Operační systém OS/400 používá hodnoty X'3F' k vymazání obrazovky.

Obecná třídící posloupnost

Třídící posloupnost použitá v programu může mít vliv na logiku programu. Následující obrázek ukazuje takový příklad.

Při použití třídící posloupnosti se sdílenými váhami Latin 1 je test znaku 3 ekvivalentní s testem znaku 4 (nejsou zobrazeny všechny znaky). Když se použije hexadecimální nebo jedinečné třídění, jsou naprosto rozdílné. Následující příklad představuje program v jazyku RPG, který používá různé třídící posloupnosti.

```

* Zdroj RPG      (Program vytvořený s třídící posloupností Latin 1)
*
C* Test znaku 3
C*
C          FLD1      IFEQ 'a'
C          ...
C* Test znaku 4
C*
C          FLD1      IFEQ 'a'
C          FLD1      OREQ 'A'
C          FLD1      OREQ ''
C          FLD1      OREQ ''
C          ...
C*
C          SETON                      LR

```

Jestliže je při kompilaci programu zadán parametr SRTSEQ s hodnotou *JOB RUN a parametr LANGID s hodnotou *JOB RUN, není v době kompilace známá tabulka třídící posloupnosti, která se použije v době běhu programu.

Pro licencované programy ILE C/400, DB2 Query Manager a SQL Development Kit for iSeries platí další speciální pokyny.

Pokyny pro produkt ILE C: Při kompilaci programů pomocí ILE C mějte na paměti, že:

- Zdrojový soubor můžete kompilovat v libovolné kódové stránce EBCDIC kromě stránky 00290.
- Je-li identifikátor CCSID 65535, předpokládá se kódová stránka 00037.
- Všechny sekundární zdrojové soubory se převedou na CCSID primárního zdrojového souboru.

Poznámka: I když se většina sekundárních zdrojových souborů převede na CCSID primárního zdrojového souboru, nejsou některé konverze podporovány. Požadujete-li podporu pro nepodporovanou konverzi CCSID, spojte se se servisním zástupcem IBM.

- Je-li CCSID sekundárních zdrojových souborů 65535, nedojde k žádné konverzi.
- Všechny moduly se vytvářejí v kódové stránce primárního zdrojového souboru. Modul je objekt operačního systému OS/400, který může být kolekcí jedné nebo více procedur a jedné nebo více definic pro externí nebo interní proměnné. Modul se kompiluje ze zdrojového kódu.
- Při vázání modulů s různými CCSID ke konverzi nedochází a mohou se vyskytnout nepředvídatelné výsledky.
- Pro C znaky, které nejsou k dispozici na všech klávesnicích, můžete použít podporu pro znakovou náhradu. Znaková náhrada obvykle používá invariantní znaky pro zobrazení proměnných znaků. Například levá hranatá závorka ([]) je znázorněna znakem (.)

Funkce knihovny run-time ILE C, které analyzují řetězce obsahující proměnné znaky, používají hodnotu kódové pozice proměnného znaku asociovanou s CCSID úlohy.

Třídící posloupnost ILE RPG: Licencovaný program ILE RPG for iSeries umožňuje uživatelům specifikovat tabulku třídící posloupnosti a používat ji při operacích porovnání s nenumernickými daty. Pro každý z podporovaných jazyků jsou se serverem dodány dvě tabulky (se sdílenými váhami a s jedinečnými váhami). Máte-li k dispozici podporu třídících posloupností, můžete na základě existujících tabulek třídících posloupností vytvořit své vlastní tabulky.

Specifikace řízení jsou specifikace, které kompilátoru ILE RPG for iSeries poskytují informace o vašem programu a vašem serveru. Třídící posloupnost, která se používá v programech ILE RPG for iSeries, určují tyto faktory:

- Specifikace řízení.
- Parametr SRTSEQ (tabulka třídící posloupnosti) v příkazech Vytvoření modulu RPG a Vytvoření vázaného RPG programu.
- Parametr LANGID (identifikátor jazyka) v příkazech Vytvoření modulu RPG a Vytvoření vázaného RPG programu.

V poli alternativní porovnávací posloupnosti (ALTSEQ) ve specifikacích řízení jsou možné tyto hodnoty:

mezera

V programu RPG se nepoužívá žádná alternativní porovnávací posloupnost. Používá se normální porovnávací posloupnost. Kompilační volby SRTSEQ a LANGID jsou ignorovány.

***NONE**

V programu RPG se nepoužívá žádná alternativní porovnávací posloupnost. Používá se normální porovnávací posloupnost. Kompilační volby SRTSEQ a LANGID jsou ignorovány.

***SRC** V programu RPG se používá alternativní porovnávací posloupnost podle tabulek zadaných na konci programu RPG. Tabulka alternativní porovnávací posloupnosti se zavádí během kompilace a provádí se podle ní řazení, třídění, porovnávání a zpracování odpovídajících polí.

Kódy operací SORTA a LOOKUP nepoužívají specifikované tabulky alternativních porovnávacích posloupností.

Parametry SRTSEQ a LANGID v příkazech Vytvoření modulu RPG a Vytvoření vázaného RPG programu jsou ignorovány.

***EXT** Alternativní porovnávací posloupnost je uvedeny mimo program RPG. Kompilátor RPG importuje externí tabulku třídící posloupnosti na základě parametrů SRTSEQ a LANGID v příkazech Vytvoření modulu RPG a Vytvoření vázaného RPG programu.

Funkce SORTA a LOOKUP s poli a tabulkami nabývají účinnosti v době kompilace a zpracování pouze tehdy, když ve specifikacích řízení uvedete D.

Tabulku třídící posloupnosti, která se má použít v programu, lze určit během kompilace nebo v době běhu úlohy. Pokud parametr SRTSEQ v příkazech Vytvoření modulu RPG a Vytvoření vázaného RPG programu:

- Je nastaven na hodnotu *HEX, nepoužije se žádná tabulka třídící posloupnosti.
- Uvádí jméno tabulky, uloží se právě tato tabulka s daným objektem typu program a použije se, až se úloha spustí. Předvolené tabulky třídících posloupností dodávané systémem pro podporované jazyky najdete v tématu Tabulky třídících posloupností.
- Je nastaven na hodnotu *LANGIDSHR nebo *LANGIDUNQ, je tabulka sdílené nebo jedinečné váhy pro jazyk určená parametrem LANGID uložena s objektem typu program. Seznam platných identifikátorů jazyka uvádí téma Identifikátory země/oblasti.
- Je nastaven na hodnotu *JOB, použije se k určení třídící posloupnosti parametr SRTSEQ kompilační úlohy. Tabulka je uložena s daným objektem typu program.
- Je nastaven na hodnotu *JOBRUN, je třídící posloupnost, která se má použít, určena atributy úlohy spouštějící kompilovaný program. Pokud se atribut úlohy SRTSEQ odkazuje na LANGID, použije se LANGID uložení s daným objektem typu program. Je-li LANGID uložení s programem také *JOBRUN, použije se LANGID úlohy, která spustila program.

Poznámky:

1. Jestliže tabulka, která se má během kompilace uložit s objektem typu program, neexistuje, použije se tabulka definující hexadecimální třídící posloupnost označená CCSID 65535.
2. Pokud jsou tabulka třídící posloupnosti a CCSID úlohy, která spouští daný program, odlišné, je tabulka konvertována na CCSID úlohy.

Kódy operací SORTA a LOOKUP

Implementace kódů operací porovnání, zpracování odpovídajících polí a řídicích polí s tabulkami třídících posloupností je stejná pro alternativní porovnávací posloupnost i pro podporu třídících posloupností. Kódy operací porovnání jsou ANDxx, COMP, CABxx, CASxx, DOUxx, DOWxx, IFxx, ORxx a WHxx. S kódy operací SORTA a LOOKUP jsou poskytovány další funkce:

SORTA

Data v poli jsou tříděna podle tabulky třídící posloupnosti.

LOOKUP

Za účelem zajištění správného vyhledávání v tabulkách se s argumenty vyhledávání v polích a tabulkách používá tabulka třídící posloupnosti.

Argumenty vyhledávání jsou s prvkem tabulky nebo pole porovnávány pomocí dané tabulky třídící posloupnosti.

Data v poli nebo tabulce jsou pomocí tabulky třídící posloupnosti ověřována vždy, když je zadána vzestupná nebo sestupná posloupnost. Pokud hodnoty parametrů SRTSEQ a LANGID rozhodnou o tom, že se tabulka třídící posloupnosti zavede znovu v době běhu programu, zavedou se při kompilaci prvky pole nebo tabulky bez ověření posloupnosti. Ověření posloupnosti se provede podle tabulky třídící posloupnosti v době běhu programu.

Třídící posloupnost ILE COBOL: Licencovaný program ILE COBOL for iSeries používá podporu třídících posloupností těmito způsoby:

- Příkaz Vytvoření modulu COBOL
- Příkaz Vytvoření vázaného programu COBOL
- Klauzule PROCESS
- Klauzule ALPHABET

Licencovaný program ILE COBOL for iSeries používá tabulky třídících posloupností, které jsou dodávány systémem nebo uživatelem.

Příkazy Vytvoření modulu COBOL a Vytvoření vázaného programu COBOL

Tyto CL příkazy mají dvě kompilační volby vztahující se k podpoře třídících posloupností: parametr SRTSEQ a parametr LANGID. Parametr SRTSEQ umožňuje uživateli specifikovat kteroukoliv ze systémem nebo uživatelem dodávaných tabulek třídících posloupností uložených v zadané knihovně. Můžete zadat, zda má být tabulka třídící posloupnosti získána v době kompilace nebo během provádění programu. Také se můžete rozhodnout mezi tabulkami se sdílenými váhami a jedinečnými váhami.

Pomocí parametru LANGID můžete zadat jeden ze systémem definovaných identifikátorů jazyka nebo můžete zadat, aby se tento parametr definoval v době běhu programu.

Význam parametrů SRTSEQ a LANGID v příkazech Vytvoření modulu COBOL a Vytvoření vázaného programu COBOL je stejný jako v příkazech Vytvoření modulu RPG a Vytvoření vázaného programu RPG tak, jak je to popsáno v tématu "Třídící posloupnost ILE RPG" na stránce 48.

Příkaz PROCESS

V příkazu PROCESS mohou být uvedeny volby podpory třídících posloupností. Syntax tohoto příkazu je shodná s příkazem Vytvoření modulu COBOL a Vytvoření vázaného programu COBOL. Jediným rozdílem je, že v případě předdefinovaných hodnot se hodnoty v parametrech příkazu PROCESS zadávají bez hvězdičky (*). Jakékoliv volby uvedené v příkazu PROCESS potlačí odpovídající volby v příkazech Vytvoření modulu COBOL a Vytvoření programu COBOL.

Klauzule ALPHABET

Jméno abecedy v klauzuli ALPHABET odstavce SPECIAL-NAMES může využívat volbu NLSSORT. Pro alternativní porovnávací posloupnosti použijte parametry SRTSEQ a LANGID kompilátoru. Jinak je význam stejný jako u volby NATIVE.

Následující řádky v jazyce COBOL jsou ovlivněny volbou NLSSORT:

- fráze PROGRAM COLLATING SEQUENCE odstavce OBJECT-COMPUTER
Při vyhodnocování výsledku nenumernických porovnávání musí být v této frázi uveden odkaz na jméno abecedy, aby program mohl používat uvedené volby třídících posloupností. Tato volba se vztahuje také na nenumernické třídění nebo operaci zatřídění. Jinak se použije hexadecimální porovnávací posloupnost.
- ALPHABET CLAUSE v odstavci SPECIAL-NAMES
Tato klauzule by měla uvádět volbu NLSSORT.
- COLLATING SEQUENCE v příkazu MERGE (nebo SORT)
Tato fráze se používá k zadání porovnávací posloupnosti, která se má použít pro nenumernická porovnávání pro jméno dat klíče v operaci MERGE nebo SORT. Je-li vynechána, definuje klauzule PROGRAM COLLATING SEQUENCE v odstavci OBJECT-COMPUTER porovnávací posloupnost, která se má použít. Pokud není uvedena ani jedna, použije se hexadecimální porovnávací posloupnost.
- jména nenumernických relací a podmínek
Zvolená tabulka třídící posloupnosti ovlivňuje výsledek určitých příkazů s využitím jmen nenumernických relací a podmínek: EVALUATE, IF, PERFORM...UNTIL, SEARCH a START. Pravdivost hodnot nenumernických porovnání závisí na odpovídajících váhách znaků v tabulce třídící posloupnosti. Pokud

například zadáte tabulku s jedinečnými váhami (LANGIDUNQ) pro francouzštinu (Latin 1), bude následující tvrzení pravdivé pro jedinou hodnotu proměnné ITEM-1, e.

```
IF ITEM-1 = "e"
```

Jestliže zadáte tabulku se sdílenými váhami (LANGIDSHR) pro francouzštinu (Latin 1), bude stejné tvrzení pravdivé pro několik hodnot proměnné ITEM-1. Všechny mají shodnou sdílenou váhu 77:

malé písmeno e (e), velké písmeno e (E),
malé písmeno e s ostrým přízvukem (acute) (é), velké písmeno e s ostrým přízvukem (acute) (É),
malé písmeno e s tupým přízvukem (grave) (è), velké písmeno e s tupým přízvukem (grave) (È),
malé písmeno e se stříškou (ë), velké písmeno e se stříškou (Ë),
malé písmeno e s přehláskou (ê), velké písmeno e s přehláskou (Ê)

Třídící posloupnost DB2 a SQL:

U interaktivního SQL je možné u příkazu STRSQL zadat parametry SRTSEQ a LANGID. Později je možné tyto parametry změnit pomocí služeb relace pro interaktivní obrazovky.

Tabulky třídící posloupnosti se používají pro všechna porovnání řetězců. Porovnání řetězců se provádí u těchto příkazů SQL:

- klauzule ORDER BY
- klauzule WHERE
- klauzule GROUP
- klauzule HAVING
- klauzule UNION a UNION ALL
- klauzule DISTINCT
- predikát BETWEEN
- predikát IN
- predikát LIKE
- skalární funkce MIN a MAX
- sloupcové funkce MIN a MAX

Kromě toho se všechny indexy nebo pohledy vytvořené příkazy CREATE INDEX nebo CREATE VIEW vytvářejí podle specifikované tabulky třídící posloupnosti.

Produkt DB2 Query Manager a produkt SQL Development Kit for iSeries

Produkt DB2 Query Manager a SQL Development Kit for iSeries nepředpokládá při předkompilaci zdroje žádný konkrétní CCSID. Předpokládá se, že v CCSID zdrojového souboru může být zakódován libovolný proměnný znak v syntaxi jazyka (jako například znaménko negace

```
¬  
).
```

Má-li, například, zdrojový soubor CCSID 00037, je znaménko negace

```
¬
```

správně interpretováno jako kódová pozice X'5F'. Jestliže však má zdrojový soubor CCSID 00500, je znaménko negace

```
¬
```

správně interpretováno jako kódová pozice X'BA'.

Literál se ukládá v CCSID zdrojového souboru.

Protože produkt DB2 Query Manager i produkt SQL Development Kit for iSeries volá při tvorbě programu SQL kompilátor odpovídajícího jazyka, je nutné brát také v úvahu obecné směrnice pro vyšší programovací jazyky.

Třídící posloupnost iSeries Access: Třídící posloupnost můžete zadat ve funkcích produktu iSeries Access, vzdáleném SQL a v přenosové funkci. Když provádíte dotazy do databází a tabulek SQL na serveru, můžete specifikovat tabulky třídících posloupností dodané systémem nebo uživatelem.

Podpora vzdáleného SQL

Při provádění dotazu můžete zadat způsob, jakým se mají vybraná data třídit. Za tím účelem musí být zadána třídící pole v klauzuli ORDER BY. Zadaná třídící posloupnost se používá také v těchto klauzulích:

- klauzule WHERE
- klauzule GROUP BY
- klauzule HAVING
- klauzule JOIN BY
- klauzule UNION
- klauzule DISTINCT
- predikát IN
- predikát LIKE
- predikát BETWEEN
- predikát RANGE
- funkce MAX
- funkce MIN

Vlastní tabulka třídící posloupnosti se načítá z atributů úlohy daného uživatele. Změna uživatelského profilu nebo atributů úlohy může mít vliv na parametry SRTSEQ a LANGID.

Podpora přenosové funkce

Při přenosu dat ze serveru iSeries na pracovní stanici můžete uvést třídící posloupnost, která se má použít na vybraná data. Tabulka třídící posloupnosti se rovněž používá v následujících operacích porovnání řetězců:

- klauzule WHERE
- klauzule GROUP BY
- klauzule HAVING
- klauzule JOIN BY
- predikát IN
- predikát LIKE
- predikát BETWEEN
- funkce MAX
- funkce MIN

V příkazu OPTION můžete zadat následující parametry související s třídící posloupností:

- SRTSEQ (tabulka třídící posloupnosti)
 - *JOB
 - *HEX
 - *LANGIDSHR
 - *LANGIDUNQ

- *LIBL/jméno-tab-tříd-posl
- *CURLIB/jméno-tab-tříd-posl
- jméno-knihovny/jméno-tab-tříd-posl
- LANGID (identifikátor jazyka)
 - *JOB
 - identifikátor-jazyka

Odpovídající třídící posloupnost můžete vybrat prostřednictvím voleb na obrazovkách produktu iSeries Access.

Jak psát globalizované aplikace používající obousměrná data

Při vývoji aplikací podporujících národní prostřední mějte na paměti následující omezení:

- Rozvržení obrazovky pro obousměrný jazyk
Prezentace dat by měla mít orientaci zprava doleva. Literály by se měly objevit na pravé straně polí, která popisují. Následující příklady ilustrují obrazovku v americké angličtině s orientací zleva doprava a stejnou obrazovku orientovanou zprava doleva.

Rozvržení obrazovky v americké angličtině zleva doprava

```
Display employee record (DSPEMPRCD)
Type choices, press enter.

Employee code ..... Code, *ALL
Field name ..... Name, *ALL
File name ..... Name
Library name ..... Name, *LIBL
Output to ..... *CONS, *PRINT
```

Rozvržení obrazovky v americké angličtině zprava doleva

```
(DSPEMPRCD) drocer eeyolpme yalpsiD
               .retne sserp ,seciohc epyT

*ALL ,edoC ..... edoc eeyolpmE
*ALL ,emaN ..... eman dleiF
   eman ..... eman eliF
*LIBL ,emaN ..... eman yrarbiL
*CONS , *PRINT ..... ot tuptuO
```

- Dlouhá pole v obousměrných jazycích
Vyvarujte se definování vstupních polí, která by přesáhla jednu řádku. Když se pole zobrazí nebo vytiskne jako celek, nebyl by výsledek takový, jaký uživatel zamýšlel.
- Umístění proměnných v obousměrných jazycích
Aplikace musí připouštět, aby se proměnné objevovaly v libovolném pořadí. Vezměme například následující zprávu v angličtině:
File &1 in library &2 not found

Po překladu do jiného jazyka by zpráva mohla vypadat takto:

```
dnuof ton &2 yrarbiL ni &1 eliF
```

V tomto případě je proměnná 2 před proměnnou 1.

- Klíčová slova CHECK(RL) a CHECK(RB) v obousměrných jazycích
Tyto volby jsou platné pouze u terminálů, u nichž je možný pohyb zprava doleva, s následujícími omezeními:
 - Indikátory volby neplatí u řídicích kódů kurzoru.
 - U těchto klíčových slov neplatí CHECK(RZ) a CHECK(RB).
 - Pole, které přesahuje jednu řádku, vydá varovnou zprávu.
 - Kontrolní číslice pro kontrolu modulo je v bajtu, který je v poli nejvíce vpravo.
 - CHECK(RL) platí pouze pro znaková pole.
- Online informace v obousměrných jazycích
Speciální obousměrné příznaky mají omezení. Při kombinování informací online nápovědy z několika skupin panelů, u nichž nemá příznak BIDI stejnou hodnotu, musí koncový uživatel použít sekvenci horkých kláves, aby přečetl informace online nápovědy s opačnou orientací.
- CCSID u obousměrných jazyků
Vzhledem k tomu, že obousměrné jazyky mají speciální jedinečné znakové sady, není vzájemná výměna dat do jiných jazyků realizovatelná. Můžete však potřebovat použít mapování dat mezi toky dat EBCDIC a ASCII. Toto mapování je nutné například tehdy, když používáte architekturu DRDA^(R) (Distributed Relational Database Architecture^(R)).
Jsou-li při vzájemné výměně dat v jazyce používajícím znaky latinky potřebné speciální znaky, které nejsou součástí invariantní znakové sady, použijte CCSID 00424 pro hebrejštinu a CCSID 00420 pro arabštinu, aby mapování proběhlo. Seznam podporovaných CCSID najdete v tématu CCSID.

Další informace o práci s obousměrnými daty uvádí téma Práce s obousměrnými daty. Toto téma také obsahuje kontrolní seznam pro obousměrná data, který poskytuje pokyny pro tvorbu aplikací s obousměrnou podporou.

Použití katalogů zpráv

Operační systém OS/400 může používat katalogy zpráv k uchování zpráv. Zprávy jsou katalogu zpráv seskupeny do sad. Každá zpráva má své jedinečné číslo v rámci sady. Katalog zpráv můžete vytvořit jako proudový soubor, člen zdrojového souboru nebo typ objektu uživatelské oblasti z jednoho nebo několika zdrojových souborů.

Jelikož můžete katalogy zpráv uložit jako proudové soubory, můžete používat adresáře za účelem oddělení zpráv pro určité produkty nebo pro verze národních jazyků.

Vytvoření nebo aktualizace katalogu zpráv pomocí příkazů GENCAT a MRGMSGCLG

K vytvoření nebo aktualizaci katalogu zpráv slouží příkazy GENCAT (Generování katalogu zpráv) a MRGMSGCLG (Začlenění katalogu zpráv). Když už katalog zpráv existuje, povede použití těchto příkazů k aktualizaci katalogu na základě porovnání původních zpráv vůči zprávám ve zdroji. Nový text zprávy nahradí určité původní zprávy, aniž by došlo ke změně ostatních zpráv v sadě. Pomocí těchto příkazů můžete zprávy přidávat do stávající sady zpráv nebo je z ní mazat. Rovněž můžete mazat sady zpráv ze stávajícího katalogu zpráv.

Další informace

Další informace o katalogách zpráv obsahují tato témata:

- Zdroj pro katalogy zpráv
- Otevření, vyjmutí a zavření katalogů zpráv

Zdroj pro katalogy zpráv: Zdroj pro katalog zpráv je buď zdrojový fyzický soubor, proudový soubor, nebo několik souborů. Zdroj obsahuje pole pro definici čísel sad, čísel zpráv, textu zprávy nebo pro specifikaci sad určených k výmazu. V následujících tématech najdete další informace a příklady týkající se katalogů zpráv.

Zdrojový formát katalogu zpráv

Katalog zpráv obsahuje pět polí pro zdrojové řádky textu zprávy. Jednotlivá pole jsou od sebe oddělena jedním znakem mezery. Jakékoliv další znaky mezer jsou považovány za součást dat následujícího pole. Další informace poskytuje téma Speciální znaky a escape sekvence (56see page).

Poznámka: Klíčová pole zadejte přesně tak, jak je to uvedeno níže, se znakem dolaru (\$) a malými písmeny. Definice maximální a minimální hodnoty jsou uloženy v knihovně QSYSINC/QSYS/LIMITS.

- **\$ komentář**

Řádka, která začíná \$, po němž následuje jedna nebo dvě mezery, je považována za komentář. Řádka komentáře by měla být umístěna hned za zprávou, k níž se vztahuje. Komentáře pro celou sadu umístěte hned pod direktivu \$set ve zdrojovém souboru.

- **\$quote C**

Tato řádka specifikuje volitelný znak uvozovek C, který se používá k ohraničení textu zprávy. Tento znak umožňuje zviditelnění koncových mezer či nulových (prázdných) zpráv ve zdrojové řádce zprávy. Standardně, případně při dodání prázdné direktivy \$quote, se nerozpoznává žádné uvození textu zprávy.

- **\$set n komentář**

Tato řádka specifikuje identifikátor sady pro zprávy, s nímž se má pracovat, dokud se neobjeví další \$set nebo konec souboru. n označuje identifikátor sady tvořený číslem mezi 1 a NL_SETMAX. V rámci jednoho zdrojového souboru uvádějte identifikátory ve vzestupném pořadí. Jejich posloupnost nemusí být souvislá. Se znakovým řetězcem, který následuje po identifikátoru sady, se zachází jako s komentářem a ignoruje se.

- **\$delset n komentář**

Tato řádka vymaže sadu zpráv n z existujícího katalogu zpráv. n udává číslo sady. S daty, která následují po čísle sady, se zachází jako s komentářem. Oba identifikátory \$set a \$delset mohou být uvedeny ve zdroji katalogu zpráv nebo v položkách pole.

- **m text zprávy**

m specifikuje identifikátor zprávy definovaný číslem mezi 1 a NL_SETMAX. Text zprávy se v katalogu zpráv uchovává s identifikátorem zprávy m, s identifikátorem sady, který byl uveden v poslední direktivě \$set. Pokud je text zprávy prázdný a je obsažen znak mezery jako oddělovač polí, uchovává se v katalogu zpráv prázdný řetězec. Stávající zprávy se z katalogu vymažou, pokud řádka zprávy neobsahuje oddělovač polí nebo text zprávy (MESSAGE TEXT) ani posun na novou řádku (NEWLINE), nebo v případě, že za řádkou zprávy následuje návrat vozíku (CR). Identifikátory zpráv musejí být ve vzestupném pořadí, nesouvislé a v rámci jedné sady. Délka textu zprávy se musí pohybovat v rozmezí 0 až NL_TEXTMAX.

Poznámka: Prázdné řádky ve zdrojovém souboru textu zprávy budou ignorovány.

Programovací formát pro zprávy

ZPRÁVY by měly splňovat tato doporučení:

- Poslední řádka všech zpráv by měla končit znaky \n.
- Druhá a zbývající řádky zprávy by měly začínat znaky \t, označujícími nastavení kurzoru.
- Všechny řádky zpráv, které pokračují na další řádce, by měly být ukončeny znaky \n\ označujícími, že zpráva pokračuje na další řádce.
- Měly by být vynechány uvozovky na začátku nebo na konci řádky. Uvozovky vymezují začátek a konec celé zprávy.

Použití několika zdrojových souborů

V parametru zdrojového souboru můžete uvést několik zdrojových souborů. Zprávy, které jsou ve všech těchto souborech obsaženy, musí splňovat stejná pravidla pro sady a zprávy, jako by byly definovány v jednom zdrojovém souboru. Například první zdrojový soubor obsahuje zprávy v sadách 1 až 3. Další zdrojový soubor musí začínat sadou 3 a mít číslo zprávy větší, než měla poslední zpráva v prvním

zdrojovém souboru. Pokud tomu tak není, musí obsahovat sady, které začínají vyšším číslem, než bylo nejvyšší číslo (sada 3) v předešlém zdrojovém souboru.

Nahrazení zpráv

Zprávy ve stávajícím katalogu zpráv je možné nahradit uvedením zdrojového souboru, jenž obsahuje stejné číslo sady a číslo zprávy, jako má text zprávy, který chcete nahradit. Všechny ostatní zprávy ve zdrojovém souboru zůstávají stejné. Chcete-li aktualizovat hodnotu \$QUOTE v katalogu, použijte stejný znak \$QUOTE v následujících zdrojových souborech.

Příklad zdroje pro katalog zpráv

Dále si můžete prohlédnout ukázkový formát pro zdroj, který se používá k vytvoření katalogu zpráv. Uvozovky vymezují jednotlivé zprávy. Z textu zprávy uloženého v katalogu zpráv byly odstraněny nadbytečné mezery. Tento příklad popisuje tři sady zpráv. Sada 2 je vymazána, zatímco sady 1 a 3 zůstanou nadále uloženy v katalogu zpráv.

```
$ Zprávy pro můj nový produkt
$quote "

$set 1

1 "Došlo k chybě.\n"
$ Další zpráva pokračuje na další řádce.
2 "Toto je velmi dlouhá zpráva, \n\
\t pro jejíž zobrazení je nezbytná další řádka. \n"
3 "Zadejte hodnotu větší než %d.\n"
4 "Soubor %c nelze v tuto chvíli použít.\n"

$set 2
1 "Došlo k chybě %d. \n"
2 "Příznak nebyl nastaven.\n"
3 "Počet argumentů musí být %d.\n"

$set 4
1 "Před použitím tohoto příkazu musíte \
nastavit správné hodnoty v okénku %c.\n"
2 "Nemáte správně aktivovanu podporu národního jazyka.\n"
10 "Zprávy by měly končit %c.\n"

$delset 2
```

Poznámka: Zpráva 2 v sadě 1 se zobrazí na dvou řádkách. Zpráva 1 v sadě 4 se zobrazí jako jednořádková zpráva.

Níže je uveden příklad použití příkazu MRGMSGCLG k vytvoření katalogu zpráv.

```
MRGMSGCLG CLGFILE('/MYPRODUCT/MESSAGES?US')
          SRCFILE('QSYS.LIB/MYLIB.LIB/MYSOURCE.FILE/US.MBR')
          CLGCCSID(*SRCCSID) SRCCSID(*SRCFILE)
          TEXT('Katalog zpráv pro USA')
```

V tomto příkladu se vytvoří katalog zpráv v proudovém souboru US v adresáři /MYPRODUCT/MESSAGES s využitím zdroje z knihovny MYLIB v souboru MYSOURCE a členu US. CCSID dat v katalogu zpráv je stejný jako příznak CCSID zdrojového souboru.

Speciální znaky a escape sekvence

Textové řetězce mohou obsahovat speciální znaky a escape sekvence definované v následující tabulce.

Popis speciálních znaků	Sekvence
\	\\
zpětný krok (backspace)	\b
návrat vozíku	\r
posun na novou stránku	\f
vodorovné nastavení kurzoru	\t
NEWLINE	\n
osmičkový bitový vzorec	\\ddd

Poznámka: Escape sekvence \\ddd sestává ze zpětného lomítka následovaného až třemi osmičkovými číslicemi, které určují hodnotu požadovaného znaku. Jestliže znak následující po zpětném lomítku není osmičková číslice, jsou zpětné lomítko i následující data zahrnuta jako součást textu.

Otevření, vyjmutí a zavření katalogů zpráv: Jakmile už máte vytvořený katalog zpráv, můžete používat následující funkce jazyka C:

CATOPEN()

Otevře katalog zpráv.

CATGETS()

Vyjme zprávu z katalogu zpráv s daným identifikátorem sady a identifikátorem zprávy.

CATCLOSE()

Zavře katalog zpráv.

C funkce CATOPEN otevře katalog zpráv. Pokud se ve jménu nenajdou žádné znaky lomítka (/), použijí se k nalezení zadaného katalogu proměnná prostředí NLSPATH a kategorie LC_MESSAGES. Jestliže jméno obsahuje jedno či více lomítek (/), interpretuje se jako jméno cesty ke katalogu, který se má otevřít.

Pokud neexistuje žádná proměnná prostředí NLSPATH nebo nelze katalog zpráv nalézt v zadané cestě NLSPATH, použije se předvolená cesta. Jestliže je hodnota příznaku oflag NO_CAT_LOCALE, může nastavení proměnné prostředí LC_MESSAGES ovlivnit předvolenou cestu. Je-li hodnota příznaku oflag nula, může ji ovlivnit i proměnná prostředí LANG.

Další informace o C funkcích a katalogách zpráv uvádí PDF ILE C/C++ Language Reference



Dodání globalizovaných aplikací

Když připravujete dodávku vaší globální aplikace, měli byste zvážit, jak by mohly otázky globalizace ovlivnit způsob instalace a použití aplikace u zákazníka. Následující témata se těmito otázkami stručně zabývají.

Hardwarová podpora vícejazyčných systémů

Hardware, v tomto kontextu znamená fyzické klávesnice, obrazovky, tiskárny a řadiče, které jsou součástí serveru iSeries. Míra, do jaké tento hardware podporuje národní jazyky, může představovat omezení stupně podpory, který s aplikací můžete poskytnout. Musíte si prostudovat příručky k hardwaru jiných firem nežli IBM a rozhodnout, jaká případná omezení tento hardware představuje.

Překlad znakových dat

Překlad znamená změnu významu znakových dat z jedné množiny pojmů, myšlenek a tvrzení v jednom lidském jazyce do kulturně podobného významu v jiném lidském jazyce. Chcete-li zajistit hladký průběh překladu, můžete se řídit některými základními pravidly. Podmnožinu těchto pravidel uvádí téma "Uživatelská rozhraní" na stránce 15.

Doručení vaší globalizované aplikace zákazníkům

Doručení aplikace zákazníkům zahrnuje proces balení, poskytnutí služeb, podpory a vyškolení uživatelů, jak s aplikací zacházet. Provedení tohoto procesu v různých zemích a světových kulturách je spojeno s nejrůznějšími úkoly. Další informace o procesech spojených s doručím aplikace zákazníkům podává téma "Proces balení a instalace" na stránce 6.



Vytištěno v Dánsku společností IBM Danmark A/S.