



Rendszerkönyvtárosok kézikönyve- Indexelés

20-as verzió

BIZALMAS INFORMÁCIÓ

Az ezen dokumentumban található információk az Ex Libris Ltd. és tagvállalatainak tulajdona, és az ezen információkkal történő minden fajta visszaélés vagy azok rosszhiszemű felhasználása pénzügyi veszteséget okoz. NE MÁSZOLJA LE EZEN INFORMÁCIÓKAT, AMENNYIBEN NEM RENDELKEZIK AZ EX LIBRIS LTD. ÍRÁSOS ENGEDÉLYÉVEL.

Ezen dokumentum korlátozott körű az Ex Libris Ltd-vel vagy annak tagvállalataival kötött szerződés alapján. A dokumentumban található információk kereskedelmi titkokat tartalmaznak, ezért bizalmas jellegű.

ELÁLLÁS

Az ezen dokumentumban található információk időnként megváltoznak. Győződjön meg arról, hogy a legfrissebb dokumentumot használja. Az Ex Libris Ltd-vel kötött szerződésben lefektetett jótállási elveken túl az ezen dokumentum használatára vonatkozó jótállási igény nem kérelmezhető. Eltérő megállapodás hiányában az Ex Libris nem felelős az ezen dokumentum használatából adódó közvetlen vagy közvetett károkért.

Ezen dokumentumban harmadik félre történő hivatkozás csupán kényelmi okokból szerepel. A harmadik fél által nyújtott információk nem részei az Ex Libris termékeknek, ezért azokért felelősséget sem vállal.

MÁRKANEVEK

Az "Ex Libris," az Ex Libris bridge to Knowledge, Primo, Aleph, Alephino, Voyager, SFX, MetaLib, Verde, DigiTool, Preservation, URM, Voyager, ENCompass, Endeavor eZConnect, WebVoyage, Citation Server, LinkFinder és LinkFinder Plus, és más nevek márkanevek vagy bejegyzett márkanevek és az Ex-Libris Ltd. vagy tagvállalatai tulajdonai.

Nevek vagy emblémák hiánya nem jelenti a szellemi jogokról történő lemondást.

Harmadik fél márkaneve az alábbiakban szerepelnek. Az Ex-Libris nem rendelkezik jogokkal ezekkel a márkanevekkel kapcsolatosan.

Az Oracle az Oracle Corporation bejegyzett márkaneve.

UNIX az Egyesült Államok és más országos bejegyzett márkaneve, az X/Open Company Ltd. kizárólagos engedélyével.

Microsoft, a Microsoft logo, MS, MS-DOS, Microsoft PowerPoint, Visual Basic, Visual C++, Win32, Microsoft Windows, a Windows logo, Microsoft Notepad, Microsoft Windows Explorer, Microsoft Internet Explorer, és a Windows NT a Microsoft Corporation bejegyzett védjegye, az ActiveX ugyanezen vállalat márkaneve.

Az Unicode és az Unicode logo az Unicode, Inc. bejegyzett márkaneve.

A Google a Google, Inc. bejegyzett márkaneve.

Copyright Ex Libris Limited, 2010. Minden jog fenntartva

Utolsó módosítás: 2010. Augusztus 2.

Weblap cím: <http://www.exlibrisgroup.com>

Tartalom

1 AZ INDEXELÉS ÁTTEKINTÉSE	6
2 AZ INDEXELÉSI FOLYAMAT	7
2.1 Indexek definiálása az ALEPH-ben	7
2.1.1 Új index definiálása	7
2.1.2 Mezők hozzárendelése az indexekhez	7
2.1.3 Index hozzáadása a pc_tab_sear.lng-hez	8
2.1.4 Index hozzáadása Web include fájlokhoz	8
2.2 Az indexelés folyamata	8
2.2.1 Az indexelési folyamat prioritásainak beállítása	9
2.3 Indexek építése	10
3 BESOROLÁSI INDEX	11
3.1 Besorolási elemek egységesítése	12
3.1.1 Normalizálás	12
3.1.2 Alkalmazott adatbázistáblák	13
3.2 Fő besorolási elem szervizek	14
3.2.1 Besorolási index (manage-02)	14
3.2.2 Pre-Enrich Bibliographic Headings Based on the Authority Database (manage-102)..	14
3.2.3 Tárgyszó alosztályok	14
3.2.4 Mikor futtassuk a besorolási indexet	14
3.3 További besorolási elem szervizek	15
3.3.1 Besorolási rekordok rendezése (p_manage_16)	15
3.3.2 Teljes hosszon rendezés (p_manage_17)	15
3.3.3 Számlálók építése logikai bázisokhoz (manage-32)	16
3.3.4 Rövid (Brief) rekordok módosítása (manage-35)	16
3.3.5 Rövid bibliográfiai rekord (manage-07)	17
3.3.6 Rendezési index (manage-27)	17
3.3.7 Kapcsolat nélküli besorolási rekordok törlése (manage-15)	17
3.4 Linkelési (kapcsolási) folyamat	18
4 SZÓINDEX	18
4.1 Szavak definiálása	20
4.2 Alkalmazott adatbázistáblák	20
4.3 Mikor építse újra a szóindexet	21
5 DIREKT INDEX	21

5.1 Direkt indexek egységesítése	22
5.2 Mikor módosítsa a direkt indexet	23
5.3 Alkalmazott adatbázistáblák.....	23
6 APAC INDEXELÉS.....	23
6.1 Besorolási indexelés	23
6.2 Szóindexelés.....	25
6.2.1 Az indexelt szöveg normalizálása	25
6.2.2 Szegmentáló rutinok definiálása indexeléshez	26
6.2.3 Szegmentáló rutinok definiálása kereséshez	27
6.2.4 Javasolt APAC indexelés és keresés szegmentálási rutin beállítások.....	28
7 INDEXELÉST TÁMOGATÓ FŐTÁBLÁK.....	32
8 RENDEZÉS ÉS SZÓTÖRDELÉS	35
8.1 Besorolási elemek és indexek rendezése.....	35
8.2 Példánylisták rendezése	41
8.2.1 Rendezési opciók	42
8.3 Szótördelés.....	44
9 BŐVÍTŐRUTINOK, TÁBLÁK ÉS BŐVÍTETT MEZŐK INDEXELÉSE	45
9.1 Rekordbővítés	46
9.2 Bővítőrutinok	49
9.3 Bővítésre vonatkozó táblák.....	90
9.3.1 Konfigurációs táblák (expand_doc_type)	90
9.3.2 tab_expand_split.....	92
9.3.3 tab_abbrev	93
9.3.4 tab_expand_duplicate_field	94
9.3.5 tab_expand_external	94
9.3.6 expand_doc_bib_z30.....	95
9.3.7 expand_doc_bib_z403	95
9.4 Bővítőmezők (virtuális mezők) indexelése	95
9.4.1 tab_expand_extract	96
9.4.2 tab_expand_join	97
9.4.3 tab_expand_join_simple	98
10 TOVÁBBI INDEXEK	98
10.1 Rövid bibliográfiai rekord (manage-07).....	98
10.2 Rendezési index (manage-27).....	99
10.3 Index aktualizálás – kiválasztott rekordokra (manage-40)	99

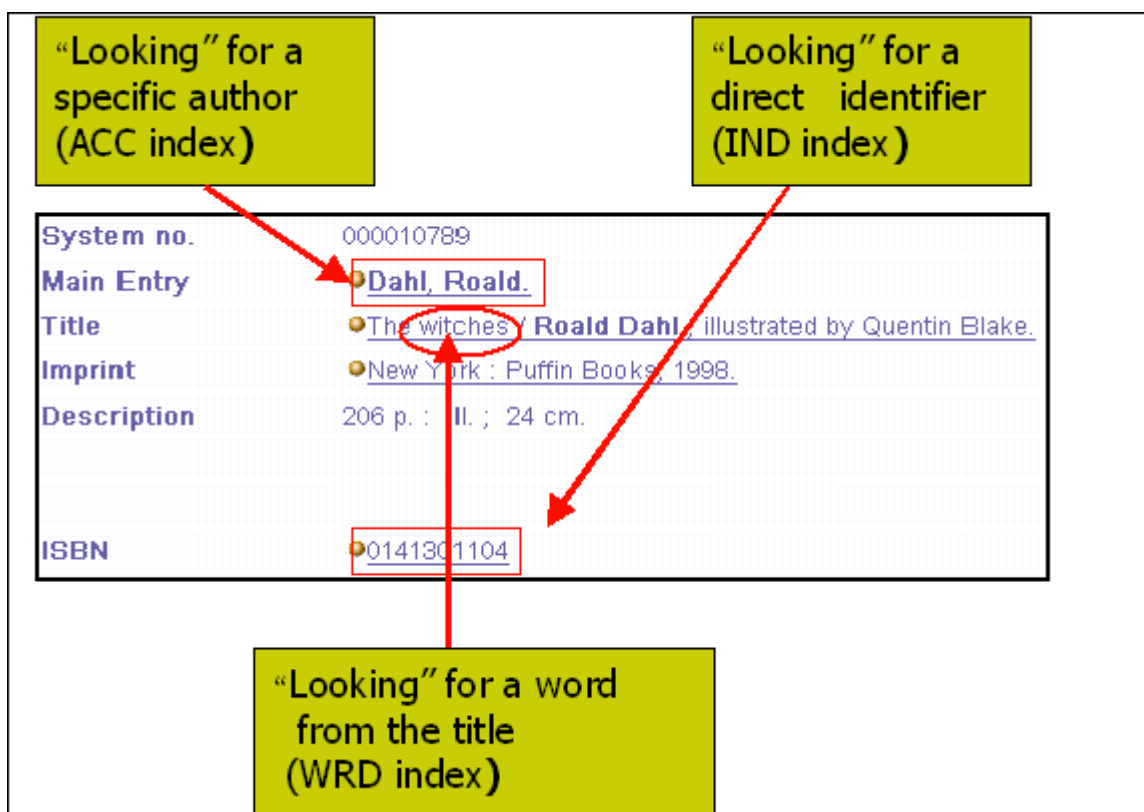
11 FELKÉSZÜLÉS AZ INDEX FELADATOKRA	99
11.1 Temp/scratch (ideiglenes) könyvtárak tisztítása	99
11.2 Oracle tárhely ellenőrzése	100
11.3 Zavart okozó feladatok megszakítása.....	100
12 PÁRHUZAMOS INDEXELÉS	100
13 INDEXELŐ SZERVIZEK	106
14 TOVÁBBI OLVASMÁNYOK	107

1 Az indexelés áttekintése

Az ALEPH a dokumentumrekordokhoz való hozzáférési utak (indexek) sokféle definiálását teszi lehetővé. Ezek a definíciók az egyes alkalmazásoknak megfelelően átalakíthatóak.

Indexek alapjai lehetnek konkrét mezők, almezők, mezők kombinációja, vagy konkrét dokumentummezőkből származó egyedi szavak.

A bibliográfiai adatbázis elérésének három fő módja van, három indextípuson keresztül: Besorolási index, Szó index és Direkt index. Minden egyes indextípusnak megvannak a hozzákapcsolt speciális alindexeik. Például a Besorolási indexhez (ACC) kapcsolódik egyebek mellett a TIT és az AUT alindex.



Az indexeket használhatják az OPAC-ban kereső végfelhasználók, valamint könyvtárosok belső célból (például feldolgozók listájához vagy riportok készítéséhez rekordok visszakeresésére).

Ez a fejezet főként rendszerkönyvtárosok számára készült. Az adat elérési módokat tekinti át. Index definíciókat és típusokat, index specifikációkat, indexelő eljárásokat és bővítő programokat tartalmaz.

Megjegyzés: a különböző indexelő szervizek a GUI Katalogizálás moduljában a Szerviz / OPAC indexek almenüben érhetőek el.

2 Az indexelési folyamat

2.1 Indexek definiálása az ALEPH-ben

Az indexek definiálásának három alaplépése az ALEPH-ben:

- A kódok és az indexek nevének definiálása a `tab00.lng` táblában.
- Mezőkódok hozzárendelése a különböző indexkódokhoz a rekord mezők és az indexek közötti kapcsolat meghatározásához.
- Az index elhelyezése a `tab/pc_tab_sear.lng` táblába és az `alephe/www_f_lng/` fájlalba helyezése azért, hogy lehetővé váljon az index böngészése/keresése a GUI és WebOPAC felület Kereső funkcióján keresztül.

2.1.1 Új index definiálása

Egy új index definiálása:

- Határozza meg, hogy milyen típusú indexet szeretne létrehozni: direkt, szó, vagy besorolási indexet.
- Adja az Indexet a `xxx01/tab/tab00.lng` táblához. Nézze meg a `tab00.lng` táblát és határozzon meg egy kódot az új indexnek (győződjön meg róla, hogy ez a kód még nincs használatban).

Megjegyzések:

- Annak ellenére, hogy a kód akár öt karakteres is lehet, általában csak három karakterből áll.
- Amennyiben az index direkt index, az „IND” kifejezést adja meg az index típusaként.
- Amennyiben az index besorolási index, az „ACC” kifejezést adja meg az index típusaként.
- Amennyiben az index szó index, keresse meg a következő W-nnn értéket és ezt használja az index típusaként. Az -nnn számoknak egyedinek és egymást követőeknek kell lenniük.
- A szó index típusának mindig W-vel kezdődnie.
- **A rendszer mindig biztosítja a rendszerszám szerinti BÖNGÉSZÉS és KERESÉS típusú hozzáférést, valamint a vonalkód alapján történő KERESÉS hozzáférést. Ezért, bár ezeket nem kell az indexelő táblában (tab11) definiálni, az indexnév 11-es oszlopban történő meghatározásához ezeket itt definiálni kell.**

2.1.2 Mezők hozzárendelése az indexekhez

Mezők hozzárendelése egy indexhez:

1. Adja hozzá az Indexet a `tab11` táblához.
2. Illessze be a `tab00.lng` táblában meghatározott kódot minden egyes mező mellé a `xxx01/tab/tab11` táblában, mely mezőt az indexbe szeretne küldeni.
3. Ha „IND” típusú indexet definiált a `tab00.lng` táblában, az indexet a `tab11_ind` táblába helyezze.
4. Ha „ACC” típusú indexet definiált, az indexet a `tab11_acc` táblába helyezze.
5. Ha „W-`nnn`” típusú indexet definiált, az indexet a `tab11_word` táblába helyezze.

2.1.3 Index hozzáadása a `pc_tab_sear.lng`-hez

Ha azt szeretné, hogy egy index kereshető legyen a GUI kereső funkciójával, akkor adja hozzá az `xxx01/tab/pc_tab_sear.lng` táblához.

Ha az indexhez böngésző módban szeretne hozzáférni, akkor illessze az „SC” (Scan) szekcióba. Az indexben szereplő kifejezések abc sorrendben jelennek meg, és görgethetőek. A besorolási elemek kifejezetten így használhatók.

Amennyiben olyan rekordokat szeretne keresni, amelyek indexelt kifejezései megegyeznek a keresőkérdelemmel, akkor illessze a FI (Find/keresés) szekcióba. A WORD, SYS és BAR indexek kifejezetten így használhatók.

2.1.4 Index hozzáadása Web include fájlokhoz

Ha azt szeretné, hogy egy index böngészhető legyen a WebOPAC felületén (abc sorrendbe rendezett keresési lista megjelenítése céljából), adja hozzá az indexet az `alephe/www_f_lng/` könyvtárban található `scan-include-2` fájlhoz.

A találati halmazban megjelenő rekordok visszakeresése érdekében adja hozzá az indexet vagy a `scan-include-2` fájlhoz, vagy az `alephe/www_f_lng/` könyvtárban található `find-code-include` fájlhoz.

2.2 Az indexelés folyamata

Az UE_01 háttérben futó eljárás frissíti a besorolási indexet és a szó indexet és a logikai bázis számlálót (`z0102`); a direkt index módosítása azonnal megtörténik.

Amikor egy katalógus rekord kerül az adatbázisba vagy módosul, akkor a rendszerszáma bekerül a `z07` táblába. Az UE_01 ellenőrzi a `z07` táblát és megvizsgálja, hogy van-e új/módosított, indexelésre váró rekord.

Miután az eljárás módosította a besorolási elemeket, és a szó indexeket, a `z07` rekord törlődik, és egy `z07a` rekord jön létre. A logikai bázis számláló frissítése után a `z07a` rekord törlődik. Egy zavartalanul működő rendszerben nem szabad túl sok rekordnak lennie a `z07` és `z07a` táblában.

Az `ue_01` eljárás három különböző al-eljárást tartalmaz:

- ue_01_a – ez az eljárás felel a besorolási elemek módosításáért (ezen felül frissíti a többit, valamint olyan indexeket és táblákat, mint a z13, z00R, z101 stb.).
- ue_01_word_parallel – ez az eljárás felel a szó indexért (teljesítmény megfontolásokból). A többi indexszel párhuzamosan építi a szavakat.
- ue_01_z102_index – ez az eljárás felel a z0102 rekordokért (logikai bázis számláló).

A logikai bázis számlálót (z0102 rekordok) az ue_08 eljárás frissítette. Megjegyzés: az ue_08 eljárás teljesítményét az ue_01_z0102_index al-eljárás nagyban növeli.

Az UE_01 eljárás két módon indítható:

- Rendszerindításkor az UTIL E/1 segédprogram meghívásával, vagy
- az eljárás automatikus indításának beállításával a \$alephe_root könyvtár aleph_startup file-jában. Ha az UE_01 folyamat nem fut, nem tud böngészni vagy keresni a frissen hozzáadott rekordok között. Mindennek ellenére az IND indexben automatikusan indexelt rendszerszámuk szerint kereshetők lesznek.

2.2.1 Az indexelési folyamat prioritásainak beállítása

Az UE_01 háttérben futó folyamat által kezelt indexelési folyamat egy prioritási szabálykészletet alkalmaz, amellyel némely rekordtípus-módosítás hamarabb indexelhető, mint a többi. Némely prioritás-szabály a rendszerbe épített (hard-coded), míg mások konfigurálhatóak.

Szabály szerint minden, a katalógusban történő módosítás egy z07 rekord létrejöttét generálja. A z07 rekord az UE_01 háttérben futó eljáráskezelés triggere. Minél alacsonyabb a z07 által kiosztott szekvenciája a rekordnak, annál magasabb a rekord prioritása az UE_01 folyamatban. A következő szabályok vonatkoznak a z07 szekvencia beállításaira:

- Az újonnan katalogizált rekordok az 1990-es szekvenciát kapják, függetlenül attól, hogy online, vagy offline jöttek létre. Minden új rekord egy elsőként létrejött-elsőként indexelt alapon indexelődik. Ez egy beépített (hard-coded) mechanizmus.
- Már létező katalógusrekordok módosításai, függetlenül attól, hogy online, vagy az OCLC szerver által módosultak, az 1998-as szekvenciát kapják. Ezzel az új rekordok a már meglévő rekordok módosításánál magasabb prioritást kapnak (1990 kevesebb, mint 1998). Ez is egy beépített (hard-coded) mechanizmus.
- Az RLIN betöltéssel módosított rekordok az 1999-es szekvenciát kapják. Ezzel az új rekordok és az OCLC által betöltött rekordok magasabb prioritást kapnak az RLIN betöltés módosításainál (mivel 1990 és 1998 kevesebb, mint 1999). Ez is egy beépített (hard-coded) mechanizmus.
- A „Feltöltés – Bibliográfiai rekordok (manage-18)” szerviz futtatásakor a katalogizáló rekordok prioritását a batch feladat 'Indexelési prioritás felülírása' paraméterével lehet beállítani. Például ha a paraméter 1985, akkor a batch feladat által betöltött rekordok magasabb prioritást kapnak, mint az újonnan online katalogizált rekordok (mivel 1985 kevesebb, mint 1990).

- Az összes betöltő batch feladathoz prioritást rendelhet az `alephe_root/aleph_start.private` egy környezeti változójának beállításával (ez a változó felel a batch feladat által létrehozott rekordok indexelési prioritásáért). Például:

```
setenv z07_p_manage_40      2000
setenv z07_p_manage_18     2001

setenv z07_p_manage_180   2001
setenv z07_p_file_90      2002
setenv z07_p_file_93      2002
setenv z07_p_file_95      2003
setenv z07_p_file_96      2004
setenv z07_p_file_97      2005
setenv z07_p_file_98      2005
setenv z07_p_file_99      2005
```

- Az UE_03 RLIN betöltő programon felül az OCLC szerver és az UE_08 démon prioritása is beállítható. Például:

```
setenv z07_ue_03          2006
setenv z07_oclc_server    2007
setenv z07_ue_08          2008
```

Megjegyzések

A manage-18 batch feladat 'Indexelés prioritás felülírása' paraméterében beállított prioritás felülírja a `z07_p_manage_18` változóban meghatározott értéket.

Az ebben a környezeti változó táblákban meghatározott prioritás felülírja az előzőekben felsorolt beépített (hard-coded) definíciókat.

2.3 Indexek építése

Az indexeket három batch eljárás építi fel:

- `p_manage_02` (besorolási index)
- `p_manage_01` (szó index)
- `p_manage_05` (direkt index)

Az indexrekordok módosításához a batch indexépítési folyamatot futtassa az indexelő táblák (pl. `tab11`, `tab00.lng`, `stb.`) konverziója és/vagy módosítása után az indexrekordok módosításához.

3 Besorolási index

Ez a fejezet a következő témákat tárgyalja:

Besorolási elemek egységesítése

Fő besorolási elem szervizek

További besorolási elem szervizek

Lineklési (kapcsolási) folyamat

A besorolási index egy olvasó, vagy könyvtáros által a Web OPAC-on böngészhető listát hoz létre. A besorolási indexet gyakran hívjuk böngészőlistának vagy ACC listának is.

A besorolási indexek rekordmezőből származó kifejezések, mint pl. szerzőségi adatok, címadatok, tárgyszó, kiadó, stb. Egy besorolási index alak állhat egy teljes mezőből, illetve egy, vagy több meghatározott almezőből is.

<i>Leader</i>	<u>LDR</u>	—	—	00936nam^^2200289^a^45e0
<i>Control No.</i>	<u>001</u>	—	—	000003328
<i>Date and Time</i>	<u>005</u>	—	—	20010809135528.0
<i>Fixed Data</i>	<u>008</u>	—	—	990126s1998^^^^nyua^^^c^^^^^^000^1^eng^ ^
<i>LC Control No.</i>	<u>010</u>	—	a	99165483
<i>ISBN</i>	<u>020</u>	—	a	0141301104
<i>System No.</i>	<u>035</u>	—	a	(OCoLC)40384393
<i>Catal. Source</i>	<u>040</u>	—	a	DLC
			c	DLC
			d	CMI
<i>LCC No.</i>	<u>050</u>	<u>00</u>	a	P27.D1515
			b	Wi 1998
<i>Dewey No.</i>	<u>082</u>	<u>00</u>	a	[Fic]
			2	21
<i>Personal Name</i>	<u>100</u>	<u>1</u>	a	Dahl, Roald.
<i>Main Title</i>	<u>245</u>	<u>14</u>	a	The witches /
			c	Roald Dahl ; illustrated by Quentin Blake.
<i>imprint</i>	<u>260</u>	—	a	New York :
			b	Puffin Books,
			c	1998

A könyvtárak dönthetik el, hogy mely bibliográfiai rekordmezők szolgáltatják az alapot a Web OPAC böngészőfunkciójához. Például lehetővé teszi a szerzők, címek, kiadók stb. szerinti böngészést is.

USMARC BIB (USM01) - Browse an Alphabetical Index

Type word or phrase:

Select index to browse:

- Publisher
- Series**
- Location
- ISSN
- ISBN
- Dewey Decimal Class
- LC Index
- LC Classification
- Thesaurus
- Word index
- Words from title

© 2003 Ex Libris

A besorolási vagy böngészőindex a Web OPAC böngészés funkcióján vagy a GUI-k keresés funkcióján keresztül érhető el.

Az egyes mezők, vagy almezők egy konkrét besorolási csoporthoz vannak rendelve. Például minden címtípus a cím-besorolási csoporthoz rendelhető. Tárgyszót egy másik, tárgyszavakra vonatkozó „alindexhez” rendelhet.

3.1 Besorolási elemek egységesítése

A besorolási elemek a böngészőlistában a besorolási elem „egységesítő szövege” szerint vannak elrendezve.

Az egységesítő szöveg három lépésben épül fel, a mező szövege alapján:

1. Szöveg megjelenítése (általában a bibliográfiai rekordból vett adat végső írásjel nélkül).
2. Szöveg megjelenítés normalizálása.
3. Normalizált szöveg átalakítása egységesített szöveggé.

A konverzió a könyvtár tab könyvtárának tab_filing táblájában és az alephe/unicode könyvtárban található tab_character_conversion_line táblában meghatározott eljárás szerint történik.

```
z01_display_text .....$$aDahl, Roald
z01_normalized_text .....$$-dahl, roald
z01 filing text .....DAHL ROALD
```

3.1.1 Normalizálás

A normalizálás az az eljárás, amely az ékezeteket, a legtöbb mondatvégi írásjelet, speciális karaktereket és az egyéb elemeket kiveszi az indexelendő mezőkből (besorolási elemekből) az egyedi besorolási elemek meghatározásához.

A besorolási elem normalizált formája a könyvtár `tab_filing` táblájában meghatározott szabályok szerint épül fel:

```
03 acc_code .....AUT
03 alpha .....I
03 filing_text .....DAHL ROALD
03 filing_sequence .....17734
02 z01_acc_sequence .....000017734
02 z01_hash .....163007586506
02 z01_aut_tag .....
02 z01_rec_key_4 \
    03 aut_library .....-CHK-
    03 aut_doc_number .....000000000
02 z01_acc_sequence_see ....000000000
02 z01_number_of_doc .....00000
02 z01_cataloger \
    03 cataloger_name .....Batch
    03 cataloger_level .....10
02 z01_open_date .....20021119
02 z01_update_date .....20021119
02 z01_cataloger_library ...
02 z01_non_filing_char .....00
02 z01_update_doc .....Y
02 z01_update_z0102 .....N
02 z01_ref_type .....
02 z01_normalized_text .....$$-dahl, roald
02 z01_display_text .....$$aDahl, Roald
```

Az egységesítő rutinok teljes listáját valamint a normalizálási rutinok magyarázatát a Rendezési és Szó-tördelési rész tartalmazza a 35. oldalon.

A normalizálás céljai a következők:

- Az azonos besorolási elemek hasonlóként kezelése.
- Annak biztosítása, hogy minden egyes besorolási elem csak egyszer szerepeljen a besorolási indexben.
- Az egyedi azonosítók alapján elkülönülő besorolási elemek megkülönböztetése.

3.1.2 Alkalmazott adatbázistáblák

Doc

Az indexelt dokumentációs tábla (bibliográfiai könyvtár esetében ez a bibliográfiai rekord.)

Z01

A z01 tábla a bibliográfiai rekordban szerepelő információból nyert besorolási elemek (bejegyzések) azon listája, melyet a felhasználó a WebOPAC-ban és a GUI kereső funkcióján keresztül böngészhet. A besorolási indexet böngésző listának vagy ACC listának is nevezzük. A felhasználó böngészhet a rekordok között Szerző, Cím, Tárty vagy bármely más, a könyvtár által meghatározott kategória szerint.

z01 a besorolási elemeket a könyvtár tab könyvtárának tab00.lng és tab11_acc tábláiban meghatározott indexelés szerint tartalmazza.

Z02

z02 tartalmazza a kapcsolatot a z01 - Besorolási hozzáférés - és a bibliográfiai rekordok között.

Z07

Katalogizálási rekord (bibliográfiai, authority, holding, stb.) létrehozásakor vagy módosításakor, a rendszerszáma megjelenik a z07 táblában. A z07-et az indexfájlok módosításának ellenőrzéséhez használja a rendszer. Az indexeket az UTIL E/1 háttérben futó feladat módosítja (a bibliográfiai indexek módosítása (Update Doc Indexes) - UE_01).

3.2 Fő besorolási elem szervizek

A Besorolási indexet a p-manage-02 hozza létre, az UE_08 bővíti, és az UE_01 módosítja.

3.2.1 Besorolási index (manage-02)

A szerviz az adatbázis besorolási indexét módosítja/frissíti.

3.2.2 Pre-Enrich Bibliographic Headings Based on the Authority Database (manage-102)

A p_manage_102 kimásolja a besorolási elemeket az authority rekordokból a z01 bibliográfiai fájlba. Ez szükségtelessé teszi az UE_08 hosszadalmas teljes futtatását, amely összekapcsolja a besorolási elemeket az authority rekordokkal.

Erről az eljárásról további információkat az ALEPH Felhasználói Kézikönyv, Authority – Batch feladatok bibliográfiai könyvtárak bővítéséhez és javításához authorityból fejezetében talál.

3.2.3 Tárgyszó alosztályok

A tárgyszó alosztályokról az ALEPH Felhasználói Kézikönyv, Authority – Authority kontroll fejezetében talál további információt.

3.2.4 Mikor futtassuk a besorolási indexet

A "Besorolási index módosítása" szervizt a következők után kell futtatni:

- Már létező indexhez új kód lett hozzárendelve

- A Besorolási indexet érintő módosítás történt a `tab00`, vagy `tab11_acc` táblában.
- Módosítás történt a `tab_filing` táblában, mely érinti a besorolási elem megjelenítését vagy normalizált formáját.

A "Besorolási index módosítása" szerviz "Futtatás eljárása" mezőjében az "újraépítés" funkciót kell elindítani, amennyiben már létező index bejegyzéseket érintő módosítások történtek.

Azt javasoljuk, hogy a besorolási indexet bizonyos időszakonként építse újra a "Besorolási index módosítása" szervizzel.

Megjegyzés

A szerviz „Újraépítés” opcióval való futtatása után mindig futtassa a Teljes hosszön rendezés (manage-17) szervizt.

3.3 További besorolási elem szervizek

3.3.1 Besorolási rekordok rendezése (p_manage_16)

Ez a szolgáltatás a `tab00.lng` táblában és a `tab_filing` táblában található szabályok szerint abc sorrendbe rendezi a besorolási elemeket.

Ezek a szabályok egy olyan "egységesítő szöveget" hoznak létre, amely alapján a sorba rendezés történik. Ezt követően történik az abc sorbarendezés az egyes elemek egységesítő szövegének első 69 karaktere szerint.

Mikor futtassuk ezt a szolgáltatást

Minden alkalommal, amikor az abc sorbarendezés szabályai megváltoztak a `tab00.lng` táblában vagy a `tab_filing` táblában (beleértve a karakter konverziós táblákat is). Például elhatározza, hogy az „u” helyett „ue” alatt szeretné az abc sorrendbe rendezést. Amennyiben a szabályok változása érinti a megjelenített, vagy normalizált szövegeket, akkor inkább a `p-manage-02` szervizt futtassa le.

Megjegyzés

A szerviz alkalmazása után mindig futtassa le a Teljes hosszön rendezés (manage-17) szervizt.

3.3.2 Teljes hosszön rendezés (p_manage_17)

Ez a szolgáltatás abc sorrendbe rendezi azokat a besorolási elemeket, melyek egységesítő szövege 69 karakternél hosszabb.

Mikor futtassuk ezt a szolgáltatást

Konverziók után, illetve akkor, amikor a hosszú besorolási elemek besorolásánál problémát gyanít. Ezen felül bármikor, ha a `tab00.lng` táblában, vagy a `tab_filing` táblában az abc sorba rendezés szabályai módosultak (beleértve a karakterkonverziós definíciókat is) – ebben

az esetben az új szabályoknak való megfelelésnek az összes besorolási elemet újra kell rendezni. Először a Besorolási rekordok rendezése (p_manage_16) szervizt futtassa le, majd a Teljes hosszon rendezés (p_manage_17) szervizt.

3.3.3 Számlálóok építése logikai bázisokhoz (manage-32)

Ez a szolgáltatás számlálókat épít logikai bázisokon.

A logikai bázisok (z0102) számlálóival a WebOPAC-ban történő böngészést teheti hétékonyabbá akkor, amikor olyan logikai bázisokban keresünk (böngészünk), melyek kisebbek, mint a teljes adatbázis 10 %-a. Az `./alephe/tab_base.lng` tábla 8-as oszlopának „Y” értéke határozza meg azt, hogy a logikai bázishoz z0102 rekordok épüljenek. Amennyiben a könyvtár nem használ logikai bázisokat, vagy a `tab_base.lng` nem tartalmaz "Y" értéket a 8-as oszlopban, akkor ez a szakasz irreleváns.

Logikai bázis böngészésekor a rendszer ezen számláló segítségével határozza meg, hogy megjelenítsen-e egy besorolási elemet vagy sem a kapcsolódó dokumentumok megkeresése, beolvasása és az eljárás mikéntjének meghatározása nélkül.

Mikor futtassuk ezt a szolgáltatást

Ezt a szolgáltatást mindig futtatni kell a besorolási index (manage-02) létrehozása után. Az érintett adatbázis táblák:

z01

z02

z0102

Z0102

A z0102 táblában valamennyi besorolási elemhez és logikai bázishoz bejegyzés jön létre. A tábla tartalmazza a normalizált szöveget, a besorolási adat `acc-sequence` azonosítóját, és a besorolási adathoz csatolt releváns dokumentumok számlálóját. A dokumentumra mutató kapcsolóelemet továbbra is a z02 (ACCD0C) tábla tartalmazza.

3.3.4 Rövid (Brief) rekordok módosítása (manage-35)

Ez a szolgáltatás rövid rekordokat hoz létre és módosít. A rövid rekord megjelenítési formátumot egy meghatározott besorolási elemhez kapcsolt rekordok al-rendszerezéséhez használhatja. Ezek a rekordok a rekordok formátuma, a besorolási index és a származási mező szerint épülnek fel.

A rövid rekordok szerkezetét a könyvtár `tab` könyvtárának `tab_z0101` és `tab_z0101_text` táblái határozzák meg.

Field to search	Actual Title
Words adjacent?	All Fields
<input type="button" value="Go"/> <input type="button" value="Clear"/>	Title Words
	Actual Title
	Author
	Subject
Limit search to:	ISSN
	ISBN
Language: <input type="text" value="all"/>	System number
	Barcode

A vonatkozó adatbázistáblák a következők:

z0101

Megjegyzés: a z0101 formátummal dolgozó könyvtáraknak hozzá kell adni a last-z0101-sequence számlálót az UTIL G/2 segédprogramhoz.

3.3.5 Rövid bibliográfiai rekord (manage-07)

Ez a szerviz az adatbázis rövid bibliográfiai rekordjait módosítja. Ezen rekordok hatékonyan és megfelelő időben biztosítanak bibliográfiai információt, jellemzően akkor, amikor a bibliográfiai adatok az adminisztratív adatok kiegészítői.

A Rövid bibliográfiai rekordot a rendszer a tab22 tábla (a könyvtár tab könyvtárban) beállításai alapján hozza létre, a rekordoknak az adatbázisba történő betöltések (az indexelési paraméter 'Full' értéke esetén), vagy a Katalogizáló modulon keresztül történő rekordhozzáadáskor, vagy módosításkor.

A Rövid bibliográfiai rekord a bibliográfiai rekord standard Oracle tábla formátumú rövidített változata. Legfeljebb hét, fix (rendszer által meghatározott) mezőt tartalmazhat (év, raktári jelzet, raktári jelzet azonosító, szerző, cím, kiadás és ISBN/ISSN), és az egyes mezők legfeljebb 100 karakterből állhatnak (kivéve a raktári jelzet azonosító mezőt, ez legfeljebb 80 karakter lehet). Továbbá, legfeljebb tizenöt, a felhasználó által meghatározott mezőt tartalmazhat, mezőnként legfeljebb 500 karakterrel.

Mikor futtassuk ezt a szolgáltatást?

A tab22 tábla olyan módosításai után, mely éri a Rövid bibliográfiai rekordokat.

A kapcsolódó adatbázis táblák az alábbiak:

z13

3.3.6 Rendezési index (manage-27)

A rendezési kulcsokat a WebOPAC felületén használják egy rekordhalmaz rendezésére. A kulcsok a bibliográfiai rekord egy mezőjéből (például cím, év) kivont adatok. A kulcsok a tab_filing és tab_character_conversion_line táblákban meghatározott szabályok szerint jönnek létre.

3.3.7 Kapcsolat nélküli besorolási rekordok törlése (manage-15)

Ez a szolgáltatás törli a rekordhoz nem kapcsolódó besorolási elemeket. A katalogizálót rekordok minden egyes módosítása új besorolási elemet hoz létre a rendszerben. A régi

besorolási rekordok megmaradnak, de a rekordra mutató link törtődik. Ez a szerviz törli az összes ilyen típusú, kapcsolat nélküli besorolási elemet.

Megjegyzés: ez az eljárás nem törli az authority kapcsolattal rendelkező besorolási rekordokat azért, hogy megmaradjanak a rekordhoz közvetlenül nem kapcsolódó keresztutalások.

3.4 Linkelési (kapcsolási) folyamat

A bibliográfiai besorolási adat és az authority rekordok kapcsolatának építését generáló folyamat az UE_08 démon. Ezt a folyamatot az UTIL E/8 segédprogram indítja.

Az UE_08 eljárás ellenőrzi a bibliográfiai könyvtár új besorolási elemeit az authority könyvtárban, valamint keresztutalásokat és/vagy többnyelvű megfelelőket ad a bibliográfiai besorolási táblákhoz.

A besorolási elem akkor válik „engedélyezetté”, amikor egy bibliográfiai adatbázisból származó besorolási elem (z01) és egy authority adatbázisból származó rekord között pontos egyezés van. Az authority rekordot az authority könyvtár besorolási indexén keresztül lehet megtalálni.

Erről az eljárásról további információ az ALEPH Felhasználói Kézikönyv Authority fejezetének az Authority adatbázis mint keresési segédlet részben található.

4 Szóindex

Ez a fejezet a következő témaköröket tárgyalja:

Szavak definiálása

Alkalmazott adatbázistáblák

Mikor építse újra a szóindexet

A szóindex az adatbázis bibliográfiai rekordjainak speciális mezőiben előforduló szavak listáját tartalmazza.

Ha egy felhasználó vagy könyvtáros a WebOPAC-ban a Keresés funkciót alkalmazza egy szóindexen, a rendszer megkeresi az összes olyan dokumentumot, mely tartalmazza a felhasználó által begépett kulcsszavakat.

A szavak speciális szócsoporthoz vannak hozzárendelve. Így a cím különböző típusaiba tartozó szavak a "szavak a címekből" csoporthoz lesznek hozzárendelve. A tárgyszóban lévő szavakat egy másik szócsoporthoz lehet kiosztani.

Web OPAC keresés alkalmazható az általános szó indexhez vagy más egyéb speciális alindexhez:

Field to search	Actual Title
Words adjacent?	All Fields Title Words Actual Title Author Subject ISSN ISBN System number Barcode
Go	Clear
Limit search to:	
Language:	all

A rendszer a rekord konkrét mezőjéből kivesszi az egyes egyedi szavakat, azokat egy szófájlban tárolja, és fenntartja a dokumentumra mutató linket.

Full View of Record	
Choose format: Standard format Catalog card Citation Name tags MARC tags	
Record 000001 out of 1	
Book Number	000010789
LC no.	P27.D1515 Wi 1998
ISBN	0141301104
Main Entry	Dahl, Roald.
Title	The witches / Roald Dahl ; illustrated by Quentin Blake.
Imprint	New York : Puffin Books, 1998.
Descr.	206 p. : ill. ; 24 cm.
Abstract	A young boy and his Norwegian grandmother, who is an expert on witches together foil a witches' plot them into mice.
Subject - Lib. Cong.	Witches - Juvenile fiction. Grandmothers - Juvenile fiction.
Subject - A. C.	Witches - Fiction. Grandmothers - Fiction.

A kivonatolt szavakat a z97 tábla tárolja, ez tartalmazza a word dictionary-t, azaz a szótárt. A szótár az összes dokumentumrekord adataiból származtatott kereshető szavakat tartalmazza.

Amikor a felhasználó végrehajt egy Keresés/Böngészés kérést a WebOPAC felületről, akkor a rendszer a – felhasználó által beírt - kulcsszavakat tartalmazó dokumentumok visszakereséséhez a szóindexet ellenőrzi.

A Szóindexet általában a WebOPAC és a GUI Keresés funkciója alkalmazza:

Basic Search	
Type word or phrase	<input type="text"/>
Field to search	All Fields
Words adjacent?	<input checked="" type="radio"/> No <input type="radio"/> Yes
Go	Clear

4.1 Szavak definiálása

Egy szó alapértelmezett definíciói az alábbiak:

- Szóköztől szóközig tartó, vagy
- Egy sor elejétől az első szóközig tartó, vagy
- Az utolsó szóköztől a sor végéig tartó karakterlánc.

Szótördelés

A szótördelő eljárások azt definiálják, hogy a rendszer speciális esetekben (pl. I.B.M.) mit tekintsen "szónak". A szótördelő rutinok a Rendezés és szótördelés fejezetben található a 35. oldalon.

A szóindex csoportok szótördelő rutinjait a könyvtár `tab` könyvtárában található `tab11_word` táblájának a 6. oszlopa definiálja.

Karakterkonverzió

A szó-építési eljárásokon felül, a szöveg szavakra tördeltése után, a karakter konverziós tábla segítségével lehet meghatározni karakterekre vonatkozó ekvivalenciákat. A rendszer a `$alephe_unicode` könyvtár `tab_character_conversion_line` táblájának 5-ös sorában a `WORD-FIX` bejegyzés alatt található karakterkonverziós táblát alkalmazza:

<code>tab_character_conversion_line</code>	
<code>WORD-FIX ##### # line_utf2line_utf</code>	<code>unicode_to_word_gen</code>

4.2 Alkalmazott adatbázistáblák

Z95

A tábla rekordjai egy dokumentum szavainak listáját tartalmazza. Az UE-01 online indexelő eljárás írja a Z95 rekordokat.

Z97

A Z97 tábla tartalmazza a szótárat. Ez tartalmazza az összes dokumentumrekordból származtatott szó listáját.

Az egyes könyvtárak meghatározhatják azt, hogy a dokumentumrekord mely mezői alkossák a WebOPAC felület Keresés/Böngészés funkciójának alapját. Például eldöntheti, hogy a rendszerben legyen-e opció címadatokból vett szavak szerinti keresésre, vagy sem. A rendszer kiemeli az egyes szavakat a rekordok címadat mezőiből és egyedi bejegyzést hoz létre az adott szónak a Z97 táblában. A Z97 tábla tartalmazza a könyvtár `tab/tab00.lng` és `tab11_word` tábláiban definiált indexelés szerinti összes kereshető szót.

Egymás melletti szavak keresése

Amennyiben a "setenv ADJACENCY 2" környezeti változó az ./alephe/aleph_start táblában definiált, akkor a szó indexelés automatikusan szópárokat alkot a z97-Szótárban az egymás melletti szavak keresése céljából.

Megjegyzés: az egymás melletti szavak keresése céljából létrehozott szópárok sok helyet foglalnak a lemezen (rekordonként négyszer annyit). Másrésztől, gyorsítja az egymás melletti szavak keresésének teljesítését.

Z98

Ez a tábla tartalmazza a szó - dokumentum kapcsolatokat. A z98 tábla tartalmaz egy bitmap-tömörítésű térképet a dokumentumokban előforduló szavakról. A bitmap karbantartja a z97 táblában a dokumentumok számára regisztrált szavak mutatóit. A WORD3 segédprogram (UTIL/F/4/word3) segítségével lehet a bitmap-et olvasni. Ez a segédprogram olvassa a bitmap-et az Y indexben tárolt X szót tartalmazó dokumentumok kereséséhez.

4.3 Mikor építse újra a szóindexet

A Word index (manage-01) célszervizt a könyvtár tab mappájában található tab00.lng vagy a tab11_word tábla olyan módosításaikor futtassa, ha az a szóindexelést is befolyásolja.

5 Direkt index

Ez a fejezet a következő témaköröket tárgyalja:

Direkt indexek egységesítése

Mikor módosítsa a direkt indexet

Alkalmazott adatbázistáblák

A direkt indexek segítségével egy konkrét rekordot kereshet vissza. A direkt index a rekord egyedi, vagy majdnem egyedi azonosítóra épül (mint pl. ISBN, ISSN, illetve polc/helyrajzi szám), és a rekord gyors elérésére alkalmazható:

<i>Leader</i>	<u>LDR</u>	—	—	00936nam^^2200289^a^45e0
<i>Control No.</i>	<u>001</u>	—	—	000003328
<i>Date and Time</i>	<u>005</u>	—	—	20010809135528.0
<i>Fixed Data</i>	<u>008</u>	—	—	990126s1998^^^^nyua^^^c^^^^^^000^1^eng^ ^
<i>LC Control No.</i>	<u>010</u>		<u>a</u>	99165483
<i>ISBN</i>	<u>020</u>		<u>a</u>	0141301104
<i>System No.</i>	<u>035</u>		<u>a</u>	(OCoLC)40384393
<i>Catal. Source</i>	<u>040</u>		<u>a</u>	DLC
			<u>c</u>	DLC
			<u>d</u>	CMI
<i>LCC No.</i>	<u>050</u>	<u>00</u>	<u>a</u>	P27.D1515
			<u>b</u>	Wi 1998
<i>Dewey No.</i>	<u>082</u>	<u>00</u>	<u>a</u>	[Fic]
			<u>2</u>	21
<i>Personal Name</i>	<u>100</u>	<u>1</u>	<u>a</u>	Dahl, Roald.
<i>Main title</i>	<u>245</u>	<u>14</u>	<u>a</u>	The witches /
			<u>c</u>	Roald Dahl ; illustrated by Quentin Blake.
<i>Imprint</i>	<u>260</u>		<u>a</u>	New York :
			<u>b</u>	Puffin Books,
			<u>1</u>	1998

Az egyes könyvtárak maguk döntenek el, hogy a rekord mely mezője legyen kereshető ezen szempont szerint.

A közvetlen hozzáférés funkció a WebOPAC böngészés és keresés menüben érhető el. A direkt indexben való böngészés megmutatja az indexet magát és a rekord a címét. Például, amikor egy olvasó vagy könyvtáros használja a böngésző funkciót a WebOPAC-ban a bibliográfiai rekord ISBN mezőjének keresésére, akkor a rendszer a keresett karakterláncot tartalmazó rekord, vagy legközelebbi rekord kereséséhez ellenőrzi a Direkt Indexet. A kereső funkció használata során a rendszer az egyező rekord(ok)at egy halmazban jeleníti meg.

USMARC BIB (USM01) - Browse an Alphabetical Index

Type word or phrase:

Select index to browse:

- Place of Publication
- Publisher
- Series
- Location
- ISSN
- ISBN
- Dewey Decimal Class
- LC Index
- LC Classification
- Thesaurus
- Word index

Basic Search

Type word or phrase

Field to search:

Words adjacent?

Limit search to:

- All Fields
- Title Words
- Actual Title
- Author
- Subject
- ISSN
- ISBN
- System number
- Barcode

Language:

5.1 Direkt indexek egységesítése

A direkt indexek az indexhez rendelt, a tab könyvtárban található tab_filing táblájában meghatározott szabályok szerint felépülő egységesítő szöveg szerint épülnek.

Az egyes direkt indexek egységesítő eljárása a könyvtár `tab` könyvtárában található `tab00.lng` tábla 5-ös oszlopán keresztül kerül meghatározásra.

5.2 Mikor módosítsa a direkt indexet

A Direkt Index Módosítása (`manage-05`) szerviz futtatása a `tab00.lng` tábla vagy `tab11_ind` tábla minden olyan módosítása után szükséges, mely érinti a direkt indexet.

Újraépítés

Egy már létező index bejegyzést érintő módosításkor az újraépítés opciót válassza a legördülő menüből.

Módosítás

Amennyiben egy már létező indexhez ad új kódot, vagy egy új indexet ad az adatbázishoz, akkor az eljárás Frissítés opcióját válassza a legördülő listából a futtatáshoz.

Megjegyzés

A direkt indexet frissíteni kell, ha az adatbázisba „partial” (részleges) módban nagy mennyiségű rekord lett feltöltve és a feltöltött rekordokon az indexelés nem futott le automatikusan.

5.3 Alkalmazott adatbázistáblák

Z11

A `z11` tábla a könyvtár `tab` könyvtárában található `tab11` és `tab00.lng` táblában definiált direkt indexeket tartalmazza.

Megjegyzés

A `z11` indexelés független az `ue_01` eljárástól; a bibliográfiai rekord hozzáadásakor vagy módosításakor automatikusan történik.

6 APAC Indexelés

6.1 Besorolási indexelés

A kínai, japán és koreai nyelvű (CJK) besorolási elemekhez speciális kiegészítők tartoznak.

Négy egységesítő rutin érhető el annak definiálására, hogy a CJK besorolási elemek a pinjin átírás, vagy karakterek szerint legyenek rendezve. Amennyiben nem alkalmazza ezen

rutinokat, akkor a besorolási elemek az Unicode érték szerint lesznek rendezve, melyhez nem szükséges további speciális indexelési beállítás. A következő CJK egységesítő rutinokat lehet alkalmazni:

- **ckj_pinyin** – a rutin egy felkiáltójelet (!) ad a CJK karakterek elé, lefordítja a karaktereket pinjin –re a z114 táblával, és az Unicode értéket tizedesjegy megjelöléssel egészíti ki. A felkiáltó jel (!) segítségével a rutin a pinjin iktató szöveghez a latin karakterektől elkülönülő szekvenciát oszt ki.
- **ckj_stroke** – A rutin a ckj_pinyin rutinhoz hasonlít annyi különbséggel, hogy az egyes karaktereket a leütési értékre fordítja le a z114 tábla alkalmazásával.
- **chi_pinyin** – a karaktereket fordítja le pinjin-re a z114 tábla alkalmazásával, illetve hozzáadja a karakterekhez az Unicode értéket (a tizedesérték jelölésével). Az Unicode karakterek az ugyanolyan pinjin értéket viselő karakterek megkülönböztetése miatt adódnak a karakterekhez. Mivel a pinjin iktató szöveg a standard latin karakterekkel együtt kapja a szekvenciát, ezt a rutint alkalmazza a 008-as nyelvi kód alapján elkülönülő böngészőlista létrehozására (pl. AUTC)
- **chi_stroke** – ez a rutin a chi_pinyin rutinnal egyezik azzal a különbséggel, hogy a karakterek a leütési értékre lesznek lefordítva a z114 tábla alkalmazásával.

Az alapértelmezett ALEPH beállítás a pinjin érték szerinti rendezést alkalmazza. Emiatt az összetett CJK nyelv használatokor CJK nyelvenként - mely a katalógusban indexelt, és felhasználó által böngészett - elkülönülő indexet kell létrehozni. A következőkben a beállítás konfigurációjáról olvashat.

- **tab_expand**

```
! 1 2 3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
```

```
ACC expand_doc_bib_lng_cjk
```

Ezek a sorok állítják be a besorolási és szóindex eljárásokat az expand_doc_bib_lng_cjk rutin futtatására. A rutin egy \$\$9 almezőt ad hozzá, a kínai, japán vagy koreai karaktereket tartalmazó mezőkhöz, mely a 008/34-36-os pozíciójának nyelvi kódját tartalmazza. A \$\$9 almezőt ezután a tab11_acc és tab11_word táblákban a CJK bejegyzések különböző indexbe történő beillesztésére használja a rendszer.

- **tab11_acc**

```
! 1 2 3 4 5 6 7 8 9 10 11
!!!!!!!!-!!!!-!!-!!!!!!!!!!!!-!!!!-!!!!-!!!!-!!!!-!!!!-!!!!-!!-!
245## 9 chi TITC -e468
245## 9 jpn TITJ -e468
245## 9 kor TITK -e468
245## 9 - TIT -e468
```

A fenti példában az egyes CJK nyelvek elkülönülő besorolási listákban lettek meghatározva.

- **tab00.lng**

```
! 2 3 4 5 6 7 8 9 10 11
!-!!!!-!!!!-!!-!!!!-!!!!-!!!!-!!!!-!!!!-!!!!-!!!!-!!!!-!!!!-!!!!
H TITC ACC 51 00 00 Chinese Titles
H TITJ ACC 52 00 00 Japanese Titles
H TITK ACC 53 00 00 Korean Titles
```


H TIT ACC 11 00 00 Titles

A fenti példában az egyes CJK besorolási elemek külön egységesítő rutinhoz lettek hozzárendelve.

- **tab_filing**

```
!1 2 3 4
!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!
51 F chi_pinyin
52 F jpn
53 F kor
```

A fenti példát az a tab00.lng tábla koordinálja, amelyben az egységesítő eljárás értéke „51” a TITC (kínai cím adatok) esetében és így tovább.

Az egységesítő kulcs létrehozásakor a „chi_pinyin” a z114 rutinnal fordítja le a karaktereket pinjin-re, majd a karakter Unicode értékét adja az egyes pinjin karakterek után azért, hogy az ugyanazon pinjin értéket viselő összetett kínai karaktereket a rendszer megkülönböztesse.

A japán és koreai nyelvekhez nincs párhuzamos szótár. Ezen nyelvek egységesítő kulcsának létrehozásakor „jpn” és „kor” használatával a rendszer a karakterek Unicode értékét veszi.

Megjegyzés

Amennyiben a tab100 pinjin szerint rendezzi a besorolási elemeket, de a tab_expand beállításai nem adnak nyelvi kóddal rendelkező \$\$9 almezőt az elemekhez, akkor a kínai karakteres besorolási elemek latin karakteres elemekkel lesznek összesorolva, mivel a besorolási elemek az általános AUT/TIT listák szerint lesznek rendezve. Mindazonáltal ilyen esetekben a pinjin szavakat a „cjk_pinyin” egységesítő eljárással különítheti el az általános szekvenciától. Ez a rutin egy felkiáltójel (!) illeszt be az egyes CJK karakterek elé, lefordítja őket pinjinre a z114 táblával, majd hozzáadja az Unicode értéket tizedesérték jelölésben. Ezzel a pinjin egységesítő szöveg a latin karakterektől elkülönülő szekvenciát kap az általános böngészőlistában.

6.2 Szóindexelés

Az APAC szavak keresése a következőkön alapul:

- Az indexelt szöveg normalizálásán
- Indexekre vonatkozó szegmentáló rutinok definiálásán
- Keresésre vonatkozó szegmentáló rutinok definiálásán

6.2.1 Az indexelt szöveg normalizálása

Az APAC szöveg indexelését megelőzően számos normalizáló rutin futtatása szükséges. Például:

Központosítás törlése

Az összes CJK és latin központosítást szóközre cseréli. Ezt a „to_blank” rutin futtatásával teheti meg.

Karakterek transliterációja

A transliterációs tábla szerint karaktereket cserél le egyenértékű karakterekre. Ezt a „char_conv” rutin futtatásával teheti meg.

Az indexelt szöveget a tab11_word és tab_word_breaking beállításával normalizálhatja. Például:

- a BIB könyvtár tab11_word táblája:

```
!1 2 3 4 5 6 7 8 9 10
!!!!-!!!!-!-!!!!!!!!!!!!-!!!!!!!!!!!!-!!-!!-!!-!!!!-!!!!!!!!!!!!
245## w 01 WTI
```

A WTI index a 245 \$\$w almezőből származó szavakból fog állni, a „01” word-breaking rutin alkalmazásával.

- a BIB könyvtár tab_word_breaking táblája:

```
!1 2 3 4
!!-!-!!!!!!!!!!!!-!!!!!!!!!!!!-!!!!!!!!!!!!-!!!!!!!!!!!!
01 # to_blank !@#$$%^()_={ } [ ] : " ; < > , . ? | \
01 # char_conv CJK_TO_NORMAL
```

A „01” szótördelő rutin a to_blank rutinnal törli a központosítási jeleket, majd a char_conv rutin alapján transliterálja a karaktereket.

A \$alephe_root/unicode-ban található tab_character_conversion_line tábla használathoz definiálni kell a char_conv táblát is. Például:

```
!!!!!!!!!!!!-!!!!-!-!!!!!!!!!!!!-!!!!!!!!!!!!-!!!!!!!!!!!!-!!!!!!!!!!!!
CJK_TO_NORMAL ##### # line_utf2line_utf cjk_to_normal
```

A cjk_to_normal táblát definiálni kell a \$alephe_root/unicode könyvtárban.

6.2.2 Szegmentáló rutinok definiálása indexeléshez

A használandó tab_word_breaking rutin a releváns tab11_word sor 6-os oszlopában definiált indexelő eljárás során használt rutin. Például:

```
1 2 3 4 5 6 7 8 9 10
!!!!-!!!!-!-!!!!!!!!!!!!-!!!!!!!!!!!!-!!-!!-!!-!!!!-
!!!!!!!!!!!!
245## w 01 WTI
```

A tab_word_breaking rutin a fent leírt normalizáló rutinnal felül a kívánt szegmentáló rutint is definiálja. Például:

```
!1 2 3 4
!!-!-!!!!!!!!!!!!-!!!!!!!!!!!!-!!!!!!!!!!!!-!!!!!!!!!!!!
01 # to_blank !@#$$%^()_={ } [ ] : " ; < > , . ? | \
01 # char_conv JAPANESE_TO_NORMALIZED
01 # morpheme_index
```

További szegmentáló rutinokért lapozzon a Javasolt APAC indexelés és keresés szegmentálási rutin beállítások részhez.

6.2.3 Szegmentáló rutinok definiálása kereséshez

A `tab_word_breaking` tábla az adott karakterlánc szegmentációjának elvét is definiálja, a keresési folyamatra.

A `tab_word_breaking` tábla alábbi szekcióit foglalja le a rendszer a FIND (keresés) művelet lekérésekor a karakterlánc keresési folyamatához.

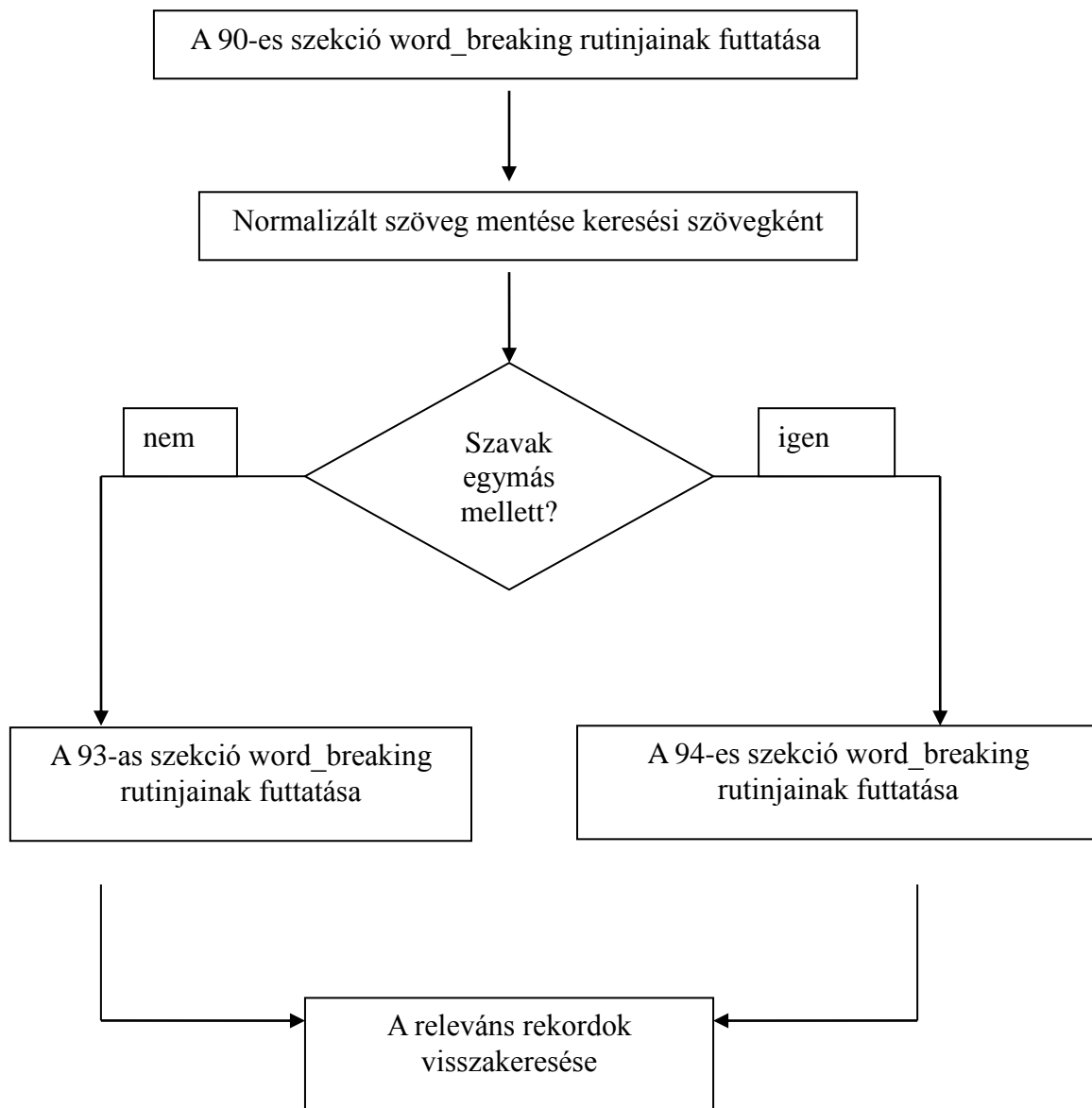
- 90-es szekció – az ebben a szekcióban beállított `word_breaking` rutin fut le a kereső kifejezés szóelemzésekor.

A lementett keresési szöveg lesz a keresési lekérés szövege a 90-es szekcióban beállított rutinok futtatása UTÁN.

- 93-as szekció - az ebben a szekcióban beállított `word_breaking` rutin fut le a kereső kérdésem, miután a 90-es szekció `word_breaking` rutinjai végrehajtottak. Az ebben a szekcióban beállított rutinok kizárólag a szavak egymás mellett keresési opció kihagyása esetén futnak le.
- 94-es szekció - az ebben a szekcióban beállított `word_breaking` rutin fut le a kereső kérdésem, miután a 90-es szekció `word_breaking` rutinjai végrehajtottak. Az ebben a szekcióban beállított rutinok kizárólag a *szavak egymás mellett* keresési opció használata esetén futnak le.

Megjegyzés: a 90-es szekció rutinjai mindig a keresési karakterláncon futnak le, míg a 93/94-es szekciók kizárólag akkor hajtódnak végre, ha a kereső karakterlánc CJK szöveget tartalmaz.

A következő ábra a keresési lekérés rendszerszegmentációjakor lezajló folyamatokat összegzi.



A következő egy példa a CJK keresésre beállított `tab_word_breaking` táblára:

```

!1 2                3                4
!!-!-!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
90 # to_blank          !@#$$%^()_={ } [ ] : " ; < > , . ? | \

93 # char_conv         CJK_TO_NORMALIZED
93 # cjk_2gram_all

94 # char_conv         CJK_TO_NORMALIZED
94 # morpheme_search
  
```

6.2.4 Javasolt APAC indexelés és keresés szegmentálási rutin beállítások

A következőkben összefoglaljuk az APAC szöveget kezelő – már létező - szegmentáló rutinokat.

- **split_cjk** – balról jobbra haladva futtatja a szegmentálást a z113 tábla (kínai szótár) szerint. Ezen felül karakterenként felosztja a szöveget.
- **cjk_to_word** – a szöveget szavakra darabolja, jobbról balra haladva a leghosszabb szó elve alapján, a z113 tábla (kínai szótár) szerint.
- **cjk_split_1** – a keresési sztringen futtatott szegmentáló rutinok definiálásakor az egyes CJK karaktereket egy-egy szóznak veszi.
- **cjk_split_3** – a szavak egy előre meghatározott szótár (z113) szerint lesznek meghatározva.
- **cjk_simplified** – a karakterek egy egyszerűsített formába lesznek lefordítva.
- **cjk_input_adj** – az egyes CJK karakterek szóznak minősülnek a „szavak egymás mellett” opciójú kereséskor
- **cjk_input** – a szöveget szavakra bontja –jobbról balra haladva – a leghosszabb szó elve szerint, a z113-as tábla (kínai szótár) alkalmazásával. A legközelebbi szó az előző szó utolsó karaktere után kezdődik.
- **cjk_2gram_lng** – ezt a rutint egy paraméterrel lehet alkalmazni. A CONC és NO-CONC értékeket veheti fel.

- cjk_2gram_lng CONC paraméterrel

A szavak a következők:

- Ugyanazon írásrendszer (kínai, hangul vagy kana) egymást követő CJK alkarakterláncok összefűzésének összes Bi-Gram szegmense
- Az összes szóközzel elválasztott nem-CJK alkarakterlánc („latin szavak”)
- Az összes szóközzel elválasztott önálló CJK karakterek
- Az összes kínai karakter
- Az összes olyan hangul karakter, amelyek egy karakternyi koreai szavak (az /alephe/unicode/tab_cjk_single_char_word táblában vannak definiálva)
- Az összes olyan normalizált katakana karakter, amely egy japán szó kana karakterének normalizált alakja (az /alephe/unicode/tab_cjk_single_char_word táblában vannak definiálva).

- cjk_2gram_lng NO-CONC paraméterrel:

A szavak a CJK szóközzel elválasztott alkarakterláncainak összes Bi-Gram szegmentálása. Más szóval a Bi-Gram szegmentálást nem előzi meg a CJK alkarakterláncok összefűzése. Ezen felül az összes „latin szó” szóznak minősül.

- **cjk_2gram_all** – a szavak a teljes szövegláncolatának összes Bi-Gram szegmensei (beleértve a nem CJK szöveget is).
- **cjk_add_single** – minden CJK karakter szóként adódik az indexhez.
- **cjk_add_space** – a különböző írásrendszerek karakterei közé egy szóközt szúr be.

- **morpheme index** – kizárólag az indexelési eljárás alatti szegmentálásra alkalmazható. Az adott mező egymást követő szavainak láncolatának összes lehetséges alkarakterláncát hozza létre.

Az összefűzött morfémák száma legfeljebb 30 lehet. Ennek eredményeként, ha a felhasználó egy 30 morfémánál hosszabb keresési kifejezést ír be, szóközök nélkül, akkor a keresés sikertelen lesz (nem lesz egyetlen egyező rekord sem.

Ha az indexelő eljárás során ezt a rutint használja a szegmentáláshoz, akkor a `ckj_morpheme_search` rutint kell alkalmaznia a keresési karakterlánc szegmentálása során.

Megjegyzés: a Bi-Gram algoritmusnak megfelelő indexeléskor a japán karakterek és a kínai ideogrammak ugyanolyan formában lesznek kezelve.

- **morpheme search** – akkor alkalmazzuk, ha a `morpheme_index` rutint használjuk az indexelési eljárás során.

Megjegyzés: a Bi-Gram algoritmusnak megfelelő indexeléskor a japán karakterek és a kínai ideogrammak ugyanolyan formában lesznek kezelve.

Megjegyzés: a morféma index nem hoz létre szópárokat; emiatt a morfémára történő keresést kizárólag a „szavak egymás mellett” opció alkalmazása nélkül használhatja (a `tab_word_breaking` tábla 93-as szekciója).

- **thai index** – kizárólag az indexelési eljárás alatti szegmentálásra alkalmazható.

A szöveget szavakra bontja – jobból balra haladva – a leghosszabb szó elve szerint, a `z117` tábla (thai szótár) alkalmazásával.

Amikor ezt a rutint alkalmazza az indexelő eljárás alatti szegmentálásra, akkor a kereső karakterlánc szegmentálásához a `thai_search` rutint kell alkalmaznia.

- **thai search** – akkor alkalmazza, ha a `thai_index` rutint használta az indexelési eljárásban.

A szöveget szavakra bontja – jobbról balra haladva – a leghosszabb szó elve szerint, a `z117` tábla (thai szótár) alkalmazásával. A legközelebbi szó az előző szó utolsó karaktere után kezdődik.

Az indexelés (a `tab11_word` táblában) és keresés („93” és „94”) szegmentáló rutinokra a következő opciókat alkalmazhatja:

- A `split_ckj` rutint alkalmazó indexeléshez

Kereséshez:

- `ckj_split_1`

vagy

- `ckj_split_3`

vagy

- `ckj_char_to_simplified`

- `ckj_to_word` rutint alkalmazó indexeléshez

Kereséshez:


```
HANJA_TO_HANGUL          ##### # line_utf2line_utf hanja_to_hangul
KANA_TO_NORMALIZED      #####   # line_utf2line_utf
kana_to_normalized
```

A `hanja_to_hangul` és `kana_to_normalized` táblákat az `$alephe_root/unicode` könyvtárban kell definiálni.

Megjegyzés

Amennyiben az egyes CJK karaktereket szeretné szóként adni a standard szavakhoz, akkor a `tab_word_breaking`-ben a `ckj_add_single`-t állítsa be. Megjegyzés: a rutinnak, használat esetén, a `ckj_to_word` és a `split_ckj` rutinok UTÁN kell következnie.

7 Indexelést támogató főtáblák

Az alábbi indexre vonatkozó táblákat tárgyalja ez a fejezet:

`tab00.lng`

`tab11_acc`

`tab11_word`

`tab11_ind`

`tab_word_breaking`

`tab_filing`

`tab_expand`

`tab_character_conversion_line`

`tab20`

`tab22`

`tab_sort`

`tab_aut`

`tab_base`

`tab00.lng`

Ez a tábla a rendszerhozzáférési kódokat és neveket definiálja. Indextípus szerint három szekcióra osztható:

ACC fájl (besorolási elemek)

WRD fájl (szavak)

IND fájl (direkt)

tab00.eng

1	2	3	4	5	6	7	8	9	10	11
H	WRD	W-001			00			01		Words
H	WTI	W-002			00			02		W-titles
H	WAU	W-003			00			02		W-authors
H	WSU	W-004			00			03		W-subjects
H	WUT	W-005			00			02		W-Unif .Titles

1	2	3	4	5	6	7	8	9	10	11
H	TIT	ACC			11	00		00		Titles
H	AUT	ACC			01	00		00		Authors
H	SUB	ACC			01	00		00		General Subjects
H	PLA	ACC			11	00		00		Place
H	PUB	ACC			01	00		00		Publisher

1	2	3	4	5	6	7	8	9	10	11
H	SYS	IND			21	00		00		System Number
H	BAR	IND			21	00		00		Barcode
H	010	IND			22	00		00		LCCN
H	ISBN	IND			21	00		00		ISBN
H	ISSN	IND			21	00		00		ISSN

tab11_acc

Ezzel a táblával lehet a rekord mezőket a besorolási indexekhez hozzárendelni. Ez a tábla legfeljebb 1500 sor lehet.

tab11_word

A tábla meghatározza a kapcsolatot a rekord mezők és a szó indexek között (egy mezőt egy vagy több szóindex csoporthoz is hozzá lehet rendelni). Egy mezőt többször is listázhat a tab11_word táblában, hogy így előfordulásenként különböző szótördelő rutinokkal indexelhesse a mezőt. Ez a tábla legfeljebb 10,000 soros lehet.

Meghatározhat olyan szócsoport kódokat is, melyek legfeljebb öt karakter hosszúak lehetnek. Megjegyzés: az összetett szó index definíciók között legyen szóköz a 9-18 oszlopokban.

tab11_ind

A tab11_ind tábla mezőket jelöl ki, határoz meg a direkt indexeknek. Ez a tábla legfeljebb 500 sor lehet.

tab_word_breaking

Ez a tábla speciális szótördelési specifikációkat tartalmaz. A szótördelési eljárásokhoz lásd a Rendezés és szótördelés fejezet megfelelő részét.

tab_filing

Ezt a tábla besorolási elemek építése során alkalmazott egységesítő és normalizációs eljárásokhoz, valamint index bejegyzések és rendezési kulcsok építése során alkalmazott egységesítő eljárásokhoz használható. Bővebb információért lásd a Rendezés és szótördelés fejezetet.

tab_expand

Azokat a bővítő eljárásokat határozza meg, amelyek egy index létrehozásakor aktiválódnak. Bővebb információért lásd a Bővítőrutinok, táblák és bővített mezők indexelése fejezetet.

tab_character_conversion_line

A karakter konverzió táblák meghatározzák a karakterek ekvivalenciáit. Valamennyi karakter a saját alapértelmezett Unicode értéke alapján van rendezve. Eltérő rendezési sorrend érdekében beállíthat egy ekvivalenciát a rendezéshez. A rendszer az `$alephe_unicode` könyvtár `tab_character_conversion_line` táblájának 5. oszlopában feltüntetett karakter konverziós táblát használja.

Például a WORD-FIX előfordulásokhoz kapcsolódó karakter konverziós táblák a word fájlok szavainak létrehozására definiálják a karakterek ekvivalenciáit.

tab22

A könyvtár `tab` könyvtárának `tab22` táblája azt határozza meg, hogy a rövid bibliográfiai rekord mely mezőket tartalmazza. A tábla segítségével az is meghatározható, hogy hogyan jönnek létre ezek a mezők. A Rövid bibliográfiai rekordot a rendszer hozza létre, e tábla definíciói szerint, a rekord adatbázisba történő betöltéskor (amikor az indexelő paraméter 'Full'-ra van állítva), vagy amikor a Katalogizálás modulon keresztül rekordokat hoznak létre vagy módosítanak.

tab_sort

A `tab_sort` tábla a rendező kulcshoz rendelt mezőket és almezőket definiálja (a rendező kulcsok rekordok egy csoportját rendezik az OPAC-ban). Legfeljebb öt mező/almező kombinációt határozhat meg az egyes rendező kulcsokhoz.

A 02 és 03 kulcsokat bizonyos szervizek a Szerző és Cím szerinti különrendezés meghatározásához használják. A Szerző és Cím szerinti rendezés aktuális módja konfigurálható. Például a Szerző rendezhető 100-as vagy 110-es mező vagy a kettő együttes kombinációja szerint. Meg kell azonban győződni arról, hogy ez a konfiguráció megtartja a 02 kulcsot a szerzőségi és a 03 kulcsot a címadatnak.

tab_aut

Az authority kontroll alá vetett besorolási indexeket határozza meg. Ez a tábla - indexenkét – azt is meghatározza, hogy melyik authority adatbázis kerüljön ellenőrzésre az egyeztetés során.

tab_base

A WebOPAC-on, és a GUI-k keresési funkcióján keresztül elérhető logikai és fizikai adatbázisokat definiálja. A logikai bázisokat egy FIND parancs beállítással definiálhat, mely előszűrőként vagy érvényességi tartományként is funkcionál. A FIND parancs legfeljebb 500 karakter hosszúságú lehet.

Annak érdekében, hogy felállítsunk egy logikai bázist, mely magába foglalja valamennyi rekordot **kivéve** a rekordok egy meghatározott csoportját, használja az *alldocuments* értéket az összes rekord meghatározásához a *not* paraméterrel együtt. Például, az *alldocuments not wst=suppressed* létrehoz egy logikai bázist, mely tartalmazza az összes rekordot, kivéve azokat, melyek tartalmazzák a *suppressed* szót a *wst* szó index csoportban.

tab20

Egy authority rekordból történő besorolási index bővítéshez definiál szabályokat, a kereszthivatkozások létrehozásához.

8 Rendezés és szótördelés

Ez a fejezet a következő témaköröket tárgyalja:

Besorolási elemek és indexek rendezése

Példánylisták rendezése

Szótördelés

8.1 Besorolási elemek és indexek rendezése

Az egyes besorolási (z01), index (z11) és rendezési kulcs (z101) rekord rendelkezik egy "egységesítő kulccsal" vagy "egységesítő szöveggel". Ez a besorolási elem vagy index rendezési célokra használt alakja. Az egységesítő kulcsok értékét rendező szabályok a könyvtár *tab_filing* táblájában találhatók.

A besorolási elemek egy további sajátossága a normalizálás. A normalizálás az az eljárás, amely az ékezeteket, a legtöbb mondatvégi írásjelet, speciális karaktereket törli a besorolási elemekből. Ennek célja az apró különbségek feloldása. Ez azokban a besorolási indexekben lényeges, ahol az egyes besorolási elemek csak egyszer tárolódnak.

A normalizációs rutinok az egységesítő rutinokkal és a szöveg megjelenítő rutinokkal együtt vannak definiálva a *tab_filing* táblában.

Egy új besorolási adat adatbázishoz való hozzáadásakor, amennyiben már létezik egy másik besorolási elem ugyanazzal a normalizált szöveggel (z01-NORMALIZED-TEXT), a rendszer egyezőnek tekinti még akkor is, ha a megjelenített szöveg (z01-DISPLAY-TEXT) eltérő. Ebben az esetben az új besorolási elemhez nem lesz besorolási rekord (z01) regisztrálva és az első besorolási elem (megjelenítendő) szövege fog megjelenni.

A besorolási elemek egységesítő kulcsa két fázisban épül fel:

1. Megjelenített szöveg (z01-DISPLAY-TEXT) a normalizált szöveghez (z01-NORMALIZED-TEXT).

2. Normalizált szöveg (Z01-NORMALIZED-TEXT) az egységesítő szöveghez (Z01-REC-KEY).

Emiatt a besorolási elem egységesítő formájának létrehozása során nem szükséges azon rutinok végrehajtása, melyeket már végrehajtott a rendszer a besorolási elem normalizált szövegének létrehozásakor.

A könyvtár `tab/tab_filing` táblája határozza meg a használatban lévő normalizációs és egységesítő rutinokat. A rutinon belül az alrutinok sorrendje lényeges: például, nem kapcsolhat almező kódhoz, ha azt megelőzően a "del subfield code"-ot határozta meg (ez törli az almezőkódot). A tábla a következő oszlopokat tartalmazza:

1-es oszlop - Azonosító

Az egységesítő rutin két számjegyű azonosítója. Ezt az azonosítót a `tab00.lng` tábla 5. oszlopában (besorolási elemekhez és direkt indexekhez), valamint a `tab01.lng` tábla 3. oszlopában (rendezési kulcsokhoz) alkalmazzuk.

2 Oszlop - Rutin alkalmazás

Ez az oszlop kizárólag a besorolási elemek (Z01) esetében releváns. Irreleváns a direkt indexek (Z11) vagy a rendezési kulcsok (Z101) esetében. A rutin alkalmazását definiálja. Az elérhető opciók az alábbiak:

D = Szöveg megjelenítő eljárások

N = Normalizált szöveg eljárások

F = Egységesítő szöveg eljárások

Megjegyzés: a `tab_filing` táblában meghatározott rutinokhoz szükséges egy 'F' rész, így még ha ez a rész nem is szükséges (mert az 'N'-hez tartozó sorok elegendőek) egy 'F' sorban még mindig lennie kell "no" egységesítő eljárásnak.

3-as oszlop - Eljárás

Ez az oszlop tartalmazza az egységesítő vagy normalizációs eljárás nevét.

4-es oszlop - Paraméterek

Paraméterek az egységesítő/normalizációs eljáráshoz (amennyiben releváns). Amennyiben a paraméterek karakterek, és egy karakter nem tartozik az ASCII tartományba, akkor ezt a karaktert gépelje Unicode rendszerben.

Az alábbiak az elérhető egységesítő és normalizáló eljárások:

abbreviation: Kiveszi a pontot az egyes karakterek közül (például I.B.M. helyett IBM). Ez a rutin csak a 7-bites ASCII tartományban lévő karakterek esetében alkalmazható.

add_prefix_hash: egy kettőskereszt (#) jelet illeszt be közvetlenül a paraméterek oszlopban meghatározott almező kód után. Főként a `fix_doc_aut_duplicate` programból származtatott besorolási elemek helyes rendezéséhez szükséges.

bbk: az orosz nyelvű egységesítő szabványok speciális eljárása, melyekben a rendezési szekvencia olyan speciális karakterekből áll, melyeket Cirill karakterek, latin karakterek és számok követnek.

char_conv: a 4-es oszlopban meghatározott eljárásnév szerint hajtja végre a konverziós eljárást. A névnek egyeznie kell az /alephe/unicode/tab_character_conversion_line 1-es oszlopában meghatározott eljárásazonosítóval.

chi: az egyes karaktereket fordítja le pinjinre, a kínai karakterek (Z114) pinjin alakját tartalmazó Oracle tábla használatával, valamint hozzáadja az Unicode értéket valamennyi karakterhez. Ezen hozzáadott Unicode értékek célja, hogy megkülönböztesse az azonos pinjin értékekkel rendelkező különböző karaktereket. Mivel a pinjin egységesítő szöveg az általános latin karakterekkel együtt van sorrendbe rendezve, ezt a rutint azon listák böngészésére kell használni, melyek a 008 MARC21 mezőből származó nyelvkódot használják a nyelvi megkülönböztetéshez, és különböznek a kínaitól (például AUTC). Külön böngésző listákat lehet létrehozni az expand_doc_bib_lng_cjk expand program segítségével.

chk: a CJK "karakterek" elé tesz egy „!” jelet, valamint a karaktereket lefordítja pinjinre, a kínai karakterek (Z114) pinjin alakját tartalmazó Oracle tábla használatával, valamint a karakterekhez hozzáadja az Unicode értéket decimális jelöléssel. Ez eredményezi a pinjin egységesítő szöveg általános latin karakterektől elkülönülő rendezését.

compress_blanks_cjk: két CJK karakter közötti szóközt törli.

hangul_call_no: Koreai könyvtárakhoz rendel egységesített példány raktári jelzet rutint (3-as oszlop). A rutin a karakterek Unicode értékét alkalmazza, majd először a hangul karaktereket illeszti be, ezt követi a latin karakter, illetve a számok. További 0, vagy 1 értéket adhat meg a tab_filing 4-es oszlopában.

0 – alapértelmezett paraméter, a hangul_call_no-n az LC rutint alkalmazza

1 – a hangul_call_no-n a Dewey rutint alkalmazza

A hangul_call_no rutin a raktári jelzet példányadat részének kronológiai szám részét rendező speciális kezelését is magába foglalja (a Z30-CALL-NO és a Z30-CALL-NO-2 \$\$i almezője). A kronológia számon belüli számok nem a szekvencia számuk szerint lesznek rendezve, hanem az évszám szerint. A rendező algoritmus ezen része a teljes évadatot tartalmazó \$\$9 almezőt használja.

A hangul_call_no rutin a tab_character_conversion_line 1-es oszlopában egy speciális bejegyzést használ: FILING_HANGUL_CALLNO. A könyvtár a tab_character_conversion_line 5-ös oszlopában bármilyen - a FILING_HANGUL_CALLNO-ra alkalmazható – konverziós táblát definiálhat. A hangul_call_no rendező rutinok a definiált karakterkonverziós táblával egyeztetnek a rendező rutin alkalmazása során.

A következőkben az ./alephe/unicode/tab_character_conversion_line FILING_HANGUL_CALLNO-re érvényes filing_hangul_callno táblájának definiálására lát példát:
FILING_HANGUL_CALLNO ##### # line_utf2line_utf filing_hangul_callno

Az ./alephe/unicode/filing_hangul_callno táblát a hangul_call_no rendező rutin által használt Unicode értékek definiálására be kell állítani.

A koreai raktári jelzetek egyes almezői különböző módosításokon mennek keresztül a raktári jelzet egységesítő értékének létrehozásakor. Emiatt, a helyes kereséshez, a keresőkérdések az almezőkódokat is tartalmaznia kell.

comma: valamennyi vessző helye üres hely lesz, a \$\$a almezőben található első vessző kivételével (főként a nevek normalizálására alkalmazható).

compress: a 4-es oszlopban található karaktereket tömöríti (pl. strip – eltávolításakor). A karaktereket tömörítheti azok paraméter oszlopban meghatározott Unicode értékeinek meghatározásával. Az alábbi példában a vessző (Unicode érték 002C) és a periódus (Unicode érték 002E) tömörítődik:

```
!1 2          3          4
!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!->
01 N compress          U+002CU+002E
```

compress_blank: üres helyeket töröl

compress_perblnk: akkor tömörít egy üres helyet amikor megelőzi egy ("."). Bizonyos LC raktári jelzetek egységesítésekor kell alkalmazni.

del_lead_space: egy almezőt követő kezdő szóközt töröl. Például: \$\$a XXX \$\$aXXX-re változik. Az adatbázis adatban nincs szóköz az almezőkód után, de a to_blank rutin létrehozhat egy szóközt. Ezek szerint a del_lead_space rutint a del_subfield elé kell tenni.

del_subfield: az almező jelet (\$\$x) törli

del_subfield_code: a "\$\$" jel megmarad, de az almező kód helyére egy kötőjel ("-") kerül. Ezt a normalizálásra használják, így a besorolási elemek még akkor is egyezni fognak, amikor az almező tartalom ugyanaz, de az almező kódok különböznek.

dewey_call_no: speciális eljárás a Dewey raktári jelzetek helyes sorrendbe tételéhez.

end_punctuation: kitörli a tab_filing tábla 4-es oszlopában felsorolt karaktereket, amikor az utolsó karakter a besorolási adatban. Lényeges, hogy ezt a funkciót leginkább a / : = stb. törlésére használja egy cím végén, és így tovább. Ezek használata a "D" típusú megjelenített szöveg konverziók rutinjaihoz javasolt.

end_sub_punctuation: Ez a rutin kitörli a tab_filing tábla 4-es oszlopában felsorolt karaktereket valamennyi olyan almezőből, melyet a besorolási elem tartalmaz.

expand_num: bővítő szám. Ez a rutin egy numerikus egységesítés számjegyeihez ad 0-át első kitöltő számjegyként. A számjegyek maximális száma meghatározható egy vagy két számjegyű számként a tab_filing tábla paraméter oszlopának (4-es oszlop) értékeivel. Amennyiben a 4-es oszlop üresen marad, akkor az alapértelmezett 7 számjegyet használja a rendszer.

Az alábbi példa a tab_filing táblából:

```
!1 2          3          4
!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
11 F expand_num          13
```

get_subfields: kizárólag az almezőket használja, illetve a 4-es oszlopban listázottak szerint a "-" használatával kihagy néhányat közülük.

get_subfields_order: ez az eljárás hasonló a get_subfields rutinhoz, azzal a különbséggel, hogy ez megtartja az almezők 4-es oszlopban meghatározott sorrendjét.

icelandic_name: megváltoztatja a 7-es és 1-es almezők sorrendjét úgy, hogy a 7-es almezőt az 1-es almező után helyezi. Az OPAC böngésző listák rendezéséhez használhatja. A következő megjelenítő szöveg például:

```
$$aAlexander $$7Alfred $$1Jonsson$$c1943-
```

az alábbi egységesítő szövegre változik:

Alexander Jonsson Alfred 1943-

jpj: lefordítja valamennyi karaktert az Unicode karakter decimális értékére. Ezáltal az egységesítő szöveg az általános karakterekkel együtt kerül rendezésre. Ezért ezt a rutint azon böngésző listákhoz kell használni, melyek a MARC21 008-as mezőjéből származó nyelvkódot használják a nyelv szerinti megkülönböztetéshez, és különböznek a japántól (például AUTJ). Megjegyzés: külön böngésző listákat is létre lehet hozni az `expand_doc_bib_lng_cjk` `expand` program segítségével.

kor: lefordítja valamennyi karaktert az Unicode karakter decimális értékére. Ezáltal az egységesítő szöveg az általános karakterekkel együtt kerül rendezésre. Ezért ezt a rutint azon böngésző listákhoz kell használni, melyek a MARC21 008-as mezőjéből származó nyelvkódot használják a nyelv szerinti megkülönböztetéshez, és különböznek a koreaitól (például AUTK). Megjegyzés: külön böngésző listákat is létre lehet hozni az `expand_doc_bib_lng_cjk` `expand` program segítségével.

lc_call_no: speciális eljárás az LC raktári jelzetek helyes rendezésére. Fontos, hogy ez a rutin az általa készített index rekordokhoz hozzáadja a következő három karaktert: ! " #. Ezért az `lc_call_no` sor után nem lehet `to_blank` vagy `compress` (tömörítő) sor, mely tartalmazza ezeket a karaktereket. Fontos továbbá, hogy ez az eljárás önmagában komplett, így további tevékenység nem szükséges. Mindazonáltal a keresés elősegítése érdekében ajánlott a `del_subfield` hozzáadása.

lc_call_no_2: ugyanaz, mint az `lc_call_no` azzal a különbséggel, hogy ez a rutin lehetővé teszi a nem-LC raktári jelzetek indexekbe történő bevonását.

mc_to_mac: cseréli a kezdő mc-t mac-ra

no: ez a rutin akkor fut, ha a `tab_filing` táblában definiált eljárás nem tartalmaz 'F' szekciót. A `tab_filing` táblában definiált eljárásokhoz szükséges egy 'F' szekció, így még ha ez a szekció nem is szükséges (mert a normalizáláshoz használt 'N' szekció elegendő) egy 'F' sornak mégis lennie kell "no" egységesítő eljárással. Például:

```
01 F no
01 N to_blank                !"()-{ }<>;:~.?/\@*%=&^_`~
01 N comma
01 N del_subfield_code
01 N char_conv                FILING-KEY-10
```

non_filing: törli a kezdő szöveget a nem-egységesítő indikátor használatával

non_numeric: törli a nem numerikus karaktereket

none: Ez a rutin felépíti a raktári jelzet kulcsot úgy, ahogy a raktári jelzetet magát is.

numbers: kiveszi a vesszőt és a pontot a számok közül (pl. 2,153 helyett 2153)

pack_spaces: az összes többszörös szóköz helyet szimpla szóközzé alakítja

subfield_mab: a MAB-authority adaton alapuló besorolási adatok egységesítéséhez használható. Az azonosító szám rendezésből történő kizárása érdekében az eljárás valamennyi almező elejére betesz három szóközt a második almezőtől kezdve, és négy szóközt ad a \$\$\$9 elejéhez. Az almező kódokat eltávolítja.

suppress: ez a program eltávolítja az << and >> jelek közötti szöveget, magukkal a karakterekkel együtt. A `tab_filing` tábla 4-es oszlopában hozzáadhat vesszővel határolt paramétereket is a `suppress` rutinhoz. Íme egy példa a `tab_filing` táblából:

```

!1 2                3                4
!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
94 suppress                88-89,<<>>

```

Ha a 88-89 paraméter meg van határozva, akkor az U+0088 és U+0089 kontroll karaktereket kell használni az << and >> helyett. A <<>> paraméter az alapbeállításnak megfelelően működik.

Ha mindkét paraméter meg van határozva, akkor a beviteli szöveg kétszer lesz elrejtve (kétszer fut le a `suppress` rutin): egyszer az << and >>-del mint határolókkal, majd ismét a U+0088 és U+0089 Unicode kódokkal, mint határolókkal (vagy fordítva).

Az alábbi paraméter kombinációk megengedettek:

<<>> Az alapbeállítással megegyező

88-89 - Elrejtés kizárólag a U+0088 és U+0089 karakterekkel mint határolókkal

88-89,<<>> (2) és azután (1)

<<>>,88-89 (1) és azután (2)

to_blank: üres helyre cseréli a 4-es oszlopban felsorolt karaktereket.

Karaktereket tömöríthet ha az Unicode értéküket a paraméterek oszlopban meghatározza. Írja be a kódot az U+<Unicode érték> formában – például „U+0153”.

to_blank_2: üres helyre cseréli a paraméterek oszlopban (4-es oszlop) felsorolt karaktereket. Csak akkor cserélődik üres helyre a karakter, ha azt szóköz követi, vagy ha a mező végén található.

Például ha ezen eljárásban a vessző van felsorolva a paraméterek oszlopában, akkor "Schiller, Friedrich" "Schiller Friedrich"-re módosul, de az "egy,kettő,három" nem változik.

to_lower: kisbetűre cserél

to_carat: az almező jelet " ^^ "-ra cseréli (a besorolási adatok hierarchikus rendezéséhez).

FIGYELEM! Bár ezzel a „\$\$aArt\$\$zZambia” besorolást a „\$\$aArt, Canadian” elé sorolja, a böngésző kérdés lekeresése szóról-szóra, karakterről-karakterre halad, az összetett üres helyeket egy üres helyé alakítva. Emiatt nem lehet az „art ^^zambia”-n egy böngészés során az előtagot elhagyni, és bár a lista hierarchikusan lesz rendezve, a kezelése nehéz lesz.

year_uu: az u-t nullára cseréli (u az év formátumban ott, ahol a dátum nem ismert, például 19uu, 197u. Fontos, hogy ezt az `expand_num` rutin előtt kell aktiválni.

A következő a cím besorolási elemekre használt `tab_filing` szekcióra egy minta:

```

11 D end_punctuation                : , = ; /
11 N to_lower
11 N to_blank                        !@#%^&*()_+~={}[ ] : " ; ? , . / ~ `
11 N pack_spaces
11 N del_subfield_code

```



```
11 F del_subfield
11 F suppress
11 F numbers
11 F to_blank          $<>
11 F expand_num
11 F non_filing
11 F compress          '
11 F pack_spaces
11 F char_conv          FILING-KEY-01
11 F cjk
```

8.2 Példánylisták rendezése

A példányok rendezési sorrendje egy egyedi táblában, az ADM könyvtárban található `tab_z30_sort` táblában van meghatározva a legtöbb modul számára.

Ez a tábla tartalmaz rendezési opciókat mind a folyóirat típusú példányokra, mind a nem folyóirat típusú példányokra valamennyi modulban, mely tartalmaz példánylistát.

A rendezési opciók két elemből állnak:

Rendezési típus - meghatározza a rendezés különböző szintjeit. Például folyóirat példányok rendezhetők évfolyamonként, azt követően folyóirat szám alapján, majd rész szám alapján.

Rendezési sorrend - meghatározza hogy a sorrend növekvő vagy csökkenő lesz.

A következő tábla tartalmazza azokat a modulokat/funkciókat és kódjait, melyekkel ez a tábla foglalkozik:

Modul/funkció	Kód
WebOpac	WWW-A
Web kurzusmodul	WWW-R
Folyóirat kliens	SERIAL
Keresés kliens	SEARCH
Kölcsönzés kliens	CIRC
Példány kliens	ITEM
Példányok kötészetre	ITEM-BIND
Szerzeményezés/Folyóirat kliens	ACQ
Szervizek	BATCH
Elvesztett példány jelentés	CIR-16
A fix_doc_create_86x eljárás alkalmazásához	86x
Navigációs ablak	TREE

8.2.1 Rendezési opciók

Az elérhető rendezési típusok az alábbiak:

Folyóirat típusú példányokhoz:

00- ha a kronologikus-i(év) szóközből áll és a számozás-a(évfolyam) szóközből áll, akkor leírás+példányazonosító (példány-szekvencia). Ha a kronologikus-i(év) –ben szerepel érték, akkor kronologikus-i(év) + számozás-a(évfolyam) + számozás-c(rész) + számozás-b(issue) + példányazonosító (példány-szekvencia).

01- ha a kronologikus-i(év) szóközből áll és a számozás-a(évfolyam) szóközből áll, akkor leírás+példányazonosító (példány-szekvencia). Ha a kronologikus-i(év) –ben szerepel érték, akkor kronologikus-i(év) + számozás-a(évfolyam) + számozás-b(issue) + számozás-c(rész) + példányazonosító (példány-szekvencia).

02 - ha a kronologikus-i(év) szóközből áll és a számozás-a(évfolyam) szóközből áll, akkor leírás+példányazonosító (példány-szekvencia). Ha a kronologikus-i(év) –ben szerepel érték, akkor hol-doc-number + kronologikus-i(év) + számozás-a(évfolyam) + számozás-b(issue) + számozás-c(rész) + példányazonosító (példány-szekvencia).

03 - alkönyvtár + példányazonosító (példány-szekvencia).

06 - alkönyvtár + gyűjteménykód + kronologikus-i(év) + kronologikus-j (év) +kronologikus-k(év) + leírás + példányazonosító (copy-id) (könyvtár-specifikus).

07 - 85x-típus + alkönyvtár + gyűjtemény + kapcsolási szám (linking-number) + ha a számozás nem üres, akkor számozás. Ha számozás üres helyekből áll, akkor leírás.

08 - 85x-típus + alkönyvtár + gyűjtemény + kapcsolási szám (linking-number) + példányazonosító (copy-id) + ha a 85x-típus 4 vagy 5, akkor supp-index-o + ha a kronológia nem üres helyekből áll és a számozás sem üres, akkor kronológia + számozás.

12 – alkönyvtár + gyűjtemény + ha a kronologikus-i(év) értéke szóköz, és a számozás-a(évfolyam) értéke szóköz, akkor leírás + ha a kronologikus-i(év) nem szóköz, akkor kronologikus-i(év) + számozás-a(évfolyam) + számozás-b(issue) + számozás-c(rész) + példány (copy).

13 – vonalkód szerint (a 98-as kódú egységesítő rutin alkalmazásával)

Nem folyóirat típusú példányokhoz:

00 - ha a számozás-a(évfolyam) nem üres helyekből áll, akkor: számozás-a(évfolyam) + számozás-b + számozás-c(rész) + alkönyvtár + gyűjtemény. Ha a számozás-a(évfolyam) üres helyekből áll, akkor: leírás + számozás-c(rész) + alkönyvtár + gyűjtemény.

01 - számozás-a(évfolyam) + számozás-b + számozás-c(rész) + leírás + alkönyvtár.

02 - ha a számozás-a(évfolyam) nem üres helyekből áll, akkor: számozás-a(évfolyam) + számozás-b + számozás-c(rész) + alkönyvtár. Ha a számozás-a(évfolyam) üres helyekből áll, akkor: leírás + alkönyvtár.

03 - ha a leírás üres, akkor: számozás-a(évfolyam) + kronologikus-i(év) + számozás-b + számozás-c(rész) + alkönyvtár + példány státusz. Ha a leírás nem üres, akkor: leírás + alkönyvtár + példány státusz.

04 - ha a számozás-a(évfolyam) nem üres, akkor: hol-doc-number + számozás-a(évfolyam)+ számozás-b + számozás-c(rész) + alkönyvtár + gyűjtemény. Ha a számozás-a(évfolyam) üres, akkor: hol-doc-number + leírás + alkönyvtár + gyűjtemény.

05 - alkönyvtár + példányazonosító (példány-szekvencia).

06 - alkönyvtár + gyűjteménykód + leírás + példányazonosító (copy-id) (könyvtár-specifikus).

12 – sub_library + gyűjtemény + ha a kronologikus-i(év) szóközökből áll és a számozás-a(évfolyam) szóközökből áll, akkor leírás + ha a kronologikus-i(év) nem szóközökből áll, akkor kronologikus-i(év) + számozás-a(évfolyam) + számozás-b(issue) + számozás-c(rész) + copy.

13 - vonalkód szerint (a 98-as kódú egységesítő rutin alkalmazásával)

Az elérhető rendezési sorrendek az alábbiak:

A - növekvő

D - csökkenő

A példány, folyóirat és kölcsönzési modulok esetében több mint egy rendezési opciót határozhat meg a `tab_z30_sort` táblában, de minden egyes opciónak speciális kódot kell adni (például CIRC-1, CIRC-2, és így tovább).

Ezeket az opciókat aztán ugyancsak definiálni kell az ADM könyvtár `pc_tab_exp_field.lng` táblájában, így azok elérhetőek lesznek legördülő menü opciókként ezen modulok példánylista ablakaiban.

8.3 Szótördelés

Szóindexeléshez egy mező egyedi szavai a szótáblába íródnak be. Alapvetően egy szó egy szóközzel határolt karaktercsoportot jelent. A könyvtár `tab_word_breaking` táblája definiálja azokat az eseteket, ahol további értékeket is figyelembe vesz a rendszer. Például egy gondolatjel minősüljön-e szóköznek, vagy tömörítve legyen úgy, mintha nem is lenne?

A szöveg egy, a `tab_word_breaking` táblában szereplő rutinban listázott jellemzőket alkalmazó szótördelésen esik át. A tábla négy oszlopból áll:

1-es oszlop: a szótördelő rutin kétszámjegyű azonosítója. Az azonosító a `tab11_word` tábla 6-os oszlopában alkalmazott.

2-es oszlop: kizárólag a CJK tartomány azonosítására szolgál.

3-as oszlop: a szótördelő eljárás neve.

4-es oszlop: a szótördelő eljárás paraméterei (amennyiben releváns).

Kiegészítők a szótördelési eljáráshoz:

2_hyphen: ez a rutin két egymás mellett lévő gondolatjelet (--) cseréli le egy szóközre. Ez a szótördelő eljárás – például – a MARC21 505-ös mezőjében használható (TARTALOM MEGJEGYZÉS), melyben a gondolatjeleket elkülönítőként alkalmazzuk. Itt cseréli le az egymást követő gondolatjeleket szóközre. Következzék az 505-ös mezőre egy példa

```
5050 L $$aHow these records were discovered -- A short sketch of the Talmuds
-- Constantine's letter.
```

compress: tömöríti (azaz kiveszi) a 4-es oszlopban felsorolt karaktereket

compress_blank: törli a szóközöket

del_subfield: az almezőjelet (`$$x`) cseréli le egy szóközre

force_delimiter: ez a rutin az almező jelet (`$$x`) cseréli le egy szóköz-z-szóközre azért, hogy több almező szavai ne legyenek egymás mellettiként kezelve

to_blank_2: azokat a karaktereket módosítja szóközre, amelyeket szóköz követ

to_blank: a 4-es oszlopban felsorolt karaktereket módosítja szóközre

subf_to_sign: a második és az azt követő almezőjeleket cseréli le egy, a 4-es oszlopban felsorolt karakterre

blank_to_carat: a szóközöket `^` jelre cseréli

numbers: a számok közötti vesszőt és pontot törli, például 100,000 vagy 100.000 -> 100000

abbreviation: az egyedülálló karakterek közötti pontot törli, például I.B.M. -> IBM

marc21_41: A MARC21 041-es mezőjében található értékek elválasztásra alkalmazható

Megjegyzések

Az eljárásokat logikai sorrendben kell listázni. Például a számokat a tömörítés/törlés, vagy a `change_to_blank` előtt kell listázni, ha szóköz vagy pont van bennük. Amennyiben nem így jár el, akkor a „számok” eljárás alkalmazása után ezek nem jelennek meg.

A szótördelő eljárások a könyvtár `tab` könyvtárában található `tab11_word` táblában (6-os oszlop) definiáltak. Egy sor többször is megismételhető a `tab11_word` táblában; ezzel többszörösen indexelhető, minden egyes alkalommal különböző szótördelő eljárással.

A következő példában a 100-as mező szavai a 01 és 02-es szótördelő eljárás szerint indexeltek:

100##	-6	01	WRD WAU
100##	-6	02	WRD WAU

Az `alephe/unicode` könyvtárban egy `unicode_to_word_gen` nevű tábla található. A rendszer automatikusan ezt a táblát alkalmazza a szótördelés építésekor. A tábla olyan értékeket tartalmazhat (és tartalmaz is) amelyek egy karaktert szóközre cserélnek (a 0020 érték hozzárendelésével) vagy egy karaktert tömörítenek/törölnek (a 0000 érték hozzárendelésével). Szóindex OPAC-ban történő böngészésekor a speciális karakterek mindig a konvertált változatban jelennek meg. Így, ha az `unicode_to_word_gen` táblát úgy állítja be, hogy az umlaut ue-ként jelenjen meg, akkor az az OPAC-ban ue-ként fog megjelenni, nem umlauttal.

Megjegyzés: a rendszer háromszor írja ki a gondolatjelet és az aposztrófot: (1) mint elkülönülő szavakat; (2) úgy, ahogy pillanatnyilag vannak (gondolatjellel/aposztróffal); (3) a gondolatjel/aposztróf tömörítéssel/törléssel.

Például: a `twenty-five` a következőképpen lesz indexelve:

```
twentyfive
twenty
five
twenty-five
```

A „gondolatjelet” és az aposztrófot az `alephe/unicode/unicode_to_word_gen` fájlban az aktuális értékükkel kell hagyni, illetve mind a gondolatjelet, mind az aposztrófot nem szabad beírni a könyvtár `tab/tab_word_breaking` fájljában szereplő egyik szótördelő eljárásba sem.

Keresőkérdés szövelemzéséhez alkalmazza a 90-es eljárást a `tab_word_breaking` fájlban.

A KKK keresőeljárás során szövelemzéshez a KKK könyvtárban szereplő `tab_word_breaking`-ben a 97-es eljárást alkalmazza.

9 Bővítőrutinok, táblák és bővített mezők indexelése

Ez a fejezet a következő témaköröket tárgyalja:

Rekordbővítés

Bővítőrutinok

Bővítésre vonatkozó táblák

Bővítőmezők (virtuális mezők) indexelése

9.1 Rekordbővítés

Az ALEPH integrált könyvtári rendszer információkat tartalmaz különböző típusú rekordokról és különböző típusú "könyvtárak"-ról (adatbázisokról).

A rendszer alapértelmezettként tartalmaz egy BIB könyvtárat a bibliográfiai adatok tárolására illetve egy ADM könyvtárat adminisztratív adatok tárolására. A legtöbb telepítésnek szintén része egy AUT könyvtár az authority adatok tárolására illetve egy HOL könyvtár a holding és lelőhely adatok tárolására.

A könyvtárak, ha úgy gondolják, akkor megjeleníthetnek a nem bibliográfiai adatbázisból származó adatokat a hozzákapcsolt bibliográfiai rekorddal együtt. A könyvtárak arra is lehetőséget kapnak, hogy a felhasználó kereshesse a bibliográfiai adatbázist egyéb adatbázisok információin keresztül, mint például a lelőhely.

Az ALEPH lehetővé teszi egy adatbázis másikból történő „adatbővítését”. Ez azért lehetséges, mert kapcsolat van a különböző adatbázisok rekordjai közt. A „kiterjesztett” adatokat meg lehet jeleníteni és/vagy indexelni. A „bővítőrutinok” rekordon belüli „adatbővítésre” is alkalmazhatók.

A „bővítőfunkció” a könyvtár `tab` mappájában található `tab_expand` táblával dolgozik. Ez a tábla minden könyvtárnál megtalálható, annak ellenére, hogy ezt leginkább a bibliográfiai könyvtárak használják.

A `tab_expand` tábla három fő szempontot definiál:

- Azt a rendszerfunkciót, amelyben a bővítő program működik.
- Azt a bővítőprogramot, amely a rekord bővíthető adatait definiálja.
- A bővítőprogram további kívánt paramétereit.

Minta a `tab_expand` táblára:

```
! 1                               2                               3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
U39-DOC      expand_doc_fmt
U39-DOC      expand_doc_join
```

A tábla magyarázata:

1-es oszlop - Rendszerkontextus

Ez az oszlop tartalmazza azt a „kontextust”, amelyben a bővítés működhet.

A következő „kontextus” funkciók használhatóak:

ACC

Kontextus: besorolási index létrehozása/módosítása (`p_manage_02`).

BUF-Z403

Kontextus: A BUF-Z403 bővítést szolgálja – ez az elektronikus forrásokhoz való hozzáférést ellenőrzi. Akkor szükséges, ha – például – a 856-os mezőt, vagy minden további elektronikus tartalommal rendelkező mezőt bővít a holding könyvtárba. Ügyeljen arra, hogy az `expand_doc_bib_z403` bővítő eljárást nem alkalmazhatja a BUF-Z403 bővítő menüvel.

CREATE-Z13

Kontextus: rövid bibliográfiai rekord létrehozása/módosítása (p_manage_07).

E-DOC-<formátum szám>

Kontextus: sajátos formátummegjelenítés. Kizárólag sajátos formátumokra alkalmazható bővítőrutinok futtatásánál alkalmazható. Például az `expand_doc_uni_merge` program kizárólag akkor működik, ha a rekord ISBD formátumban jelenik meg.

Az megjelenítési formátum formátumszámának egyeznie kell a kívánt formátum `edit_doc.lng` táblában definiált formátumszámával. A fent említett `expand_doc_uni_merge` példában ha az ISBD formátumot 038-asként definiálta az `edit_doc.lng` táblában, akkor a `tab_expand` táblát a következők szerint kell definiálni:

GUI-ACCREF

Kontextus: Authority rekord megjelenítés a bibliográfiai besorolási elemből (keresés modul).

GUI-BRIEF

Kontextus: rövid megjelenítés (keresés modul).

GUI-DOC-D

Kontextus: teljes megjelenítés (keresés modul).

GUI-DOC-P

Kontextus: teljes nyomtatás (keresés modul).

HOL-LOC

Kontextus: a holding könyvtárban használjuk az `expand_doc_hol_loc_1_a` és az `expand_doc_hol_loc_2_a` bővítőprogramokkal.

HOLDING

Kontextus: példánylista megjelenítése.

INDEX

Kontextus: direkt index létrehozása/módosítása (p_manage_05).

PRE-MERGE

Kontextus: egy egyesített (union) katalógusban/nézetben szereplő rekord összevont megjelenítéséhez ad bővített mezőket.

PRINT-CAT

Kontextus: katalógusnyomtatás (p_print_04).

PRINT-CUST

Kontextus: egyéni formátum nyomtatása (p_print_01).

PRINT-COL

Kontextus: oszlopos formátum nyomtatása (p_print_08).

PRINT-REC

Kontextus: „nem preferált” besorolási elemű katalógusrekordok nyomtatása (print-05).

RET

Kontextus: rekordok visszakeresése (p_ret_01) és rendezése (p_ret_21).

SORT-DOC

Kontextus: rendezési kulcs létrehozás/módosítás (p_manage_27).

U39-DOC

Kontextus: UTIL F/4/DOC segédprogramon keresztüli rekordmegjelenítés.

UE-08

Kontextus: UE-08 (UE-08 eljárásokhoz authority rekordok bővítésére).

WEB-ACCREF

Kontextus: bibliográfiai besorolási elemből történő authority rekord megjelenítés (Web OPAC).

WEB-BRIEF

Kontextus: rövid megjelenítés (Web OPAC).

WEB-FULL

Kontextus: teljes megjelenítés (Web OPAC).

WEB-FULL-1

Kontextus: teljes megjelenítés – 01-es formátum (Web OPAC).

WEB-MAIL

Kontextus: teljes nyomtatás – mail (Web OPAC).

WEB-SAVE

Kontextus: teljes nyomtatás – mentés (Web OPAC).

WEB-SCNIND

Kontextus: címadatok megjelenítése direkt index böngészésekor (Web OPAC).

WORD

Kontextus: szóindex létrehozása/módosítása (p_manage_01).

Z00R

Kontextus: egy Z00R rekord létrehozása (p_manage_07, katalogizálás, UE_01).

Z39-SERVER

Kontextus: Z39-SERVER.

2-es oszlop – Bővítő program

Az a bővítő program, amely meghatározza a bővítendő adatokat.

3-as oszlop - Paraméterek

Bizonyos bővítőrutinok olyan további információkat igényelnek, mint például a mezőkódok. Ebben az oszlopban definiálhatja a bővítő programokhoz a további paramétereket.

Megjegyzés: az egyes bővítő programok dokumentációja jelzi, hogy igényel-e paramétereket, vagy sem (példaként lásd `expand_doc_sort_field` rutint).

9.2 Bővítőrutinok

A következő bővítőprogramok érhetők el:

`expand_doc_acronym_title`

`expand_doc_adm_bib`

`expand_doc_adm_hol`

`expand_doc_aut_aut`

`expand_doc_bib_001`

`expand_doc_bib_852_1`

`expand_doc_bib_852_title`

`expand_doc_bib_880_n`

`expand_doc_bib_accref`

`expand_doc_bib_accref_1`

`expand_doc_bib_adm`

`expand_doc_bib_avail`

`expand_doc_bib_hol`

`expand_doc_bib_hol_ana`

`expand_doc_bib_hol_usm`

`expand_doc_bib_hol_usm_2`

`expand_doc_bib_inv`

`expand_doc_bib_lng_cjk`

`expand_doc_bib_loc_1_a`

`expand_doc_bib_loc_1_b`

`expand_doc_bib_loc_1_b2`

`expand_doc_bib_loc_1_c`

`expand_doc_bib_loc_1_c2`

`expand_doc_bib_loc_3_a`

`expand_doc_bib_loc_4_a`

expand_doc_bib_loc_4_b
expand_doc_bib_loc_4_c
expand_doc_bib_loc_5_c
expand_doc_bib_loc_cleanup
expand_doc_bib_loc_dedup
expand_doc_bib_loc_disp
expand_doc_bib_loc_n
expand_doc_bib_loc_usm
expand_doc_bib_local_notes
expand_doc_bib_multi_lng
expand_doc_bib_ndu
expand_doc_bib_psts
expand_doc_bib_psts_disp
expand_doc_bib_tab04
expand_doc_bib_z30
expand_doc_bib_z30
expand_doc_bnu_initials
expand_doc_course
expand_doc_crs_bib
expand_doc_date_yrr
expand_doc_del_fields
expand_doc_deleted
expand_doc_duplicate_field
expand_doc_extract
expand_doc_extract_holding
expand_doc_fix_abbreviation
expand_doc_fmt
expand_doc_fmt_mgu
expand_doc_hld_stmt
expand_doc_hol_852_disp
expand_doc_hol_86x
expand_doc_hol_bib
expand_doc_hol_loc_1_a

expand_doc_hol_loc_2_a
expand_doc_hol_z30_86x
expand_doc_isbn_13
expand_doc_ismn_13
expand_doc_issn_isbn
expand_doc_join
expand_doc_join_filter
expand_doc_acronym_title
expand_doc_adm_bib
expand_doc_adm_hol
expand_doc_aut_aut
expand_doc_bib_001
expand_doc_bib_852_1
expand_doc_bib_852_title
expand_doc_bib_880_n
expand_doc_bib_accref
expand_doc_bib_accref_1
expand_doc_bib_adm
expand_doc_bib_avail
expand_doc_bib_hol
expand_doc_bib_hol_ana
expand_doc_bib_hol_usm
expand_doc_bib_hol_usm_2
expand_doc_bib_inv
expand_doc_bib_lng_cjk
expand_doc_bib_loc_1_a
expand_doc_bib_loc_1_b
expand_doc_bib_loc_1_b2
expand_doc_bib_loc_1_c
expand_doc_bib_loc_1_c2
expand_doc_bib_loc_3_a
expand_doc_bib_loc_4_a
expand_doc_bib_loc_4_b

expand_doc_bib_loc_4_c
expand_doc_bib_loc_5_c
expand_doc_bib_loc_cleanup
expand_doc_bib_loc_dedup
expand_doc_bib_loc_disp
expand_doc_bib_loc_n
expand_doc_bib_loc_usm
expand_doc_bib_local_notes
expand_doc_bib_multi_lng
expand_doc_bib_ndu
expand_doc_bib_psts
expand_doc_bib_psts_disp
expand_doc_bib_tab04
expand_doc_bib_z30
expand_doc_bib_z30
expand_doc_bnu_initials
expand_doc_course
expand_doc_crs_bib
expand_doc_date_yrr
expand_doc_del_fields
expand_doc_deleted
expand_doc_duplicate_field
expand_doc_extract
expand_doc_extract_holding
expand_doc_fix_abbreviation
expand_doc_fmt
expand_doc_fmt_mgu
expand_doc_hld_stmt
expand_doc_hol_852_disp
expand_doc_hol_86x
expand_doc_hol_bib
expand_doc_hol_loc_1_a
expand_doc_hol_loc_2_a

expand_doc_hol_z30_86x

expand_doc_isbn_13

expand_doc_ismn_13

expand_doc_issn_isbn

expand_doc_join

expand_doc_join_filter

Az `expand_doc_join_filter` program a könyvtár `tab` könyvtárának `tab_expand_join_filter` táblájával használható.

Ez a tábla maximum négy mezőből hoz létre virtuális mezőt. A `tab_expand_join_filter` tábla meghatározza melyik mezőket és azok mely almezőit kell egyesíteni, a sorrendjüket és azt, hogy az eredménymezőnek mi legyen a neve. Lehetőséget ad arra, hogy szűrő értékeket állítsunk be, és csoportos értéket az egyesítéshez. Csak azok a mezők lesznek egyesítve, melyek tartalmazzák a szűrő értéket és azonos csoportos értékük van.

További részletekért nézze meg a: `tab_expand_join_filter` táblát.

expand_doc_join_permute

expand_doc_join_simple

expand_doc_last_cat

expand_doc_link_to_ros

expand_doc_open_cat

expand_doc_own

expand_doc_primo_plk

expand_doc_ros_id

expand_doc_rotate

expand_doc_section

expand_doc_sort

expand_doc_sort_field

expand_doc_sort_loc_a

expand_doc_sort_loc_b

expand_doc_sort_loc_x

expand_doc_split

expand_doc_split_external

expand_doc_split_sub1

expand_doc_sysno

expand_doc_type

expand_doc_uni_merge

expand_doc_union_add_852

expand_doc_union_exclude_lib

expand_doc_yr

Megjegyzés: néhány bővítőprogram *usm* és *mab* utótagokkal szerepel. Ezek a kiterjesztések a MARC formátumtól függő bővítőprogramokban szerepelnek (pl. MARC21, MAB, UNIMARC stb.).

expand_doc_acronym_title

Az `expand_doc_acronym_title` rutin a közös szavakkal rendelkező hosszú címadatok csonkolt indexelését teszi lehetővé (például: A ... egyesület jelentése). Ez a bővítés egy olyan új indexet hoz létre, amely segítségével a felhasználó könnyebben kereshet sorozati címekre. A rutin veszi a címadat első szavának első négy betűjét, hozzákapcsolja a második szó első három betűjét, majd a harmadik szó első két betűjét, végül a negyedik szó első betűjét. Amennyiben a szavak rövidek, akkor kizárólag a meglévő betűket fogja használni. Például: a „Gen. hosp psych” „genhosps”-ként lesz indexelve.

Olyan, a rendezésből – a bibliográfiai könyvtár `tab03` táblájában felsorolt - kihagyandó szavak, mint az „és”, „a” illetve az „az” (angol nyelvben a „the” „and” „to”) nem számítanak a „csonkolt címadat” felépítésekor.

Ez a bővítés szóindexeléskor is alkalmazható.

Az indexelendő rekordformátumot, mező- és almezőkódot, illetve az alkalmazandó `tab_word_breaking` eljárást a 3-as oszlopban kell definiálni.

```
! 1          2          3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD           expand_doc_acronym_title   SE,245##a,92
WORD           expand_doc_acronym_title   BK,245##a,92
```

A bővítésnek a `tab_word_breaking` eljárást a következő formában kell alkalmaznia:

```
92 # del_subfield
92 # to_blank !          @$%^^( )_={ } [ ] : " ; < > , . ? | \
92 # compress           '
92 # to_lower
```

Megjegyzés: az 1-es oszlopban szereplő szótördelőrutin száma definiálható.

A következő példában egy keresést/böngészést támogató szóindex jön létre:

```
tab11_word:
! 1          2 3          4          5          6 7 8 9 10 11
12
!!!!!!!!-!!!!!!-!-!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!-!-!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ACT##          a          92          WAC
tab00.lng:
! 2          3 4 5 6 7 8          9 10          11
!-!!!!!!-!!!!!!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-!-
H WAC          W-026          00 00          W-act.
```

expand_doc_adm_bib

Az `expand_doc_adm_bib` program bibliográfiai adatot ad az adminisztratív rekordhoz.

Speciális BIB mezőknek a bővített ADM rekordba foglalásához, vagy onnan történő kizárásához használja az ADM könyvtár `tab_expand` táblájának 3-as oszlopát.

A `tab_expand` tábla 3-as oszlopában található paraméterek vesszővel elválasztott mezők, számuk legfeljebb 10 lehet (mindegyik 5 karakter, tartalmazhat # jelet, pl. 100##), ezzel emelhetjük be, vagy zárhatjuk ki az ADM rekordba/ból.

Konkrét mezőket egy eléjük írt mínusz jellel (-) lehet kizárni, pl. -245##,100##.

Amennyiben a 3-as oszlop üresen marad, a BIB rekord ÖSSZES mezője az ADM rekordhoz adódik.

A következő példában a bővített ADM rekordból kimarad a 260## és a 100## mező.

```
! 1                2                3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC      expand_doc_adm_bib      -260##,100##
```

Megjegyzés: ezt az `expand` programot az adminisztratív könyvtár (xxx50) `tab_expand` táblájában is definiálni kell. Ennek az `expand` programnak akkor van szerepe, amikor `short_doc`-ot (rövid dokumentum rekordot/z13) hoz létre az ADM könyvtárban (a `manage-07` szervizt használva).

A bővítő rutin viselkedése teljesen megegyezik az `expand_doc_hol_bib` rutinéval.

expand_doc_adm_hol

Az `expand_doc_adm_hol` `expand` rutin az ADM könyvtár `tab_expand` táblájában egy HOL adatot ad az ADM rekordhoz.

A `tab_expand` tábla 3-as oszlopa legfeljebb 10, vesszővel elválasztott mezőt tartalmazhat (mindegyik 5 karakter hosszú lehet és tartalmazhat kettőskeresztet, például, 85###). Ha a 3-as oszlop üres, akkor a HOL rekord összes mezője az ADM rekordhoz adódik.

Konkrét mezőket egyelőre írt mínusz jellel (-) lehet kizárni, pl. -85###,-86###).

Példa az `expand_doc_adm_hol`-ra a `tab_expand`-ban (ADM könyvtár):

```
! 1                2                3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC      expand_doc_adm_hol      85###,86###
```

Amennyiben az `expand_doc_adm_hol` rutinra a `tab_expand` 3-as oszlopában a „MERGE-TYPE=” változót alkalmazza, akkor az ADM rekorddal több, mint egy HOL rekord egyesítése lehetséges. Például ha az `expand_doc_adm_hol` rutinban ha több mint egy HOL rekord kapcsolódik az ADM rekordhoz, akkor az első HOL rekord lesz az ADM rekordba illesztve, majd a második HOL rekord a már bővített ADM rekordba lesz beillesztve és így tovább.

Megjegyzés: ezen bővítőrutin viselkedése megegyezik az `expand_doc_hol_bib`-bel.

expand_doc_aut_aut

Többnyelvű alkalmazásoknál, az `expand_doc_aut_aut` program meghatározza a besorolási rekord „lásd még” típusú authority rekordját. A program hozzáadja a besorolási elem összes

alakját a "fő" szerzőségi rekordból. Ez a program építi ki a „lásd még” mezőt az összes nyelvhez a bővebb (broader), szűkebb (narrower) és „lásd még” (see also) kifejezésekhez.

expand_doc_bib_001

Az `expand_doc_bib_001` program építi fel a 001-es mezőt, ami a rekord rendszerszámát tartalmazza. A mező kizárólag akkor épül fel, ha a 001-es mező még nem szerepel a rekordban.

expand_doc_bib_852_1

Az `expand_doc_bib_852_1` program a 852 MARC21 lelőhely mezővel bővíti a bibliográfiai rekordot. A mező a holding rekordból kerül ki és/vagy a példány rekord (z30) információi alapján jön létre. Ha a holding rekord rendelkezik 866 (bibliográfiai egység megnevezése) MARC21 mezővel, akkor a mező hozzá lesz kapcsolva ahhoz a holding rekordból származó 852-es mezőhöz, amellyel a bibliográfiai rekordot bővíti.

Az alkönyvtár (\$\$b almező) és a gyűjtemény (\$\$c almező) kódok azokba \$\$4 almezőbe és az \$\$5 almezőbe kerülnek, ahol az alkönyvtár kód és a gyűjteményi kód – a `tab_sub_library` (alkönyvtár definíciók) és `tab40` (gyűjtemény definíciók) tábla használatával – nevekre cserélődik. A példányvonalkódok a 852\$\$p almezőbe kerülnek.

A második raktári jelzet (z30-CALL-NO-2) bővítése ugyanazon almező szerint történik, mint ami az általános raktári jelzet (z30-CALL-NO) bővítésénél használatos, azonban nagybetűvel. Például:

```
852 $$bULINC$$cGEN$$HHG939.5 D38 1970$$bLincoln  
Library$$cGeneral$$HHG939.5 D38 1970
```

A bővítendő holding rekordok szűrésére definiálhatja a mezőt, almezőt és almezőtartalmat. Ezt a könyvtár `tab_expand` táblájának paraméterek oszlopában (3-as oszlop) teheti meg.

Például ha a `tab_expand` tábla a következő sort tartalmazza:

```
PRINT-REC expand_doc_bib_852_1 852##,b,ULINC
```

Amikor a bibliográfiai rekordot a holding adatokkal bővíti, akkor a holding adat kizárólag akkor kerül átemelésre, ha a 852-es mező \$\$b almezője az 'ULINC' értéket viseli. Az ezzel a definícióval nem egyező holding rekordok nem lesznek átemelve.

A szűrődefiníciók formátuma a következő:

```
MEZŐ, Almező, TARTALMAK
```

expand_doc_bib_852_title

Az `expand_doc_bib_852_title` program kiemeli a \$\$a, \$\$b, \$\$c almezőket a főcím mezőből (245 MARC21 mező) és beemeli azt a 852 MARC 21 mezőbe (\$\$a, \$\$b, \$\$c) valamint létrehoz egy új CLN mezőt. Például, ha bibliográfiai rekord a következő mezőket tartalmazza:

```
24510 L $$aWall shadows; :$$ba study in American prisons,  
852 L $$bUELECS$$hhE183.8.B7$$iL494
```

akkor az `expand_doc_bib_852_title` program a következő új CLN mezőt hozza létre:

```
CLN01 L $$hhE183.8.B7$$iL494$$1Wall shadows; : $$2a study in American prisons,
```

Amennyiben több 852-es mező is szerepel a rekordban, akkor ezeket a rendszer egyesével veszi, és ugyanazt a 245-ös mezőt adja mindegyikhez.

expand_doc_bib_880_n

Az `expand_doc_bib_880_n` program két új összefűzött mezőt hoz létre, melyek a `$$$6`-os almező segítségével kapcsolódnak össze. A `$$$6` almező tartalmazza azt az adatot, amely két – ugyanolyan tartalmú, de különböző írásrendszerű - mezőt kapcsol össze. A két új mező a következő módokon lesz összefűzve: egyszer mint [latin megjelenítés] + [anyanyelvi megjelenítés], egyszer pedig mint [anyanyelvi megjelenítés] + [latin megjelenítés]. Az összefűzés almezőnként történik egymás utáni megjelenítéssel, a megkülönböztetést az almező nagy- illetve kisbetűs kódja végzi. Például ha rekord a következő linkelt mezőket tartalmazza:

```
245 10 $$$601$$$a$osei to kako$$$bNihon Sosei Kako Gakkai shi.
```

```
245 10 $$$601$$$1$a<cím japán írással>$$$b<alcím japán írással>.
```

akkor az `expand_doc_bib_880_n` program a következő két új virtuális mezőt hozza létre:

```
245 10 $$$603$$$a$osei to kako$$$A<cím japán írással>:$$$bNihon Sosei Kako Gakkai shi. $$$B<alcím japán írással>.
```

```
245 10 $$$603$$$a<cím japán írással>$$$A$osei to kako$$$b<alcím japán írással>. $$$BNihon Sosei Kako Gakkai shi.
```

expand_doc_bib_accref

Az `expand_doc_bib_accref` program nem-preferált alakokat ad a bibliográfiai rekordhoz a kereszthivatkozásokból történő szöfelépítéshez. Ez a kiegészítő teszi lehetővé, hogy a preferált vagy a nem-preferált alakra történő keresésnél is ugyanazt az eredményt kapjuk.

Az `expand_doc_bib_accref`-et kizárólag a WORD rendszerfunkcióval együtt lehet alkalmazni.

expand_doc_bib_accref_1

Az `expand_doc_bib_accref` rutinhoz hasonlóan működik azzal a különbséggel, hogy a kereszthivatkozások adatainak `expand_doc_bib_accref_1`-gyel történő bővítése a bibliográfiai könyvtár releváns `Z01`-es rekordjának `acc` kódja alapján elnevezett sorokat szűr be.

Amennyiben – például- a releváns `Z01` rekord `acc` kódja `AUT`, akkor a bővített adatok sora az `AUT` nevet kapja. Az `expand_doc_bib_accref_1`-et kizárólag a WORD rendszerfunkcióval együtt lehet alkalmazni.

expand_doc_bib_adm

A program veszi az `ADM` rekord mezőit, és a kapcsolódó bibliográfiai rekordokat kibővíti velük.

Megjegyzés: ez a bővítés kizárólag az egy `ADM`-es környezetben működik.

expand_doc_bib_avail

Az `expand_doc_bib_avail` bővítő program a példányok és holdingok elérhetőségi adatait hozza elő. A program speciális társrekord (`Z300`) függvénye.

A bővített adat egy 'AVA' mezőben jelenik meg, melynek a következők az almezői:

```
$$$a ADM könyvtárkód
```

```
$$$b Alkönyvtár kód
```

§§c Gyűjteménykód – amennyiben egy alkönyvtárban különböző gyűjteményekben több példányról van szó, akkor kizárólag az alkönyvtár első gyűjteménye lesz megjelenítve.

§§d Raktári jelzet - amennyiben egy alkönyvtárban különböző gyűjteményekben több példányról van szó, akkor kizárólag a gyűjtemény első raktári jelzete lesz megjelenítve.

§§e Elérhetőség státusa – „elérhető” és „nem elérhető” lehet. Akkor lesz elérhető státusz rendelve a mezőhöz, ha az összes példány és a nem elérhető példányok számának különbsége pozitív. Akkor lesz nem elérhető státusz rendelve a mezőhöz, ha az összes példány és a nem elérhető példányok számának különbsége negatív vagy nulla. Amennyiben a rekordhoz nincs példány csatolva (kizárólag holding rekordok), akkor a státusz 'elérhető' lesz.

§§f Példányok száma (az egész alkönyvtárban, nem csak a lelőhelyen).

§§g Nem elérhető példányok száma (az egész alkönyvtárban, nem csak a lelőhelyen).

§§h Több évfolyam változó (Y/N) – amennyiben az első példány Z30-ENUMERATION-A értéke üres vagy 0, akkor Y, minden más esetben N.

§§i Kölcsönzések száma (az egész alkönyvtárban, nem csak a lelőhelyen).

§§j A példány gyűjteménykódja.

§§t Az 'e' almező tartalmának fordítását tartalmazza, az ./error_lng/expand_doc_bib_avail táblában megadott üzeneteknek megfelelően.

Egy példány nem elérhető, ha igaz rá a következő feltételek egyike:

- Kölcsön van (Z36 rekordja van)
- Az előjegyzési polcon van (Z37_status=S)
- Feldolgozási státusszal rendelkezik és nincs a Z30-DEPOSITORY-ID változóban meghatározott értéke.

Feldolgozási státusszal rendelkező példányok 'nem elérhető'-nek számítanak. Megjegyzés: a tab_expand tábla 3-as oszlopában határozhat meg feldolgozási státuszokat, amelyekhez tartozó példányok 'elérhető'-ként lesznek kezelve.

Amennyiben a polcra visszakerülési időt is számításba szeretné venni, akkor a tab_expand tábla 3-as oszlopában a RESHELVING=Y értéket adja meg. Ez legfőképp a valós idejű elérhetőségi funkcióra (elérhetőségi X szerviz) alkalmazható.

expand_doc_bib_hol

Az expand_doc_bib_hol program a bibliográfiai rekordhoz ad holding adatokat (holding rekordot).

Megjegyzés: ez a bővítő rutin ugyanúgy működik mint az expand_doc_hol_bib rutin.

A program argumentumai (a tab_expand 3-as oszlopa) a következők:

1. MERGE_TYPE=egyesítés száma. Megjegyzés: az ADM vagy BIB rekordba több HOL rekordot is beolvashat.
2. SUP-HOL=Y/N. A kapcsolódó bib dokumentum összes holding mezővel történő bővítését teszi lehetővé még akkor is, ha a holding rekordban szerepel az STA=SUPPRESSED mező.

3. Olyan mezők, mint a 856-os

expand_doc_bib_hol_ana

Az `expand_doc_bib_hol_ana` program a BIB rekordban egy 'a' = „ANA” almezős LKR mezőre keres, majd veszi a kapcsolódó BIB rekordszámot a 'b' almezőből, ezután pedig a holding rekord adatait kibővíti (az `expand_doc_bib_hol` program működésével megegyező módon).

Paraméter definiálásához alkalmazhatja a `tab_expand` tábla 3-as tábláját úgy, ahogy azt az `expand_doc_bib_hold` táblában tenné.

Egy példa:

A 10-es BIB rekordban az LKR mező `$$aANA$$b000000020` almezővel rendelkezik.

A 20-as BIB rekordnak van egy 30-as számú HOL rekordja.

A 10-es BIB rekordon alkalmazott bővítőprogram a 30-as HOL rekord mezőit fogja bemásolni a BIB rekordba.

expand_doc_bib_hol_usm

Az `expand_doc_bib_hol_usm` program létrehoz egy új 866 mezőt úgy, hogy a holding rekord 866 MARC21 mezőjét (szöveg szerinti holdingok - alap bibliográfiai egység) veszi és összefűzi azt a 852 MARC21 mezőjével (lelőhely). A program végül a holding rekordból hozzáadja az új 866 mezőt a bibliográfiai rekordhoz.

Megjegyzés: ha a holding rekord `STA` mezőt tartalmaz és a mező tartalma "DISPLAY" akkor csak a Web OPAC-on fog megjelenni. Amennyiben a rekord nem tartalmaz `STA` mezőt, a rekord meg fog jelenni.

Megjegyzés

Szabályozni lehet a 852-es mező `$$z` almezőjének megjelenését a 866 mezőben. Ehhez a `tab_expand` 3-as oszlopába a következő példa szerinti `SUPPRESS_SF_Z` paramétert kell beilleszteni:

```
! 1                               2                               3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC      expand doc bib hol usm      SUPPRESS SF Z
```

expand_doc_bib_hol_usm_2

Az `expand_doc_bib_hol_usm_2` program létrehoz egy új 866 mezőt úgy, hogy a holding rekord 866 MARC21 mezőjét (szöveg szerinti holdingok - alap bibliográfiai egység) összefűzi a 852 MARC21 mezőjével (lelőhely).

expand_doc_bib_inv

Az `expand_doc_bib_inv` program egy `INV` mezőt épít fel a holding rekord leltári számából (`Z30-INVENTORY-NUMBER`). Az `INV` mező szerkezete a következő:

```
INV $a [leltári szám]
```

expand_doc_bib_lng_cjk

Az `expand_doc_bib_lng_cjk` program az egyes CJK mezőihez ad egy `$$9`-es almezőt a nyelvi kóddal (chi, jpn és kor) a kínai, japán és koreai nyelvek megkülönböztetéséhez. Ez az almező

a CJK nyelvek külön indexelését végzi el. A bővítőprogram az összes CJK karakteres mezőhöz hozzáadja az almezőt.

expand_doc_bib_loc_1_a

Ez a program importálja a holding könyvtár kódját (xxx60), a holding rendszerszámát, indikátorát és almezőit a MARC21 852 (lelőhely) mezőjéből az ideiglenes PS1 mezőbe.

expand_doc_bib_loc_1_b

Ez a program importálja a példányokat az ideiglenes PS1 mezőbe az ADM link típusok alkalmazásával. Ez bibliográfiai (xxx01) és Kurzus (xxx30) könyvtárak esetében használatos.

expand_doc_bib_loc_1_b2

Ez a program importálja a példányokat az ideiglenes PS1 mezőbe az ITM link típus alkalmazásával. Ez Kurzus (xxx30) könyvtárak esetében használatos és abban az esetben, ha a bibliográfiai könyvtár ITM linkeket (pl. analitikus rekordok) használ.

Megjegyzés: az expand_doc_bib_loc_1_b és az expand_doc_bib_loc_1_b2 bővítőprogramokhoz nem tartozik z16 úgy, mint expand_doc_bib_loc_usm és az expand_doc_bib_psts esetében.

expand_doc_bib_loc_1_c

Annak érdekében, hogy a feldolgozási státusz prioritást élvezzen a példány státusszal szemben ez a program elraktározza a példány feldolgozást a §§e almezőben. Ha a példány nincs feldolgozás alatt, akkor az bővítőrutin a kölcsönzési státuszt beemeli a §§d almezőbe.

A PST rekordot közvetlenül a holding rekordból hozza létre, kihagyva az ideiglenes PS1 mező képzését akkor, ha nincsenek példányok kapcsolva a holding rekordhoz.

Ezt a programot azokon a helyeken alkalmazza, ahol a példányok és a holding rekordok össze vannak kapcsolva.

expand_doc_bib_loc_1_c2

Ez a program az expand_doc_bib_loc_1_c programhoz hasonlít azzal a kivétellel, hogy ha a holding rekordhoz nem kapcsolódnak példányok, akkor ez nem közvetlenül a holding rekordból hozza létre a PST rekordot.

Ezt a programot olyan helyeken használja, ahol a példányok és a holding rekordok nincsenek összekapcsolva.

expand_doc_bib_loc_3_a

A következő almezőket adja (a kódokat cseréli le nevekre) a rekordhoz megjelenítés céljából:

§§3 – Anyagtípus (megjelenítési forma)

§§4 – Alkönyvtár neve

§§5 – Gyűjtemény neve

§§6 – Példány kölcsönzési státusza (megjelenítési forma)

§§7 – Példány feldolgozási státusz (megjelenítési forma)

expand_doc_bib_loc_4_a

Az `expand_doc_bib_loc_usm` program eljárásával megegyezően hoz létre egy `LOC` mezőt.

expand_doc_bib_loc_4_b

Az `expand_doc_bib_psts` program eljárásával megegyezően hoz létre egy `PSTS` mezőt.

expand_doc_bib_loc_4_c

Az `expand_doc_bib_loc_usm` program eljárásával megegyezően hoz létre `SBL`, `LOC` és `STS` mezőt a linkelt példány és holding rekordokhoz.

expand_doc_bib_loc_5_c

Az `expand_doc_bib_loc_usm` program eljárásával megegyezően hoz létre `SBL`, `LOC` és `STS` mezőt a linkelt előfizetés rekordokhoz.

expand_doc_bib_loc_cleanup

A közbenső `PS1` mezőket törli a program.

expand_doc_bib_dedup

Az `expand_doc_bib_loc_dedup` program megelőzi azt, hogy a példányokkal megegyező raktári jelzetű folyóiratelőfizetések lelőhelye duplikálódjon. A lelőhelyek duplikálásának megelőzéséhez a `tab_expand` táblában az `expand_doc_bib_loc_usm` sor után kell beszúrni az `expand_doc_bib_dedup` sort.

expand_doc_bib_loc_disp

Az `expand_doc_bib_loc_disp` program az `expand_doc_bib_loc_usm` program által létrehozott `LOC` mező `$$b`, `$$c` és `$$o` almezőit bővíti ki a `$$4` (alkönyvtár), `$$5` (gyűjtemény) és `$$3` (anyag típus) almezők hozzáadásával, melyekben a kódokat nevekre cseréli.

expand_doc_bib_loc_n és expand_doc_sort_loc_x

A következő bővítő programokat arra használja, hogy lelőhelyadatokat illesszen a bibliográfiai indexekbe és megjelenítésbe. A bővítőprogramok ezen elemei integrálják az `expand_doc_bib_loc_usm` és az `expand_doc_bib_psts` funkcióit.

Ez a bővítő mechanizmus közbenső `PS1` mezőket hoz létre; a `PS1` mezők rendezve és duplum mentesen bekerülnek a `PST` mezőkbe. A `PST` mező kódjai (például: `alkönyvtár`) a megjelenítési formákat bővítik.

A `PST` mező struktúrája:

első indikátor: raktári jelzet típusa (0-8).

második indikátor: nincsen meghatározva, üres marad.

`$$0` [PST mező eredete].

Ha a mező a példány rekordból származik, akkor az almező `Z30`-t tartalmaz. (`$$0Z30`).

Ha a mező a holding rekordból származik, akkor az almező `HOL`-t tartalmaz (`$$0HOL`).

`$$1` [az eredő rekord egyedi azonosítója]

Ha a mező példány rekordból származik, akkor az almező tartalmazza a kapcsolt adminisztratív rekord rendszerszámát és a példány sorszámát, (például: \$\$1100005921000010). Formátuma: <Z30-DOC-NUMBER> <Z30-ITEM-SEQUENCE>

Ha a mező holding rekordból származik, akkor az almező tartalmazza a holding könyvtár kódját és a holding rekord rendszerszámát, (például: \$\$1USM60-000001909). Formátuma: <könyvtárkód>-<holding rendszerszám>

\$\$b [alkönyvtár kód].

\$\$c [gyűjteményi kód].

\$\$d [példány státusz] ha nincsen példány feldolgozási státusz.

\$\$e [példány feldolgozási státusz] ha van példány feldolgozási státusz a példányrekordban.

\$\$f [ideiglenes lelőhely változó].

Ha az alkönyvtár, gyűjtemény és a raktári jelzet adatok ideiglenesek (az ideiglenes lelőhely jelölődoboz kijelölt), akkor az almező tartalma Y (\$\$fY).

Ha az elhelyezés nem ideiglenes, akkor az almező tartalma N (\$\$fN).

\$\$h [raktári jelzet] ha a raktári jelzet típusa 0-3 vagy 6-8.

\$\$j [raktári jelzet] ha a raktári jelzet típusa 4.

\$\$l [raktári jelzet] ha a raktári jelzet típusa 5.

\$\$n [raktári jelzet típusa]

\$\$o [anyag típus - a tab25.lng tábla 1. oszlopa] (például: BOOK).Bármely anyagtípus, amelynek az első három karaktere ISS, mint pl. ISSBD, \$\$OISSUE lesz.

\$\$r [kapcsolt holding rekord]

A példányhoz kapcsolódó holding rekord holding könyvtár kódját és rendszerszámát tartalmazza (például: \$\$1USM60-000001909). Formátuma: <könyvtárkód>-<holding rendszerszám>

\$\$y [másolat száma]

\$\$3 [anyagtípus - megjelenítési forma: a tab25.lng tábla 3. oszlopa] (például: Book).

\$\$4 [alkönyvtár neve]

\$\$5 [gyűjtemény neve]

\$\$6 [példány kölcsönzési státusza - megjelenítési forma]

\$\$7 [példány feldolgozási státusz - megjelenítési forma]

A PST mező csak a kapcsolt példány és holding adatokból jön létre, a kapcsolt előfizetési rekordból nem.

A programok listája a következő:

expand_doc_bib_loc_1_a

expand_doc_bib_loc_1_b

expand_doc_bib_loc_1_b2
expand_doc_bib_loc_1_c
expand_doc_bib_loc_1_c2
expand_doc_sort_loc_a
expand_doc_sort_loc_b
expand_doc_bib_loc_3_a
expand_doc_bib_loc_4_a
expand_doc_bib_loc_4_b
expand_doc_bib_loc_4_c
expand_doc_bib_loc_5_c
expand_doc_bib_loc_cleanup
expand_doc_hol_loc_1_a
expand_doc_hol_loc_2_a

Ezen felül ügyeljen arra, hogy egy z07 rekord akkor generálódik egy példányhoz kapcsolt bibliográfiai rekordhoz, ha a következő példányrekord mezők egyike módosul:

Z30-SUB-LIBRARY
Z30-MATERIAL
Z30-ITEM-STATUS
Z30-COLLECTION
Z30-CALL-NO-TYPE
Z30-CALL-NO
Z30-CALL-NO-KEY
Z30-CALL-NO-2-TYPE
Z30-CALL-NO-2
Z30-CALL-NO-2-KEY
Z30-DESCRIPTION
Z30-INVENTORY-NUMBER

Ez biztosítja azt, hogy a bővített bibliográfiai rekord a kapcsolódó példány módosításakor frissüljön.

Következzék egy példa egy olyan könyvtári beállításra, amelyben a példányok holding rekordokhoz kapcsolódnak:

```
! 1                2
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
XXX-XXX          expand_doc_bib_loc_1_b
XXX-XXX          expand_doc_bib_loc_1_b2
```

```

XXX-XXX      expand_doc_bib_loc_1_c
XXX-XXX      expand_doc_sort_loc_b
XXX-XXX      expand_doc_bib_loc_2_a
XXX-XXX      expand_doc_bib_loc_3_a
XXX-XXX      expand_doc_bib_loc_cleanup

```

A következő pedig egy példa egy olyan könyvtári beállításra, ahol a példányokhoz nincs holding rekord kapcsolva:

```

! 1          2
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
XXX-XXX      expand_doc_bib_loc_1_a
XXX-XXX      expand_doc_bib_loc_1_b
XXX-XXX      expand_doc_bib_loc_1_b2
XXX-XXX      expand_doc_bib_loc_1_c2
XXX-XXX      expand_doc_sort_loc_a
XXX-XXX      expand_doc_bib_loc_2_a
XXX-XXX      expand_doc_bib_loc_3_a
XXX-XXX      expand_doc_bib_loc_cleanup

```

exp_doc_bib_loc_usm

Az `expand_doc_bib_loc_usm` program a holding rekord `Z30` (példányrekord), `Z16` (előfizetés rekord) és `852` mezőjéből (lelőhely) négy mezőt épít fel: `SBL`, `LOC`, `STS` és `PST`.

A `$$$HOL` almezős `PST` mezők alapján egy `HLD` mező is létrejön. Ez a mező a releváns példányok, elektronikus lelőhely, összesített holding adatok, index holdingok és melléklet holdingok linkjét tartalmazza. Ezt a mezőt **KIZÁRÓLAG** a teljes rekordnézetben lehet megjeleníteni. A rövid megjelenítésben nem működik.

Az `OPTIMIZE=N` programargumentum a `tab_expand` 3-as oszlopához adva az összes példányt kibővítve veszi, beleértve azokat is, amelyeknek különböző lelőhelyadattal rendelkeznek. Ha az `OPTIMITE=N` értéket alkalmazzuk, a futtatási idő nem optimalizált.

Egy példa:

```

!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD          expand_doc_bib_loc_usm          OPTIMIZE=N

```

A `Kurzus` modulban az `expand_doc_bib_loc_usm` kizárólag azokon a rekordokon működik, melyek `xxx30` doc-ja egy (`Z103`) `ADM` típusú linkkel kapcsolódik a példányhoz; az `ITM` típusú linkeken nem működik:

-egy `Z103_lkr_type` `ADM`-mel rendelkeznek, az `xxx30` azon oktatói példányai, amelyek az `xxx50` példányhoz közvetlenül kapcsolódnak.

-egy `Z103_lkr_type` `ITM`-mel rendelkeznek azok az `xxx30` rekordok, amelyek egy `xxx01` bib rekordon keresztül kapcsolódnak az `xxx50` példányhoz.

Az `expand_doc_bib_loc_usm` egy beégetett meghívásbeállítás az alábbiakhoz:

```

expand_doc_bib_loc_n_x programok.
expand_doc_bib_loc_1_b egy ADM-típusú linken keresztül olvas be példányokat;

```


expand_doc_bib_loc_1_b2 egy ITM-típusú linken keresztül olvas be példányokat;
expand_doc_bib_loc_1_b2 nem az expand_doc_bib_loc_usm által meghívott
programok egyike.

Ahhoz, hogy mindkét LKR típust tartalmazza (ADM és ITM), az xxx30 tab_expand-nak meg kell határoznia mind a _b és _b2 rutinokat:

```
CREATE-Z13 expand_doc_course
CREATE-Z13 expand_doc_bib_loc_1_a
CREATE-Z13 expand_doc_bib_loc_1_b
CREATE-Z13 expand_doc_bib_loc_1_b2
CREATE-Z13 expand_doc_bib_loc_1_c
CREATE-Z13 expand_doc_sort_loc_b
CREATE-Z13 expand_doc_bib_loc_2_a
CREATE-Z13 expand_doc_bib_loc_3_a
CREATE-Z13 expand_doc_bib_loc_4_a
```

Az SBL mező szerkezete

Indikátorok – mindkettő definiálatlan, és egy szóközt tartalmaz.

\$a[alkönyvtár kód]

Az LOC mező szerkezete:

1. indikátor: raktári jelzet típus (0-8).

2. indikátor: definiálatlan, szóközt tartalmaz

\$b [alkönyvtár kód]

\$c [gyűjtemény kód]

\$h [raktári jelzet] ha a raktári jelzet típus 0-3 vagy 6-8.

\$j [raktári jelzet] ha a raktári jelzet típus 4.

\$l [raktári jelzet] ha a raktári jelzet típus 5.

\$o [anyagtípus]

Az STS mező szerkezete

Indikátorok – mindkettő definiálatlan, és egy szóközt tartalmaz.

\$a[példánystátusz kód]

Ez a program ugyanazt a környezeti változót alkalmazza, amelyet az ALEPH akkor használ, amikor a linkelt holding rekord 852-es mezőjéből automatikusan frissíti a z16 (előfizetés rekord) és a z30 (példányrekord) rekordokat. A program a 852-es mezőnek kizárólag az aleph_start fájlban definiált correct_852_subfields környezeti változóban meghatározott almezőit bővíti. Ezen a módon a példány és a holding rekord raktári jelzetei folyamatosan kezelve lesznek a bővítés során, amikor egy listában egyesülnek.

Ezen felül ügyeljen arra, hogy a példányhoz kapcsolódó bibliográfiai rekordhoz egy z07 rekord generálódik akkor, amikor következő példányrekord-mezők egyike módosul:

-Z30-SUB-LIBRARY

-Z30-MATERIAL

-Z30-ITEM-STATUS

-Z30-COLLECTION

-Z30-CALL-NO-TYPE

-Z30-CALL-NO

-Z30-CALL-NO-KEY

-Z30-CALL-NO-2-TYPE

-Z30-CALL-NO-2

-Z30-CALL-NO-2-KEY

-Z30-DESCRIPTION

-Z30-INVENTORY-NUMBER

A bibliográfiai rekordhoz akkor is generálódik egy z07, ha a linkelt előfizetés rekord módosul.

Ez biztosítja azt, hogy a bővített bibliográfiai rekord módosuljon akkor, amikor a linkelt példányra vonatkozó adatok, vagy a linkelt előfizetési adatok módosulnak.

A PST mező szerkezete

Lásd fenn.

expand_doc_bib_local_notes

A holding rekordban tárolt helyi mezőkóddokkal bővíti a bibliográfiai rekordot – megjelenítési és indexelési célból. A holding rekordban tárolt helyi mezőkódok egy olyan társult környezetben szerepelnek, ahol több intézmény oszt meg egy bibliográfiai rekordot, és az egyik intézmény szeretne mások által nem látható helyi mezőkódokat alkalmazni. A helyi intézmény konfigurálja azt, hogy mely helyi mezőkódok kerüljenek a bibliográfiai rekordból a holding rekordba (lásd a `fix_doc_create_hol_local_notes` bejegyzését) illetve a holding rekord mely helyi mezőkódjai jelenjenek meg az OPAC-ban.

Rekord teljes megjelenítésekor (WEB-FULL rutin a `tab_expand` táblából) illetve indexeléskor a program az `alephe` könyvtár `tab_base.conf` táblájával egyeztet, és kizárólag a kiválasztott bázis helyi mezőkódjai lesznek indexelve és megjelenítve.

A következő táblák érintettek:

A `tab_expand` a bibliográfiai könyvtár (`xxx01`) `tab` könyvtárában:

Azokat az előfordulásokat határozza meg, amelyekben a program lefut. A következő mintában a holding rekordból származó helyi mezőkódok bibliográfiai rekorddal történő együttes megjelenítésének beállítását mutatja be:

```
WEB-FULL expand_doc_bib_local_notes
```

A bibliográfiai könyvtár (xxx01) `tab_expand_local_notes.conf` táblája:

Ez a tábla definiálja a holding rekord tulajdonosi kódját az OPAC-ban történő helyi mezőkódok megjelenítéséhez. A tábla a bővítendő helyi mezőkódokat is definiálja. A következő mintában az `xxx01_AA` a holding rekordokban szereplő olyan helyi mezőkódokat jeleníti meg, ahol az `OWN` mezőkód értéke `AA`. Az `xxx01_PUB` mind az `AA`, mind a `BB` tulajdonosok esetén megjeleníti a helyi mezőkódokat:

```
[xxx01_AA]
owners list = AA
owner tag = OWN
owner subfield = a
owner alternative tag = 590,690
owner alternative subfield = 9
mapping section = LCN-2-BIB

[xxx01_PUB]
owners list = AA,BB
owner tag = OWN
owner subfield = a
owner alternative tag = 590,690
owner alternative subfield = 9
mapping section = LCN-2-BIB
```

A bibliográfiai könyvtár (xxx01) `tab` mappájában található `tab_mapping` tábla.

A helyi mezőkódokat képezi le az új, indexelendő, vagy megjelenítendő virtuális mezőkódokba. A következő mintában, amennyiben a holding rekordban 690-es mező szerepel, akkor a program ezzel a mezővel bővíti a bibliográfiai rekord virtuális LCS mezőjét:

LCN-2-BIB	541## abcde	LCN abcde	Y Y
LCN-2-BIB	541## fho39	LCN fho39	Y Y
LCN-2-BIB	561## ab39	LCN ab39	Y Y
LCN-2-BIB	590## ab9	LCN ab9	Y Y
LCN-2-BIB	690## ab9	LCS ab9	Y Y

Megjegyzés: az új virtuális mezők indexelésénél és/vagy megjelenítésénél elengedhetetlen a standard megjelenítési és indexelési táblák (pl. `edit_doc_999.lng`, `tab11_word`) további testreszabása.

expand_doc_bib_multi_lng

A többnyelvű alkalmazásoknál, az `expand_doc_bib_multi_lng` program újabb nyelvi mezőket ad a rekordhoz. A program az összes olyan besorolási elemhez ad (z01 rekordok) nyelvi mezőt, amelyek ugyanahhoz az authority rekordhoz kapcsolódnak és a besorolási mező nem kereszthivatkozás.

Megjegyzés: az `expand_doc_bib_accrref` tartalmazza az `expand_doc_bib_multi_lng`-t. Ennek következtében nincs szükség arra, hogy ezt felsorolja a WORD rendszer funkció alatt. Az

`expand_doc_bib_multi_lng` programot csak az ACC rendszer funkciók alatt szükséges feltüntetni.

A BIB rekord bővítése után ahhoz, hogy kitöltesse a rendszerrel a \$\$6-os almezőt, a `tab_expand` tábla 3-as oszlopában az `add_sf6=Y` paramétert határozza meg.

expand_doc_bib_ndu

Az `expand_doc_bib_ndu` program a következő MARC21 mezőkhöz hoz létre egy virtuális TIT5 mezőt:

130, 245, 730 (összes almezőből)

240, 242, 243, 246, 770 (a \$a almezőből)

700, 710, 711, 773, 780, 785 (a \$t almezőből)

A program kitörli az írásjeleket, nagy kezdőbetűket épít a szövegbe és eltávolítja a kezdő névelőket (amennyiben az egységesítésnél rejtve vannak).

expand_doc_bib_psts

Az `expand_doc_bib_psts` program PSTS mezőt épít a holding rekord z30 (példány rekord), z16 (előfizetési rekord) és 852-es mezőjéből. Ez az eljárás mutatja a példány rekord feldolgozási státuszát, amennyiben elérhető, csak úgy, mint a raktári számot, gyűjteményt és alkönyvtárat.

A PSTS mező szerkezete:

Indikátorok - mindkettő meghatározatlan, mindkettő üresen marad:

\$b [alkönyvtár kód]

\$c [gyűjtemény kód]

\$h [raktári szám] ha a raktári jelzet típusa 0-3 vagy 6-8.

\$j [raktári szám] ha a raktári jelzet típusa 4.

\$l [raktári szám] ha a raktári jelzet típusa 5.

\$d [példánystátusz] amennyiben nincs példány feldolgozási státusz.

\$e [példány feldolgozási státusz] ha van példány feldolgozási státusz.

Megjegyzés: a példány feldolgozási státuszt az \$\$e almező tárolja. Amennyiben a példány nincs feldolgozás alatt, akkor a bővítőrutin a példány kölcsönzési státuszát tárolja a \$\$d almezőben.

Ez a program ugyanazt a környezeti változót használja, mely akkor van használatban, amikor az ALEPH automatikusan módosítja a z16 (előfizetési rekord) és z30 (példány rekord) táblát a kapcsolódó holding rekord 852 mezőjéből. A program csak azon 852 mező almezőjét bővíti, mely mező az `aleph_start` fájlban meghatározott `correct_852_subfields` környezeti változóban van meghatározva. Ily módon a bővítés során a példány és a holding rekordok raktári jelzetei következetesen lesznek kezelve egy listába történő egyesítésükkor.

Megjegyzés: az `expand_doc_bib_psts` program kiemeli az ADM és ITM linkekkel a bibliográfiai rekordhoz kapcsolódó példányokat. Az ITM linkek kiemelése elkerülhetetlen, ugyanis a bővítő program így fogja megjeleníteni a kurzus könyvtárat függetlenül attól,

hogy az ADM rekord a korzusedokumentumhoz, vagy a bibliográfiai dokumentumhoz linkelt.

Ez az expand rutin átugorja a rejtett holding rekordokat (STA\$\$aSUPPRESSED).

Megjegyzés: példányrekord következő mezőinek egyikének módosításakor a példányhoz kapcsolódó bibliográfiai rekordhoz generálódik továbbra is egy Z07 rekord:

-Z30-SUB-LIBRARY

-Z30-MATERIAL

-Z30-ITEM-STATUS

-Z30-COLLECTION

-Z30-CALL-NO-TYPE

-Z30-CALL-NO

-Z30-CALL-NO-KEY

-Z30-CALL-NO-2-TYPE

-Z30-CALL-NO-2

-Z30-CALL-NO-2-KEY

-Z30-DESCRIPTION

-Z30-INVENTORY-NUMBER

A kapcsolódó előfizetési rekord módosításkor is generálódik egy Z07 a bibliográfiai rekordhoz.

Ez biztosítja a kiemelt bibliográfiai rekord módosítását akkor, amikor a kapcsolódó példányra vagy a kapcsolódó előfizetési információra vonatkozó információt módosítják.

expand_doc_bib_psts_disp

Az expand_doc_bib_psts_disp program a PSTS mező azon \$\$b és \$\$c almezőit jeleníti meg, amelyeket az expand_doc_bib_psts hozott létre, és hozzáadja azon \$\$4 (alkönyvtár név) és \$\$5 (gyűjtemény név) almezőket, amelyekben a kód felcserélődik a névre.

Megjegyzés: az expand_doc_bib_psts program az indexelésre, míg az expand_doc_bib_psts_disp program a megjelenítésre alkalmazható.

expand_doc_bib_tab04

Az expand_doc_bib_tab04 program elsődlegesen a könyvtár tab könyvtárában található tab_expand Z39_SERVER változó előfordulásakor alkalmazható. Ez a program alkalmas arra, hogy lefordítsa az alfabetikus mezőkódokat szám értékekre (például LOC-ból 952).

Megjegyzés: A Z39 protokoll nem ismeri a nem numerikus mezőkódokat és ALEPH mezőket (mint például LOC, CAT, Z30 és így tovább) ezért ezeket az expand_doc_bib_tab04 programmal konvertálni kell.

A program a könyvtár tab könyvtárában található tab04 táblát használja. A tab04 tábla tartalmazza azokat az értékeket, amely megadja, hogy egy katalogizálási mezőkódhalmaz mely másik halmazba kerüljön konvertálásra. Az expand_doc_bib_tab04 azokat az elemeket

használja, amelyek a 90-es konverziós rutinhoz kapcsolódnak. A következő a tab04 tábla egy mintája:

```
90 Z####          9#### N
90 CAT##          956 N
90 #####          ##### N
```

Megjegyzés: a minta utolsó sorának mindig jelen kell lennie.

expand_doc_bib_z30

Ez a program a bibliográfiai rekordot bővíti a példányadatokkal. Az expand_doc_bib_z30 program a könyvtár tab könyvtárában található expand_doc_bib_z30 táblával együtt alkalmazható. A bővítő program egy új, példányadatot tartalmazó virtuális mezőt – z30-1 (a könyv típusú példányok) vagy z30-2 (folyóiratpéldányok) – hoz létre. A táblával definiálhatja a bővítendő példányrekord mezőket, illetve meghatározhatja az új bővített mező almezőszerkezetét.

A tab_expand tábla paraméter oszlopában határozhatja meg, hogy a program egyeztessen-e a tab15.lng táblával, amely az új mezők szerkezetét határozza meg, illetve azt, hogy mely mezők jöjjenek létre. A következő paraméterek közül választhat:

CONF – ez az opció használható arra, hogy meghatározzuk, hogy az alapértelmezett expand_doc_bib_z30 tábla helyett más táblát használunk. A következő egy minta a változó alkalmazására:

```
!      1              2              3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!->
U39-DOC      expand_doc_bib_z30      CONF=ex1_z30
```

Amennyiben a CONF nincs beállítva, akkor az alapértelmezett expand_doc_bib_z30 táblát alkalmazza a rendszer.

TAB15 – Ez a paraméter használható annak meghatározására, hogy a tab15.lng tábla 10-es oszlopával kell-e egyeztetni. Ez az oszlop határozza meg azt, hogy a példány megjelenjen-e a WebOPAC-ban. Amennyiben a tab_expand tábla TAB15 változója Y értéket visel, akkor a tab15.lng tábla – bibliográfiai rekordokhoz csatolt példányokra vonatkozó - 10-es oszlopának N értéke esetén a bővítés nem hoz létre új virtuális mezőket ezekhez a példányokhoz. A tab15.lng táblának meghatározásaitól függetlenül az alábbi opciókkal hozhat létre új virtuális mezőket a rendszerrel:

- a TAB15 változónak N értéket adjon, különben nem lesz definiálva
- a tab_expand tábla 3-as oszlopában szerepeljen at ALL érték.

Minta a paraméter alkalmazására:

```
! 1              2              3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!->
WEB-FULL      expand_doc_bib_z30      TAB15=Y
```

TAG – Ez a paraméter használható arra, hogy meghatározza azt, hogy egy eltérő, új bővített mezőt kell létrehozni az alapértelmezett z30-1 (könyv típusú példányok) vagy z30-2 (folyóiratpéldányok) mezők helyett.

Minta a paraméter alkalmazására:

```
! 1                               2                               3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!->
U39-DOC   expand_doc_bib_z30           TAG=LOC
```

Amennyiben ezt a paraméteret nem állítja be, akkor a z30-1 és a z30-2 lesz alkalmazva.

Megjegyzés: Különös figyelemmel alkalmazza ezt a programot, ugyanis a puffereket nagy valószínűséggel túlterheli, ezzel is teljesítménybeli problémákat okozva. Ezt a programot ne alkalmazzák azok a könyvtárak, amelyekben egy bibliográfiai rekordhoz nagy mennyiségű példányrekord kapcsolódik (többnyire a folyóiratpéldányok egyenkénti listázása miatt).

expand_doc_bib_z403 (ADAM-mal működik)

Az `expand_doc_bib_z403` programmal a bibliográfiai rekordot bővítheti az objektum adatainak információjával. Az `expand_doc_bib_z403` program a könyvtár `tab` könyvtárában található `expand_doc_bib_z403` táblával együtt alkalmazható. A bővítőprogram létrehoz egy új – z403 – virtuális mezőt, amely az objektum adatait tartalmazza. A tábla ezen felül az új bővített mező almezőszerkezetét is meghatározza.

A program alapértelmezéseit a következő paraméterekkel módosíthatja a `tab_expand` táblában:

- TAG – Ez a paraméter használható arra, hogy meghatározzuk, hogy az alapértelmezett z403 mező helyett új bővített mezőt hozunk létre.
- USAGE-TYPE – Ez a paraméter használható arra, hogy meghatározzuk a bővítőablakban megjelenő objektum típusait. Alapértelmezettként a program csak azoknak az objektumoknak az információit bővíti, ahol a z403-USAGE-TYPE értéke VIEW. Ez a paraméter használható arra, hogy meghatározzuk a bővítendő további objektumtípusokat. Az összes objektumtípushoz a z403-USAGE-TYPE-nak az ALL értéket adja. Több mint egy, de nem az összes típus megjelenítéséhez vagy ismételje meg a sort a `tab_expand` táblában, vagy ismételje meg a paramétert ugyanabban a sorban (pl. USAGE-TYPE=INDEX,USAGE-TYPE=VIEW).
- CONF – Ez a paraméter határozza meg, hogy az alapértelmezett `expand_doc_bib_z403` tábla helyett más táblát használunk.
- DISPLAY-LINK – A paraméter „N” értéke esetén az objektum adatai nem bővülnek akkor, ha az objektum Lejárati dátuma lejárt és/vagy a Megjelenés OPAC-ban változó értéke No.

expand_doc_bnu_initials

Ez a bővítő program az UNIMARC 701, 702, 711, 712 és 200-as mezőikhez ad egy virtuális \$\$\$ almezőt. A virtuális almező ezen mezők pinyin fordításaiból épülnek fel, melyek a \$\$A almezőben találhatóak. Az almező a \$\$A almező tartalmának kezdőbetűit tartalmazza (a létrehozó és/vagy címadat kezdőbetűi). Megjegyzés: ez a bővítőprogram kizárólag a kínai nyelvű telepítéseken alkalmazható, és lehetővé teszi a rekordok kezdőbetű szerinti visszakeresését.

expand_doc_course

Az `expand_doc_course` programot a Kurzus modul implementálásakor kell alkalmazni. A programnak szerepelnie kell a Kurzus könyvtár (xxx30) alatti (rendszerfunkció) összes `tab_expand` táblában.

expand_doc_crs_bib

Az `expand_doc_crs_bib` program akkor hasznos, amikor egy BIB rekordot egy másik BIB rekordból bővítettünk, és a kurzus rekordnak mind a kapcsolódó és a bővített dokumentum BIB rekordot tartalmaznia kell. A programnak szerepelnie kell a Kurzus könyvtár (xxx30) alatti (rendszerfunkció) összes `tab_expand` táblában.

Az `expand_doc_crs_bib` beállításánál definiálja a `tab_expand` tábla 3-as oszlopában azokat a mezőket, amelyekkel a kurzus dokumentumot bővíteni szeretné.

Például: `./xxx30/tab/tab_expand`

```
CREATE-Z13                expand_doc_crs_bib                331-2,1##-2
```

expand_doc_date_yrr

Az `expand_doc_date_yrr` bővítő rutin a BIB rekordhoz kapcsolódó időszakok indexelését könnyíti meg. A bővítés az USMARC rekordok (a 008-as mező 06, 07-10 és 11-14-es pozíciói) illetve az UNIMARC rekordok (a 100-as mező \$\$a almezőjének 08, 09-12 és 13-16-os pozíciói) 1. év és 2. év részében szereplő időszakokra hoz létre `YRR` mezőket.

expand_doc_del_fields

Ez a bővítő program a rekord összes mezőjét törli, a `tab_expand` tábla 3-as oszlopában meghatározott mezők kivételével. A 3-as oszlopban a mezőket vesszővel válassza el egymástól.

Példa:

```
! 1                2                3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
TEST                expand_doc_del_fields                245##,260##,500##
```

Ebben az esetben kizárólag a 245##, 260##, 500## és AVA## mezők maradnak meg.

expand_doc_deleted

Ez a rutin egy törölt rekord összes bővített sorát törli. Ezt a rutint az indexelésre releváns szekciók utolsó sorához kell adni.

expand_doc_duplicate_field

Az `expand_doc_duplicate_field` program egy mezőt duplikál egy új mező + indikátor hozzárendelésével. Az `expand_doc_duplicate_field` programot a könyvtár `tab` könyvtárában található `tab_expand_duplicate_field` táblával együtt kell alkalmazni. Ez a tábla definiálja a duplikálandó mezőt és az új, hozzárendelendő mező kódját. Például ha a `tab_expand_duplicate` mező a következő sort tartalmazza:

```
! 1    2
!!!!-!!!!
260## IMP
```

akkor a 260-as mező egy új, hozzárendelt `IMP` mezőbe fog duplikálódni. Az `expand_doc_duplicate_field` eljárást és az `expand_doc_duplicate` táblát használhatja akkor, ha az `expand_doc_split` alkalmazása problémát okoz (azaz nem őrzi meg a forrásmezőt). Amikor az `expand_doc_split` olyan mezőn alapul, amelyet az `expand_doc_duplicate_field` program hozott létre, akkor a forrásmező megmarad. A fenti példában az `IMP` mezőt úgy dolgozhatja fel az `expand_doc_split` táblával a későbbiekben, hogy nem veszíti el az eredeti 260-as mezőt.

expand_doc_extract

Az `expand_doc_extract` program a könyvtár `tab` könyvtárában található `tab_expand_extract` táblával együtt alkalmazható. Ez a tábla definiálja az almezők kibővítését az indexeléshez. Például ha a könyvtára „kronológiai alosztályok” besorolási listát szeretné a „tárgyszó további bejegyzései – helységnevek”-hez létrehozni, akkor definiálhatja, hogy a 650-es MARC21 mező `$$y` almezője egy új mezőkóddá bővüljön (például `y650`). A virtuális mezőt ezután indexelheti, vagy megjelenítheti.

Megjegyzés: a `tab_expand_extract` 4-es oszlopa határozza meg azokat az almezőelőfordulás-számokat, ahányszor az új virtuális mező létrejön. A tábla egy kivonata:

```
! 1 2 3 4
!!!!-!-!!!!-!
650## y Y650 1
```

expand_doc_extract_holding

Ez a program a holding rekordból a 852-es mezőt a kapcsolódó bibliográfiai rekordba helyezi át. Ezen felül a holding rekord 866/7/8-as mezőjének `$$a` és `$$z` almezőit az új 852-es mezőhöz adja egy `$$3`-as almezőben. Megjegyzés: minden egyes hozzáadott almező és mező közé három csillag (***) kerül. Az új mező `$$a` almezője a könyvtár kódja szerint épül fel (az első három pozíció, például USM). Például, ha a holding rekord a következő mezőket tartalmazza:

```
86631 L $$aav. 37-52$$zSome issues missing
```

akkor a 852-es mező a következőképpen adódik a kapcsolt bibliográfiai rekordhoz:

```
8520 L $$aUSM$$bUELECS$$hhE183.8.B7$$iL494$&
#36;3av. 37-52***Some issues missing
```

A `tab_expand` tábla paraméteroszlopa két programargumentumot tartalmazhat: az első a holding könyvtár (`xxx60`) `tab_fix` táblájának egy szekcióját hívja meg, míg a második azokat a példány feldolgozási státuszokat listázza, amelyeket rejtettként vesz a rendszer (a bővítésnek ki kell hagynia a példányokhoz kapcsolt holding rekordokat). Megjegyzés: ezen bővítő rutin alkalmazásához a `library_relation` táblában egy ADM USM60 USM50 értékeket tartalmazó sornak kell lennie. Például ha a `tab_expand` táblát a következőképpen definiálta:

```
! 1 2 3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC expand_doc_extract_holding HOL,OI,BP,CA,CL,CT
```

akkor a HOL a holding könyvtár `tab_fix` táblájának egy szekcióját jelenti, míg az OI, BP, CA, CL és CT a rejtett példány feldolgozási státuszok lesznek. Legfeljebb 10 példány feldolgozási státuszt határozhat meg.

Ezen felül a program nem bővíti azokat a holding rekordokat, ahol az `STA` mező értéke „SUPPRESSED”. Továbbá a program nem ellenőrzi az SE formátumú rekordokat.

Megjegyzés: a kettőskeresztet (#) jokerkarakterként használhatja, illetve ezzel határozhat meg értékeket felvehető listákat mint pl: A#, vagy #B. Ez a beállítás kiszűri az összes A-val kezdődő, vagy B-re végződő feldolgozási státuszkódot.

expand_doc_fix_abbreviation

Az `expand_doc_fix_abbreviation` program a rövidítéseket módosítja teljes szövegre. A rutinnal egy rekord bármely szöveges karakterláncát lecserélheti egy másik szövegre. Két opció közül választhat:

- A rekordhoz egy új duplikált mező adódik a szöveg nem-rövidített formájával együtt.
- A rövidített szöveg teljes szövegre cserélődik az eredeti mezőben.

A `tab_expand` tábla paraméteroszlopában (3-as oszlop) határozhatja meg azt, hogy a rekordhoz egy új duplikált mező adódik a szöveg nem-rövidített formájával együtt vagy a rövidített szöveg teljes szövegre cserélődik az eredeti mezőben.

A lehetséges értékek az ADD és a REPLACE. Az ADD érték a rekordhoz új mezőt ad, a nem-rövidített formával, míg a REPLACE lecseréli a rövidített alakot a teljes szövegre, a mező duplikálása nélkül. Amennyiben az oszlopot üresen hagyja, akkor a rendszer duplikálja a mezőt (mint az ADD értéknél). Minta a `tab_expand` táblából:

```
! 1      2      3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC   expand_doc_fix_abbreviation   ADD
```

Az `expand_doc_fix_abbreviation` programot a könyvtár `tab` könyvtárában található `tab_abbrev` táblával együtt használja. A `tab_abbrev` tábla tartalmazza a „rövidítések” listáját és azt a teljes alakot, amelyre az új - rekordhoz hozzáadott - virtuális mezőben szereplő rövidített alak lecserélődik.

Minta a `tab_abbrev` táblából:

```
! 1      2      3      4
!!!!!!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
225## Y 1st          FIRST
043## Y u-at---      Australia
043## Y u-at-ne      Australia New South Wales
```

A fenti minta alapján a következő dokumentum:

```
043 L $$au-at-ne
1001 L $$aWilliams, David
2451 L $$aThe 1st man
```

a következő módon bővül:

```
043 L $$au-at-ne
1001 L $$aWilliams, David
2451 L $$aThe 1st man
043 L $$aAustralia New South Wales
2451 L $$aThe FIRST man
```

Megjegyzés: ezt a programot `fix` programként is használhatja (a `tab_fix` alkalmazásával a `tab_expand` helyett). A különbség annyi, hogy a `fix` programként történő alkalmazáskor (indexelésre és/vagy megjelenítésre) a virtuális mezők helyett a mezők teljes alakja ténylegesen hozzáadódik a rekordhoz.

expand_doc_fmt

Az `expand_doc_fmt` program a rekord formátumából (`FMT` mező) épít fel egy `TYP` mezőt.

A TYP mező felépítése:

Indikátorok – mindkettő definiálatlan és szóközt tartalmaz

\$a[rekord formátumkód]

\$b[rekord formátum név]

expand_doc_fmt_mgu

Az `expand_doc_fmt_mgu` program egy TYP – azaz rekordtípus – mezőt épít fel. A TYP mező az LDR (06-os és 07-es pozíciók), 006 (00-ás pozíció), 007 (00-ás és 01-es pozíció) és 008 (23-as pozíció) mezők kódolása szerint jön létre.

A TYP mező szerkezete:

Indikátorok – mindkettő definiálatlan és szóközt tartalmaz

\$a[<rekord típusa>]

A TYP mező az alábbiak egyikéből generálódik:

```
<Electronic Resource>
<Web Resource>
<Microform>
<Serial>
<Electronic Journal>
<Web Journal>
<Microform Serial>
<Computer File>
<Map>
<Digital Map>
<Score>
<Sound>
<Archive/MSS>
<Visual>
<Graphic>
<Kit>
<Realia>
```

expand_doc_hld_stmt

Az `expand_doc_hld_stmt` rutin 863/4/5 számozást és időrendet generál a HOL rekordokban, a HOL rekordhoz kapcsolódó példányrekordokból. Az így generált mezőket a későbbiekben összesítő holding kimutatás létrehozására alkalmazhatja.

Bővítő rutinként az összesítő holding kimutatások a rekordmegjelenítéskor menet közben épülnek fel. Megjegyzés: egy ugyanilyen programot, a `fix_doc_hld_stmt`-t alkalmazhat a katalogizáló modulban (miután a HOL könyvtár (xxx60) `tab_fix` és `fix_doc.eng` tábláit beállította). A `fix_doc_hld_stmt` program lefuttatása után a 863/4/5 mezők hozzáadódnak a HOL rekordhoz.

Az alábbi példányrekordokat kell a 863/4/5 mezők generálásánál számításba venni:

Azok a példányok, amelyek HOL kapcsolat mezőjében HOL no. szerepel.

Olyan példányok, melyek számozás mezője nagyobb, vagy egyenlő a HOL rekord s63/4/5 (kezdőpont mező) mezőjének \$\$a almezőjében szereplő értéknel. Amennyiben nem szerepel a rekordban s63/4/5 mező, akkor a példányrekord számozás mezője nem lesz ellenőrizve. További információkért lásd az 5. bekezdést.

Példánycsoportok:

Azok a példányok lesznek figyelembe véve a holding kimutatásban, melyeknek azonos a Kapcsolati szám azonosítójuk van a §8-as almezőben és azonos a Példányazonosítójuk (ha van); amennyiben egy HOL rekordhoz több, mint egy Példányazonosító kapcsolódik, akkor példányazonosítónként külön holding kimutatás jön létre. További részletekért lásd az 5-ös bejegyzést.

„Kezdőpont” és „Példányazonosító”:

A HOL rekord s63/4/5-ös speciális mezője definiálja a példány kezdőpontját és a példányazonosítóját a holding kimutatás generálásához. Ezt a mezőt azok a könyvtárak alkalmazhatják, amelyek már rendelkeznek példányrekordra vonatkozó holding kimutatás mezőkkel. A kötet kezdeti pontjához az s63/4/5 mezőt §a almezővel, míg a példányazonosítóhoz §t almezővel alkalmazza.

Például: ha az „s63” mező a §a almezőben „3”-as értéket visel, akkor kizárólag az „3”-as (vagy annál magasabb) számozási szinttel rendelkező példányok szerepelnek a holding kimutatásban.

Amennyiben az „s63”-as mezőben a §t almező a „2”-es értéket tartalmazza, akkor kizárólag azok a példányok szerepelnek a holding kimutatásban, melyeknek a Példányszám mezőjében (Z30-COPY-ID) „2”-es szerepel.

s63 L §a3 §t2

C6340 L §81.1§a3§b2-4§i2002§j03-07§wg§t2§9Y

C6340 L §81.2§a4§b1-2§i2003§j01-03§wg§t2§9Y

C6340 L §81.3§a4§b4-5§i2003§j07-09§wg§t2§9Y

Megjegyzés: A HOL rekordban több „s63”-as mező is szerepelhet. Ebben az esetben a holding kimutatás mezők (s63/4/5) halmaza jön létre minden egyes olyan „s63”-as mező után, amely §t (Példányazonosító) almezőt tartalmaz.

Például:

86340 L §81.13§a1§i2000-2000§t1§9Y

86340 L §81.14§a2§b1-3§i2001§j01-05§wg§t1§9Y

86340 L §81.15§a2§b5-6§i2001§j09-11§t1§9Y

86340 L §82.13§a1§i2000§t2§9Y

86340 L §82.14§a2§b1§i2001§j01§wg§t2§9Y

86340 L §82.15§a2§b3§i2001§j05§wg§t2§9Y

A rutin a következő eljárásokat hajtja végre:

A §9-es almezőben „y” értékkel szereplő holding kimutatások (s63/4/5 mezők) törlődnek.

(Megjegyzés: §9Y almező jelzi, hogy a holding kimutatásokat minden egyes eljárás során újra kell generálni. Egy holding kimutatás (s63/4/5) mező a generáláskor mindig tartalmazza a §9Y almezőt. Ha a könyvtárnak olyan holding kimutatás mezői vannak amelyeket szeretne megtartani az eljárás futtatásakor, a mezőknek §9N értéket kell tartalmazniuk.

A „számba vett” példányrekordok halmaza (ahogy azt a fenti 4-es szekcióban van definiálva) számozás és időrend szerinti tartományok létrehozására lesznek tömörítve.

Például:

86340 L §82.1§a1§b1-6§i2000§j01-06§t00001§9Y

86340 L §82.2§a2§i2001§t00001§9Y

Az egyes tartományok külön-külön 863/4/5-ös mezőkbe íródnak, a megfelelő kimaradást jelző indikátorral a \$\$\$w almezőben, amennyiben szükséges. A \$\$\$w almező „g” értékkel egy „hiány” miatti kimaradást jelent (pl. olyan folyóiratszámokra, amelyek nem érkeztek be). A \$\$\$w almező „n” értéke egy nem hiányra utaló kimaradást jelent (pl. olyan folyóiratpéldányok, amelyek nem kerültek kiadásra).

Például:

```
C6340 L $$$82.3$$$a3$$$b2-5$$$i2002$$$j03-09$$$wg$$$t1
```

A fenti példában a \$\$\$w almező „g” értéke egy hiányra utaló megszakítást jelent. A 3-as évfolyam 1-es és 6-os számú füzetek nem érkeztek be.

Kimaradást jelző indikátor (\$\$\$w) jön létre azokban a példányokban, amelyek nem szerepelnek az összesítő holding kimutatásban. Az NA, vagy NP feldolgozási státuszú, illetve a leképezett példányokat a rendszer úgy veszi, mintha az NA vagy NP feldolgozási státuszuk az ADM könyvtárban (xxx50) található tab_hld_stmt tábla alkalmazásával jöttek volna létre.

Mivel a különböző intézmények különböző N-en érkezett be vagy Nem lett kiadva feldolgozási státuszkódot alkalmazhatnak, ezért a tab_hld_stmt táblát a különböző kódok egységes „NA” vagy „NP” státuszkóddá való átalakítására alkalmazhatja.

A tábla alkalmazásával egy konkrét alkönyvtárat és/vagy gyűjteményt és/vagy példánystátuszt és/vagy példány feldolgozási státuszt és/vagy kimaradást jelző indikátort képezhet NA vagy NP példány feldolgozási státusszá. Az ilyen típusú példányok kimaradnak a holding kimutatásból.

Például: ha egy könyvtár például ki szeretné hagyni a hiányzó példányokat a holding kimutatásból, akkor az MI (hiányzó) feldolgozási státuszt a következőképpen kell a tab_hld_stmt táblában NA-ra leképeznie:

```
! 1 2 3 4 5 6
!!!!!!-!!!!!!-!!!-!!!-!!!
##### ##### ## MI # NA
```

Az expand_doc_hld_stmt-t a HOL könyvtár tab_expand táblájában a 2-es oszlopban kell listázni, például:

```
! 1 2 3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!>
WEB-FULL expand_doc_hld_stmt
WEB-FULL expand_doc_hol_86x
```

A rutin 863/4/5 mezőkódokat hoz létre az UTIL F/4/doc és a WEB FULL megjelenítésben.

Megjegyzés: a 863/4/5 mezők helyett egyéb mezőkódot is definiálhat azért, hogy a rekordban már meglévő 863/4/5-ös mezők és a rutin által generált mezőkódok között különbséget tudjon tenni.

Például, ha a tab_expand tábla a következőképpen definiált:

```
! 1 2 3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC expand_doc_hld_stmt CMPRS_TAG=C63
```

akkor a holding kimutatásban a mezőkód c63 lesz:

```
C6340 L $$$81.1$$$a1$$$i2000$$$t2$$$9Y
C6340 L $$$81.2$$$a2$$$b1-5$$$i2001$$$j01-09$$$wg$$$t2$$$9Y
```

C6340 L \$\$81.3\$\$a3\$\$b2-4\$\$i2002\$\$j03-07\$\$wg\$\$t2\$\$9Y

Ezt jellemzőt akkor használhatja, ha generált adatot ellenőriz, és szeretné őket ekvivalenstíni az aktuálisan rekordba írt adatoktól.

Amennyiben olyan példányrekordokra szeretne összesítő holding kimutatásokat generálni, amelyek nem kapcsolódnak előfizetési rekordhoz, vagy nincs az előfizetési rekordjuk Pédányazonosító mezőjében az érvényes érték, akkor a:

GET_Z30_BY=HOL

paramétert a tab_expand tábla 3-as oszlopában a következőképpen kell definiálni:

```
!      1      2      3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WEB-FULL      expand_doc_hld_stmt      GET_Z30_BY=HOL
```

Azon példányrekordokra, amelyek előfizetési rekordhoz kapcsolódnak, illetve amelyeknek a Pédányazonosító mezőjükben szerepel érték (mind a példány- mind az előfizetési rekordban), akkor a:

GET_Z30_BY=COPY_ID

paramétert a következőképpen kell definiálni a tab_expand tábla 3-as oszlopában:

```
!      1      2      3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WEB-FULL      expand_doc_hld_stmt      GET_Z30_BY=COPY_ID
```

Megjegyzés: Amennyiben a tab_expand 3-as oszlopában nem definiál paramétert, akkor a rendszer az alapértelmezett paramétert alkalmazza:

GET_Z30_BY=COPY ID

expand_doc_hol_852_disp

Az expand_doc_hol_852_disp program 852-es MARC21 elhelyezés mező \$\$b és \$\$c almezőit bővíti egy \$\$4 és \$\$5 almezővel, melyekben az alkönyvtár és gyűjtemény kódok nevekkkel vannak helyettesítve.

expand_doc_hol_86x

Az expand_doc_hol_86x program szöveges holding kimutatás mezőket (866, 867 és 868) hoz létre a holding rekord 853-855 (felirat és minta) illetve 863-865 (számozás és időrend) mezői alapján.

A szöveges holding kimutatás mezők az ANSI/NISO Z39.71 – 1999 szabványnak (bibliográfiai példányok holding kimutatásai) megfelelően jönnek létre, központozás nélkül.

Amennyiben a holding rekord a következő mezőket tartalmazza:

```
85320 L $$81$$av.$$bno.$$u9$$vr$$i(year)$$j(month)$$wm
86332 L $$81.1$$a1-4$$i1941-1943
86332 L $$81.2$$a6-86$$i1945-1987
```

Akkor a szöveges holding kimutatás mezői a következőképpen jönnek létre:

```
86631 L $$av.1(1941)-v.4(1943)$$81
86631 L $$av.6(1945)-v.86(1987)$$81
```

Megjegyzés: Amennyiben a holding rekord 866/7/8 \$\$8 almezőjében ugyanaz a szekvenciaszám szerepel, mint a 863/4/5 mező \$\$8 almezőjében, akkor a program nem hoz létre egy újabb 866/7/8 mezőt.

Amennyiben a holding rekord 866/7/8 §§8 almezőjében egy 0 értékű szekvenciaszám szerepel, akkor sem hoz létre egy újabb 866/7/8 mezőt a program.

Megjegyzés: ezt a bővíítőporgramot a holding könyvtár (XXX60) tab_expand táblájában kell definiálni.

expand_doc_hol_bib

Az expand_doc_hol_bib program holding rekordhoz ad bibliográfiai adatot.

A következő táblák definiálják azt, hogy a holding rekord hogyan módosuljon/egyesüljön:

A holding könyvtár (XXX60) tab könyvtárában szereplő tab_expand tábla:

A paraméterek oszlop (3-as oszlop) határozza meg az alkalmazandó rutin nevét.

A 3-as oszlop szintaktikája a következő:

MERGE-TYPE=<a tab_merge tábla első oszlopában szereplő rutin neve>

Kivonat a tab_expand táblából:

```
!      1              2              3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WEB-FULL      expand_doc_hol_bib      MERGE-TYPE=WEB-FULL
```

A holding könyvtár (XXX60) tab könyvtárának tab_merge táblája:

A tab_merge tábla 1-es oszlopának egyeznie kell a tab_expand 3-as oszlopával.

A tab_merge tábla 2-es oszlopa azonosítja az alkalmazandó egyesítő programot.

A 3-as oszlop tartalmazza a tab_merge_overlay táblából alkalmazott egyesítő szekciót.

Kivonat a tab_merge táblából:

```
! 1 2 3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WEB-FULL      merge_doc_overlay      99
```

A holding könyvtár (XXX60) tab mappájának tab_merge_overlay táblája:

Ez a tábla definiálja azt, hogy a holding rekord miként módosuljon (egyesüljön), ha már létezik.

A tábla a tab_merge 3-as oszlopában beállított egyesítést tartalmazza.

Kivonat a tab_merge_overlay táblából:

```
!1 2 3              4
!!-!-!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
99 1 Y #####
99 2 Y 245##
99 2 Y 1####
```

A fenti példa szerint a holding rekord az összes holding rekordban katalogizált mezőt megjeleníti, a bibliográfiai rekord címadatával és elsődleges besorolási elemével kiegészítve, a következőképpen:

```
FMT      L HO
LDR      L ^^^^^nx^^a22^^^^^1i^4500
```

```

008 L 0207042u^^^8^^4001uueng0000000
LKR L $$aHOL$$lUSM01$$b000000120
CAT L $$aYOHANAN$$b10$$c20020704$$lUSM60$$h1317
8520 L $$bULINC$$cGEN$$hB819$$i.G3 1968
10010 L $$aGabriel, Leo,$$d1902-
2451 L $$aExistenzphilosophie :$$bKierkegaard, Jaspers,Heidegger, Sartre.
Dialog der Positionen /$$cLeo Gabriel. --

```

A `tab_expand` 3-as oszlopában szereplő bővítő paraméterek legfeljebb 10, vesszővel elválasztott mezőt tartalmazhatnak (mindegyik 5 karakter lehet, beleértve a kettőskereszteket is, pl. 85###). Amennyiben a 3-as oszlopot üresen hagyja, akkor a HOL rekord összes mezője hozzáadódik az ADM rekordhoz. Egy konkrét mezőt a mezőlistából a mínusz (-) jellel hagyhat ki (pl: 85## -86##).

Egy egyesített rekorddal is bővíthet már bővített rekordot. Például: az `expand_doc_lib1_lib2` esetében bővítheti a lib2 rekordot majd egyesítheti a lib1 rekorddal. Ezt a LIB2 `tab_expand` táblájában egy speciális szekció alkalmazásával teheti meg, a „LIB1-LIB2” alakban.

Az `expand_doc_hol_bib` esetében ha a „HOL-BIB” szekciót definiálja a BIB könyvtár `tab_expand` táblájában, akkor a BIB rekord lesz először bővítve (ezen szekció használatával), és kizárólag ezután adódnak/egyesülnek a mezői a bővített HOL rekordhoz/rekorddal (a HOL könyvtár `tab_expand` táblájának 3-as oszloptartalmának megfelelően).

Amennyiben az `expand_doc_bib_hol`-ra a `tab_expand` 3-as oszlopában a „MERGE-TYPE=” változót alkalmazza, akkor a BIB rekordba egynél több HOL rekordot is beolvaszthat. Például ha az `expand_doc_bib_hol`-ban több, mint egy HOL rekord kapcsolódik a BIB rekordhoz, az első HOL rekord lesz a BIB rekordba olvasztva, aztán a második HOL rekord a már egyszer bővített/egyesített BIB rekorddal fog egyesülni és így tovább.

expand_doc_hol_loc_1_a

Ez a program keresi vissza a holding rekordhoz kapcsolt összes példányadatot.

expand_doc_hol_loc_2_a

Amennyiben az összes `expand_doc_hol_loc_1_a` programmal visszakeresett – holding rekordhoz csatolt példány egy ideiglenes helyen van, akkor ez a program létrehoz egy STA mezőt, \$\$aSUPPRESSED_TEMP almezővel. A SUPPRESSED (rejtett) holding rekordok nem jelennek meg a Web OPAC példánylistában.

Megjegyzés: mind az `expand_doc_hol_loc_1_a`, mind az `expand_doc_hol_loc_2_a` a holding könyvtár `tab` mappájában található `tab_expand` könyvtárban a HOL-LOC bővítőrutin alatt definiáltak.

expand_doc_hol_z30_86x

Az `expand_doc_hol_z30_86x` program holding mezőket (v66, v67 és v68) hoz létre a holding rekord 853-855 (felirat és minta) és 853X-855X (számozás és időrend) mezőinek alapján a kapcsolódó példányrekordok adataival (a leírás mezőből) együtt. Egy példányhoz egy holding mező jön létre.

A v66 a 853/853X, a v67 a 854/854X míg a v68 a 855/855X mezőkből jön létre.

Amennyiben a holding rekordban a következő mezők szerepelnek:

```
85331 L $$av.$$bno.$$u3$$vr$$i (year) $$j (month) $$wt$$ypm01,05,09
```


853X L \$\$a2\$\$b1\$\$i2000\$\$j1\$\$320000101\$\$80

akkor a holding mező a következőképpen jön létre:

V6651 L \$\$av.2:no.1(2000:Jan.)\$\$80

Megjegyzés: ezt a bővítő programot a holding könyvtár (xxx60) tab_expand táblájában kell definiálni.

Ezen felül a példányokat a státuszuk szerint szűrheti is. Ehhez a tab_expand tábla paraméterek oszlopát kell alkalmaznia. Legfeljebb 30 példánystátuszt definiálhat (ezeket vesszővel kell elválasztani). A bővítő program kizárólag az oszlopban szereplő státuszú példányokhoz hoz létre új mezőt. Amennyiben egy mínusz (-) jelet tesz az első pozícióba, akkor a holding mezők nem jönnek létre azokhoz a státuszokhoz, melyeket az oszlop tartalmaz.

expand_doc_isbn_13

Az expand_doc_isbn_13 bővítő eljárás egy 13 számjegyű ISBN számot tartalmazó mezőt ad a rekordhoz, ha a rekord egy 10 számjegyű ISBN számot tartalmaz, illetve ugyanezt teszi fordított esetben is. Az expand_doc_isbn_13-at a tab_expand „INDEX” szekciójához kell adni, majd - a bővítés alkalmazásához – a direkt indexet (p_manage_05) újra kell építeni.

expand_doc_ismn_13

Az expand_doc_ismn_13 rutin a 13 számjegyű ISMN kódokra (az USMARC-ban a 024-es mező 2-es indikátora illetve a MAB-ban az 541-es mező) és a 10 számjegyű ellenpárjukra alkalmazható. A 10 számjegyű ISMN kódot konvertálja 13 számjegyűvé és fordítja.

Az eljárást a tab_expand tábla 3-as oszlopában is definiálni kell, mint paramétert – ez az ISMN mező indexkódja lesz (a tab00.eng táblából származó ISMN mező index kód neve). Amennyiben nem ír be paramétert, akkor az ISMN lesz az alapértelmezett index kód.

Következzék egy példa az expand_doc_ismn_13 tábla tab_expand-ban történő beállítására:

```
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>  
INDEX expand_doc_ismn_13 ISMN
```

expand_doc_issn_isbn

Az expand_doc_issn_isbn program virtuális 020/022-es mezőket hoz létre olyan 020/022-es mezőkből, melyek zárójelek között lévő karaktereket tartalmaznak. A program eltávolítja mind a zárójeleket, mind a karaktereket.

Például ha a bibliográfiai rekord a következő mezőket tartalmazza:

020 L \$\$a0226123693 (pbk. : v. 2 : alk. paper)

akkor a program a következő új virtuális mezőt hozza létre:

020 L \$\$a0226123693

expand_doc_join

Az expand_doc_join program a könyvtár tab könyvtárában található tab_expand_join táblával együtt alkalmazható. A tábla kettő, vagy több MARC mezőből hoz létre egy virtuális mezőt. A tab_expand_join tábla meghatározza azokat a mezőket és almezőket, amelyek egyesülni fognak, illetve a sorrendjüket és a létrejövő új mező nevét is.

Például ha a könyvtár a szerzőségi adatok és címadatok listáját szeretné létrehozni, akkor megadhatja a MARC21-es 100-as és 245-ös mezőinek egyesítését egy új mezőkód alatt (például TMP01). Ezután elküldheti az új virtuális mezőkódot a \$\$t almezőt tartalmazó 7XX mezőkkel együtt egy közös szerző/cím listába.

Amennyiben a mezőknek több előfordulása is van, akkor az egyesítés párok szerint történik. Például ha a tab_expand_join táblában az 595-ös és 596-os mezők egy új 599-es mezőben történő egyesítését definiálja, és a rekordban van „595 a1”, „595 a2”, „596 b1”, „596 b2”, „596 b3” mező, akkor a rendszer az „599 a1 b1” és „599 a2 b2” mezőket hozza létre. Az „596 b3”-at azonban figyelmen kívül hagyja, mert nincs 595-ös párja.

Megjegyzés: amennyiben az eredeti mezőnek nincsenek almezői, akkor az adatokat átírhatja egy új mezőbe (például a MARC21 001-es mezőjének egyesítésekor).

A következő példában a 001-es mező tartalma a \$\$x almezőbe kerül átírányításra, a 001-es és 245-ös mezőből létrehozandó új virtuális mező (SYS01) megalkotásakor.

```
! 1      2      3      4      5      6      7
!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!
SYS01 001          x      245## a t
```

Például ha a katalogizáló rekord a következő mezőket tartalmazza:

```
001    L 000003333
24504 L $$aThe Yearbook of Medicine
```

akkor a bővített új mező a következőképpen jön létre:

```
SYS01 L $$x000003333$$tThe Yearbook of Medicine
```

Ha a \$data_tab/tab_expand tábla 3-as oszlopába a TYPE=ALL paramétert határozza meg, akkor az kettő, vagy több MARC mezőből hoz létre egy virtuális mezőt.

Példa az ilyen funkcióval felruházott tab/tab_expand táblára:

```
! 1          2          3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
U39-DOC      expand_doc_join          TYPE=ALL
```

Ezt a funkciót a könyvtár tab könyvtárában található tab_expand_join táblájában is alkalmazhatja. A tab_expand_join tábla határozza meg azokat a mezőket és almezőket, amelyeket egyesíteni lehet, ezek sorrendjét, és az újonnan létrejött mező nevét is. A különbség annyi, hogy ez a program a táblában definiált összes mező kombinációjára létrehoz egy új virtuális mezőt. Például ha három mezőt szeretne egyesíteni, és az első háromszor ismétlődik a rekordban, a második szintén háromszor ismétlődik, míg a harmadik kétszer, akkor ez a program 18 új virtuális mezőt fog felépíteni.

Az expand_doc_join 4-es oszlopa (az új almező kódjára vonatkozó oszlop) a következő módszerek szerint működik:

Vesz egy, vagy több almezőt a szövegből, majd kicseréli őket a vonatkozó új almezők számára. Például:

Veszi abc-t és módosítja xyz-re.

Veszi az egész szöveget (vagy néhány almezőt) és egy új almezőre módosítja őket. Például:

Veszi *abcd*-t és *x*-re módosítja.

Pár almezőt leszámítva (- jel) veszi az összeset, és egy almezőre cseréli. Például:

Veszi a *-cd*-t, amiből *x* lesz.

Vesz egy pár almezőt (vagy a teljes szöveget) és kevesebb új almezőre cseréli. Az összeset lecseréli az elsőre. Például:

Veszi *abcd*-t és *xy*-ra módosítja. Ez veszi az *abcd*-t és *x*-re módosítja.

Az összes almezőt veszi, egy pár definiált almező kivételével, majd a definiált almezőket lecseréli egy pár új almezőre úgy, hogy az szöveget az első új almezőre módosítja. Például:

Veszi *-abcd*-t és *xy*-ra módosítja. Ez veszi az összes szöveget, kivéve *abcd*-re, majd módosítja *x*-re.

Példa egy olyan `tab_expand_join` konfigurációra, amely az `expand_doc_join` eljárásom keresztül új virtuális mezőket ad a bibliográfiai rekordhoz.

```
MAY00 260## abc      xyz      245## abcd t
MOD00 260## abc      s        245## abcd t
DEZ00 260## -a       d        245## abcd t
GOS00 260## abc      gd       245## abcd t
ROD00 260## -b       os       245## abcd t
```

expand_doc_join_filter

Az `expand_doc_join_filter` program a könyvtár `tab` könyvtárában található `tab_expand_join_filter` táblával együtt alkalmazható.

Ez a tábla legfeljebb négy mezőből hoz létre egy virtuális mezőt. A `tab_expand_join_filter` tábla határozza meg azokat a mezőket és almezőket, amelyeket egyesíteni szeretne, valamint a sorrendjüket és az új mező elnevezését is. Ezen felül szűrőértékeket és az egyesítés csoportértékét is beállíthatja. Kizárólag azok a mezők lesznek egyesítve, melyek tartalmazzák a szűrőértéket és megegyezik a csoportértékük.

További információkért lásd a `tab_expand_join_filter-t`.

expand_doc_join_permute

Ez a bővítő eljárás a `tab_expand_join` táblát használva létrehozza egy mező összes lehetséges permutációját.

Például ha a `tab_expand_join` tábla a következő beállítást tartalmazza:

```
! 1      2      3      4      5      6      7
!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!! ->
AU700 700## abc  abc  245## ab  fg
AU700 700## abc  abc  740## a   f
```

és ha a rekordban a következő sorok szerepelnek:

```
000000508 LDR      L 00000cas^^2200589^^^4500
000000508 001      L 000000508
000000508 005      L 20040725104218.0
000000508 008      L 750907c19349999pauqr1p^^^^^^0uu1a0engdd
```

000000508 1102 L \$\$aMusic Library Association.
 000000508 24500 L \$\$aNotes.
 000000508 26000 L \$\$a[Philadelphia, PA, etc.] :\$\$bMusic Library Association.
 000000508 300 L \$\$av. :\$\$bill., ports., facsims. ;\$\$c24-29 cm.
 7001 L \$\$aO'Meara, Eva Judd,\$\$seed.
 7001 L \$\$aFox, Charles Warren,\$\$d1904-\$\$seed.
 7001 L \$\$aHill, Richard S.\$\$q(Richard Synyer),\$d1901-1961,\$\$seed.
 7400 L \$\$5wid\$\$aNotes (Online)
 7400 L \$\$aNotes (Journal)

akkor a mezők a következő módon bővülnek:

AU700 L \$\$aO'Meara, Eva Judd,\$\$fNotes.
 AU700 L \$\$aFox, Charles Warren,\$\$fNotes.
 AU700 L \$\$aHill, Richard S.\$\$fNotes.
 AU700 L \$\$aO'Meara, Eva Judd,\$\$fNotes (Online)
 AU700 L \$\$aO'Meara, Eva Judd,\$\$fNotes (Journal)
 AU700 L \$\$aFox, Charles Warren,\$\$fNotes (Online)
 AU700 L \$\$aFox, Charles Warren,\$\$fNotes (Journal)
 AU700 L \$\$aHill, Richard S.\$\$fNotes (Online)
 AU700 L \$\$aHill, Richard S.\$\$fNotes (Journal)

A virtuális mezőhöz vett szöveg a tab01.eng táblában definiált egységesítő indikátornak megfelelően módosul.

expand_doc_join_simple

Az `expand_doc_join_simple` program a könyvtár tab könyvtárában található `tab_expand_join_simple` táblával együtt alkalmazható. A tábla egy virtuális mezőt hoz létre egy mező egy konkrét, vagy összes előfordulására, majd egyesíti egy másik mező egy, vagy összes előfordulásával. A `tab_expand_join_simple` meghatározza az egyesítendő mezőket és azok almezőit, a sorrendjüket illetve az újonnan létrejövő mező nevét is.

Megjegyzés: az `expand_doc_join` program indexeléskor, míg az `expand_doc_join_simple` a megjelenítéshez szükséges virtuális mezők létrehozásánál hasznos.

expand_doc_last_cat

Az `expand_doc_last_cat` program egy olyan LAS nevű mezőt ad a rekordhoz, amely megegyezik az utolsó CAT mezővel. Ezzel a bővítéssel – például – olyan visszakeresési feladatokat futtathat, amelyek kizárólag egy meghatározott katalogizáló által, vagy egy meghatározott órában módosított rekordokat vesznek találatnak.

expand_doc_link_to_ros

Az Aleph és Rosetta közötti integrációhoz. Az Aleph Web OPAC-ban ezzel a rutinnal linkelheti a Rosetta tartalom aggregátort azokhoz a rekordokhoz, amelyek Rosettába lettek exportálva. További információért olvassa el a *Hogyan integráljon Aleph-et Rosettával* című Aleph dokumentációt.

expand_doc_open_cat

Az `expand_doc_open_cat` program az első `CAT` mezőből hoz létre egy `CDATE` mezőt.

expand_doc_own

Az `expand_doc_own` program a rekord összes mezőjéhez hozzacsatolja az `OWN` mező tartalmát. A tartalom a `$$1` almezőbe kerül.

expand_doc_primo_plk

Az `expand_doc_primo_plk` a céldokumentumból a forrásra mutató link részleteit szolgáltatja. Egy új `PLK` mezőt hoz létre, amely egy dokumentum 'DN' linkjét írja le. Ezzel az új mezővel a Primoban publikált dokumentumok linkelhetik a céldokumentumot a forrásukhoz (így az 'UP' és 'DN' linkek működni fognak).

expand_doc_ros_id

Az Aleph és Rosetta közötti integrációhoz. Ez a bővítő program hozza létre a `cms_id` mezőkódot (Aleph BIB rendszerszám). További információért olvassa el a *Hogyan integráljon Aleph-et Rosettával* című Aleph dokumentációt.

expand_doc_rotate

Az `expand_doc_rotate` program egy virtuális 600-as mezőt épít fel a MARC21 600-as mező (további tárgyszó – személynév) `$$a` (személynév) és `$$t` (cím) almezőiből. Az új 600-as mezőben a `$$t` almező a `$$a` elé kerül.

Például: `600 $$a $$c $$d $$t` -ből az `expand_doc_rotate` egy `600 $$t $$a` mezőt ad a rekordhoz.

expand_doc_section

Az `expand_doc_section` program a `tab_expand` tábla paraméterek oszlopát használja a `tab_expand` (a bővítőrutin – 1-es oszlop) azon szekciójának meghatározására, amelyet a program lefutathat. Ez a program akkor hasznos, ha egyszerre több bővítő programot szeretne futtatni több, a rendszerben található előfordulásra (több bővítő rutinra). Az egyes rutinokra történő bővítőprogram-csoportok definiálása helyett a csoportot elég egyszer meghatározni, majd az összes többi programelőfordulás erre mutathat az `expand_doc_section` program segítségével. Egy példa a `tab_expand` táblában szereplő `expand_doc_section` programra:

```
!      1                      2                      3
!!!!!!!!!!!!-!!!!!!!!!!!!-
!!!!!!!!!!!!->
U39-DOC      expand_doc_section      SECTION1
WEB-FULL     expand_doc_section      SECTION1
WEB-BRIEF   expand_doc_section      SECTION1
GUI-BRIEF   expand_doc_section      SECTION1
GUI-DOC-D   expand_doc_section      SECTION1

SECTION1     expand_doc_bib_loc_usm
SECTION1     expand_doc_yr
SECTION1     expand_doc_type          tab_type_config
```

expand_doc_sort

Az `expand_doc_sort` program a könyvtár `tab` könyvtárában található `tab_expand_sort` táblával együtt alkalmazható. A program egy konkrét mező almezőit rendezi sorba a táblában meghatározott elv szerint.

expand_doc_sort_field

A program gy konkrét mezőt rendez sorba a könyvtár `tab_expand` táblájának 3-as oszlopában szereplő paramétereknek megfelelően. A következő példában a 260-as MARC21-es mező a `$$b` almező tartalmának megfelelően lesz sorba rendezve (a kiadó neve, terjesztő, stb):

A `tab_expand` tábla a következőképpen van definiálva:

```
!      1                2                3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
U39-DOC      expand_doc_sort_field      260##,b
```

A 260-as mezőket a következőképpen katalogizálták:

```
260 L $$aBogota :$$bOveja Negra,$$c1999.
260 L $$aLondon :$$bHeadline Book Publishing,$$c1994.
```

Az `expand_doc_sort_field`-re vonatkozó `tab_expand` tábla definíciói szerint a mezők a következő módon lesznek sorba rendezve:

```
260 L $$aLondon :$$bHeadline Book Publishing,$$c1994.
260 L $$aBogota :$$bOveja Negra,$$c1999.
```

expand_doc_sort_loc_a

Ez a program egyedi módon rendez `PS1` mezőket (példány + holding) egy- az egyes egyedi `PS1` mezőkhöz adott - `PST` mező létrehozásával. A `PS1` egyediségét a rendszer alkönyvtáron, gyűjteményen, raktári jelzeten és státuszon egyezteti.

Ezt a programot azok a könyvtárak alkalmazhatják, amelyekben a példányok és a holding rekordok nincsenek összekapcsolva.

expand_doc_sort_loc_b

Ez a program egyedi módon rendez `PS1` mezőket (példány + holding) egy- az egyes egyedi `PS1` mezőkhöz adott - `PST` mező létrehozásával. A `PS1` egyediségét a rendszer alkönyvtáron, gyűjteményen, raktári jelzeten, státuszon és anyag típuson egyezteti. Ügyeljen arra, hogy közvetlenül az `expand_doc_bib_loc_1_c` által jönnek létre azok a holding rekordok, amelyekhez nem kapcsolódnak más `PST` mezővel rendelkező példányok.

Ezt a programot azok a könyvtárak alkalmazhatják, amelyekben a példányok és a holding rekordok összekapcsolódnak.

Ügyeljen arra, hogy az `expand_doc_sort_loc_a` és az `expand_doc_sort_loc_b` egymást kölcsönösen kizáró programok, így egyszerre nem futtathatók.

expand_doc_spilt

Az `expand_doc_spilt` program a könyvtár `tab` mappájában található `tab_expand_split` táblával együtt alkalmazható. A program egy mezőkód tartalmát almezők szerint darabolja fel úgy, hogy az összes almezőt veszi egyiktől a másikig. Például ha a `$$a` mezőt határozza meg a táblában, akkor a `$$a $$b $$c $$a $$a $$c` a következőképpen oszlik el:

```
$a $b $c
$a
$a $c
```

Mivel a darabolás az almező összes előfordulásán lezajlik, a program több mezőt is létrehoz. A „darabolás” az almező összes adatát tartalmazza, illetve az almezőből származó összes

adatot is egészen az almező legközelebbi előfordulásáig, illetve a legutolsó sorig. A mezőkód létrejövő szekcióit virtuális „kimeneti” mezőkódokba lesznek átirányítva.

Ha a `tab_expand_split` tábla a következő sort tartalmazza:

```
      1      2      3      4
!!!!-!-----!!!!-!!!!
700## t      A700 T700
```

akkor az alábbi mező:

```
700 L $$aMendelssohn-Bartholdy, Felix$$tLider ohne Worte,$$mpiano
```

a következőképpen darabolódik fel:

```
A700 L $$aMendelssohn-Bartholdy, Felix
T700 L $$tLider ohne Worte,$$mpiano
```

expand_doc_split_external

Ez a program a – könyvtár `tab` mappájában található - `tab_expand_external` táblával együtt egy mezőkód tartalmát darabolja fel úgy, hogy egy többször előforduló, külső lelőhelyet (856, 505 stb) tartalmazó almezőt külön mezőkódos mezőkre osztja fel (a meghatározott almező összes előfordulására).

A program meghatározott almezők mellett duplikálja az összes almezőt, és egy új mezőkódú mezőbe illeszti be őket.

Ügyeljen arra, hogy a program által létrehozott új sorok az eredeti mezőkódot fogják tartalmazni, míg az eredeti sor rejtve marad.

A `tab_expand_external` táblát mezők felülvizsgálatára is alkalmazhatja: külső lelőhelyeket tartalmazó almezők többszörös előfordulását keresheti vele, a következőképpen:

```
! 1 2
!!!!-!-
856## u
505## u
```

Például ha a katalogizáló rekord a következő mezőt tartalmazza:

```
856 L
$$uhttp://jefferson.village.virginia.edu/pmc/contents.all.html$$uhttp://lcweb
2.loc.gov/ammem/ead/jackson.sgm
```

akkor az új, bővített mezők a következőképpen jönnek létre:

```
856 L $$uhttp://jefferson.village.virginia.edu/pmc/contents.all.html
856 L $$uhttp://lcweb2.loc.gov/ammem/ead/jackson.sgm
```

expand_doc_split_sub1

Az `expand_doc_split`-tel egyező program. Mindkettő - a könyvtár `tab` mappájában szereplő – a `tab_expand_split` táblával egy mezőkód tartalmát darabolja fel külön mezőkódokra almezőnként. A különbség az, hogy az `expand_doc_split_sub1` program futtatásának eredményeképpen párosított mezők egy \$\$1 almezőt kapnak, az eredeti párosítást mutató szekvenciaszámmal.

Például ha a bibliográfiai rekord az alábbi mezőket tartalmazza:

```
7001 L $$aShakespeare, William,$$d1564-1616.$$tPlays.
7001 L $$aDonne, John.$$tSonnets
```

és a `tab_expand_split` tábla a következőképpen definiált:

```
! 1 2 3 4
!!!!-!-----!!!!-!!!!
700## t A700 T700
```

akkor az `expand_doc_split_sub1` program a következő mezőket hozza létre újonnan:

```
A700 L $$aShakespeare, William,$$d1564-1616.$$1001
A700 L $$aDonne, John.$$1002
T700 L $$tPlays.$$1001
T700 L $$tSonnets$$1002
```

A bővítő rutin a rövid rekordok (z0101) helyes felépítéséhez szükséges. A rövid rekordok felépítése a \$\$1 almezőn múlik, ugyanis az elválasztott mezők párosítása ez alapján történik. Ha nem alkalmazza ezt a programot, akkor a fenti példára a rövid rekordok a következőképpen jönnek létre:

```
$$aShakespeare, William,$$d1564-1616.$$tPlays.
$$aDonne, John.$$tPlays.
```

Az expand_doc_split_sub1 program alkalmazásakor a rövid rekordot építő program azt a T700-at preferálja, amelyben az az eredeti A700-zal egyező \$\$1 almező szerepel, amely a rövid rekord építését indította el. A rövid rekordok a következőképpen jönnek létre:

```
$$aShakespeare, William,$$d1564-1616.$$tPlays.
$$aDonne, John.$$tSonnets.
```

expand_doc_syno

Az expand_doc_syno program elsősorban a könyvtár tab mappájában szereplő tab_expand Z39_SERVER változójára vonatkozik. Ez a program egy virtuális SYS mezőrekordot hoz létre a Z39.50 szerverre. Ez a program a könyvtár tab mappájában található tab04 táblával együtt működik. A tab04 tábla a „SYS” mezők kód numerikus mezők kódá történő átalakításának specifikációit állítja be. Ügyeljen arra, hogy ennek mindig meg kell jelennie a 90-es konverziós rutin alatt is. A következő egy kivonat a tab04 táblából:

```
90 SYS##          903## N
90 #####          ##### N
```

Ügyeljen arra, hogy ezen minta utolsó sora mindig jelen legyen. Ezen felül ügyeljen arra is, hogy a programot az expand_doc_bib_tab04 program előtt, a Z39_SERVER bejegyzés alatt kell definiálni, a következőképpen:

```
!      1          2
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
Z39_SERVER   expand_doc_syno
Z39_SERVER   expand_doc_bib_tab04
```

expand_doc_type

Ezzel a programmal új mezőt hozhat létre a tab_expand tábla paraméterek oszlopában paraméterként definiált konfigurációs tábla specifikációi szerint (lásd a konfigurációs táblák expand_doc_type részét). Ez a program a táblával együtt akkor alkalmazható, ha – például – egy - a rekordban szereplő mező(k) tartalmán alapuló - rekord formátumát tartalmazó mezőt szeretne létrehozni (például az LDR és a 008-as mezők kombinációja). A következő példában az új mező - egy LDR és 008-as mező értékein lefuttatott egyeztetés alapján - a „FILM” karakterláncot fogja tartalmazni:

```
TYP      Film          LDR F06-01 EQUAL g
          008 F33-01 EQUAL m
```

A fenti példában a TYP mező (\$\$aFilm) akkor jön létre, ha az LDR mező 06-os pozíciója egy 'g'-t, és a 008-as mező 33-as pozíciója egy 'm'-et tartalmaz. A mező felépítése a következő:

```
TYP L $$aFilm
```

Ügyeljen arra, hogy a konfigurációs tábla nevét (például tab_type_config) paraméterként a könyvtár tab_expand táblájának 3-as oszlopához paraméterként adhatja.

Az expand_doc_type-ot ismétlődő mezőkkel is alkalmazhatja. Ez különösen az EXIST funkcióra hasznos. Például:

```
!!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!!!!!!!!!-
!!!!!!!!!!!!-!!!!!!!!!!!!!!
TYP      EL Electronic material          856## u          EXIST
```

Ügyeljen a következőkre:

Ha a tab_expand második paramétere üres, vagy Y, a következő módon:

```
!      1          2          3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
```



```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD          expand_doc_type          tab_type_config.eng,Y

```

akkor a program egyezteteti az első mezőkód+almező feltétellel egyező mezőkódot. Ennek megfelelően az EQUAL, N-EQUAL vagy MATCH funkciókat nem alkalmazhatja az ugyanazzal az almezővel rendelkező ismétlődő mezőkre. Például:

```

TYP    Publisher          260## b          EQUAL  Ocllc

```

A fenti sor nem működik az ilyen típusú rekordokkal, mivel ennek csak az első sorát veszi:

```

000000000 L 26001 $$aNew York$$bRandom House
000000000 L 26001 $$aNew York$$bOCLC

```

Amennyiben a tab_expand második paramétere „N” a következőképpen:

```

!      1          2          3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD          expand_doc_type          tab_type_config.eng,N

```

akkor a program a rekord összes ismétlődő mezőjére és almezőjére keres.

Például:

A tab_type_config a következő sorokat tartalmazza:

```

! 1      2          3          4          5          6
      7
!!!!!!!!-!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!!!!!!!!!-!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!
LOC      LOC-1          852## b          EQUAL      WID
LOC      LOC-2          852## b          EQUAL      HIL
LOC      LOC_3          852## b          EQUAL      MED
LOC      LOC-4          852## b          EQUAL      LAW

```

Egy olyan rekordra, amely a következő mezőket tartalmazza:

```

852 L $$bLAW
852 L $$bMED
852 L $$bHIL
852 L $$bWID

```

A bővítés eredménye pedig a következő:

```

LOC L $$aLOC-1
LOC L $$aLOC-2
LOC L $$aLOC_3
LOC L $$aLOC-4

```

expand_doc_uni_merge

UNIMARC-os könyvtárak alkalmazzák linkelt rekordok egyesítésére, valamint együttes megjelenítésére. Például linkelt rekordok rövid formátumban történő megjelenítéséhez a tab_expand-ban a következő sornak kell szerepelnie:

```

!      1          2
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
WEB-BRIEF    expand_doc_uni_merge

```

expand_doc_union_add_852

Egy egyesített katalógusban egy, a SID \$\$b almezőjének értékét tartalmazó „ál” 852-es mezőt ad a rekordhoz olyan esetekben, ahol csak 856-os mezők vannak (852 nélkül). A programot a tab_expand tábla PRE-MERGE szekciójában kell elhelyezni.

Ügyeljen arra, hogy a z124 megjelenítéséhez egy egyező z124-nek is lennie kell.

expand_doc_union_exclude_lib

A program megelőzi az azonosítatlan felhasználóktól, Web OPAC-on keresztül egyesített rekordok letöltését. A tab_expand 3-as oszlopát alkalmazhatja ezen bővítőrutin beállítására – azaz azokat a közreműködő könyvtárakat adhatja meg, amelyeket ez a bővítő rutin töröl. Az adott könyvtárak SID, 852, 856 és 866-os mezőit paraméterként megadva törölheti a könyvtárakat. Amennyiben egynél több könyvtárádot ad meg a 3-as oszlopban, akkor a

kódokat vesszővel válassza el egymástól.

expand_doc_yr

Egy olyan virtuális YR mezőt épít fel, amely a tab_expand tábla 3-as oszlopában meghatározott paraméterek szerinti kiadási évet tartalmazza:

```
YR $a [évszám]
```

A következő példában a tab_expand tábla úgy lett beállítva, hogy az évszám a MARC21-es 008-as mezőből kerüljön ki. Ha a 008-as mezőben nem szerepel kiadási évszám, akkor az új YR mező tartalma a MARC21-es 260-as mező \$\$c almezőjéből kerül ki:

```
! 1 2 3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
U39-DOC expand_doc_yr 008,260##c
```

Ha (például) 2001 a kiadási év, akkor az új mező a következőképpen épül fel:

```
YR L $$a2001
```

A program az UNIMARC 100-as mezőből is veheti az évszámot:

```
! 1 2 3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
U39-DOC expand_doc_yr 100
```

9.3 Bővítésre vonatkozó táblák

9.3.1 Konfigurációs táblák (expand_doc_type)

A könyvtár tab mappájában található tab_type_config azon tábla mintája, mely az expand_doc_type programban használható. A tábla új mező meghatározását végzi. További hasonló táblákat adhat hozzá, melyek ezután paraméterként listázhatók a tab_expand (3-as oszlopban) táblában.

A következő példában az új virtuális mező mind az LDR, mind a 008-as mező értékei szerint a 'FILM' karakterláncot fogja tartalmazni:

```
TYP Film LDR F06-01 EQUAL g
008 F33-01 EQUAL m
```

A fenti példában a TYP mező (\$\$aFilm) akkor jön létre, ha az LDR mező 06-os pozíciója 'g', míg a 008-as mező 33-as pozíciója 'm' értéket visel.

Következzék egy minta a konfigurációs táblára:

```
! 1 2 3 4 5 6 7
8
!!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!!!!!!!!!-!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!
982 Media LDR F06-01 EQUAL [g,k,r,o]
982 BK Book LDR F06-01 EQUAL a
982 Film LDR F06-01 EQUAL g
008 F33-01 EQUAL m
982 Videorecording LDR F06-01 EQUAL g
008 F08-01 EQUAL v
```

Ügyeljen arra, hogy az expand_doc_type program által használt táblák nyelvfüggőek. Ha – például – a tab_expand tábla a következőképpen van meghatározva:

```
! 1 2 3
!!!!!!!!!!!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!>
WORD expand_doc_type tab_type_config
```

akkor a rendszer a következő, nyelv szerinti sorrend alapján keres egyező tab_type_config-ot:

\$data_tab/tab_type_config.<lng>

\$data_tab/tab_type_config.<control_lng> (ahogy az az aleph_start-ban van definiálva)

\$data_tab/tab_type_config

Magyarázat a konfigurációs táblákhoz (például tab_type_config):

1-es oszlop – Cél mezőkód

A célmezőt az expand_doc_type hozza létre a 2-es és/vagy 3-as oszlop tartalmával.

2-es oszlop – Formátumkód

A formátum kódja, pl. BK (könyvekre). Az itt beírt érték az expand_doc_type által létrehozott új mező \$\$a almezőjébe kerül. Amennyiben az oszlopot üresen hagyja, akkor a formátum neve (3-as oszlopértéke) kerül az új mező \$\$a almezőjébe. Például ha a tábla a következő sort tartalmazza:

```
TYP BK Book                LDR F06-01  EQUAL  a
```

akkor az új TYP mező – amennyiben az LDR mező 06-os pozíciója 'a' értéket visel – a következő szerkezetben adódik a rekordhoz:

```
TYP  L $$aBK$$bBook
```

Amennyiben az oszlopot üresen hagyja, akkor az új mező a következőképpen jelenik meg:

```
TYP  L $$aBook
```

3-as oszlop – Formátum neve

A formátum neve, például Book/Könyv. Amennyiben a 2-es oszlopban szerepel egy formátumkód, akkor a formátumnév az új mező \$\$b almezőjébe kerül. Amennyiben nincs formátumkód meghatározva, akkor a formátumnév az új mező \$\$a almezőjébe kerül. Például ha a táblában a következő sor szerepel:

```
TYP BK Book                LDR F06-01  EQUAL          a
```

akkor az új TYP mező – amennyiben az LDR mező 06-os pozíciója 'a' értéket visel – a következő szerkezetben adódik a rekordhoz:

```
TYP  L $$aBK$$bBook
```

Amennyiben a 2-es oszlopban nem szerepel formátumkód, akkor az új mező a következőképpen jön létre:

```
TYP  L $$aBook
```

4-es oszlop – Mezőkód

A rekord azon mezője, amely meghatározza az új mezőbe bővített anyagtípust. A következő sorban az LDR (06-os pozíció 'a' értékkel) definiálja azt, hogy a rekord egy könyvre vonatkozik:

```
TYP  BK Book                LDR F06-01  EQUAL          a
```

5-ös oszlop – Almező(k) vagy meghatározott hosszúságú mezőpozíció

Ez az oszlop tartalmazza azokat az almezőkódokat, vagy meghatározott hosszúságú mezőpozíciókat (a 4-es oszlopban meghatározott mezőét), amelyet ellenőrizni kell. A következő sorban a program az LDR mező 06-os pozícióját ellenőrzi:

```
TYP  BK Book                LDR F06-01  EQUAL          a
```

A következő sorban a program a 490-es mező \$\$a almezőjét ellenőrzi:

```
TYP  Thesis                4901# a      MATCH [masters*,education*]
```

Pozícionált feltételeket is meghatározhat ebben az oszlopban, a következő formátumban:

S<almező><előfordulás>FNN-NN

Például:

```
!!!!-!!-!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!-!!!!-!!!!!!!!!!!!-!!!!!!!!!!!!-
```

```
!!!!!!!!!!!!!!
```

```
TYP  AB Book                100  Sa1F06-01  EQUAL          x
```

Ebben a példában ha a 100-as mezőben „a” az almező, és csak egyszer fordul elő, illetve a 6-os pozíció 1-es hosszúsága x értéket tartalmaz, akkor a kibővített TYP mező az AB book értéket fogja tartalmazni.

6-os oszlop – Egyezési kritérium

Az 5-ös oszlopban definiált almező vagy meghatározott hosszúságú mezők tartalmával kapcsolatos egyezési kritériumokat definiálja ez az oszlop. A következő opciók közül választhat: EQUAL, N-EQUAL, EXIST, N-EXIST, MATCH és N-MATCH.

Alkalmazása:

- EQUAL (N-EQUAL) – direkt egyezésre ellenőriz.
- EXIST (N-EXIST) – a mező tartalmának ellenőrzése nélkül ellenőrzi, hogy létezik-e a mező. Például ha egy rekordnak van egy 027-es MARC21-es mezője, akkor a rekord egy technikai jelentés (a tartalma irreleváns).
- MATCH (N-MATCH) – egy kisbetű/nagybetűre nem érzékeny egyezést ellenőriz.

7-es oszlop - Tartalmak

Ez az oszlop tartalmazza az egyeztetéshez használt mező vagy meghatározott hosszúságú mezőpozíció tartalmát (a 6-os oszlopban definiált egyezési kritériumok szerint). Használja a []-t, ha az egyeztetésben több értéket szeretne megadni (legfeljebb 15 különböző összehasonlítási értéket adhat meg). Az értékek közötti kapcsolat típusa OR / VAGY. A következő sorban az egyeztetés az LDR mező 6-os pozíciójának 'e' vagy 'f' értéke szerint történik:

TYP Map LDR F06-01 EQUAL [e, f]

8-as oszlop – Kisbetű/nagybetű között különbséget tevő egyeztetés

Az kisbetűs/nagybetűs egyeztetés változója. Az értékek:

Y=az egyeztetés függ a kisbetűtől/nagybetűtől

N=nem tesz különbséget a kis- és nagybetű között

Ügyeljen arra, hogy ha ezt az oszlopot üresen hagyja, akkor az alapéretelmezett egyeztetés nem tesz különbséget a kis- és nagybetű között.

9.3.2 tab_expand_split

Az expand_doc_split program specifikációit határozza meg, amely program egy mezőt darabol fel, a tab_expand_split tábla 2-es oszlopában meghatározott almezők szerint. A feldarabolás egy almező összes előfordulása szerint történik, így egy mezőt többször hoz létre a program.

A „feldarabolás” az összes almezőtartalmat magába foglalja, illetve az almezők – a következő almezőelőfordulásig, vagy az almező végéig tartó – tartalmát is. Például ha a \$\$a almezőt határozta meg a táblában, akkor a \$\$a \$\$b \$\$c \$\$a \$\$a \$\$c a következőképpen alakul:

```
$a $b $c
$a
$a $c
```

Következzék egy példa a tab_expand_split táblára:

```
! 1 2 3 4
!!!!-!-----!!!!-!!!!
260## b PLA PUB
700## t 100 240
```

A tab_expand_split tábla magyarázata:

1-es oszlop – Mezőkód és indikátorok

A bővítőprogram által kezelt dokumentummező kódja és indikátorai. A 4. és 5. pozícióban az # jellel a mezőkód további számjegyeinek csonkolását jelezheti (például a 245## jelenthet 2451, 2452, 24501-et stb.)

2-es oszlop

Azt az almezőkódot tartalmazza, amely szerint a feldarabolás történik.

3-as oszlop – „-ig” mezőkód

Az almezőtördelés szövegére tartozó kimeneti mezőkódját és indikátorát jelöli.

4-es oszlop – „Után” mezőkód

Az almezőtördelés utáni szöveghez tartozó kimeneti mezőkódot és indikátorokat jelöli.

A mintatábla alapján a következő mező:

```
260 $$aBoston: $$bLittle, Brown, and company,$$c1933.
```

az alábbiakra lesz feldarabolva:

```
PLA $$aBoston:  
PUB $$bLittle, Brown, and company,$$c1933.
```

9.3.3 tab_abbrev

A tab_abbrev tábla az expand_doc_fix_abbreviation programmal együtt működik.

Az expand_doc_fix_abbreviation program egy duplikált virtuális mezőt ad a dokumentumhoz ott, ahol a szó, vagy kifejezés rövidített formája (rövidítés) a teljes szóra, vagy kifejezésre lett kicserélve. Gyakorlatilag egy bibliográfiai rekordban bármilyen szöveges karakterláncot ki lehet cserélni a programmal egy másikra. A tab_abbrev tábla a „rövidítések” listáját, illetve az expand_doc_fix_abbreviation program által hozzáadott – és a rövidített alakok lecseréléséhez szükséges - új virtuális mezők teljes alakjait tartalmazza. A szövegcsere mezőkódonként (indikátorral) lehet definiálni.

Ügyeljen arra, hogy ez a program fix programként is alkalmazható (a tab_expand helyett a tab_fix táblát használva). A különbség annyi, hogy fix programként alkalmazva a virtuális mezők hozzáadása helyett (indexeléshez és/vagy megjelenítéshez) a teljes alakot tartalmazó mezők közvetlenül a rekordhoz adódnak.

Következzék egy minta a tab_abbrev táblára:

```
! 1      2      3      4  
!!!!!!-!-!!!!!!-  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
225## Y 1st          FIRST  
043## Y u-at---     Australia  
043## Y u-at-ne     Australia New South Wales
```

A tábla magyarázata:

1-es oszlop – Mezőkód és indikátorok

A program által kezelt mezők kódját és indikátorait tartalmazza. A mezőkódokat # jellel csoportosíthatja/helyettesítheti. (például a 2###-vel a 2-essel kezdődő mezőkódokat lehet meghatározni).

2-es oszlop – Alkalmazási kód

Válassza ki a megfelelő alkalmazási kódot. A következő lehetőségek közül választhat:

- X – a rövidítés fixekből kihagyandó mezőkódokat határozza meg. Ennek megfelelően az X sorában nem szerepelhet szöveg a 3-as és 4-es oszlopban.
- N – a szövegcsere által figyelmen kívül hagyandó mezőkódot vagy mezőkódcsoporthoz definiálja. Egy N sor akkor szükséges, ha ezt egy # karaktert tartalmazó Y sor követi. Például ha a tábla következő sora az összes 2-essel kezdődő mezőkódú mező esetében a Ft. -ot tartalmazza a FORT-ra:

```
2#### Y Ft. FORT
```

és ha a tábla a következő sorokat tartalmazza:

```
245## N Ft. FORT
```

2#### Y Ft. FORT

akkor a program a 245-ös mező kivételével az összes 2-essel kezdődő mezőben lecseréli az Ft.-t FORT-ra.

- Y – a szövegcsere lezajlását definiálja

3-as oszlop - Rövidítés

A módosítandó szöveget tartalmazza.

4-es oszlop – Bővített alak

A 3-as oszlopban meghatározott szöveg módosított alakját tartalmazza.

A minta tábla alapján a következő dokumentum:

```
043 L $$au-at-ne
1001 L $$aWilliams, David
2451 L $$aThe 1st man
```

az alábbiakra módosul:

```
043 L $$au-at-ne
1001 L $$aWilliams, David
2451 L $$aThe 1st man
043 L $$aAustralia New South Wales
2451 L $$aThe FIRST man
```

9.3.4 tab_expand_duplicate_field

Az expand_doc_duplicate_field program specifikációit definiálja ez a tábla. A program egy új mezőt duplikál egy új mezőkód + indikátorok alatt.

Következzék egy minta a tab_expand_duplicate_field-re:

```
! 1      2
!!!!-!!!!
260&335;# IMP
```

Az expand_doc_duplicate_field eljárás és az expand_doc_duplicate tábla az expand_doc_split rutin alkalmazásakor fellépő problémát (azaz nem marad meg a forrásmező) kezeli. Amikor az expand_doc_duplicate programmal létrejött mezőn alapul az expand_doc_split rutin, akkor a forrásmező megmarad. A fenti példában az IMP mezőn a későbbiekben is futtathatja az expand_doc_split-et úgy, hogy az eredeti 260-as mező közben megmarad.

Magyarázat a tab_expand_duplicate_field táblához:

1-es oszlop – Input mezőkód és indikátorok

A bővítő programmal duplikálni kívánt dokumentummező kódját és indikátorait tartalmazza.

2-es oszlop – Kimeneti mezőkód és indikátorok

A duplikálás utáni mezőkódot és indikátorokat tartalmazza.

A mintatábla alapján a következő mező

```
2600 L $$aBoston, :$$bRoberts Bros
```

az alábbiak szerint duplikálódik:

```
IMP L $$aBoston, :$$bRoberts Bros
```

9.3.5 tab_expand_external

Ez a tábla az expand_doc_split_external programmal együtt alkalmazható.

Az expand_doc_split_external program egy külső lelőhelyet tartalmazó almezőkkel rendelkező mező (856, 505 stb.) tartalmát darabolja fel a megjelölt almező összes előfordulása

szerint elkülönülő mezőkre.

A `tab_expand_external` táblával definiálja a program által vizsgált mezőket (külső lelőhelyet tartalmazó almezők keresése céljából). A következőképpen jelenik ez meg:

```
! 1 2
!!!!-!-
856## u
505## u
```

Magyarázat a táblához:

1-es oszlop – Input mezőkód és indikátorok

A bővítő program által vizsgált mezőkódokat és indikátorokat definiálja.

2-es oszlop – Kimeneti mezőkód és indikátorok

Azt az almezőkódot tartalmazza, amely egy mezőben többször is előfordul.

A mintatábla szerint a következő mező esetében:

```
5054 L $$:a505 FORMATTED CONTENTS
NOTE$$:uhttp://lcweb.loc.gov/marc/bibliographic/ecbdnot2.html#mrcb555$$:uhttp://www
.loc.gov/standards/mets
```

a bővített mezők a következőképpen jönnek létre:

```
5054 L $$:a505 FORMATTED CONTENTS
NOTE$$:uhttp://lcweb.loc.gov/marc/bibliographic/ecbdnot2.html#mrcb555
```

```
5054 L $$:a505 FORMATTED CONTENTS NOTE$$:uhttp://www.loc.gov/standards/mets
```

9.3.6 expand_doc_bib_z30

Az `expand_doc_bib_z30` program által bővített példányrekordból származó adatokat definiálja ez a tábla, a következők szerint: definiálja példányrekordból vett mezőket; azokat a eseteket, amelyekben ezeket a mezőket venni kell (folyóiratpéldányokra, másodpéldányokra stb); a bővített mezőkben mely almezők adatai tárolódnak; illetve néhány konkrét példánymező esetén azt, hogy a kódokat nevekre cserélje-e, vagy sem (például az alkönyvtárkód lecserélhető az alkönyvtár nevére).

9.3.7 expand_doc_bib_z403

A tábla az `expand_doc_bib_z403` program által bővítendő objektumadatokról vett adatokat definiálja, a következők szerint: az objektum adatrekord (z403) mely mezőit vegye a program; a bővített mezők mely almezőadatai tárolódnak; és azt, hogy objektum adatrekordjának konkrét mezőiben cserélődjenek-e a kódok nevekre (például az alkönyvtárkód lecserélhető az alkönyvtár nevére).

Ügyeljen arra, hogy a tábla 2-es oszlopa többnyire a z403 rekord mezőjét tartalmazza (például `z403-sub-library`) egy érvényes 856 URL mező létrehozásához: emiatt az oszlopban szereplő bővített mezőt és `url-t` a `tab_expand` táblában 856-nak kell beállítani. Ezt az opciót olyan különleges esetekre alkalmazhatja, mint például az X és Z39.50 szerverek.

9.4 Bővítőmezők (virtuális mezők) indexelése

Indexeléshez a következő lépéseket kövesse:

1. Állítsa be a megfelelő bővítőtáblát. Például:

`tab_expand_extract`

```
! 1 2 3
!!!!-!-!!!!
650## a SUBJ
```

2. Adja a a bővítőmezőt a tab11-hez.

Hívja meg a programot a tab_expand tábla ACC/WORD/INDEX bejegyzésében:

Kulcsszóindexekhez a programot a tab_expand WORD bejegyzéséhez kell adni

Besorolási elemekhez a programot az ACC bejegyzéshez adja

3. Megjelenítéshez a WEB-FULL, WEB-FULL-1, WEB-BRIEF, GUI-DOC, GUI-BRIEF bejegyzésekhez kell adni.

Például:

tab_expand

```
WORD          expand_doc_fmt_mgu
WORD          expand_doc_join
!
ACC           expand_doc_bib_loc_usm
ACC           expand_doc_bib_ndu
ACC           expand_doc_join
!
INDEX expand_doc_extract
INDEX expand_doc_bib_loc_usm
SORT-DOC      expand_doc_bib_loc_usm
!
WEB-BRIEF     expand_doc_join_simple
!
```

9.4.1 tab_expand_extract

Ez a tábla az almezők indexelésre történő kivonatolását definiálja. Például ha a „tárgyszavak további bejegyzései – földrajzi alakok” kronológiai alosztályának besorolási elem-listáját szeretné létrehozni, akkor definiálhatja azt, hogy a MARC21 650-es mező \$\$y almezője egy új mezőkódként jelenjen meg (pl y650). A virtuális mezőt ezután indexelheti, vagy megjelenítheti.

A tab_expand_extract tábla 4-es oszlopában az újonnan létrejövő virtuális mezőhöz tartozó almezők előfordulási számát is meghatározhatja. Például a 650-es mező \$\$y almezőjének kizárólag az első megjelenését alkalmazhatja az új mező létrehozásához. Következzék a táblára egy példa:

```
! 1 2 3 4
!!!!-!-!!!!-!
650#:#y CHRON 1
```

A tábla magyarázata:

1-es oszlop – Input mezőkód és indikátorok

A program által bővítendő almezőt tartalmazó mező kódja és indikátorai.

2-es oszlop

Egy új mezőkódba bővítendő és a program által veendő almezők.

3-as oszlop – Kimeneti mezőkód és indikátorok

Az új mezőkódot tartalmazza az indexeléshez.

A fenti táblaminta szerint a következő \$\$y almező:

```
650-0 L $$y19th century$$aProtestants$$zFrance.
```

a CHRON virtuális mezőt bővíti a következőképpen:

```
CHRON L $$y19th century
```

9.4.2 tab_expand_join

Ez a tábla kettő, vagy több MARC mezőből hoz létre egy virtuális mezőt. A tab_expand_join tábla determinálja az egyesítendő mezőket és almezőit, a sorrendjüket, illetve a létrejövő mező nevét is.

Például ha a címadatok és szerzőségi adatok besorolási elemeit szeretné létrehozni, akkor a 100-as és 245-ös mező egy új mezőkód alatti (pl. AUTIT) egyesítését definiálhatja, mely után az új virtuális mezőkódot elküldheti egy szerző/címadat listába. A virtuális mező indexelési célokat szolgál.

Következzék egy minta a tab_expand_join táblából:

```
! 1      2      3      4      5      6
!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!-!!!!!!
AUTIT 100## a          245## abcd
```

Magyarázat a tab_expand_join táblához:

1-es oszlop – az új virtuális mező kódja és indikátorai

2-es oszlop – az új virtuális mező építéséhez az első mezőkód

3-as oszlop – az egyeztetéshez vett almezők

Amennyiben ezt az oszlopot üresen hagyja, akkor az összes almezőt veszi a rendszer.

Amennyiben konkrét almezőket szeretne választani, akkor írja be az almezőkódokat

4-es oszlop – új almező kódja

5-ös oszlop – második mezőkód az új virtuális mezőkód építéséhez

6-os oszlop – az egyeztetés során vett almezők

Amennyiben ezt az oszlopot üresen hagyja, akkor az összes almezőt veszi a rendszer.

Amennyiben konkrét almezőket szeretne választani, akkor írja be az almezőkódokat

Megjegyzés

Egy mező több előfordulása esetén az egyesítés párok szerint történik. Például ha a tab_expand_join-ban az 595 és 596 mezők 599-es mező „i” almezőjében való egyesítését definiálja, és a rekord „959 a1”, „595 a2”, „596 b1”, „596 b2”, „596 b3” mezőket tartalmaz, akkor a rendszer „599a1 b1” és „599 a2 b2” mezőket hoz létre. Az „596 b3” mezőt figyelmen kívül hagyja, mivel nincs 595 párja.

A fenti minta alapján a következő mezők:

```
10010 L $$aOrcibal, Jean.24510 L $$aLouis XIV et les protestants, :$$b"La cabale des
accommodeurs de reli
```

egy AUTIT mezőkód alatt lesznek összefűzve az alábbi módon:

```
AUTIT L $$aOrcibal, Jean.$$aLouis XIV et les protestants, $$b"La cabale des
accommodeurs de religion", la caisse des conversions, la revocation de l'dit de
Nantes.
```

9.4.3 tab_expand_join_simple

Ez a tábla egy mező egy, vagy minden előfordulása esetén hoz létre egy virtuális mezőt, amelyben egy másik mező egy, vagy összes előfordulásával fűzi össze. A

tab_expand_join_simple tábla determinálja az egyesítendő mezőket és almezőit, a sorrendjüket, illetve a létrejövő mező nevét is. A virtuális mező megjelenítési célokat szolgál.

Következzék egy minta a tab_expand_join táblából:

```
! 1      2      3      4      5      6      7      8      9
!!!!-!!!!-!!!-!!!!-!!!!-!!!!-!!!-!!!!-!!!!
ATS02 100## AA  a      a      245## AA  y
```

Magyarázat a tab_expand_join_simple táblához

1-es oszlop – az új virtuális mező kódja és indikátorai

2-es oszlop – az első mezőkód az új virtuális mezőkód építéséhez

3-as oszlop – az első mezőkód előfordulás; nn indexhez, AA az összeshez

4-es oszlop – az egyeztetés során vett almező

5-ös oszlop – az új almező kódja

6-os oszlop – az új virtuális almező építéséhez vett második mezőkód

7-es oszlop – Második mezőkód előfordulás; nn indexhez, AA az összeshez

8-as oszlop – az egyeztetés során vett almező

9-es oszlop – új almezőkód

A fenti minta alapján a következő mezők:

```
10010 L $$aOrcibal, Jean.
```

```
24510 L $$aLouis XIV et les protestants, :$$b"La cabale des accommodeurs de religion", la caisse des conversions, la revocation de l'dit de Nantes.
```

egy ATS02 mezőkód alatt az alábbi formában lesznek összefűzve:

```
ATS02 L $$aOrcibal, Jean.$$aLouis XIV et les protestants, :$$b"La cabale des accommodeurs de religion", la caisse des conversions, la revocation de l'dit de Nantes.
```

10 További indexek

Ez a fejezet a következő témaköröket tárgyalja:

- Rövid bibliográfiai rekord (manage-07)
- Rendezési index (manage-27)
- Index aktualizálás – kiválasztott rekordokra (manage-40)

10.1 Rövid bibliográfiai rekord (manage-07)

A rövid bibliográfiai rekord a teljes rekord egy standard Oracle táblaformátumban szereplő rövidített változata. Legfeljebb hat (rendszer-definiált) meghatározott hosszúságú mezőt (évszám, raktári jelzet/raktári jelzet kulcs, szerző, cím, megjelenési adatok és ISSN/ISBN) tartalmazhat mezőnként legfeljebb 100 karakterrel (kivétel ez alól a raktári jelzet kulcs mező, amely legfeljebb 80 karakter hosszú lehet). Ezen felül 15 felhasználó által definiált, egyenként legfeljebb 500 karakterből álló mezőt tartalmazhat.

A rövid bibliográfiai rekord célja bibliográfiai adatok gyors és hatékony megadása,

különösen azokban az esetekben, ahol a bibliográfiai adatokat az adminisztrációs adatok mellett kell megjeleníteni.

A rövid bibliográfiai rekordot a `tab22` tábla definíciói szerint a rendszer építi fel.

Mikor módosítsa a rövid bibliográfiai rekordokat?

Futtassa ezt a szervizt minden egyes alkalommal, amikor a `tab22` táblában olyan módosításokat hajt végre, amelyek kihatnak a rövid bibliográfiai rekordokra.

A rövid bibliográfiai rekordokat frissíteni kell, ha nagy mennyiségű rekordot tölt fel az adatbázisba, „részleges” módban, mivel a rendszer nem épít fel automatikusan rövid bibliográfiai rekordokat ezekre a dokumentumokra.

10.2 Rendezési index (manage-27)

Ez a szerviz módosítja az adatbázis rendezési indexét.

A `z101` tábla a rendezési kulcsokat tartalmazza. Amikor a rövid rekordok listája a WebOPAC-ban megjelenik, akkor a `z101` rekordokkal lesz a lista rendezve. A rendezési kulcsok felépítéséhez alkalmazott mezők a könyvtár `tab` mappájában található `tab/tab_sort` táblában vannak meghatározva.

10.3 Index aktualizálás – kiválasztott rekordokra (manage-40)

Ez a szerviz írja a kívánt dokumentumszámokat a `z07` Oracle táblába, amely után a rekordok a könyvtár szokásos frissítő eljárásán (`ue_01`) keresztül újraindexelődnek.

Mikor futtassa ezt a szervizt?

Akkor, ha a válsztott rekordokat újra szeretné indexelni.

11 Felkészülés az index feladatokra

A `p_manage_01`, `p_manage_02`, `p_manage_07` (a `z00r -re`) és a `p_manage_32` nagy tárhelyet követel akkor, ha nagy adatbázison futtatja őket, illetve ez az első eset, hogy egy meghatározott könyvtárban teszi ezt, így a futtatás hosszú időt vehet igénybe.

11.1 Temp/scratch (ideiglenes) könyvtárak tisztítása

Győződjön meg arról, hogy az `xxx01 $data_scratch`, az `xxx01 $data_files` és a `$TMPDIR` könyvtárakat megtisztította az összes oda nem illő ideiglenes fájljuktól. Amennyiben az ezt megelőzően elvégzett tárhely kalkulálás során a háttértárakon, amelyekben a `$data_scratch`, `$data_files` vagy a `$TMPDIR` szerepel, nincs elég hely, akkor helyezze át ezeket a könyvtárakat ideiglenesen egy másik helyre.

11.2 Oracle tárhely ellenőrzése

Az `UTIL A/17/11` (Oracle táblák tárhelyellenőrző segédprogramja) győződik meg arról, hogy a vonatkozó fájlok (`Z97/Z98` a szó-, `Z01/Z02` a besorolási- míg a `Z11` a direct indexekhez) a maximális számú terjedelmük (505, ahogy azt az Ex Libris szolgáltatja) vagy a maximális táblatárhely méretük közelében vannak. Amennyiben így van, akkor az `xxx01/file_list`-ben meghatározott terjedelmet vagy a táblatárhely(ek) méretet növelni kell a feladat futtatását megelőzően.

11.3 Zavart okozó feladatok megszakítása

Biztonsági mentő, vagy egyéb normál esetben futtatott feladatok futtatásakor a `manage_nn` index feladatot is futtatni kel. Biztosítania kell azt, hogy az indexelő feladattal ütköző feladatokat ne fussanak. Az automatikus futtatás mechanizmusai az ALEPH feladtdémon és a Unix cron. Az ALEPH feladtdémont az `UTIL E/15` segédprogrammal szakítható meg, illetve egyenként kikommentelhet feladatokat az `UTIL E/16` segédprogramon keresztül. Az ütköző cron feladatokról tájékozódjon a Unix rendszeradminisztrátornál.

12 Párhuzamos indexelés

Párhuzamos indexelést az online ALEPH rendszerrel párhuzamos OPAC index újraépítésekor alkalmazhat, indexlétrehozás közbeni időkiesés nélkül. Ez az eljárás az indexelő paraméterek módosítását, illetve az aktuális indexek elvesztése nélkül az eredmények ellenőrzését is lehetővé teszi.

Az indexelés egy külön könyvtárban zajlik le. Ez a könyvtár az aktuális könyvtárban szereplő dokumentumokra mutató linkkel van beállítva, míg az indexek az indexelő könyvtárban találhatóak. Az indexelés végeztével az aktuális könyvtárban egy, az indexelő könyvtárban lévő indexre mutató link is létrejön.

A következő szekció példáiban az `USM01` az aktuális könyvtár, míg az `USM21` az indexelő könyvtár.

Az `USM21` könyvtárban történő indexelési eljárás megkezdéséhez azt feltételezzük, hogy a fájllistában az összes index tábla (például, a szótáblákhoz ez `Z97`, `Z98` stb. lesz) lokálisként definiált, míg a dokumentumok fájl az `USM01` logikai szinonímájaként szerepel. Ezek után futtathatja az `USM21`-ben az indexelő feladatot (például a `p_manage_01`-et). A szerviz a logikai szinonímán keresztül beolvassa a rekordokat az `USM01`-ből, azonban az indextáblákat helyben/lokálisan hozza létre. Az indexelő könyvtár indexbeállításait módosíthatja is, így külön index kódokat hozhat létre, vagy külön iktató eljárásokat alkalmazhat.

Miután sikeresen felépítette az indexet, ellenőrizze azt az `USM21`-ben a WebOPAC-on keresztül.

Végül, amennyiben új indexet szeretne létrehozni, alkosson egy logikai szinonímát az `USM01`-ből az `USM21`-be az összes vonatkozó táblára. Így semmiféleképpen nem lesz kiesési idő.

A szó (`W-nnn`) és direkt (`IND`) index újraindexeléséhez létezik egy - a többi indexeléshez nem szükséges - önálló eljárás. A besorolási (`ACC`) index újraindexelése azonban számos

más indexelő feladat futtatását igényli.

A következő lépésekben aktiválhatja a párhuzamos indexelést:

1. Nyisson meg egy könyvtárat

Nyisson egy új BIB könyvtárat az indexelendő könyvtárral párhuzamosan.

2. Adja az indexelő könyvtárat a library_relation-hoz

Az /alephe/tab könyvtárban szereplő library_relation tábla definiálja a különböző könyvtárak szerinti kapcsolatot. Párhuzamos indexeléshez a PID kapcsolatot kell definiálni a következő példa szerint:

```
PID USM21 USM01
```

Ezen felül az indexelő könyvtár, az ADM és a HOL könyvtárak közötti kapcsolatot is definiálni kell pontosan úgy, ahogyan az aktuális könyvtár a rá vonatkozó ADM és HOL könyvtárak közötti kapcsolatot is.

```
ADM USM01 USM50 USM51  
HOL USM01 USM60
```

```
ADM USM21 USM50 USM51  
HOL USM21 USM60
```

```
PID USM21 USM01
```

3. A könyvtár fájllistájának igazítása

Az egyes ALEPH könyvtárak gyökérkönyvtárában található egy file_list elnevezésű konfigurációs tábla. Ez a tábla listázza az összes könyvtári Oracle táblát, a méretüket, terjedelmüket és a leőhelyüket. Ebben a konfigurációs táblában definiálhatja azt, hogy egy könyvtár egy másik könyvtár Oracle tábláját használja a saját Oracle tábla helyett. Ezt a másik könyvtárra mutató link beállításával teheti meg, egy logical symbol (logikai szimbólum – LS) definíció alkalmazásával.

Alapértelmezésben az új könyvtár gyökérkönyvtárában az aktuális könyvtár file_list fájljának egy másolata kell, hogy szerepeljen. Ebben a stádiumban a file_list az aktuális könyvtár file_list-jében felsorolt Oracle táblákkal és azon definícióival megegyezik. A későbbiekben – igény szerint - a definíciók módosulnak mind az aktuális, mind az indexelő könyvtárban.

Az indexelő könyvtárban a z52 szekvencia számok tábláknak mindig lokálisnak kell lenniük:

```
TAB z52                10K                10K                ts0  
IND z52_id            10K                10K                ts1
```

Az indexelő könyvtárban a bibliográfiai dokumentumok táblára mindig logikai szimbólumot alkalmazzon, mind a z00, mind a z103 tábla esetében:

```
LS z00                usm01  
LS z103               usm01
```

Ezen felül a z00 és a z103-ra vonatkozó TAB és IND sorokat ki kell kommentelni (egy „!” jelet téve eléjük).

A párhuzamos indexelés elindítása előtt a fenti táblákat ki kell hagyni az indexelő könyvtárban (a párhuzamos könyvtár).

Az SQL – táblákat ejtő – parancsok kiadása előtt (ahogy azt majd lentebb látja) a következő leválogatást hajtsa végre:

```
>>s+ usm21  
SQL-USM21> select count(*) from Z00 ;  
SQL-USM21> select count(*) from Z103 ;
```

Mindkét leválogatás eredménye 0 kell, hogy legyen. Amennyiben nem így van, akkor

vagy:

- (1) nem a párhuzamos könyvtárban van
- (2) a párhuzamos könyvtárból az élesre irányítva

Lépjen ki azonnal

Az első esetben tegye az „s+”-t a helyes, párhuzamos könyvtárhoz.

A második esetben futtassa az A/17/5/1 segédprogramot annak ellenőrzésére, hogy a logikai szimbólumok léteznek-e már. Amennyiben igen, akkor ne tegyen mást. A tény, hogy a szinoníma létezik, azt jelzi, hogy a tábla ejtve volt. (Az Oracle nem engedélyezi egy párhuzamos könyvtárban már meglévő tábla esetében szinoníma létrehozását).

Amennyiben a leválogatás eredménye 0, akkor folytassa a táblaejtés SQL parancsával:

```
SQL-USM21>drop table Z00;  
SQL-USM21>drop table Z103;
```

A `file_list` módosítása és a táblák dobása után hozza létre a párhuzamos könyvtárban a logikai szinonímákat (UTIL A/17/5/2).

Megjegyzés

Szóindex felépítésekor ha szinonímákat (Z970) alkalmaz, akkor a Z00 és Z103-ra a fent említettek érvényesek a Z970-re is

Ne felejtse el ellenőrizni, hogy van elég tárhely a párhuzamos könyvtár táblatárhelyében történő indexek építéséhez (O/14/1 segédprogram). Az A/17/11/2 segédprogrammal nézheti meg az aktuális könyvtár (USM01) indextábláinak méretét. (Ügyeljen arra, hogy a második eredménye a KB-ben, míg a az elsőé az MB-ben van). Az új táblák a párhuzamos könyvtár táblatárhelyeivel egyenértékű helyet igényelnek (hacsak nem módosítja az indexek vagy az indexelt mezők számát).

4. A könyvtár Z52 táblájának igazítása

A párhuzamos indexelés elkezdése előtt győződjön meg arról, hogy a következő számlálókat úgy definiálta, ahogy azok itt fel vannak sorolva. Az UTIL/G/2 segédprogrammal adhat hozzá hiányzó számlókat, illetve módosíthatja a számlálók értékeit.

A szükséges értékek a következők:

```
last-doc-number (az aktuális könyvtár last-doc-number értékével egyezzen)  
last acc-number (0 értéket adjon)  
last-long-acc-number (0 értéket adjon)  
last-similar-acc-num (0 értéket adjon)  
last-word-number (0 értéket adjon)
```

5. Indexelő konfigurációs táblák beállításai

Definiálja az indexelő könyvtár `tab` mappájában található indexelő táblák beállításait. Választhat: vagy az indexelő könyvtár `tab` mappájába bemásolja az aktuális könyvtár indexelésre alkalmazott tábláit, vagy a `path_convert` konfigurációs táblával átirányíthatja a rendszert az aktuális könyvtár konfigurációs tábláihoz. Ez az opció kizárólag akkor alkalmazható, ha nem akarja az indexelő beállítást módosítani.

Az indexelés elősegítéséhez módosíthatja a `$data_tab` értékeket az indexelőkönyvtárban. Amennyiben ezt teszi, győződjön meg arról, hogy ezeket a módosított táblákat a 9. lépésben átmásolja az aktuális könyvtárba.

6. Átmeneti indexelő módosítások mentése

Mialatt az indexelő feladatok futnak, az aktuális könyvtárban új és módosított rekordok indexelődnek az ue01 és z07 mechanizmusokon keresztül és a régi indexeken. Ezeket a z07 rekordokat kell menteni az új indexek felépítése utáni újraindexeléshez. Ahhoz, hogy ezeket a rekordokat le tudja menteni, az indexelő eljárást megelőzően aktiválja az E/5/1 segédprogramot az aktuális könyvtárban. Az UTIL E/5 segítségével az ue_01 által kezelt z07 bejegyzések előzménytábláját hozhatja létre. Ezek a tárolt bejegyzések később felhasználhatók az ugyanazon rekordon az ue_01 segéprogram ismételt végrehajtásához.

7. Az indexelő feladatok futtatása

Szóindexek

Az adatbázis szóindexének újraépítéséhez futtassa a p_manage_01-es szervizt.

Direkt indexek

Az adatbázis direkt indexének újraépítéséhez futtassa a p_manage_05-es szervizt.

Besorolási indexek

Opcionális:

- Az authority adatbázis alapján történő bibliográfiai besorolási index előbővítéséhez futtassa a p_manage_102 szervizt.
- Futtassa a p_manage_02-t.
- Indítsa az UE-08-at a bibliográfiai besorolási elemek – authority rekord kapcsolat kiépítéséhez.

Megjegyzés

A p_manage_102 futtatásához ebben a stádiumban nem szükséges az UE-08-at elindítani.

Futtassa az AUT könyvtárakban a p_manage_105 a kapcsolat nélküli hivatkozások hozzáadásához.

Futtassa a p_manage_17-et a teljes hosszon rendezéshez.

Futtassa a p_manage_35-öt a rövid rekordok (z0101) létrehozásához.

Futtassa a p_manage_32-öt a logikai bázisokra vonatkozó számlálók építéséhez.

Megjegyzés: a 17-es verzió óta a p_manage_32 a párhuzamos könyvtárban is futtatható.

Megjegyzés

Besorolási (böngésző) indexek újraépítésekor a fent felsorolt további indexelő eljárásokat is futtatnia kell.

Ha az AUT adatbázisa nem tartalmaz kapcsolat nélküli besorolási elemeket, akkor nem kell a p_manage_105-öt futtatni.

Amennyiben nincs logikai bázisa, vagy a tab_base.lng tábla 8-as oszlopában a bázisok bármelyikére nem adta meg az y értéket, akkor nem szükséges a p_manage_32 futtatása.

Amennyiben nem alkalmaz rövid rekordokat, akkor nem kell a p_manage_35-öt futtatnia.

Egyéb futtatandó indexelő feladat

Futtassa a p_manage_07-et a rövid bibliográfiai rekordok módosításához.

Futtassa a p_manage_27-et a rövid indexek módosításához.

8. Az új index ellenőrzése

Az indexelő könyvtárat adja az /alephe/tab/tab_base.lng fájlhoz majd adja a könyvtárat a bázis listához, a WebOPAC számára (/alephe/www_f_lg/base_list). A WebOPAC-hoz való hozzáféréshez válassza ki az indexelő könyvtárat, majd ellenőrizze az új indexet. Amennyiben minden megfelelőnek tűnik, akkor folytassa a 9-es lépéssel.

Megjegyzés

Ezen teszt végrehajtásakor a lelőhely nem jelenik meg az OPAC rövid, vagy teljes megjelenítésben. (Mivel nincs a párhuzamos könyvtárra vonatkozó ADM, vagy HOL). Amennyiben ez a párhuzamos indexelés-futtatás csak a kulcsszavakra vonatkozik, akkor a böngészés és a teljes megjelenítés böngésző linkjei nem működnek a teszt végrehajtásakor. Amennyiben a párhuzamos indexelés ezen futtatása csak a böngészésre vonatkozik, akkor a kulcsszó nem működik a teszt végrehajtásakor.

Amennyiben a könyvtár egyesített katalógust, vagy egyesített nézetet alkalmaz, akkor ügyeljen arra, hogy ezek nem működnek a teszt végrehajtásakor, hacsak nem adja hozzá az LS értékeket a z120 és a z127-re vonatkozóan.

9. Új indexek alkalmazása

Miután végzett az összes indexeléssel, állítsa meg az összes futó démont az aktuális könyvtárban (UTIL E).

1-es opció – logikai szinonímák

Ez az opció a logikai szinonímákat alkalmazza az aktuális könyvtárból az indexelő Oracle táblákra mutató linkhez az új indexek alkalmazására.

A következő lépés az aktuális (rég) indexből az új indexbe történő váltáshoz szükséges. Hozzon létre egy aktuális könyvtár Oracle táblából az indexelő könyvtárra mutató linket úgy, hogy az aktuális könyvtár file_list-ben szereplő definíciót egy logikai szinonímára módosítja.

A következő példa megmutatja a besorolási elemek újraindexelése utáni új beállításokat:

Cserélje le:

```
TAB z01      2M      1M      ts0
IND z01_id   1M      1M      ts1
IND z01_id2 300K    100K    ts1
IND z01_id3 200K    100K    ts1
IND z01_id4 200K    100K    ts1
IND z01_id5 200K    100K    ts1
IND z01_id6 200K    100K    ts1
```

A következőre:

```
LS z01      USM21
```

Cserélje le

```
TAB z02      400K    100K    ts0
IND z02_id   300K    100K    ts1
IND z02_id1 400K    100K    ts1
```

A következőre:

```
LS z02 USM21
```

A következő példa mutatja be a szavak újraindexelése utáni új beállításokat:

Cserélje le:

```
TAB z95      1M      1M      ts0
IND z95_id   1M      1M      ts1
```



```
TAB z97      2M      1M      ts0
IND z97_id   1M      1M      ts1
IND z97_id1  1M      1M      ts1
IND z97_id2  1M      1M      ts1
IND z97_id3  1M      1M      ts1
TAB z98      3M      1M      ts0
IND z98_id   2M      1M      ts1
TAB z980     1M      1M      ts0
IND z980_id  1M      1M      ts1
```

A következőre:

```
LS z95      USM21
LS z97      USM21
LS z98      USM21
LS z980     USM21
```

A következő leválogatást hajtsa végre mielőtt az SQL táblaeldobást a fentiekre végrehajtaná (az SQL parancsok továbbiakban lesznek bemutatva):

```
>>s+ usm21
SQL-USM21> select count(*) from Z01 ;
SQL-USM21> select count(*) from Z02 ;
```

Ellenőrizze, hogy ezek a táblák nem üresek.

Dobja az aktuális könyvtárban a releváns z táblákat (fenn említve) az SQL parancs alkalmazásával a következő példa szerint:

```
>>s+ usm01
SQL-USM01>drop table Z01;
SQL-USM01>drop table Z02;
```

stb.

Ezután hozza létre az indexelő könyvtárra a logikai szinonímákat az aktuális könyvtárban alkalmazott UTIL A/17/5 segédprogram alkalmazásával.

Megjegyzés

Ha ezt az opciót használja, akkor a következő újraindexelésnél létre kell hoznia egy további indexelő könyvtárat (azért, hogy mind a BIB dokumentumok könyvtár, mind az indexelő könyvtár hozzáférhető maradjon).

2-es opció - Oracle import

Azok a telepítések, melyekben van Oracle DBA expertise választhatják az új indexek (azaz az Oracle táblák) bemásolását az indexelő könyvtárból az aktuális táblába.

A következő műveleteket kell végrehajtania függetlenül attól, hogy az 1-es vagy a 2-es opciót választotta:

- Ha az új index a p_manage_02 eljárás eredményeképpen jött létre, akkor módosítsa a last-acc-number és a last-similar-acc-number számlálókat az aktuális könyvtárban (az UTIL G/2 segédprogrammal) ugyanarra az értékre, amely az indexelő könyvtár számlálójára vonatkozik.
- Ha a „Besorolási index módosítása” eljárásban a p_manage_02 szerviz lefutott, és a Duplikálási mód: Igen (Duplicate mode: yes), akkor módosítsa a last-long-acc-number számlálót az aktuális könyvtárban (az UTIL G/2 segédprogrammal) ugyanarra az értékre, amely az indexelő könyvtár számlálójára vonatkozik.
- Ha az új index a p_manage_01 eljárás eredményeképpen jött létre, akkor módosítsa a last-word-number számlálót az aktuális könyvtárban (az UTIL G/2 segédprogrammal) ugyanarra az értékre, amely az indexelő könyvtár számlálójára vonatkozik.

- Ha az új index a `p_manage_35` eljárás eredményeképpen jött létre, akkor módosítsa a `last-z0101-sequence` számlálót az aktuális könyvtárban (az `UTIL G/2` segédprogrammal) ugyanarra az értékre, amely az indexelő könyvtár számlálójára vonatkozik.
- Ha az új index a `p_manage_17` eljárás eredményeképpen jött létre, akkor módosítsa a `last-long-acc-number` számlálót az aktuális könyvtárban (az `UTIL G/2` segédprogrammal) ugyanarra az értékre, amely az indexelő könyvtár számlálójára vonatkozik.
- Amennyiben az indexelés elősegítéséhez módosította az indexelő könyvtárban található `$data_tab` táblákat (lásd fenn az 5-ös lépést), akkor másolja be a módosított táblákat az aktuális könyvtárba. Abban az esetben, ha néhány fájlban korábbi időjelzés szerepel, mint azokban, amelyeket le szeretne cserélni, akkor az `x/7` segédprogrammal tisztítsa ki az `utf_files-t`.

Az újraindexelő eljárás utolsó művelete ebben a szekcióban a démonok újraindítása (`UTIL E...`) az aktuális könyvtárban.

10. Azon rekordok újraindexelése, amelyek időközben módosultak

Aktiválja az aktuális könyvtárban a `UTIL/E/5/2` segédprogramot ahhoz, hogy az indexelő könyvtárban futtatott indexelés során módosított rekordokat is belevegye az indexelésbe. Ez a segédprogram átmásolja a lementett `Z07H` rekordokat a `Z07-be`, miközben törli a duplikátumokat. Az aktuális könyvtárban folyamatban lévő `ue_01` eljárás indexeli újra a `Z07-ben` tárolt rekordokat.

13 Indexelő szervizek

Minden egyes szerviz a neve alapján azonosítható a batch naplóban és a batch sorban.

Word index (manage-01)

Ez a szerviz az adatbázis szóindexét módosítja. Zárolja az ALEPH rendszert és kizárólag akkor futtatható, ha a könyvtár zárva van.

Direkt index módosítása (manage-05)

Ez a szerviz módosítja az adatbázis direkt indexét. Zárolja az ALEPH rendszert és kizárólag akkor futtatható, ha a könyvtár zárva van.

Besorolási index (manage-02)

Ez a szerviz módosítja az adatbázis besorolási indexét. Zárolja az ALEPH rendszert és kizárólag akkor futtatható, ha a könyvtár zárva van.

Ügyeljen arra, hogy a szerviz futtatásakor a „Az egész besorolási index újraépítése” módban futtatnia kell a „Teljes hosszon rendezés” (manage-17) szervizt a helyes rendezéshez.

Rendezési index (manage-27)

Ez a szerviz módosítja az adatbázis rövid indexét. Zárolja az ALEPH rendszert és kizárólag akkor futtatható, ha a könyvtár zárva van.

Besorolási rekordok rendezése (manage-16)

Ez a szerviz a besorolási elemeket rendezi a `tab00.lng` és a `tab_filing` táblában meghatározott rendezési szabályok szerint.

Ezek a szabályok egy „iktató szöveget” hoznak létre, amely szerint a besorolási elem rendezve lesz. A besorolási elemek ekkor az egyes bejegyzések iktató szövegének első 69 karaktere alapján lesznek sorba rendezve.

A teljes hosszon rendezés (manage-17) szerviz rendezi azokat a besorolási elemeket, amelyek iktató szövege 69 karakternél hosszabb.

A szerviz futtatása után mindig futtassa le a teljes hosszon rendezés (manage-17) szervizt is. Ez a szerviz zárolja az ALEPH rendszert és kizárólag akkor futtatható, ha a könyvtár zárva van.

Teljes hosszon rendezés (manage-17)

A teljes hosszon rendezés (manage-17) szerviz rendezi azokat a besorolási elemeket, amelyek iktató szövege 69 karakternél hosszabb. Ez a szerviz zárja az ALEPH rendszert és kizárólag akkor futtatható, ha a könyvtár zárva van.

Rövid bibliográfiai rekord (manage-07)

Ez a szerviz módosítja az adatbázis rövid bibliográfiai indexét. Zárja az ALEPH rendszert és kizárólag akkor futtatható, ha a könyvtár zárva van.

Ügyeljen arra, hogy ha a könyvtár `tab` mappájában található `tab100` tábla `CREATE-Z00R` változója `Y` értéket visel, akkor ez a funkció módosítja/hozza létre a `Z00R` rekordokat is. A `Z00R` tábla az adatbázis összes dokumentumában mezőire külön-külön `Z00R` rekordot tartalmaz. Ezek az adatok statisztikai célokra alkalmazhatók.

Index aktualizálás – kiválasztott rekordokra (manage-32)

Ez a szerviz a logikai bázisok számlálóit építi fel. A besorolási index (manage-02) felépítése után kell futtatni akkor, ha az adatbázisbeállítások ezt a számlálót használják.

Rövid (brief) rekordok módosítása

Ez a szerviz módosítja és hozza létre rövid rekordokat. Zárja az ALEPH rendszert és kizárólag akkor futtatható, ha a könyvtár zárva van.

Kapcsolat rekordok között (manage-12)

Ez a szerviz hozza létre az adatbázis rekordjai közötti linkeket. Zárja az ALEPH rendszert és kizárólag akkor futtatható, ha a könyvtár zárva van.

14 További olvasmányok

A következő ALEPH dokumentumok olyan anyagokat tartalmaznak, melyek nem szerepelnek az Indexelés fejezetben. Ezek a Ex Libris Documentation Center weboldalon érhetők el. Fontos, hogy a jelenlegi változat és ezen dokumentumok között eltérések lehetnek.

Hogyan futtassuk az Indexelő folyamatokat - Ez a dokumentum bemutatja, hogy hogyan futtassuk az olyan indexelő folyamatokat, mint a szó indexek és a besorolási indexek. Olyan kérdésekkel foglalkozik, mint az naplózás kikapcsolása, eljárások száma, lemez terület és fájl elhelyezkedés, könyvtár feloldása mialatt egy folyamat fut, folyamatok monitorozása, hibaelhárítás és futtatási idő kiszámítása.

Index Építés / Párhuzamos Futtatás - Ez a dokumentum bemutatja, hogy hogyan használható a párhuzamos futtatás az ALEPH-ben az indexelés idejének csökkentésére, bizonyos kikötések mellett. Kulcs koncepciókat mutat be, mint például a körök ciklusok, folyamat időtartama, lemez terület kalkuláció és az olvasó megismertetése a háttérmunkákkal. Ugyancsak felhívja a figyelmet arra, ha egy speciális feladat egy része párhuzamosan nem hajtható végre.