



DOKUMENTATION

Online-Schnittstelle Normdaten-Versorgung

INHALTSVERZEICHNIS

<u>VORBEMERKUNG.....</u>	4
<u>VORAUSSETZUNGEN FÜR DIE NUTZUNG DER SCHNITTSTELLE</u>	4
<u>DATENFORMAT, SATZTYP, SATZSTATUS</u>	4
<u>HARVESTING DER DATEN.....</u>	5
STEUERUNG DER HARVESTING-REQUESTZYKLEN	6
SELEKTIVES LADEN	7
REIHENFOLGE DER VERARBEITUNG	8
PLAUSIBILITÄTSPRÜFUNGEN UND ZUSÄTZLICHE VERARBEITUNGSSCHRITTE.....	9
PND-, GKD-, SWD-DATEN	9
Löschungen	9
Umlenkungen	9
TIT - TITELDATEN	9
Löschungen	9
Umlenkungen	9
LOK – LOKALDATEN.....	10
Korrekturen	10
Löschungen	10
KONFIGURATION.....	10
<u>LOGFILES / DATENFILES</u>	14
<u>STARTEN UND BEENDEN DES HARVESTING-DAEMONS</u>	15

Vorbemerkung

Die Benutzerdokumentation für die „Online-Schnittstelle Normdaten-Versorgung“ beinhaltet folgende Aspekte:

- Voraussetzungen für die Nutzung der Schnittstelle
- Datenmodell und Datenfluss
- Konfiguration und Logfiles

Eine Beschreibung der Protokollsyntax und detaillierte Informationen zu den im Rahmen der Datenübermittlung verwendeten Datenformaten ist nicht Bestandteil dieser Dokumentation. Weitere Informationen hierzu finden Sie unter den nachfolgenden Adressen:

MABxml: <http://www.ddb.de/professionell/mabxml.htm>

OAI-PMH v.2.0: <http://www.openarchives.org/OAI/openarchivesprotocol.html>

Voraussetzungen für die Nutzung der Schnittstelle

Die Online-Schnittstelle Normdaten-Versorgung ist ein Zusatzmodul zu ALEPH 500 und wird gesondert lizenziert. Die Kommunikation erfolgt über TCP/IP.

Die Schnittstelle berücksichtigt, dass die Daten in den Zielsystemen als Bestandteil des aktiven Bestandes gehalten werden und nicht zwangsläufig in Libraries gespeichert werden, die ausschließlich ZDB-Daten beinhalten. Werden die ZDB-Titeldaten selektiv in eine Library geladen, so ist der vollständige ZDB-Titeldatenbestand zusätzlich in einer weiteren Library der Installation vorrätig zu halten. Dies ist erforderlich, damit bei Lokalsatzneuaufnahmen der zugehörige Titelsatz aus dem Gesamtbestand in den Teilbestand übernommen werden kann, wenn er dort noch nicht vorhanden ist.

Datenformat, Satztyp, Satzstatus

Die von der Deutschen Bibliothek zur Verfügung gestellten Daten werden im Format MABxml übergeben und sind nach UTF-8 kodiert. Für die Verarbeitung werden die im MABxml-Satz hinterlegten Inhalte berücksichtigt. Die Informationen des OAI-Headers zum Datensatz werden nicht verwendet.

Folgende Satztypen können geliefert werden:

h, y, u, v	Titeldaten (set: zdb.titel)
l	Lokaldatensätze (set: zdb.exemplare)
k, w	Köperschaftsdaten (set: gkd)
p, t	Personendaten (set: pnd)
r, s, x	Schlagwortdaten (set: swd)

Anmerkung: Üblicherweise werden die Satztypen y, u, v, w, t und x nicht geliefert.

Pro Satztyp sind folgende Satzstatus vorgesehen:

Satzstatus	Satztyp
n = Korrektur oder Neuaufnahme	Alle Satztypen
d = Löschung	Alle Satztypen
u = Umlenkung	h, y, u, k, p, r, s

Umlenkungen und Löschungen werden erst dann im Repository bereitgestellt, wenn die Vorgänge im ZDB-Datenbestand abgeschlossen sind. Das Erfassen einer Umlenkung oder Löschung innerhalb der ZDB löst noch keine Bereitstellung aus. Erst die Durchführung des Vorgangs aktiviert den Zeitstempel für das Harvesting. Die Reihenfolge der Datensätze im Set entspricht nicht zwingend der Änderungsreihenfolge.

Harvesting der Daten

Die Daten der ZDB, GKD, PND und SWD werden von der Deutschen Bibliothek auf Basis des Open Archives Initiative Protocol for Metadata Harvesting, Version 2.0 in einem so genannten Repository zur Verfügung gestellt. Dieses Repository kennt folgende Sets:

- zdb:titel (ZDB-Titeldaten)
- zdb:titel:online:frei (Lizenzfreie Zeitschriften)
- zdb:exemplare (ZDB-Lokaldaten)
- gkd (GKD-Daten)
- pnd (PND-Daten)
- swd (SWD-Daten)

Der ALEPH 500 Harvester ist in Form eines Daemons (ue_13) implementiert. Der Daemon holt in konfigurierbaren Zeitabständen die aktuellen Änderungen. Die Setbezeichnungen und die Adresse des Repositories sind in einer Konfigurationsdatei hinterlegt. Beim ersten Start des ue_13 (UTIL/E/13) wird der konfigurierte Startzeitpunkt als „von“-Parameter verwendet. Der „bis“-Parameter wird auf Grundlage der ebenfalls konfigurierbaren Harvesterschnittweite (maximale Zeitstempeldifferenz zwischen „von“- und „bis“-Parameter ein und desselben Requests) ermittelt. Der ue_13 speichert den „von“-Parameter für den anschließenden Requestzyklus in der Datei `$data_files/util_e_13_next_run`, sobald die Verarbeitung eines Requestzyklus abgeschlossen ist. Damit ist sichergestellt, dass bei Beendigung und Neustart des ue_13 der Harvestingprozess nahtlos wieder aufsetzt.

Hinweis: Für Testzwecke kann die Datei `$data_files/util_e_13_next_run` gelöscht werden, um den „von“-Parameter über die Konfiguration manuell zu bestimmen.

Das Repository unterstützt Zeitstempel auf Sekundenbasis (UTC).

Steuerung der Harvesting-Requestzyklen

Es ist vorgesehen, dass jeder Harvesting-Requestzyklus mit der Abfrage von Lokaldaten beginnt. Bitte beachten Sie, dass die Reihenfolge, in der die Sets abgefragt werden, über die Konfiguration beeinflusst werden kann. Bitte übernehmen Sie, die in der ausgelieferten Musterkonfiguration für die MAB01 vorgegebene Reihenfolge, um Dateninkonsistenzen zu vermeiden.

Jeder Request beinhaltet einen „von“- und einen „bis“-Parameter. Der „bis“-Parameter kann in der Zukunft liegen. Er wird basierend auf dem „von“-Parameter berechnet, indem die Harvesterschnittweite hinzuaddiert wird.

Beispiel:

„von“-Parameter: 2004-11-05T10:00:30Z
Harvesterschnittweite: 1 Stunde
„bis“-Parameter: 2004-11-05T11:00:30Z

Für den ersten Request eines Harvesting-Zyklus wird der „von“-Parameter der Datei `$data_files/util_e_13_next_run` oder – falls diese Datei nicht vorhanden ist – der Konfigurationsdatei `$data_tab/tab_ue13.conf` entnommen. Der „bis“-Parameter wird errechnet. Liegt der Abfragezeitraum vollständig in der Vergangenheit - gemessen am Antwortzeitpunkt des Repositories (ResponseDate), wird derselbe Abfragezeitraum auch für die nachfolgenden Sets des Zyklus verwendet. Liegt der „bis“-Parameter in der Zukunft, wird als „bis“-Parameter für die nachfolgenden Sets des Zyklus der vom Repository gesandte Antwortzeitpunkt (ResponseDate) gesetzt. Somit können innerhalb eines Zyklus keine Titel- und/oder Körperschaftsdaten angefordert werden, die nach der Bearbeitung des Lokalsatz-Requests geändert wurden.

Für den nachfolgenden Zyklus wird der „von“-Parameter mit dem „bis“-Parameter des vorhergehenden Zyklus belegt. Tritt innerhalb einer Zyklusbearbeitung ein Fehler auf, wird der vollständige Zyklus wiederholt.

Beispiel (Auszüge aus dem ue_13-Logfile in kursiver Schrift):

```
2004-11-09 10:08:47 [log] Fetching 'http://oai.ddb.de/zdbrep.php?verb=ListRecords&metadataPrefix=mabxml-1&set=zdb:exemplare&from=2004-11-09T09:20:00Z&until=2004-11-09T10:20:00Z'...
```

Das Harvesting sei erfolgreich. Als Antwortdatum sendet das Repository:
responseDate=2004-11-09T10:02:06Z

Daraus wird die Anfrage für das zweite Set (Titeldaten) formuliert:

```
2004-11-09 10:10:29 [log] Fetching 'http://oai.ddb.de/zdbrep.php?verb=ListRecords&metadataPrefix=mabxml-1&set=zdb:titel&from=2004-11-09T09:20:00Z&until=2004-11-09T10:02:06Z'...
```

Für denselben Zeitraum werden schließlich noch Körperschaftsdaten geharvestet:

```
2004-11-09 10:11:02 [log] Fetching 'http://oai.ddb.de/zdbrep.php?verb=ListRecords&metadataPrefix=mabxml-1&set=gkd&from=2004-11-09T09:20:00Z&until=2004-11-09T10:02:06Z'...
```

Der nächste Requestzyklus beginnt ...

```
2004-11-09 10:11:41 [log] Fetching 'http://oai.ddb.de/zdbrep.php?verb=ListRecords&metadataPrefix=mabxml-1&set=zdb:exemplare&from=2004-11-09T10:02:06Z&until=2004-11-09T11:02:06Z'...
```

Alle Log- und Datenfiles eines Requestzyklus werden mit dem Tagesdatum und der laufenden Nummer des Zyklus gekennzeichnet. Diese Nummer wird ebenfalls über die Datei `$data_files/util_e_13_next_run` verwaltet. Nach dem Löschen der Datei beginnt der Zähler wieder bei „00001“, ein Überlauf setzt ihn auf „00000“ zurück.

Selektives Laden

Das Repository liefert alle Daten eines Sets, die ein im Abfragezeitraum liegendes Änderungsdatum aufweisen. Es sollen im Zielsystem aber nicht immer alle Daten geladen werden. Für die Online-Schnittstelle stehen dieselben Selektionsmechanismen zur Verfügung, die bereits aus der Verwendung der Batchladeroutinen bekannt sind (`p_manage_42`, `p_manage_20`). Im Rahmen der Batchlieferungen ist sichergestellt, dass zu jeder Lokalsatzneuaufnahme innerhalb derselben Lieferung auch der verknüpfte Titelsatz bereitgestellt wird. Dies kann für die Online-Lieferung nicht gewährleistet werden. Daher ist es erforderlich, die Titeldaten innerhalb einer Library der Installation vollständig vorzuhalten, wenn zusätzlich selektiv in eine zweite BIB-Library geladen werden soll. Lokaldaten können sowohl vollständig als auch auszugsweise in unterschiedlichen Libraries gespeichert werden.

Beim selektiven Laden von Titeldaten werden zunächst nur die Datensätze berücksichtigt, für die in der Library ein korrespondierender Datensatz ermittelt werden kann (Korrekturen). Bei Umlenkungen muss zumindest die Umlenkquelle vorhanden sein. Das Umlenkeziel kann anhand der ZDB-Identifikationsnummer aus der Library, die den vollständigen ZDB-Titelbestand enthält, importiert werden.

Soll nur ein Teilbestand der Lokaldaten vorgehalten werden, wird geprüft, ob das gelieferte Sigel in einer Konfigurationstabelle (`tab_sigel`) hinterlegt ist. Nur in diesem Fall wird der Lokalsatz verarbeitet. Kann zu dem Lokalsatz kein Titeldatensatz in der zugehörigen BIB (`library_relation`) ermittelt werden, wird er anhand der ZDB-Identifikationsnummer aus der Titellibrary, die alle ZDB-Titeldatensätze enthält, als Neuaufnahme importiert.

Die Ziel-Libraries und die Art der Übernahme (selektiv oder gesamt) sowie der Code der Library, die den vollständigen Titeldatenbestand enthält, werden in der Konfiguration hinterlegt. Falls für eine Datenart sowohl eine vollständige als auch eine selektive Verarbeitung in unterschiedlichen Ziellibraries vorgesehen ist, ist dafür

Sorge zu tragen, dass in der Konfiguration das vollständige Laden vor dem selektiven Laden aufgerufen wird.

Reihenfolge der Verarbeitung

Aufgrund der Abhängigkeiten innerhalb der Daten (auch setübergreifend) ist die Verarbeitung nicht in beliebiger Reihenfolge möglich. Titel- und Lokaldaten müssen zusammenhängend verarbeitet werden, da abhängig vom Satzstatus Sätze beider Sets in einer festgelegten Reihenfolge zu Laden sind. Körperschaftssätze können vor oder nach den Titel- und Lokaldaten importiert werden. Die Verarbeitung der Datensätze erfolgt analog zum p_manage_42 bzw. p_manage_20:

	Datenart	Satzstatus	Verarbeitung
1.	GKD PND SWD	n, d	Korrektur / Neuzugang: Anhand der Nomdaten-ID wird geprüft, ob bereits ein korrespondierender Satz existiert. Abhängig vom Ergebnis wird der Satz als Neuzugang oder Korrektur geladen. Soll selektiv geladen werden, bleiben Neuaufnahmen unberücksichtigt. Löschung: Der zu löschende Satz wird über die Normdaten-ID ermittelt und abhängig von der Konfiguration gelöscht oder nur protokolliert.
2.	GKD PND SWD	u	Umlenkung: Für die Verarbeitung werden sowohl Quelle als auch Ziel über die Normdaten-ID ermittelt. Abhängig vom Ergebnis der Plausibilitätsprüfung wird der Zielsatz mit einem Umlenkungsfeld angereichert und die Quelle wird gelöscht oder der Vorgang wird protokolliert.
3.	TIT	n	Korrektur / Neuzugang: Anhand der ZDB-ID wird geprüft, ob bereits ein korrespondierender Satz existiert. Abhängig vom Ergebnis und von der Verarbeitungsart (selektiv oder gesamt) wird der Satz als Neuzugang oder Korrektur geladen oder übergangen.
4.	LOK	n	Korrektur / Neuzugang: Anhand der ZDB-ID wird geprüft, ob bereits ein korrespondierender Satz existiert. (Für das selektive Laden werden nur Lokalsätze mit in der Konfiguration hinterlegten Sigeln verarbeitet) Abhängig vom Ergebnis wird der Satz als Neuzugang oder Korrektur geladen. Falls beim selektiven Laden der korrespondierende Titeldatensatz nicht ermittelt werden kann, wird dieser aus der Titellibrary, die alle ZDB-Titeldaten enthält, übernommen.
5.	LOK	d	Löschung: Der zu löschende Satz wird über die ZDB-ID ermittelt und abhängig vom Ergebnis der Plausibilitätsprüfungen gelöscht oder protokolliert.
6.	TIT	d	Löschung: Der zu löschende Satz wird über die ZDB-ID ermittelt und abhängig vom Ergebnis der Plausibilitätsprüfungen gelöscht oder protokolliert.
7.	TIT	u	Umlenkung: Für die Verarbeitung werden sowohl Quelle als auch Ziel über die ZDB-ID ermittelt. Ist bei selektiver Verarbeitungsart das Umlenkziel nicht im

			Bestand vorhanden, wird es aus der Library, die alle ZDB-Titeldaten enthält, bezogen. Abhängig vom Ergebnis der Plausibilitätsprüfungen wird die Umlenkung durchgeführt, wobei alle abhängigen Daten automatisch umgelenkt oder protokolliert werden.
--	--	--	---

Plausibilitätsprüfungen und zusätzliche Verarbeitungsschritte

Die aktivierten Plausibilitätsprüfungen und bibliothekarischen Details zum Update werden nachfolgend pro Datenart aufgeführt.

PND-, GKD-, SWD-Daten

Analog zum aktuell verwendeten p_manage_20 findet eine Überprüfung auf „Ansetzungsdubletten“ statt. Wird beim Laden einer Korrektur oder einer Neuaufnahme ein Datensatz mit der gleichen Ansetzungsform ermittelt, kann eine entsprechende Kennzeichnung in den Ladesatz eingefügt werden. Ändert sich beim Update die Ansetzungsform, wird automatisch ein „COR“-Feld erstellt.

Löschungen

Per Parameter kann gesteuert werden, ob Löschungen im Bestand durchgeführt oder ob Löschsätze nicht verarbeitet sondern nur protokolliert werden.

Umlenkungen

Umlenkungen werden nur dann durchgeführt, wenn Umlenkziel und Umlenkquelle ermittelt werden können. Fehlt einer der Sätze wird der Vorgang bei der Gesamtverarbeitung protokolliert.

TIT - Titeldaten

Bei der Verarbeitung der Titeldaten können Korrekturen und Neuanlagen ohne zusätzliche Plausibilitätsprüfungen oder die Manipulation abhängiger Daten erfolgen.

Löschungen

Eine Löschung wird nur dann durchgeführt, wenn

- keine abhängigen Titel vorhanden sind (DN-, SRD-, AND- Link),
- keine Lokalsätze zu dem Titel existieren (HOL Link),
- keine Exemplare (Z30) oder Erwerbungsätze (Z68) zu dem Titel existieren,
- es keine zugehörigen Z300-Sätze (Kopie der Exemplarsätze aus ALEPH-Lokalsystemen im Zentralkatalog) gibt.

PAR-Links verhindern den Löschvorgang nicht, werden aber protokolliert. Eventuell verknüpfte ADM-Sätze ohne weitere administrative Daten werden ebenfalls gelöscht.

Umlenkungen

Es wird anhand der ZDB-Identifikationsnummern der Ziel- und Quellsatz ermittelt. Fehlt einer der beiden Sätze im Bestand und kann – im Falle der selektiven

[Debug] Debug-Level (für Tests und Analyse)

[Sets] Definition der Sets und der angeschlossenen Verarbeitung

[Main] Parameter

BASE-URL	URL des Repositories
DELETE-XML-FILES	N/Y – Für Test und Analyse können die geharvesteten Daten in \$alephe_scratch hinterlegt werden. Das Format der Daten ist abhängig von der Verarbeitung (SERVICE-NAME). Der Service oairep.pl legt die Daten im MAB2-Format ab.
DELETE-LOG-FILES	N/Y – Für Test und Analyse können die Logfiles der im Bereich [Sets] angeschlossenen Weiterverarbeitung in \$alephe_scratch hinterlegt werden.
METADATA-PREFIX	Format der Metadaten. Belegung: mabxml-1
RECALL-PERIOD	Harvesterschnittweite in Stunden (Format nn). Anhand dieser Angabe wird der „bis“-Parameter für den ersten Request innerhalb eines Zyklus errechnet. Überschreitet eine Anfrage diesen Zeitraum, wird sie in mehrere Zyklen aufgeteilt.
SERVICE-NAME	Verarbeitender Service. Belegung: oairep.pl
START-DATE	Startdatum („von“-Parameter) in der Form: jjjj-mm-ttTss:mm:ssZ (Beispiel: 2004-10-27T08:10:00Z)
TIMEOUT	Timeout in Sekunden (Format nnn). Erfolgt innerhalb des Timeouts keine Rückmeldung vom Repository wird der Request abgebrochen und erneut versandt. Hinweis: Soll ein länger zurückliegender Zeitraum berücksichtigt werden, muss der Timeout-Wert entsprechend hoch gesetzt werden (mind. 240).
WAIT-TIME	Wartezeit zwischen zwei Harvesting-Zyklen in Sekunden (Format nnn). Übersteigt die Verarbeitungszeit die WAIT-TIME wird der nächste Zyklus direkt anschließend an die Verarbeitung der aktuellen Response gestartet.

[Debug] – Parameter

VERBOSE-MSG	Debug-Level. Die Werte 0 – 2 sind möglich.
-------------	--

[Sets] - Definitionen

Im Bereich [Sets] werden die Sets angegeben, die vom Harvester zu berücksichtigen sind und die Verarbeitungsroutinen zum Laden der Daten. Für die Übernahme der ZDB-Daten inklusive Körperschaftsdaten sind die Sets in der nachfolgenden Reihenfolge anzugeben:

```

LOK = zdb:exemplare
TIT = zdb:titel
EJN = zdb:titel:online:frei
GKD = gkd
PND = pnd
SWD = swd

```

Das Harvesten der lizenzfreien Online-Zeitschriften (Set „EJN“) ist nicht erforderlich, wenn in der Folgeverarbeitung (p_oairep_42) die Auswahl der Titeldaten nicht auf den eigenen Bestand beschränkt wird. Die lizenzfreien Zeitschriften sind auch Bestandteil des Sets „zdb:titel“ sind.

Den Setnamen des Repositories wird ein maximal fünfstelliger interner Code zugewiesen. Der Code wird benötigt, um die jeweilige Verarbeitung pro Set zu steuern. Die Verarbeitungsroutinen werden mithilfe der Angabe *<code>-RUN-*nn** angeschlossen. Der Platzhalter „*nn*“ steht für eine zweistellige Zahl von „01“ bis „10“. Bei Bedarf können so mehrere aufeinander folgende Läufe für ein Set aktiviert werden. Für eine korrekte Verarbeitung der ZDB-, GKD-, PND-, und SWD-Daten sind die Codes wie in dem obigen Beispiel zu wählen, damit die nachfolgenden Verarbeitungsroutinen auf die korrekten Eingabedateien zugreifen.

Für die Verarbeitung der Titel- und Lokaldaten steht die Routine p_oairep_42 zur Verfügung. Sie lädt sowohl Titel- als auch Lokaldaten, da die Abhängigkeiten innerhalb der Daten und die Reihenfolge der einzelnen Verarbeitungsschritte dies erfordern. Die Routine p_oairep_20 wird für das Laden der Körperschaftsdaten verwendet.

Parametrisierung von p_oairep_42:

```

csh -f p_oairep_42 <bib_library>, <input_file>, <check_local_data>, <char_conv>,
<cat_level_bib>, <025_id>, <cat_level_hol>, <merging_set>, <central_local_upd>,
<cataloger>, <entire_bib_stock>

```

<i>bib_library</i>	Librarycode der Titellibrary
<i>input_file</i>	Eingabedatei; Konstante <datetime>
<i>check_local_data</i>	N/Y – Y für selektives Laden , Default ist N
<i>char_conv</i>	Konstante MAB_TO_UTF
<i>cat_level_bib</i>	Cataloger-Level für den Aufbau des CAT-Feldes im Titelsatz
<i>025_id</i>	Y – Die ZDB-IDN wird im Feld 001 sowohl in der Eingabedatei als auch im Datenbestand erwartet (Index: IDN). N – Die ZDB-IDN wird im Feld 025z sowohl in der Eingabedatei als auch im Datenbestand erwartet (BIB-

	Index: 025; HOL-Index: IDZ)
<i>cat_level_hol</i>	Cataloger-Level für den Aufbau des CAT-Feldes im Lokalsatz
<i>merging_set</i>	Merging Set (COL 1) für tab_merge. Muss identisch für Titel- und Lokaldaten sein.
<i>central_local_upd</i>	Konstante: N
<i>cataloger</i>	Cataloger für den Aufbau des CAT-Feldes
<i>entire_bib_stock</i>	Beim selektiven Laden: Library-Code der Titel-Library, die alle ZDB-Titeldaten enthält.

Parametrisierung von p_oairep_20:

```
csh -f p_oairep_20 <active_library>,<input_file>,<fix_type>,<trace_level>,  
<z01_code>,<expand_code>,<char_conv>,<cataloger>,<cataloguer_level>,<check_dup>,  
<merge_type>,<check_local_data>,<perform_delete>
```

<i>active_library</i>	Library-Code der Körperschaftsdatei
<i>input_file</i>	Eingabedatei; Konstante <datetime>_<setname>
<i>fix_type</i>	Einsprung für tab_fix
<i>trace_level</i>	Trace-Level für die Protokollausgabe (ALL oder keine Belegung)
<i>z01_code</i>	Browseindex, der bei der inhaltlichen Dublettenprüfung herangezogen werden soll. Es wird dabei automatisch das Unterfeld \$\$9 für die Dublettenprüfung entfernt.
<i>expand_code</i>	Feld im Eingabsatz, das für die inhaltliche Dublettenprüfung herangezogen werden soll. Es wird dabei automatisch das Unterfeld \$\$9 für die Dublettenprüfung entfernt.
<i>char_conv</i>	Konstante MAB_TO_UTF
<i>cataloger</i>	Cataloger für den Aufbau des CAT-Feldes
<i>cataloger_level</i>	Cataloger-Level für den Aufbau des CAT-Feldes
<i>check_dup</i>	Y/N – Dublettenprüfung (match2) aktivieren? (Default: Y)
<i>merge_type</i>	Merging Set (COL 1) für tab_merge.
<i>check_local_data</i>	Y/N – Sollen nur bereits vorhandene Sätze bearbeitet werden? (Default: N)
<i>perform_delete</i>	Y/N – Sollen Löschungen durchgeführt oder nur protokolliert werden? (Default: Y)

Beispiel für eine [Sets]-Definition :

```
TIT-RUN-01 = csh -f $aleph_proc/p_oairep_42 MAB01,<datetime>,Y,  
MAB_TO_UTF,,N,,OAIREP-42,N,, > <logfile>  
GKD = gkd
```

```
GKD-RUN-01 = csh -f $aleph_proc/p_oairep_20
MAB11,<datetime>_<setname>,MAB,ALL,GEN,TMP01,MAB_TO_UTF,BATCH-
UPD,90,Y,OAIREP-20,N,N, > <logfile>
```

Mit der Angabe > <logfile> wird das Gesamtprotokoll des Laufs in eine Datei mit folgenden Namensaufbau umgeleitet:

```
$alephe_scratch/ue_13.jjjjmmtt.<Requestzyklusnummer>_<Set-Code>.log
```

Die Routinen p_oairep_42 und p_oairep_20 nutzen die Konfigurationstabellen für den p_manage_42 und p_manage_20. Die erforderlichen Anpassungen sind daher an dieser Stelle nicht dokumentiert, sondern der Dokumentation zum p_manage_20 und p_manage_42 zu entnehmen.

Logfiles / Datenfiles

Das Logfile des Harvesting-Daemons ue_13 ist im scratch-Verzeichnis der Library zu finden, in der der Daemon gestartet wird (run_e_13.<pid>). Abhängig vom definierten Debug-Level erfolgt die Protokollierung mehr oder minder detailliert.

Die Logfiles der einzelnen Ladeschritte werden für den p_oairep_42 im scratch-Verzeichnis der Library, in der die Importverarbeitung durchgeführt wird, hinterlegt. Hinsichtlich Bezeichnung und Aufbau entsprechen die Logfiles denen des p_manage_42, wobei zusätzlich die Requestzyklusnummer im Namen enthalten ist. Das Gesamtlogfile des p_oairep_42 bzw. -20 wird - sofern es erhalten bleiben soll - in \$alephe_scratch gespeichert. Der Dateiname ist – wenn die Ausgabe nach <logfile> erfolgt, wie folgt aufgebaut:

```
ue_13.jjjjmmtt.<Requestzyklusnummer>_<Set-Code>.log
```

Beispiel:

```
ue_13_20041109.00002_tit.log
ue_13_20041109.00002_gkd.log
```

Die Datenfiles für die Ladeläufe – sofern sie erhalten bleiben sollen – werden im \$alephe_scratch-Verzeichnis abgelegt. Der Dateiname ist wie folgt aufgebaut:

```
ue_13.jjjjmmtt.<Requestzyklusnummer>_<Set-Code>
```

Beispiel:

```
ue_13_20041109.00002_lok
ue_13_20041109.00002_tit
ue_13_20041109.00002_gkd
ue_13_20041109.00002_pnd
ue_13_20041109.00002_swd
```

Wurde der Debuglevel auf „2“ gesetzt, so werden zusätzlich auch die geharvesteten Daten im Originalformat (XML) in \$alephe_scratch hinterlegt. Der Dateiname ist wie folgt aufgebaut:

```
ue_13.jjjjmmtt.<Requestzyklusnummer>_<Set-Code>.dbg
```

Beispiel:

```
ue_13_20041109.00002_lok.dbg  
ue_13_20041109.00002_tit.dbg  
ue_13_20041109.00002_gkd.dbg  
ue_13_20041109.00002_gkd.pnd  
ue_13_20041109.00002_gkd.swd
```

Die Dateien bleiben nur erhalten, wenn in `$data_root/tab_ue13.log` hinterlegt ist, dass „Logfiles“ nicht automatisch gelöscht werden.

Starten und Beenden des Harvesting-Daemons

Der Daemon kann über das UTIL-Menue gestartet und beendet werden.

Start: UTIL/E/13

Stop: UTIL/E/14

Dies muss in der Library erfolgen, in der auch die Konfiguration für den Daemon hinterlegt ist. Wird der Daemon beendet, so führt der Prozess den begonnenen Harvestingzyklus durch und beendet sich erst, wenn alle konfigurierten Verarbeitungen abgeschlossen sind. Dies kann – insbesondere beim Harvesting über einen längeren Zeitraum – einige Zeit in Anspruch nehmen.

Für einen automatischen Start (startup) des ue_13 kann die Variable `OAIREP_STARTUP_LIBS` in `$alephe_root/aleph_start.private` genutzt werden. (Ein entsprechender Agent für die Einbindung des Prozesses in ein SUN-Cluster ist zurzeit in der Entwicklung.)

Der Daemon wird automatisch mit dem Sperren / Entsperren der Library gestoppt bzw. gestartet. Ein Neustart erfolgt nur, wenn die Library in der Umgebungsvariable `OAIREP_STARTUP_LIB` (`$alephe_root/alph_start`) gesetzt ist.