
1 AMU-Datenbank

1.1 Inhalt

Dieses Dokument enthält Informationen und Anweisungen, die für die Arbeit mit der AMU-Datenbank notwendig sind.

1.2 Zielgruppe

Das Dokument ist bestimmt für die Personen, die Installationen und Wartungsarbeiten an der AMU ausführen, sowie die System-Administratoren des Kunden bei der Arbeit mit dem AML-System unterstützen. Kenntnisse von dem Betriebssystem OS/2 und der AMU-Software und Grundkenntnisse über Datenbanken werden vorausgesetzt.

1.3 Datenbankaufbau

Datenbankname ABBA

Datenbankname	ABBA
Kommentar	ABBA/2 Management Unit Archive
Codepage	850

Die Datenbank wird über das UPM geschützt. Die Autorisierung erfolgt mit der User-ID "AMUADMIN"

Die Datenbank ist gegliedert ab der AMU Version 2.40 in drei Tabellen:

- COORDINATES (Stellplätze für Speichertürme und Linearregale),
- SCOODINATES (Laufwerke, E/A-Einheiten und Problembox)
- POOL (Reinigungs- und Scratchmedien-Verwaltung)

Qualifikationsmerkmal (Qualifier): AMU

Die Auswahl der Tabellen erfolgt z.B. im Query Manager mit

- AMU.COORDINATES
- AMU.SCOORDINATES oder
- AMU.POOL

Ab der AMU Version 2.40 sind in den Tabellen AMU.COORDINATES UND AMU.SCOORDINATES folgende Spalten vorhanden:

Spalte	Erklärung	Typ	Werte und Beispiele
COORDINATE	zehnstellige logische Koordinate des Stellplatzes als Ordnungsmerkmal der Tabelle	10 Zeichen	z. B. T002050110
VOLSER	1 bis 16-stellige Volume Serial Number (VSN) zur Identifizierung des Mediums (physikalisch auf den Medien durch ein Barcode-label repräsentiert).	16 Zeichen	z.B.GR0007
CTYPE	Verwendungstyp des Stellplatzes	1 Zeichen	S: Storage D: HACC-Dyn. A: AMU-Dyn. P: Problembox F: Foreign N: Clean
CATTR	Aktueller Status des Mediums auf diesem Stellplatz	12 Zeichen	O: Occupied E: Ejected M: Mounted J: In Jukebox R: Reverse Side Mounted Y: Empty T: Temp Here A: Temp Away
COWNER	Zuordnung der Zugriffsrechte der Roboter	1 Ziffer	1: Rob. 1 2: Rob. 2 3: Rob 1+ 2
USECOUNT	Anzahl der Zugriffe auf den Stellplatz (nicht auf das Medium)	Ganzzahl	
CRASHCOUNT	nicht verwendete Spalte	Ganzzahl	0
MEDIA	Medientyp	2 Zeichen	z. B. C0, O1, V2 .. (☞ AMU Reference Guide)

Spalte	Erklärung	Typ	Werte und Beispiele
BUDSTATE	Sicherungszustand bei Verwendung der Dual-AMU	1 Zeichen	C: Clean D: Dirty S: Suspicious W: Working
VOWNER	nicht verwendete Spalte	2 Zeichen	
TIMESTAMP	Zeitpunkt des letzten Update-Befehls auf diese Koordinate	17 Zeichen	Format: jjjj-mm-tt- hh.mm.ss.nnnn nn (N= Microsekunden)
VTYPE	Scratch-Zustand	1 Zeichen	U: Undefined S: Scratch
LOCKED	Zuordnung in Pools für spezielle Zugriffsrechte	16 Zeichen	z. B. DEFAULT_L
COUNTER	Zähler für die Cleanmedienverwaltung und Laufwerksreinigung	Ganzzahl	

1.4 SQL-Kommandos

1.4.1 Öffnen der Datenbank zur Bearbeitung

```
startdbm
dbm start using database abba
```

1.4.2 Ansehen der Datenbank

Für die Darstellung der vollständigen Tabelle wird empfohlen, die fensterdarsellung mit `mode 150,40` zu ändern: Am besten den Bildschirm (Fenster) auf ein breites Format stellen mit

Syntax

```
dbm select Spalte from Tabelle [where Spalte = | like
Wert]
```

Parameter	Erklärung
Spalte	Name der Spalte (☞ Datenbankaufbau) oder * für alle
Tabelle	Name der Tabelle mit Qualifier (z. B: AMU.COORDINATES)
Wert	Inhalt der Zelle, bei Zeichen in Hochkommas z.B: 'C0'

Beispiele

Abfrage der vollständigen Tabelle mit Speichertürmen und Linearregalen

```
dbm select * from amu.coordinates
```

Abfrage der vollständigen Tabelle mit E/A-Einheiten, Problembox und Laufwerken

```
dbm select * from amu.scoordinates
```

Um nur einzelne Felder anzusehen, kann man statt des * einzelne Felder angeben (z.B. nur Koordinate und Budstate).

```
dbm select coordinate,budstate from amu.coordinates
```

Um nach der Volser 'KW4711' zu suchen

```
dbm select * from amu.coordinates where volser =  
'KW4711'
```

Um nach freien Stellplätzen in allen Linearegalen zu suchen

```
dbm select * from amu.coordinates where coordinate  
like 'L%' and catr = 'E'
```

Um nach in der Datenbank definierten Laufwerken zu suchen

```
dbm select * from amu.coordinates where coordinate  
like 'D%'
```

Der Inhalt der Tabelle AMU.COORDINATES soll in die Datei COOR.TXT ausgegeben werden:

```
dbm -r(coor.txt) select * from amu.coordinates
```

1.4.3 Ändern in der Datenbank

Der Update-Befehl bietet die Möglichkeit einfache Änderungen, auch Massenänderungen (bis maximal 500 Datensätze) durchzuführen.

Syntax

```
dbm update Tabelle set Spalte=Wert [where Spalte =|like  
Wert]
```

Parameter	Erklärung
Spalte	Name der Spalte (☞ Datenbankaufbau)
Tabelle	Name der Tabelle mit Qualifier (z. B: AMU.COORDINATES)
Wert	Inhalt der Zelle, bei Zeichen in Hochkommas z.B: 'C0'

Beispiele

Alle Koordinaten des ersten Segments der Turmes 1 als leer deklarieren.

```
dbm update amu.coordinates set cattr='Y' where coordi-  
nate like 'T00101%'
```

Volser in Datenbank als mounted kennzeichnen:

```
dbm update amu.coordinates set cattr='M' where volser  
= 'KW4711'  
dbm update amu.scoordinates set cattr='O' and  
volser='KW4711' where coordinate = D901010101'
```

Will man nun also für einen ganzen Turm alle Koordinaten auf EMPTY setzen, darf nicht verwendet werden:

```
dbm update amu.coordinates set cattr='Y'  
where coordinate like 'T001%'
```

stattdessen

```
dbm update amu.coordinates set cattr='Y'  
where coordinate like 'T0010%'  
dbm update amu.coordinates set cattr='Y'  
where coordinate like 'T0011%'  
dbm update amu.coordinates set cattr='Y'  
where coordinate like 'T0012%'  
dbm update amu.coordinates set cattr='y'  
where coordinate like 'T0013%'
```

1.4.4 Backup und Restore der Datenbank

(☞ AMU Reference Guide)

1.4.5 Export der Tabellen in ASCII-Dateien

Zum Exportieren der ABBA-Datenbank müssen die beiden Tabellen der Datenbank exportiert werden. Hier wird im Gegensatz zum Backup nur der Inhalt der Tabellen abgezogen. Der Backup sichert zusätzlich noch alle anderen Informationen, die zu einer Datenbank gehören (z.B. Name, Codepage, Tabellenaufbau, Zugriffsrechte und sonstige Verwaltungsinformation).

Der erste Befehl exportiert die Tabelle AMU.COORDINATES in die Datei DB_COO.ASC. Meldungen während des Exportierens werden in die Datei DB_COO.MSG geschrieben. Die Dateien werden auf dem Verzeichnis angelegt, in dem die Kommandos ausgeführt werden. Der zweite Befehl exportiert die Tabelle AMU.SCOORDINATES.

```
dbm export from abba to db_coo.del of del messages  
db_coo.msg select * from amu.coordinates  
dbm export from abba to db_scoo.del of del messages  
db_scoo.msg select * from amu.scoordinates
```



Hinweis:

Ab DB2/2 V2.1 dürfen die Extensions .ASC nicht mehr verwendet werden. Bitte benutzen Sie die Extensions .DEL wie im Beispiel angegeben.

1.4.6 Import der ASCII-Dateien

Import in bestehende Tabellen mit gleicher Konfiguration, d.h. alle Datensätze sind vorhanden, aber nicht up to date:

```
dbm import to abba from db_coo.del of del replace
into amu.coordinates messages im_coo.msg
dbm import to abba from db_scoo.del of del replace
into amu.scoordinates messages im_scoo.msg
```

Import in leere Tabellen:

```
dbm import to abba from db_coo.asc of del insert into
amu.coordinates messages im_coo.msg
dbm import to abba from db_scoo.asc of del insert
into amu.scoordinates messages im_scoo.msg
```



Hinweis:

Siehe Hinweis bei Export der Tabellen in ASCII-Dateien

1.4.7 Erstellen von Commandfiles

Wenn mit einem Commandfile (Batchfile) mehrere Datenbankbefehle nacheinander ausgeführt werden sollen, so muß vor dem Wort DBM noch CALL stehen. Das liegt daran, daß DBM.CMD eben selbst ein Commandfile ist und Commandfiles nur mit CALL in Commandfiles aufgerufen werden können.

Beispiel:

Eine Datei dbselect.cmd enthält:

```
startdbm
call dbm start using database abba
call dbm select * from amu.coordinates
```

1.5 Hinweise zu Konfiguration

1.5.1 Welche Geräte sind in der Datenbank konfiguriert?

Nachdem die Koordinaten der Geräte immer eine erste Koordinate haben, die das Segment, das Fach und den Stellplatz im Fach angibt, kann man sich das wie folgt zunutze machen:

Für die Tabelle mit den Türmen und Linearregalen:

```
dbm select * from amu.coordinates where coordinate  
like '%010101'
```

Für die Tabelle mit EA-Bereich, Problemboxen und Laufwerken

```
dbm select * from amu.coordinates where scoordinate  
like '%010101'
```

Damit bekommt man immer nur die erste Koordinate eines Gerätes angezeigt.



Information

Falls die Systemkonfiguration (AMUCONST.INI) keine Einträge für ein gefordertes Gerät (auch z.B. Linearregal, oder die Kombination Segment mit Medium) enthält, wird von der AMU als Voreinstellung (Default) ein Segment mit einem Stellplatz (01 01) angenommen.

1.5.2 Datenbank und graphischer Konfiguration

Die grafische Konfiguration der AMU enthält die Türme, Linearregale, EA-Einheiten, Problemboxen und Laufwerke, sowie die Volser Ranges und die Logial Ranges. Die Datenbank hält für jedes dieser Geräte für jeden Stellplatz einen Datensatz.

Während des laufenden Betriebs (Produktion) benötigt die AMU die Konfigurationsdaten nicht! (mit kleine Ausnahmen) Die Konfigurationsdaten werden nur bei der Erstellung und bei Änderungen in der Datenbank benötigt. Die Änderungen in der Konfiguration werden nicht automatisch in der Datenbank nachgezogen. Dies soll immer ein bewußter Schritt sein.

-
-
- Funktion Create Archive zum Anlegen des Archivs von Anfang an.
 - Funktion Update Devices vergleicht die Konfiguration (INI-Files) mit der der Datenbank und gleicht die Datenbank der Konfiguration der INI-Files an. D.h. neue Geräte werden in der Datenbank hinzugefügt, nicht mehr vorhandene aus der Datenbank gelöscht. Das geht übrigens auch, wenn in einem Turm ein anderes Mediensegment eingebaut wird.
 - Function Update EIF , wenn Definitionen im EA-Schacht geändert werden.

1.6 Online-Hilfe

1.6.1 SQL-Befehle

Eine Hilfe zu den Befehlen erhält man mit

```
dbm ? Schlüsselwort
```

1.6.2 SQL-Fehlern

Eine Hilfe zu den SQL-Fehlern erhält man mit

```
dbm ? <SQL-Fehler>
```

1.7 Systemzeitkonflikt - Timestamp conflict

Oft kommt es zu der Situation, daß nach dem Einspielen einer neuen Software die AMU einen Fehler beim ersten Datenbankzugriff meldet. (SQL0805)

Das besagt, daß hier auf die bestehende Datenbank ABBA neue Programme zugreifen, die noch nicht beim Datenbanksystem angemeldet wurden.

Man kann diese mit den beiden mitgelieferten Commandfiles ARCBNDIT.cmd und ARCGRANT.cmd (früher ARCBNDIT.cmd und GRANTACC.cmd) beim Datenbanksystem anmelden. Dazu muß man aber als AMUADMIN eingeloggt sein und den Datenbankmanager starten.

Beispiel

```
logon /l amuadmin /p=xxxxxxxxxx
startdbm
arcbndit
arcgrant
```

Das gleiche Problem gibt es natürlich auch, wenn man wieder in der AMU-Version zurückgehen muß und läßt sich natürlich auf die gleiche Art und Weise lösen.

Eine andere Möglichkeit, wie man zu diesem Fehler kommen kann, ist die, daß die *.bnd-Dateien nicht zur AMU-Version passen.

1.8 Migration vom Datenbank-Backup

Nach einem Update des Datenbankmanagers funktioniert die normale Prozedure Restore Database vom DB/2 nicht mit einem Backup aus einer älteren Version. Dafür muß der Backup-File migriert werden:

Bei dem DB2/2 V2.1 ist das Programm DB2RESDB im Verzeichnis C:\SQLLIB\MISC.

Syntax: I

```
db2resdb <dbname> <source drive> <target drive>
```

Beispiel

```
db2resdb abba a c
```

Danach kann der normale Restore-Vorgang von C: ausgeführt werden

1.9 Die DB-Icons sind verschwunden

Mit DB2/2 V2.1 gibt es auf c:\sqllib\misc das Programm DB2ICO. Dieses stellt einen fehlenden DB2/2-Ordner wieder her!

1.10 Zugriffsrechte

Im DB/2 Version 1 wurden die Benutzer

- AMUADMIN mit Administratorrechten und
- AMUUSER mit Benutzerrechten eingetragen

Ab DB2/2 V2.1 wurde von IBM das User Profile Management (UPM) für DB2 abgeändert. Damit muß bis zur AMU-Version 3.0

- AMUADMIN mit Administratorrechten und
- AMUUSER mit Administratorrechten eingetragen sein

Ab AMU Version 3.0 wird nur noch AMUADMIN benutzt (AMUSER entfällt)