

## VORWORT FÜR UNI-MAN VERSION 3.0d

Liebe MIDI-asten!

Sie haben soeben die Version 3.0d des Programmes UNI-MAN – "the Most Extensive Generic SYSEX-Toolkit" – angeschafft. Dieses Softwarepaket sollte bei jedem MIDI-asten immer in Reichweite sein, da Sie hiermit die meisten Probleme auf dem Gebiet der MIDI/SYSEX Daten lösen können.

Damit sich auch wirklich jeder MIDI-Freund dieses Softwarepaket zulegen kann, haben wir es preislich sehr attraktiv gestaltet. Sie, die sie gerade erst UNI-MAN angeschafft haben, wissen natürlich, daß uns das auch geglückt ist. Für den Preis dieser gut ausgerüsteten Werkzeugkiste können Sie nur schwer einen Manager oder Editor für nur einen Synthesizer kaufen. UNI-MAN ist bis heute zweifellos das stärkste Softwarepaket aus der "PM-TOOLS" Sound Series. Wenn Sie mehr über die anderen PM-TOOLS wissen wollen, fragen Sie bitte Ihren Händler oder nehmen Sie direkt Kontakt auf mit ZADOK PRODUKTS.

Sollten Sie UNI-MAN noch nicht kennen, dann haben Sie wohl – mit Verlaub – die letzten paar Jahre auf dem Mars biwakiert oder sonstwo, wo MIDI höchstens als Kleidergröße bekannt ist und in jedem Fall keine MIDI-Literatur zu kaufen ist. Dieses Programm bedarf also eigentlich nicht mehr der Erläuterung. Am besten lesen Sie die oft sehr positiven Tests in der internationalen Fachpresse nach, als da wären:

Keyboards (D); Atari-ST Nieuws; Sound on Sound (E); Fachblatt (D) . Letzteres nennt UNI-MAN sogar "das Universal-Genie"! Dabei wurde damals nur die Version 2.2 mit einigen wenigen Adaptern getestet, während Sie nun die wesentlich verbesserte Version 3.0 in Händen halten mit sage und schreibe 50 Adaptern.

## **Um Mißverständnissen Vorzubeugen....**

Dieser UNiverselle MANager, abgekürzt UNI-MAN, (das Paradeferd aus Zadoks PM-TOOLS Sound Series) wird in der Fachpresse immer als universeller Editor dargestellt. Dies ist nicht ganz zutreffend, sonst hätten wir das Kind ja UNI-EDI getauft. Grundsätzlich können Sie mit UNI-MAN Probleme lösen, die bei der Arbeit mit MIDI-Geräten und Synthesizern auftreten können. Ganz nebenbei fallen dabei auch noch Soundeditoren und Bankmanager an, und das ist sicher kein Nachteil.

Auf jeden Fall können Sie von jedem MIDI-Gerät einen Dump auf Diskette ablegen. Darüberhinaus können Sie auch mit dem MIDI-Monitor (UNI-MON) sehen, was eigentlich alles auf der MIDI-Leitung passiert, oder mit dem Accessory (auch PRG), UNI-DUMP wirklich alle Arten Daten über MIDI empfangen oder versenden.

So sind noch viele andere praktische MIDI-Hilfsprogramme in dieser Werkzeugkiste zu finden. UNI-MAN ist also ein MIDI-SYSEX-TOOLKIT, auf das kein ernsthafter MIDI-ast verzichten kann.

Nicht umsonst wird UNI-MAN auf der ganzen Welt in Studios, Musikhochschulen und Musikschulen, an Univeritäten als Studierhilfe, bei diversen Soundwarehäusern zur Bearbeitung ihrer Soundlibraries und bei MIDI-Workshops und in Home-Studios eingesetzt.

Wir wünschen Ihnen viel Spaß und Erfolg mit dem UNI-MAN 3.0d Toolkit.

Ihr ZADOK-Team

## INHALTSANGABE

	Seite
VORWORT zu UNI-MAN 3.0d	1
Um Mißverständnissen vorzubeugen	2
INHALTSANGABE	3
Inhalt der Programmdiskette	4
Wie wird UNI-MAN 3.0d installiert?	5
EINLEITUNG	6
Das Update 3.0 (Allgemeine Bemerkungen)	7
Receive-, Send- und Save-Bank-Parameter	9
Zur Erstellung von Device-Adaptors	10
DIVERSE UPDATES 1	11
Neue Programmiertechnik	11
Erweiterte grafische Möglichkeiten	12
Zusätzliche Tastaturfunktionen	12
UNI-LINK VERSION 3.0	13
UNI-MAN VERSION 3.0d und die DUMMY-BANK (VORAB)	14
Dummy-Bank nach der Exclusive Bank	15
Wie wird eine solche nachfolgende Dummy-Bank erstellt?	15
Overall-Data	16
Bank-Receive Adaptor	17
Sound/Patch-Receive-Adaptor	17
Sound/Patch-Send-Adaptor	18
Die Dummy-Bank an sich	19
UNI-CON 3.0	20
DIVERSE UPDATES 2	24
Individual Parameter Handling	24
'Jump To' Fixed Parameter	24
Anpassung der 'Edit-Formulas'	24

**Inhalt der Programm-Diskette:**

UNI-MAN 3.0d hat keinen Hardware-Kopierschutz (Schlüssel) mehr und ist auch sonst vielfach weiterentwickelt. Deshalb ist der Inhalt der Diskette auch verändert worden. Das Programm UNI-BOY haben sie für diese Version nicht mehr nötig.

Inhalt der Diskette UNI-MAN 3.0d:

UNI-DUMPACC	UNI-CHAR.PRG
UNI-CON.PRG	UNI-DUMFPRG
UNI-LINK.PRG	UNI-MAN.PRG
UNI-MON.PRG	UNI-NUM.PRG
UNI-DUMFRSC	UNI-MON.RSC
UNI-NUM.RSC	

Der Ordner "RESOURCE"

ALL-MAN.RSC	ALL-RESTRSC
UNI-CON.RSC	UNILINK.RSC

Der Ordner "UNI-MAN"

INITD7V	STANDARD.DA
UNI-LINK.HF	STANDARD.NP
UNI-MAN.PIC	UNI-CHAR.SC1
UNI-CON.SC1	UNI-MAN.SC1
UNIMAN.SC2	UNI-MAN.SC3
INITSEQ	

**SEHR WICHTIG:** Bitte verändern Sie absolut niemals irgendetwas am Inhalt dieser Diskette. Auch Extensions oder das Desktop dürfen nicht verändert werden. Andernfalls kann die Funktion des Programmes gestört werden. Achten Sie also immer darauf, daß der Schreibschutz der Diskette aktiv ist!

## Installation des Programmes:

**WICHTIG:** Wenn Sie Programme als ACC benutzen und danach wieder andere Programme in Ihren Computer laden, müssen Sie zuerst das ACC im Desktop aktivieren und wieder verlassen. Tun Sie das nicht, könnte es passieren, daß das eingeladene Programm abstürzt oder nicht korrekt funktioniert. Bitte denken Sie also daran!

Benutzung von Diskette mit UNI-DUMP als Accessory:

1. Schieben Sie die Master-Diskette in Laufwerk A;
2. Schalten Sie den Monitor ein
3. Schalten Sie den Computer ein

Die Accessories werden automatisch beim "Booten" geladen und sind dann wie alle andere ACC:s unter dem Atari-Symbol zu finden.

ACHTUNG: Auch das UNI-MONPRG und UNI-NUMPRG können Sie als ACC benutzen. Also möchten Sie UNI-MONPRG und UNI-NUMPRG auf eine andere Diskette kopieren, und die zusammengehörende "RSC"-Files.

Benutzung von Diskette ohne Accessories:

1. Legen Sie eine andere Diskette in Laufwerk A;
2. Schalten Sie den Monitor ein
3. Schalten Sie den Computer ein
4. Wenn das gewohnte Desktop erscheint und die Laufwerkslampe erloschen ist, wechseln Sie auf die UNI-MAN Diskette.
5. Starten Sie mit einem Doppelklick das gewünschte Programm.

Benutzung mit einer Harddisk:

1. Kopieren Sie den ganzen Inhalt der Diskette nach Drive C: (Rootdirectory der Harddisk)
2. Starten Sie das gewünschte Programm mit einem Doppelklick, während sich die Original-Diskette in Laufwerk A: befindet.

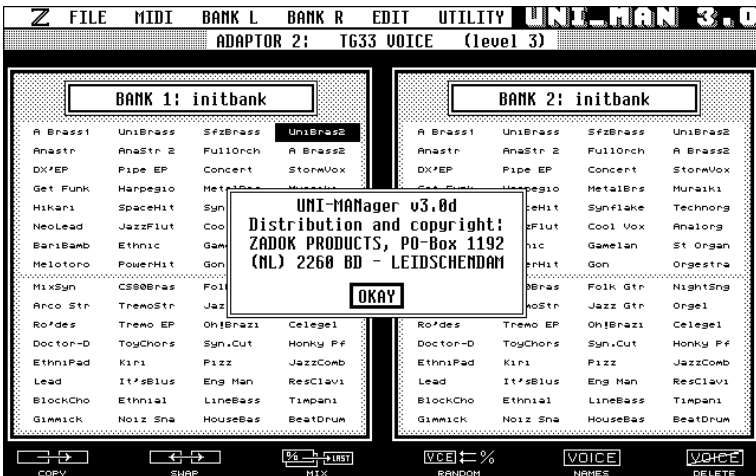
Beim nächsten Kaltstart werden die ACC:s automatisch installiert. Achten Sie bei aktivierten ACC's darauf, daß sich beim Starten der Programme immer die Originaldiskette in Laufwerk A: befindet. Denken Sie daran, daß Sie das ACC, nun eben noch einmal aktivieren müssen, bevor Sie andere Programme starten!

**ACHTUNG:** Wenn Sie (einige) Accessories nicht benutzen wollen, dann können Sie – ausschließlich auf der Harddisk! – , also nicht auf der Diskette, den Extender ACC, verändern (z.B. ACX). Das entsprechende Accessory wird dann beim nächsten Systemstart nicht mehr geladen.

## EINLEITUNG

Als UNI-MAN (es ist nun schon ein paar Jahre her) als erster vollständiger Universal-Manager auf den Markt kam, hat das bei Vielen (neben Bewunderung) auch Skepsis hervorgerufen. Da ja jeder Synthesizer seinen eigenen 'Dialekt' spricht, gab es anfänglich wenig Vertrauen in diese Software. Man hätte zwar gerne ein einziges Programm gehabt, mit dem alle MIDI-Geräte zu managen wären, nur müßte dies doch erst einmal seine Funktionstüchtigkeit unter Beweis stellen.

Seit der Präsentation des universalen UNI-MAN Systems hat zwischen den Beteiligten (Programmierer, Presse und natürlich die UNI-MAN-Anwender) ein reger Austausch (z.B. von Verbesserungsvorschlägen) stattgefunden. Auch das Arbeiten unter verschiedenen Bedingungen mit einem derartigen Programm hat eine Fülle von Erfahrungen gebracht. Es hat sich gezeigt, daß eine Anzahl von Anpassungen nicht nur wünschenswert, sondern in manchen Fällen sogar notwendig waren. Unabhängig davon haben die Updates für mehr Durchschaubarkeit und eine enorme Zunahme der Möglichkeiten gesorgt. Vor allem die Implementierung der Bibliotheksfunktionen mit umfassenden Random-, Mix-, Search- und Sortiermöglichkeiten stellte für den Anwender eine wichtige Neuerung dar.



Das alles hat dazu geführt, daß mit der Version 3.0 nun ein ausgereiftes Komplettsystem zur Verfügung steht. UNI-MAN hat sich in allen Anwendungsgebieten einen festen Platz bei den täglichen Arbeiten erobert, und das sowohl im professionellen als auch im Amateurbereich. Im Gegensatz zur Anfangsperiode erreichen uns immer mehr Reaktionen von zufriedenen Anwendern.

Daraus geht u.a. hervor, daß eine der Stärken von UNI-MAN darin zu sehen ist, daß er die Kommunikation selbst mit MIDI-Geräten ermöglicht, für die ansonsten **nie** Software geschrieben wurde bzw. geschrieben worden wäre.

Mittlerweile wächst das Interesse an UNI-MAN im In- und Ausland. Es ist für uns eine große Genugtuung, daß UNI-MAN in der renommierten Zeitschrift 'FACHBLATT MUSIKMAGAZIN' in einer Leserwahl zum besten und wichtigsten MIDI-Programm des Jahres 1990 den dritten Platz erreichte (nach den Sequenzern von C-Lab und Steinberg). (*Siehe Fachblatt Nr.3 vom März '91 und für den Test 'UNIVERSALGENIE' Fachblatt Nr1 vom Januar '91.*)

## UNI-MAN Update 3.0 (Allgemeine Bemerkungen)

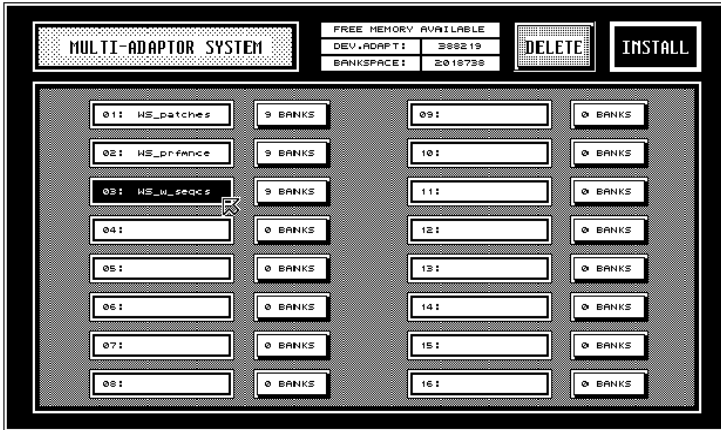
Dadurch, daß die Programmierung von UNI-MAN entlang einer bestimmten (geplanten) Entwicklungslinie verlief, war es möglich die Bedienungsanleitung so zu schreiben, daß nur geringfügige bzw. gar keine Änderungen beim Wachsen des Systems notwendig sind. Philosophie und Konzept bleiben genauso unberührt wie die Adaptors, die bereits in Gebrauch sind. Das ist nur möglich, weil von Anfang an genug Raum für weitere Entwicklungen gelassen wurde. Die zu erwartenden (vorgesehenen) Änderungen zur Version 3.0 sind bereits zum größten Teil umschrieben und bedürfen deshalb keiner näheren Erklärung. Insoweit jedoch Verbesserungen auf Anwenderniveau vorgenommen wurden (sowohl detaillierter als auch struktureller Art), werden diese unten erläutert. Auch programmiertechnische Änderungen (wie Speicherplatzeinsparung etc.) sollen kurz erwähnt werden.

Die in dieser ergänzenden Anleitung beschriebenen Änderungen gegenüber der Version 3.0 stellen die logische Konsequenz der Veränderungen auf dem Synthesizermarkt dar. Anfänglich war die Speicherstruktur von MIDI-Devices ziemlich einfach. Diese Schlichtheit war für ein universales Programm nicht nur willkommen, sondern sogar Bedingung. Vor allem kam diese Sachlage in Form eines durchsichtigen Konzeptes dem Anwender zugute. In den letzten Jahren kamen jedoch immer mehr Geräte mit umfangreicheren Speicher-Konfigurationen und komplizierten Strukturen auf den Markt. Dies hat uns zu einer noch flexibleren Programmstruktur veranlaßt.

Bis zur Version 2.2 von UNI-MAN war es weder möglich noch erforderlich, mit einem spezifischen Speicherbereich kommunizieren zu können. Bis dahin war der Grundgedanke hinter UNI-MAN, daß alle system-exklusiven Daten innerhalb des Programmes unverändert bleiben. In der Praxis bedeutete dies, daß eine einmal empfangene Bank zu jeder Zeit genau die System Exclusive-Daten enthielt, die auch durch das MIDI-Device abgegeben wurden.

Beim Eingeben des Requests im BANK RECEIVE ADAPTOR mußte also auch festgelegt werden, in welchen Speicherbereich des Devices diese Bank beim Senden ('COMPUTER ==> MIDI DEVICE') zu gehen hatte. Das war ebenso einfach wie zweckmäßig. Wenn es vereinzelt (z.B. bei KAWAI K1) um mehr als einen Bankspeicher ging, wurde derselbe Adaptor nochmals geladen und danach der Request mit einer kleinen Änderung versehen. Es mußten also genausoviele Adaptors geladen sein wie vorhandene Bankspeicher. Jeder Device-Adaptor korrespondierte so mit dem entsprechenden Bankspeicher.

Auf diese Weise entstand ein komfortabler und übersichtlicher Manager/Editor, der vom M.A.S. aus ähnlich einer Datenbank zu handhaben war. Dieser Weg konnte deshalb beschritten werden, weil Adaptors und Banks sehr klein waren. Der Nachteil dabei war daß nur die Banks (und damit die Sounds/Patches) jeweils eines Adaptors zur Verfügung standen. Das Vertauschen von Sounds/Patches zwischen den verschiedenen Adaptors mußte daher via Diskette erfolgen.



Wie bereits erwähnt, hat einerseits der Umfang der Soundbänke von MIDI-Devices immer mehr zugenommen und ist andererseits die Speicher-konfiguration komplizierter geworden. Außerdem ist es (anders als früher) häufig möglich, direkt mit Data-Cards zu kommunizieren. In zunehmendem Maße lassen sich über MIDI diverse Codes, beispielsweise für Modus-Umschaltungen oder Programmchanges usw. mitsenden. Vor allem beim Einsatz mehrerer gleichartigen Geräte, die über unterschiedliche Device-Nummern angesprochen werden und außerdem über mehr als eine Bank verfügen, ist die Koordination schwieriger.

Schließlich zeigte sich, daß die Sendegeschwindigkeit bei größeren Bänken kritischer ist als früher. Größere Datenmengen, die mit hoher (sprich: maximaler) Geschwindigkeit verschickt werden, kann nicht jedes Gerät gleichermaßen bewältigen. Obendrein kann ein Device eine kurze Pause zwischen zwei zusammengehörigen Dumps verlangen und ein anderes gerade

### nicht

Diese Entwicklungen haben dazu geführt, daß UNI-MAN strukturell verändert wurde. Dank der in dieser ergänzenden Bedienungsanleitung beschriebenen Veränderungen, ist es nun **doch** möglich, als Anwender auf die RECEIVE- und SEND-Parameter Einfluß zu nehmen. Dadurch wird jetzt nur noch ein einziger Device-Adaptor pro Parametergruppe (z.B. Patches) benötigt.

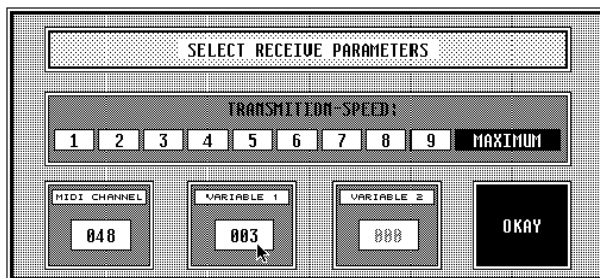
Sowohl beim Empfangen als auch beim Senden einer Bank kann bestimmt werden, wo die Daten herkommen bzw. abgelegt werden sollen. Auch läßt sich der MIDI-Kanal bzw. die Device-Nummer einstellen. Um schließlich auf die Send-Parameter einer Bank auf der Diskette Einfluß nehmen zu können, gilt dasselbe beim Abspeichern einer Bank ('SAVE-BANK'). Folglich kann der Anwender bestimmen, auf welchem MIDI-Kanal und wohin z.B. 'UNI-DUMP' die Bank sendet. Weil der Header bei 'RECEIVE-', 'SEND-' und 'SAVE-BANK' angepaßt wird, kann außerdem jede Bank - und damit jeder Sound/Patch - im selben Adaptor geladen werden. Dieser Punkt stellt neben der enormen Speicherplatzersparung möglicherweise den größten Vorteil dieser Arbeitsweise dar. Andererseits bleibt die bislang übliche Methode ohne Einschränkungen anwendbar.

Für die Version 3.0 ist ein erweiterter Device-Adaptor erforderlich. Der Anwender wird damit aber keine Probleme haben. Um einen Adaptor in das neue Format zu konvertieren, reicht es aus diesen einmal in UNI-MAN 3.0 zu laden und danach - mit den richtigen RECEIVE- und SEND-Parametern - auf die Diskette zu speichern.



## Receive-, Send-, und Save-Bank-Parametern

Sobald eine dieser Optionen gewählt wird, erscheint automatisch eine Dialogbox, worin die obengenannten Parameter einzustellen sind.



In der Regel kann die Sendegeschwindigkeit maximal sein. Manche MIDI-Devices aber können größere Datenmengen nicht so schnell verarbeiten und erfordern deshalb eine niedrigere Sendegeschwindigkeit. Das Gerät zeigt das meist als 'CHECKSUM-ERROR' oder andere Mitteilungen wie z.B. 'COMMUNICATION ERROR' u.ä. an. Manchmal kann es sogar passieren, daß MIDI-Devices abstürzen ('TOTAL ERROR') und jegliche weitere Operation verweigern.

Die einzig richtige Lösung: das Gerät etwa 10 Sekunden abschalten und die Übertragung nochmals – mit einer niedrigeren Sendegeschwindigkeit – starten. Der richtige Wert läßt sich nur durch Ausprobieren ermitteln. Wenn ein Dump aus mehreren Datenblöcken besteht, kann es sein, daß eine kurze Pause zwischen den Bulks nötig ist. Wählt man hier eine niedrige Zahl, wird automatisch eine Pause eingefügt, die mit der niedrigeren Sendegeschwindigkeit übereinstimmt.

Die richtige Sendegeschwindigkeit wird eigentlich schon beim Erstellen des Device-Adaptors bestimmt. UNI-MAN merkt sich in der Version 3.0 diese Angabe, sodaß der Anwender sich darum keine Gedanken zu machen braucht.

Unten in der Box können drei Parameter eingestellt werden, die sich auf den Request der zu empfangenden Bank und den Header der zu sendenden Bank beziehen. Im Prinzip bestreichen diese drei Parameter denselben Einstellbereich und können für jeden beliebigen Zweck eingesetzt werden. Der Bequemlichkeit halber wurden ihnen die wohl am häufigsten vorkommenden Namen, nämlich 'MIDI-CHANNEL', 'VARIABLE 1' und 'VARIABLE 2' gegeben. Der Bereich läuft im Dezimalsystem von 0 bis einschließlich 207. Dieser Umfang wurde gewählt, um dem Request z.B. einen Program-Change über jeden gewünschten MIDI-Kanal mitgeben zu können. Als Beispiel hierfür gilt der DW/EX 8000 von KORG. Weil einige Devices dies erfordern, können neben dem MIDI-Kanal noch **zwei** Variablen installiert werden. *(Siehe z.B. KAWAI K1 Adaptor).*

Der Parameter 'MIDI-CHANNEL' korrespondiert mit dem Symbol 'FC' im Request, die Parameter 'Variable-1' und 'Variable-2' mit 'FA' und 'FB'. Durch das Setzen dieser Symbole im Request kann der Anwender auf einfache Weise festlegen, an welcher Stelle der betreffende Parameter einstellbar sein soll.

## Zur Erstellung von Device-Adaptors

Dieser Abschnitt richtet sich speziell an die Benutzer, die sich für die Erstellung von Device-Adaptors interessieren. NI-MAN 3.0 weiß automatisch, an welcher Stelle und mit welchem Wert der Header von den im Speicher vorhandenen Banks angepaßt werden soll. Auch braucht der Anwender sich zur Einstellung der o.a. Parameter nicht mit Fragen wie Paketformat, Checksum über den Header oder nicht, Hereinholen der Bank als Ganzes oder als Anzahl Sounds/Patches usw. auseinanderzusetzen. Dies alles wird automatisch korrekt verwaltet. Wenn **einzelne** Sounds/Patches mit einem Edit-Buffer kommunizieren (oder einem Speicherplatz, der als solcher fungiert) sind diese Fragen diesbezüglich sowieso belanglos. Auch die Sendegeschwindigkeit spielt dann keine Rolle.

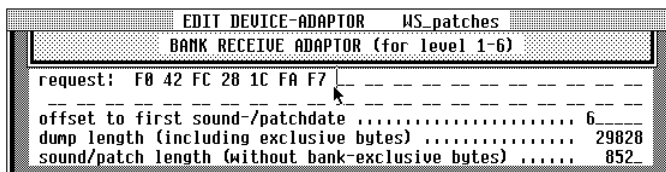
Zur Verdeutlichung dient in diesem Fall die KORG Wavestation als Beispiel. Sehen wir uns zunächst den Request beispielsweise einer kompletten Bank von Patches an: 'FO 42 3m 28 1C Ob F7'

FO	system exclusive status byte
42	KORG ID (identification number)
3m	format id (m = MIDI-Channel)
28	wavestation id
1C	all patch dump
Ob	(b = bank number)
F7	end of exclusive

Wie den meisten bekannt sein wird, steht der MIDI-Kanal bzw. die Device-Nummer (wie in diesem Falle auch) an dritter Stelle. Dies ist also die Stelle, wo der Programmierer des Device-Adaptors 'FC' eingeben sollte. Die Zahlen im Request sind hexadezimal und müssen daher in Werte umgesetzt werden, die der Musiker in der Dialogbox eingeben kann. Für MIDI-Kanal 1 (Zahlenwert: 0) müßte der Anwender also hier die Dezimalzahl '48' eingeben ('3m' entspricht  $3 \times 16 + 0 = 48$ ; diese Berechnungen können auch mit dem Accessory UNI-NUM durchgeführt werden). Beim Speichern des Device-Adaptors wird dieser Wert behalten und wird daher später automatisch an der richtigen Stelle erscheinen.

An sechster Stelle des Headers steht der kürzlich durch die I.M.A. (International MIDI Association) freigegebene Bankparameter. Der Wert '0' korrespondiert mit RAM 1, '1' mit RAM 2, '2' mit dem ROM und '3' mit der CARD. In diesem Falle reicht also eine Variable aus, um den Speicherplatz anzugeben. Hier sollte der Adaptorprogrammierer 'FA' eingeben.

Falls mehrere Requests notwendig sind, müssen diese Anpassungen auch überall durchgeführt werden.



## DIVERSE UPDATES 1

### Neue Programmieretechnik

Die Anwendung neuester Programmieretechniken ermöglichte es, den Umfang des Programms UNI-MAN drastisch zu reduzieren. UNI-MAN ist auf der Diskette um fast 200 Kbyte kürzer geworden. Das bedeutet, daß jetzt alle Programme des UNI-MAN Systems auf einer einzigen Diskette Platz finden. Außerdem geht damit ein deutlicher Speicherplatzgewinn innerhalb des Computers einher. Zwar ist der Nettogewinn nicht so spektakulär wie bei der Programmlänge, aber doch ansehnlich. Weiterhin wurden beide Programme (abhängig von der gemessenen Option) zwischen 66 und 100 Prozent schneller. Das bedeutet, daß die meisten Operationen kaum die Hälfte der Zeit zur Durchführung benötigen, die sie früher brauchten. Das ist umso erfreulicher, als vollgepackte Universalprogramme nun mal nicht die schnellsten sein können; z.B. müssen sie in komplizierten Scannerschleifen auf "1001 Möglichkeiten" Rücksicht nehmen. Hinzu kommt noch, daß – im Interesse des Musikers – oft mit Speicherplatz gewuchert wird und nur ein kleiner Rest für Betriebssystem und Programm bleibt. Am deutlichsten zeigt das die Option 'EDIT DEVICE ADAPTOR'. Beim ersten Durchgang der verschiedenen Dialogboxen braucht der Rechner noch einige Sekunden Zeit (für die sog. Garbage Collection). Danach ist der Rechner schneller weil sein Speicher bereits 'aktualisiert' ist.

### Erweiterung der Grafischen Möglichkeiten

Um die Flexibilität in der Darstellung weiter zu erhöhen, wurde die Liniengrafik um die Möglichkeit der Wiedergabe von Negativwerten erweitert. Es kann z.B. wünschenswert sein, alle Detunewerte (die ja i.d.R. sowohl einen positiven wie auch einen negativen Wertebereich überstreichen) im Bild zu haben. Von der Version 3.0 an kann zu diesem Zweck im GRAPHICAL WINDOW hinter dem Parameter GRAPHICAL\_ZERO ein negativer Offset eingegeben werden. Ein Beispiel ist zu finden im Adaptor 'PATCHES' der KORG Wavestation.

**ENTER GRAPHICAL PARAMETERS**

GRAPHICAL WINDOW NUMBER:

GRAPHICAL WINDOW NAME :

GRAPHICAL ZERO (+/-127):

OKAY

CANCEL

RECIPROCAL VALUE	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
VERTICAL-AXIS →	<input type="text" value="0"/>	<input type="text" value="60"/>	<input type="text" value="142"/>	<input type="text" value="224"/>	<input type="text" value="306"/>	<input type="text" value="1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	
GRAPHIC POINT	<input type="text" value="01"/>	<input type="text" value="02"/>	<input type="text" value="03"/>	<input type="text" value="04"/>	<input type="text" value="05"/>	<input type="text" value="06"/>	<input type="text" value="07"/>	<input type="text" value="08"/>	<input type="text" value="09"/>	<input type="text" value="10"/>
VALUE/HOR-AXIS →	<input type="text" value="0"/>	<input type="text" value="20"/>	<input type="text" value="20"/>	<input type="text" value="20"/>	<input type="text" value="20"/>	<input type="text" value="20"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	<input type="text" value="-1"/>	
RECIPROCAL VALUE	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
POINT-NAME →	<input type="text" value="DetA"/>	<input type="text" value="DetB"/>	<input type="text" value="detC"/>	<input type="text" value="detD"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	

LEVEL/DEPTH/  
GAIN/ETC.

TIME/RATE/  
FREQ./ETC.

(e.g. A D S R)

## Zuzätzliche Tastaturfunktionen

Im MANAGER:

TASTE	FUNKTION
[F1]	Sound/Patch oder Bank kopieren
[F2]	Sound/Patch vertauschen
[F3]	Sound/Patch oder Bank mixen
[F4]	Sound/Patch oder Bank randomisieren
[F5]	Sound/Patch (um)benennen
[F6]	Sound/Patch löschen
[F7]	Bank im Librarian speichern
[F8]	Librarian in Bank laden
[F9]	Sound/Patch oder Bank vergleichen
[F10]	Sequencer
[L]	Funktionen laden
[S]	Funktionen speichern
[E]	Edit-Modus aufrufen
[M]	M.A.S. aufrufen (siehe [ESC])
[F]	Diskette formatieren
[D]	Directory zeigen / File löschen
[P]	Bank ausdrucken
[T]	MIDI Thru
[ESC]	M.A.S. aufrufen

Im M.A.S.:

[D]	Device-Adaptor löschen
[RETURN]	Device-Adaptor installieren
[CURSOR]	auf/ab Device-Adaptor selektieren
[CURSOR]	links/rechts Device-Adaptor ↔ Banks
[-], [+]	auf 'Banks' Anzahl der Banks einstellen

Im Editor:

[SHIFT] + [J]	Parameter für Sprung markieren
[J]	Markierten Parameter anspringen

## UNI-LINK VERSION 3.0

In Anbetracht der Veränderungen, die im Zusammenhang mit dem Device-Adaptor und seiner Verarbeitung vorgenommen worden sind, wurde es nötig, auch UNI-LINK anzupassen. Nur die Version 3.0 von UNI-LINK verarbeitet einen 3.0 Adaptor korrekt und konstruiert einen Adaptor, der die o.a. Funktionen unterstützt.

Sicherheitshalber wollen wir einige Punkte noch einmal anhand eines Beispiels erläutern. Dazu betrachten wir wieder die Korg Wavestation. Dieser Synthesizer verfügt über drei Speicherbereiche:

1. RAM-1
2. RAM-2
3. CARD

Jeder der drei Speicherbereiche umfaßt:

1. PERFORMANCES
2. PATCHES
3. WAVESEQUENCES

Bis zur Version 2.2 von UNI-MAN würde man für eine vollständige Editierung der Wavestation neun (3 x 3) Adaptors benötigen. Da sowohl Adaptors als auch Banks ziemlich groß sind, war es nicht möglich, eines der Sets aus drei Adaptors im Speicher eines 1040 ST unterzubringen, geschweige denn arbeiten zu lassen. Dank der Speicherplatzeinsparung durch die neue Programmierertechnik und der o.a. Bank-Parameter ist es nun doch möglich.

In dem Ordner WAVESTATION befinden sich folgende Adaptors:

1. Multi-Setup-WS
2. Patches-WS
3. Performances-WS
4. Wave-Sequences-WS
5. Total-WS

Außerdem finden sich noch eine Anzahl Requests im Ordner 'REQUESTS' und die beiden Konfigurationen MEGA-2-4.CFG und 1040-BOYCFG. Die Requests sind gedacht, um auch ein SYSTEM-SETUP oder eventuelle GLOBAL-Daten mittels der Option 'RECEIVE MIDI-FILE' oder mit UNI-DUMP auf Diskette schreiben zu können. Es ist übrigens auch sehr gut möglich, bei Bedarf einen kleinen LEVEL-6-Adaptor zu erstellen, mit dem sich auch diese Daten editieren lassen. Beispiele befinden sich auf der "DATA DISK". In der Praxis wird der Bedarf allerdings nicht so groß sein, da es sich durchweg um einmalige Einstellungen handelt.

Beim Arbeiten mit der Wavestation muß eine Besonderheit berücksichtigt werden. Es ist können nämlich innerhalb einer PERFORMANCES-Bank (z.B. RAM-1) ausschließlich Patches und Wavesequences aus derselben Performance-Bank benutzt werden. Soundbanks sollten also nicht isoliert ausgetauscht werden, sondern nur in Zusammenhang mit dem kompletten Speicherbereich (von ungefähr 180.000 Bytes ohne CARD).

Da es also unsinnig wäre, **nur** Patches oder **nur** Performances auszutauschen, wurde mit UNI-LINK 3.0 ein Device-Adaptor erstellt, der den **kompletten** Inhalt von RAM-1, RAM-2 oder CARD empfängt. Selbst Adaptors, die wie in diesem Fall mit **mehreren** Bänken arbeiten, brauchen jetzt nur noch einmalig angefertigt werden. Möglich wird das, da UNI-MAN automatisch **alle** Header entsprechend den vom Anwender eingegebenen Parametern anpaßt, bevor gesendet wird.

Trotz der Tatsache, daß die zugehörigen Bänke so groß sind, hat ein solcher Adaptor in einem 1040 ST bequem Platz. Werden MIDI-Kanal und/oder Speicheradresse einmalig **vor** dem Senden oder Empfangen der Daten eingestellt, wird UNI-MAN die Werte automatisch für alle Performances, Patches und Wavesequences übernehmen, damit sie alle innerhalb des angegebenen RAM-Speichers bleiben.

Bei manchen älteren Software-Versionen kann es passieren, daß die Wavestation mit der maximalen Sendegeschwindigkeit von UNI-MAN nicht Schritt hält. In diesem Falle wird es erforderlich sein, eine geringere Sendegeschwindigkeit einzustellen. Auch bei Verwendung der Option 'SEND MIDI FILE' ist dieser Parameter von Belang. In den meisten Fällen wird eine solche Verzögerung jedoch nicht notwendig sein.

## UNI-MAN 3.0d UND DIE DUMMY-BANK (VORAB)

Die im folgenden hinsichtlich des Updates zur UNI-MAN Version 3.0d erläuterten Zusammenhänge sind hauptsächlich für die Benutzer von Interesse, die sich mit der Konstruktion von Device-Adaptors beschäftigen wollen. Wer UNI-MAN ausschließlich als ManagerEditorData-

FileSound-CreatorLibrarian usw. benutzen will, braucht sich mit diesem Teil **nicht** auseinanderzusetzen. Andererseits schadet es sicher nicht, diese Abschnitte **doch** zu lesen, wenn man ein gewisses Interesse für das Thema MIDI System Exclusive mitbringt; es ist dann sogar durchaus zu empfehlen!

## Dummy-Bank Hinter der Exclusive-Bank

In diesem Abschnitt geht es um das Erstellen einer DUMMY-BANK, die sich **hinter** der eigentlichen "Exclusive"-Bank befindet. Die folgenden Maßnahmen benötigen die UNI-MAN Version 3.0d, da mit dem Ziel größerer Universalität im Vergleich zur Version 2.2 umfangreiche Änderungen vorgenommen wurden.

Im UNI-MAN Handbuch ist mehrfach die Rede von einer sog. DUMMY-BANK. Dabei handelt es sich wie Sie wahrscheinlich bereits wissen um eine Bank, die Daten enthält, welche **nicht** zu den eigentlichen Sound-Daten gehören.

Das prägnanteste (und gleichzeitig geläufigste) Beispiel ist die Dummy-Bank, die der eigentlichen "System Exclusive Bank" **voran** geht und beispielsweise Soundnamen oder Anmerkungen zum Sound enthalten kann. Von dieser Möglichkeit wird häufig Gebrauch gemacht, wenn das MIDI-Device selber nicht mit Sound/Patch-Namen arbeitet, oder der Musiker eine andere Form der Klassifikation benötigt. Alle Bedingungen für eine korrekt funktionierende Dummy-Bank werden im Handbuch mehrfach ausführlich erörtert.

Es mag jedoch vorkommen, dass es wünschenswert ist, **nach** den eigentlichen Sound-Daten extra Daten zu senden. Das ist z.B. bei Synthesizern der Fall, die noch einen WRITE REQUEST verlangen, nachdem sie die Sound-Daten empfangen haben (z.B. der D70 von Roland). Erst nach dem Empfang dieses Write Request wird dann z.B. das Display aktualisiert und der Sound als definitiv betrachtet. Wichtig ist, dass oft nur Einzelsounds, nicht jedoch die komplette Bank einen solchen Write Request benötigen. Dabei können die Write Requests sogar je Sound unterschiedlich sein, wenn jeder Sound seine eigene Speicher-Adresse hat.

Es kann auch erforderlich sein, das angeschlossene MIDI-Device nach dem Empfang der Daten in einen spezifischen "Mode" zu schalten bzw. zurückzusetzen. In diesem Fall sind die zusätzlichen Daten durchgehend identisch.

## Wie wird in UNI-MAN solch eine Dummy-Bank Realisiert?

Der Einfachheit halber nehmen wir einmal an, dass das angeschlossene MIDI-Gerät eine Bank abgibt, die alle Sounds/Patches enthält. Als Beispiel nehmen wir den Roland D70. Wie gesagt, verlangt dieser Synthesizer nach dem Empfang eines Sounds einen Write-Request von 13 Bytes Länge. In einer Bank befinden sich 64 PERFORMANCES. Da der Rest ausführlich im Handbuch dargelegt wird, beschränken wir uns hier auf die Details, die mit der Dummy-Bank in Zusammenhang stehen.

```

EDIT DEVICE-ADAPTOR   D70_PRFMNC
BANK RECEIVE ADAPTOR (for level 1-6)
request: F0 41 10 39 11 00 25 41 00 70 40 62 F7  -----
offset to first sound-/patchdate ..... 8_____
dump length (including exclusive bytes) ..... 16034
sound/patch length (without bank-exclusive bytes) ..... 241_

```

## Overall-Data

Als erstes mu unter "OVERALL DATA" sowohl bei "NUMBER OF DUMPS PER BANK" wie auch bei "NUMBER OF DUMPS PER SOUND/PATCH" eine "2" eingegeben werden. Vor allem, wenn **nur** die Sound/Patch-Daten mit einem Write-Request versehen werden müssen, wird man geneigt sein, allein die zweite Frage mit "2" zu beantworten. Das ist dann korrekt, wenn die betreffenden Daten tatsächlich in der ersten Bank vorhanden sind. Im Falle einer separaten Dummy-Bank jedoch mu UNI-MAN seine Daten aus dieser zweiten Bank holen, von daher mu auch für die erste Frage eine "2" eingetragen werden.

```

EDIT DEVICE-ADAPTOR  D70_PRFMNC
-----
OVERALL DATA (for all levels)
-----
device name ..... D70_PRFMNC
file extender ..... PRF
-----
number of snds/patches (max.128)... 64_
name-offset in sound/patch ..... 0__
length of sound-/patchname ..... 10
number of dumps per bank ..... 2_
number of dumps per sound/patch ... 2_
convert patch before send ..... 0__
-----
level: 1=bank, 2=patch, 3=parantr 3
preconvert to 8/7 bitformat ..... 0__
maximum paran's/bytes in editor ... 400_
-----
EXIT                                INSTALL
  
```

Auf diese Weise entstehen in einem solchen Fall:

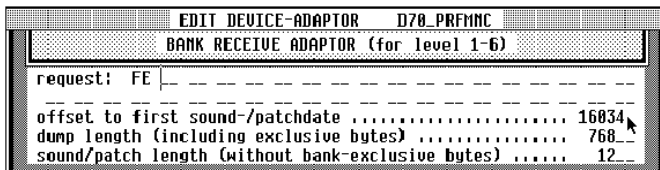
- Zwei BANK-RECEIVE-ADAPTORS
- Zwei SOUND/PATCH-RECEIVE-ADAPTORS
- Zwei SOUND/PATCH-SEND-ADAPTORS

In den jeweils letzten Device Adaptor-Boxen (in diesem Fall den zweiten) müssen immer die Daten für die Dummy-Bank eingetragen werden.



### Bank-Receive-Adaptor

Im Prinzip gelten dieselben Regeln wie bei einer Dummy-Bank, die den Sounddaten **voran** geht. Bedingung ist, da die Dummy-"BANK LENGTH" größer "0" ist und der "OFFSET TO FIRST SOUND/PATCH DATA-BYTE" im Bank-Receive-Adaptor der Länge der vorangehenden Dummy-Bank(s) entspricht. Wenn z.B. die ursprüngliche Banklänge 16034 Bytes beträgt, gilt für die zweite (Dummy-)Bank ein Offset von 16034.



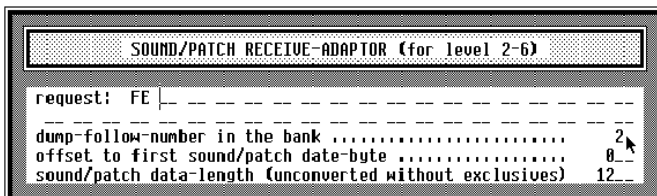
Um zu vermeiden, da UNI-MAN beim Empfang einer Bank auf die Daten der zweiten (Dummy-)Bank wartet, mu als Request für die Dummy-Bank "FE" eingetragen werden (*siehe Abschnitt 6.2.11 des Handbuches*).

Der Write-Request besteht aus 13 Bytes, wovon das letzte ("F7") durch UNI-MAN selbst erzeugt wird. Daher beträgt die Länge pro Data-String (des Write-Requests) nur 12 Bytes. Die Dump-Länge berechnet sich somit aus  $64 \times 12 = 768$ . Entsprechend ist natürlich die Sound/Patch Length ebenfalls "12".

Die "DUMP-FOLLOW-NUMBER" der Dummy-Bank ist immer "2", da sie (in diesem Fall) ja **hinter** den eigentlichen Exclusive-Daten kommt.

### Sound/Patch-Receive-Adaptor

Auch für den zweiten Sound/Patch-Receive-Adaptor gilt, da als Request "FE" eingegeben werden mu. Die zweite Bank bleibt dann beim Datenempfang unberührt. Da die Dummy-Bank keinen Request benötigt und enthält, wird für den "OFFSET TO FIRST SOUND/PATCHDATA" eine "0" eingegeben.



Die Dumplänge beträgt wieder  $64 \times 12 = 768$  und für die Sound/Patch Data-Länge wird wie gehabt "12" eingetragen.

## Sound/Patch-Send-Adaptor

Im zweiten Sound/Patch-Send-Adaptor **mu** in die Request-Zeile "00" eingetragen werden, um UNI-MAN anzuzeigen, da kein Header vorhanden ist.

SOUND/PATCH SEND ADAPTOR (for Level 2-6)	
header:	00
dump-follow-number in the bank .....	2
converting-table follow-number .....	2
checksum precount (pos. or neg. offset) .....	0
number of headerbytes to add -> checksum .....	0
program-change (0=no, 1=before, 2=after) .....	2
program-change channel .....	16
program-change sound-/patchnumber .....	1
send-message for data-upgrade by device (0=off, 1=on) ..	0
SOUND/PATCH SEND ADAPTOR: 2	
OKAY	

(Es gibt noch eine zweite Möglichkeit, den Dummy-Bank-Adaptor zu gestalten, die universeller, aber auch komplizierter ist und dabei von einer Reihe praktischer Eigenschaften UNI-MANs profitiert. Diese Ausführung wird unten in der **Anmerkung** beschrieben.) Da hier kein "FE" vorkommt, werden die Daten dieser Bank tatsächlich **mitgesendet**. Auf diese Weise erreicht man, da diese zusätzlichen Daten **nicht** mit der Bank, **wohl aber** mit den Sound gesendet werden. Falls das MIDI-Gerät eine kurze Pause zwischen den Dumps benötigt, kann dies durch Eingabe von einem oder mehreren "FD" in der Header-Zeile erreicht werden. Diese Pause sollte wirklich nur im Bedarfsfall eingefügt werden, da sie ansonsten zu unnötigen Wartezeiten führt.

Die restlichen Daten entsprechen wieder dem SOUND/PATCH-RECEIVE-ADAPTOR.

Oft arbeitet man in UNI-MAN mit einem selbstdefinierten Edit-Buffer. Das kann z.B. der Soundspeicher Nr 1 am MIDI-Gerät sein. Diese Methode ist vor allem praktisch bei Geräten, die nicht über einen eigenen Edit-Buffer verfügen. In diesem Fall hat das zur Folge, da die Write Request-Daten immer exakt gleich sind. Alle Sounds werden dann an dieselbe Adresse gesendet. Für diejenigen, die einen Device-Adaptor perfektionieren wollen, ist es ohne weiteres möglich, ihn so abzuändern, da jeder Sound/Patch an seine eigene Adresse gesendet wird. Die Sound/Patch-Daten stehen dann im Gerät immer auf dem richtigen Platz.

**ANMERKUNG:** Ein Dummy-Send-Adaptor, mit dem an jede gewünschte Adresse gesendet werden kann, und der automatisch die korrekte Checksum über den Request erstellt, sieht folgendermaßen aus:

Als HEADER (wo oben "00" eingegeben wurde) kann der unveränderliche Teil (über den auch keine Checksum berechnet wird) eingetragen werden. Bei den D70 Performances ist das "F0 41 10 39 12". Die Dummy-Bank besteht dann aus 64 x "address" MSB und LSB gefolgt von "size" MSB und LSB. Da in unserem Beispiel für alle Performances dieselbe Adresse benutzt wird, sind das immer die Zahlen "04 58 6A 00 00 01".

Nun **mu** noch im zweiten Bank-Receive-Adaptor der passende Checksum-Modus angegeben werden ("2"), damit UNI-MAN automatisch für jede zu sendende Performance eine Checksum generiert und ein "F7" anhängt. Dieses System funktioniert auch, wenn für jede Performance eine andere Adresse eingegeben wird.

## Die Dummy-Bank Selbst

Unter anderem, um selber (ohne programmiertechnische Kenntnisse) eine Dummy-Bank erstellen zu können, wird mit dem UNI-MAN Toolkit das Hilfsprogramm UNI-CON geliefert. Im obigen Beispiel besteht die Dummy-Bank aus 64 Write Requests á 12 bzw. 6 Bytes. Der Einfachheit halber gehen wir von einer Dummy Bank aus, die den kompletten Write-Request (12 Bytes) enthält. Im Prinzip müssen diese  $64 \times 12 = 768$  Bytes aneinandergelinkt und auf Diskette gespeichert werden. Anschließend mu dieses File hinter die eigentliche Bank "gehängt" werden.

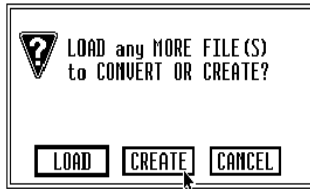
Diese Operation mu nur einmal vorgenommen werden, da UNI-MAN die Daten im Rahmen der INIT-BANK ablegt und von daher automatisch mit jeder neuen Bank abspeichert.

Es ist auch möglich, die ganze Manipulation in einem Rutsch durchzuführen. Sie können hier die Methode wählen, die ihnen mehr liegt.

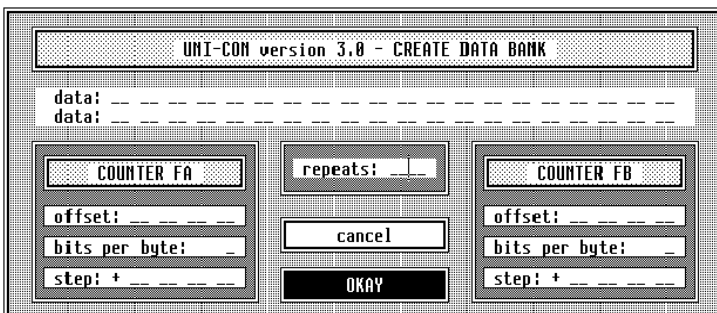
## UNI-CON 3.0

Starten Sie zunächst das Programm UNI-CON. Achten Sie darauf, da die Versionsnummer von UNI-CON 3.0 oder höher ist. In früheren Versionen ist die hier besprochene Funktion nämlich nicht vorhanden. Wählen Sie die Option "Specials". Wie gewohnt erscheint die Dateiauswahlbox. An dieser Stelle bestimmen Sie, ob eine Bank geladen wird, die der Dummy-Bank vorangeht oder ob die Dummy-Bank zuerst erstellt wird. In unserem Fall geht es um die Init-Bank wie sie vom D70 abgelegt wurde. Um alles in einem Rutsch zu erledigen, klicken Sie die D70 Bank an und danach "OKAY". Die Bank wird dann geladen.

Im anderen Fall wählt man "Cancel". Beide Male erscheint dann dieselbe Frage: "LOAD any MORE FILE(S) to CONVERT or CREATE?". Wählen Sie nun "CREATE".



In der Dialogbox, die dann erscheint, (siehe Abbildung "CREATE DATA BANK") kann die Dummy-Bank erstellt werden. Die Abbildung läßt einen gewissen Schwierigkeitsgrad erwarten, jedoch ist das Gegenteil der Fall, da die Daten der MIDI-Implementation ohne weiteres "passen". Um zu vermeiden, daß beim Aufbau des Device-Adaptors Daten, deren Länge ein Byte überschreitet, mühsam umgerechnet werden müssen, können die Werte in ihrer ursprünglichen Form bestehen bleiben. Bedenken Sie außerdem, daß wir immer noch über das Erstellen eines Device-Adaptors sprechen! Diese Operationen haben also nichts mit dem Gebrauch von UNI-MAN als MANAGER und/oder EDITOR usw. zu tun.



In den obersten zwei Zeilen können 40 Bytes zu einem "Data-String" zusammengestellt werden. In der Regel werden hier die Daten eingegeben, die – wie in unserem Fall – den Write-Request formen. Es kann sich jedoch genauso um eine "Mode-Umschaltung", Programmwechsel, Wiederherstellung eines Ausgangszustandes, Display Update-Befehl oder ähnliches handeln.

Ist in dem so aufgebauten Data-String ein "Counter" (Zählbyte) erforderlich, so geben Sie an der entsprechenden Stelle ein "FA" und/oder "FB" ein. Diese Counter können unten links und rechts installiert werden. Sie arbeiten völlig unabhängig voneinander. Normalerweise wird ein Byte pro Counter ausreichend sein, es sind jedoch auch Situationen denkbar, wo zwei, drei oder gar vier Bytes für den Counter nötig sind. Letzteres kann vorkommen, wenn eine Adresse verändert werden soll, die größer ist als eine Zahl, die sich in ein, zwei oder drei Bytes darstellen lässt. Im Beispiel (das in vielen Fällen zutrifft) arbeiten wir immer mit derselben Adresse, soda auf diesem Platz kein Counter nötig ist. Da hier auch sonst kein Counter gebraucht wird, nur der Data-String mit der Anzahl der Wiederholungen eingegeben werden.

Neben den Count-Bytes FA und FB stehen noch weitere vier Zähler zur Verfügung: FC, FD, FE und FF. Sie stellen jeweils einen **1 byte**-Wert dar und werden automatisch hoch- bzw. runtergezählt. Gedacht sind sie, um MIDI-Datenwerte in einem Sysex-String zu **initialisieren**. Daher stehen diese

Count-Bytes für folgende Werte:

FC: von 127 nach 1

FD: von 127 nach 0

FE: von 1 nach 127

FF: von 0 nach 127

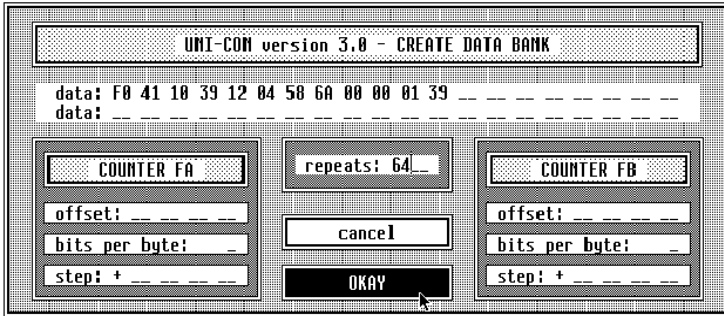
Jeder dieser Zähler kann innerhalb desselben Sysex-Strings bis zu vier Mal gebraucht werden.

Das Arbeiten **mit** einem Counter ist etwas schwieriger, aber wie gesagt können die Daten direkt aus der MIDI-Implementation (oder Bedienungsanleitung) des Gerätes übernommen werden.

Je Counter lässt sich bestimmen, in welchem Format der Zähler laufen soll. Oft ist das in "7 bit hexadecimal". Ein häufig vorkommendes Datenformat ist auch das "Nibble" (4 bits pro Byte). Das benutzte Format wird meistens in der MIDI-Implementation angegeben. Außerdem werden diese Begriffe im UNI-MAN Handbuch ausführlich erläutert (*siehe Abschnitte 6.3 und 6.5*).

Die Größe der Veränderung pro Schritt wird ganz unten in ebenfalls maximal vier Bytes festgelegt. Dabei ist zu beachten, da der hier eingestellte Wert im Zahlenbereich des gewählten Datenformats bleibt. Nun bleibt noch für jeden Zähler die Polarität (+ oder -) zu bestimmen. Damit wird festgelegt, ob der Counter auf- oder abwärts zählt. Ein Mausklick auf das Zeichen schaltet die Polarität um.

Ganz in der Mitte des Bildschirms wird schließlich eingegeben, wie oft der String hintereinander erzeugt werden soll. Meistens stimmt der Wert mit der Anzahl Sounds/Patches überein; so auch hier im Beispiel.



Wenn erst eine Bank geladen worden ist, wird die erzeugte Dummy-Bank neben der bereits vorhandenen Bank angezeigt. Im Hauptbildschirm erscheinen dann zwei Files, die auf die bekannte Weise aneinander "gehängt" werden. Im anderen Falle befindet sich nur die gerade zusammengestellte Bank im Speicher und wird als "CREATED ....." angezeigt. In dieser Situation ist es erforderlich, in einem zweiten Durchgang die zwei Bänke zu einer Init-Bank zusammenzufügen. Es ist natürlich auch möglich, die System Exclusive-Bank zu laden, **nachdem** die Dummy-Bank erzeugt wurde. Achten Sie dann auf die richtige Reihenfolge (*siehe hierzu Abschnitt 76*).

**ANMERKUNG:** Im Gegensatz zu früheren Versionen ist es ab UNI-CON 3.0 möglich, auch mit "Byte-Strings", die kürzer als 17 Bytes sind, zu arbeiten.

UNI-MAN sorgt dafür, da beim Abspeichern einzelner Sounds auf Diskette der zugehörige Write-Request mitgeschrieben wird. Das kann im Zusammenhang mit UNI-DUMP von Bedeutung sein. Auf diese Weise lassen sich Sounds/Patches für die Ablage auf bestimmten Plätzen im MIDI-Gerät vorbereiten. Ist das nicht erwünscht, gibt es noch einen anderen Weg (*s.u. zur Behandlung von Banks*).

Um zu vermeiden, da die Sound/Patch-Adressen der Dummy-Bank überschrieben werden, ignoriert UNI-MAN diese Daten beim Laden einzelner Sounds. Dadurch bleibt die Adressierung durch UNI-MAN stets korrekt.

Als zusätzliches Beispiel betrachten wir die Erstellung einer Dummy-Bank mit Write-Requests für den ROLAND RA-50 (*siehe Abbildung "WRITE PATCHES RA-50"*). Dabei handelt es sich um einen Realtime-Arranger, der mit einer umfangreichen MIDI-Implementation versehen ist. Die Variable "FA" wird hier eingesetzt, um die Patch-Nummer hochzuzählen, während "FB" die Checksum über den Request darstellt.

Da die Patch-Nummer immer um "1" erhöht wird, muß die Checksum jedesmal um "1" vermindert werden, um korrekt zu sein. Der Checksum-Wert "3F" ist das von 128 subtrahierte Zweierkomplement der Summe über die Adresse "40 01 00" und der Patch-Nummer-Variablen "FA". Diese Daten können direkt der Anleitung des RA-50 entnommen werden.

Bei einigen Geräten können Probleme auftauchen, wenn von UNI-DUMP Daten mit einer **angehängten** Dummy-Bank gesendet werden. Da UNI-MAN alle Daten, die sich auf eine Bank beziehen, in einem File ablegt, kann es vorkommen, daß die Daten nicht in der (für das Senden einer Bank) richtigen Reihenfolge stehen. Zwar sendet UNI-DUMP keine Daten, die nicht dem System Exclusive-Format entsprechen, aber es **kann** dennoch sein, daß die Write-Requests einer Bank verkehrt behandelt werden, da einige MIDI-Geräte dazu neigen, alle Sound/Patch-Speicherplätze mit demselben (zuletzt gesendeten) Sound/Patch zu füllen. Sollte dieses Problem auftreten, gibt es verschiedene Lösungsmöglichkeiten:

**Lösung 1:** Laden Sie den Device-Adaptor (mit der zugehörigen Init-Bank) in UNI-LINK. Dieses Programm entfernt alle Dummy-Banks von einer SysEx-Bank. Die dann abgelegte Init-Bank läßt sich problemlos (mit UNI-DUMP) abschicken.

**Lösung 2:** Erstellen Sie einen Device-Adaptor, der soundweise arbeitet. Dabei kommt hinter jeden Sound/Patch ein Write-Request. Eine derartige Bank wird auf jeden Fall korrekt übermittelt und angenommen. Auch diese Bank muß nach der oben beschriebenen Methode nur einmalig als Init-Bank angelegt werden (siehe EX8000 und SPX900/1000). Dies ist eine ausgesprochen solide Lösung.

**Lösung 3:** Schreiben Sie die Bank von UNI-MAN aus auf Diskette und löschen Sie dann die Dummy-Bank mit Hilfe von UNI-CON.

UNI-CON version 3.0 - CREATE DATA BANK

data: F0 41 1F 2D 11 40 01 00 FA FB F7 \_ \_ \_ \_ \_ \_ \_ \_  
data: \_ \_ \_ \_ \_ \_ \_ \_

COUNTER FA	repeats: 128_	COUNTER FB
offset: 00 _ _ _ _	cancel	offset: 3F _ _ _ _
bits per byte: 7	OKAY	bits per byte: 7
step: + 01 _ _ _ _		step: - 01 _ _ _ _

## DIVERSE UPDATES 2

### Individual Parameter Handling

Eine gewichtige Erweiterung versetzt UNI-MAN nun in die Lage, nicht nur Sound/Patch- bzw. bankweise, sondern auch **per Parameter** operieren zu können. Auch Geräte, die ausschließlich durch Exclusive-Strings pro Parameter editierbar sind, lassen sich also ab jetzt mit UNI-MAN bearbeiten. Ebenso werden damit Realtime-Prozesse möglich.

### 'JUMP TO' Fixed Parameter

Auf vielfachen Wunsch wurde im Editor eine Funktion zur Erhöhung des Bedienungskomforts implementiert: Markierung eines beliebigen Parameters mit der Möglichkeit, diese Marke direkt anzuspringen. Häufig erfordert die Editierung bestimmter Parameter den Rückgriff auf andere Werte. Stehen diese Parameter nicht nahe beieinander im Editor, so kostet es Zeit und Mühe, zwischen den beiden hin- und herzuspringen. Das ist durch die JUMP TO MARKER-Funktion nun nicht mehr der Fall.

Beispiel: Ein ROLAND D50 Klang besteht aus COMMON-Daten und zwei PARTIALS für UPPER und LOWER. Um einen Partial ungestört zu hören, muß der andere zeitweilig ausgeschaltet werden. Mit der Partial-Mute Funktion (COMMON-Parameter) lassen sich Partials ein- und ausschalten. In diesem Fall ist es also wünschenswert, den Parameter Partial-Mute auf einem JUMP TO-Button zu haben, sodaß er dann direkt in der mittleren Zeile erscheint. Denn dann erst ist es möglich, direkt nach dem Sprung die gewünschte Mute-Einstellung mit der Maus oder den [+] und [-] Tasten zu wählen und anschließend wieder zurückzuspringen.

Indem man mit der **rechten** Maustaste das "JUMP TO" Feld anklickt, wird der gerade in der mittleren Zeile befindliche Parameter markiert. Zu genau diesem Parameter springt UNI-MAN, wenn später mit der **linken** Maustaste auf das "JUMP TO" Feld geklickt wird. Sie werden feststellen, daß diese Funktion insbesondere bei intensivem Gebrauch des Editors sehr nützlich ist.

### Anpassung der 'EDIT-FORMULAS'

Im Formel-Editor (*siehe Abschnitt 6.5 'EDIT FORMULAS' im Handbuch*) besteht nun eine zusätzliche Möglichkeit, (rechnerische) Zusammenhänge zwischen verschiedenen Parametern herzustellen. Ab der Version 3.0d können Sie nämlich in der Formel auch auf in der Converting-Table bereits bearbeitete Parameter zurückgreifen. Das geschieht ganz einfach durch Eingabe eines negativen Wertes hinter 'READ BYTE'. Der damit bestimmte Parameter kann anschließend wie gewohnt weiter verrechnet werden. Ebenso ist es möglich, mit [SHIFT] und 'READ BYTE' den gerade gefundenen Wert zum aktuellen Wert zu addieren. In der Formel erscheint dann z.B.: +Rb-22 (Addition des Wertes, der 22 Bytes zurück gefunden wird).