

1 Einleitung

1.1 Kurzbeschreibung des Produkts

TOM-REF ist ein Data Dictionary System, das Daten über Anwendungen im BS2000 speichert und verfügbar hält. Es wird bevorzugt im Bereich kommerzieller Software eingesetzt. Wegen seiner guten Verwendbarkeit in allen Phasen des Entwicklungsprozesses dient es als Datenbasis für die Planung, die Erstellung, den Betrieb, die Wartung und Pflege von Software.

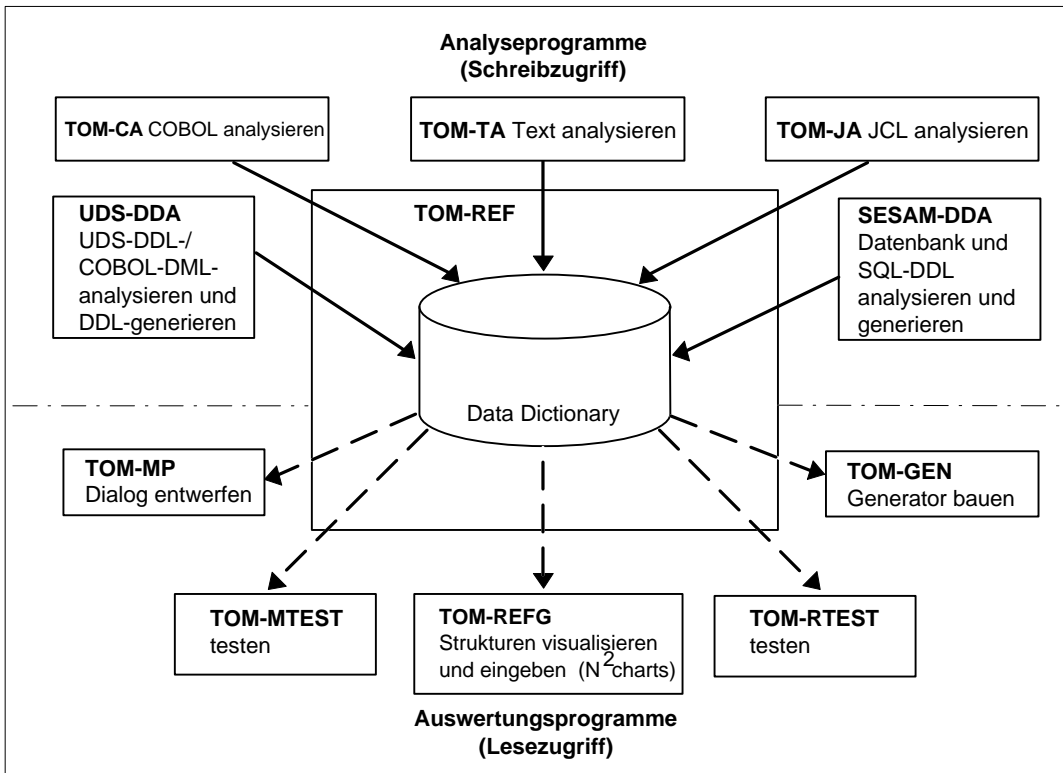


Bild 1 Funktionsverbund mit dem Data-Dictionary-System TOM-REF

TOM-REF dokumentiert die Systemstruktur einer Anwendung durch die Beschreibung einzelner Objekte (Programme, Module, Daten usw.) und durch Beziehungen, die zwischen diesen Objekten bestehen (Objekt ruft anderes auf, Objekt enthält anderes Objekt).

Die wesentlichen Leistungen von TOM-REF umfassen die einheitliche Beschreibung der Objekte (insbesondere soweit sie Daten repräsentieren) und die Bereitstellung von Verwendungsnachweisen (z.B. welche Module eine bestimmte Datendefinition benutzen).

Der Standard-Strukturkatalog von TOM-REF sieht Objekte mit Beziehungen aus nahezu dem gesamten Anwendungsspektrum des BS2000 vor. Für den eigenen Bedarf können Sie zusätzlich Objekte und Beziehungen mit fest umgrenztem Attributvorrat anlegen.

Über Menüs können im Dialog alle Funktionen von TOM-REF aufgerufen werden. Diverse Funktionen für Masken-Ein/Ausgaben und Druckausgaben, sowie Servicefunktionen sind integriert.

Die Eingaben zu TOM-REF erfolgen entweder manuell im Dialog oder durch maschinelle Auswertung von Texten, Prozeduren, Quellprogrammen, Formatdefinitionen und Datenbanken über entsprechende Analyse-Programme, z.B. TOM-CA oder TOM-JA.

Für den besseren Überblick über die im Data Dictionary gespeicherten Daten sorgt das Zusatzprodukt TOM-REFG, indem es Strukturen in Form von N²-Charts sichtbar macht. Über diese komfortable Oberfläche sind auch Eingaben möglich.

TOM-REF ist einerseits als persönliches Werkzeug für den einzelnen Entwickler ausgelegt. Andererseits kann mit TOM-REF zusätzlich ein zentrales Data Dictionary installiert werden und mit dem Entwickler-Data Dictionary im Verbund eingesetzt werden. Damit wird die Konsistenz der Einträge im Data Dictionary innerhalb eines Entwicklungsteams sichergestellt. Auf ein zentrales Dictionary kann lesend zugegriffen werden, um dortige verbindliche Datendefinitionen zu übernehmen und Doppeldefinitionen (Objekte gleichen Namens und Objekttyps) innerhalb eines Softwaresystems zu vermeiden.

Als Bestandteil einer Software-Produktionsumgebung erfüllt TOM-REF im Verbund mit anderen Werkzeugen die wichtige Aufgabe, alle anfallenden Daten zu speichern und zu verwalten und somit in den Phasen des Software-Entwicklungsprozesses für alle Tools verfügbar zu machen.

Damit ist TOM-REF sowohl für den Entwurf als auch für die Entwicklung, den Betrieb, Wartung und Pflege von Anwendungssoftware einsetzbar.

1.1.1 Strukturkatalog

Der Standard umfaßt Teilstrukturkataloge für Kommandosprache (JCL), Programm-Design und -Realisierung, Teilhaberbetrieb UTM, Datenhaltung DVS/LEASY, Datenbanken SESAM/SQL und UDS/SQL und den Funktionsentwurf. Die Objekttypen enthalten typneutrale Attribute wie Name, Erstellungsdatum, Änderungsdatum, Verantwortlicher, beschreibender Text. Letzteres Attribut kann erheblich ausgedehnt werden. Bibliothekselemente mit beliebiger Text- bzw. Informationsmenge sind anschließbar. Außerdem enthalten Objekttypen jeweils spezifische Attribute. Beziehungstypen enthalten ebenfalls typneutrale und spezifische Attribute.

Hilfstypen für mehrstellige Beziehungen und Synonyme stehen zur Verfügung.

Benutzereigene Erweiterungen sind möglich. Neue Objekttypen mit fest umgrenztem spezifischen Attributvorrat (4 Attribute mit je 32 Byte Länge) sind anlegbar, Beziehungstypen ebenfalls, allerdings ohne neue, über den Standard hinausgehende Attribute.

Beschreibbar sind somit Prozesse (z.B. System, Programm, Modul, Entry) und Daten (z.B. Datenbank, Datei, Satz, Maske, Feldgruppe, Feld).

1.1.2 Datenhaltung

Die Daten werden in einer TOM-Datenbasis gehalten. Sie wird nach den TOM-Namenskonventionen benannt und besteht aus:

- Der Bibliothek (PLAM-Bibliothek) mit den einzelnen Elementen, die unverbunden abgespeichert werden (Texte, Quellprogrammcode, Prozeduren usw.), und den Lang-Texten zu den Objekten im Data Dictionary.
- Dem Data Dictionary (ISAM-Dateien) mit der Darstellung der Systemstruktur, die sich u.a. aus den Metadaten der Bibliothekselemente zusammensetzt und den systemweit verwendeten Datendefinitionen. Die Speicherungsformen sind Objektbeschreibungen mit ihren Beziehungen sowie den inversen Beziehungen gemäß dem Strukturkatalog.

Servicefunktionen dienen der Initialisierung, Reorganisation und Sicherung der Datenhaltung.

Eine Import- oder Incorporate-Funktion erlaubt die Zusammenführung von Data Dictionary-Datenbeständen.

1.1.3 Ein-/Ausgabe

Die direkte Ein-/Ausgabe steuern Sie über Bildschirmmasken. Zusätzlich können Sie mit Analyse-Programmen Anwendungen im Data Dictionary ablegen. Folgende Zusatzprodukte sind erhältlich:

- TOM-TA für die Textanalyse
- TOM-JA für die Analyse von BS2000-Dialog- und BS2000-Enter-Prozeduren
- TOM-CA für die Analyse von COBOL/COLUMBUS-COBOL-Programmen
- UDS-DDA für die Analyse von UDS-DDL (Data Definition Language) und UDS-COBOL-DML (Data Manipulation Language)
- SESAM-DDA für die Analyse von und den Vergleich mit einer SESAM/SQL-Datenbank, sowie die Analyse von SQL-Quelldateien für Datenbankaufbau und -änderung

Sie können Verwendungsnachweise aller Art (z.B. Verwendung von Modulen durch Module, von Daten durch Module, von Daten durch Daten) erstellen und am Bildschirm ausgeben oder drucken.

1.1.4 Unterstützung der Software-Entwicklung

Ein Data Dictionary können Sie zentral und privat im Verbund betreiben. Durch die Integrationsmöglichkeit in die TOM-Umgebung und Steuerung über TomDoors-M/-TI können Sie TOM-REF in eine Software-Produktionsumgebung einbetten.

Die einzelnen Phasen im Software-Lebenszyklus werden zusammen mit weiteren Produkten unterstützt:

- Der Funktionsentwurf wird mit eigenen Objekttypen unterstützt (Funktionshierarchie, Kommunikationsstrukturen).
- Für den Entwurf und die Beschreibung des Dialogs bietet TOM-REF ebenfalls Standard-Objekttypen an. Die Abfolge der Maskenformate ist dokumentierbar, einschließlich der Übergangskriterien von einem Format zum anderen und zurück, z.B. Feldinhalte und Funktionstasten. Das Maskenprototyping TOM-MP benutzt diese Daten, um Sequenzen von Maskenformaten mit Testdaten zu präsentieren.
- Wird der Datenentwurf mit TOM-REF durchgeführt, kann aufbauend darauf mit Hilfe eines entsprechenden Zusatzprodukts (UDS-DDA oder SESAM-DDA) die Datenbank aus den Datenbankdaten in TOM-REF generiert werden.

- TOM-REF enthält einen COPY-Strecken-Generator. Er erzeugt aus den Data Dictionary-Einträgen Daten-Definitionen (auch mit Initialisierung) auf Satzebene und für Datenübertragungen (Move-Statements) zwischen zwei Datensätzen.
- Auf der TOM-REF-Datenhaltung setzen mit TOM-MTEST (Testrahmen für Modultest) und SSGEN (Schnittstellengenerator der TOM-TI-Toolbörse) weitere Generatoren der TOM-Familie auf.
- Mit TOM-GEN lassen sich für den eigenen Bedarf auf einfache Weise zusätzliche Generatoren erstellen.

1.2 Zielsetzung und Zielgruppe des Handbuchs

Das Handbuch TOM-REF V3.0 richtet sich an Software-Entwickler, die im BS2000 arbeiten und Daten über ihre Software-Projekte sinnvoll verwalten wollen.

Dieses Handbuch setzt Kenntnisse von TomDoors-M voraus. Für die Nutzung von TOM-REF sind - abhängig von der konkreten Aufgabenstellung - Kenntnisse der BS2000-Kommandosprache, von COBOL, Columbus-COBOL, SESAM/SQL V2, UDS oder UTM erforderlich.

1.3 Konzept des Handbuchs

Obwohl TOM-REF auch unabhängig von TomDoors-M aus der BS2000-Kommandoebene aufgerufen werden kann, beschränkt sich die Beschreibung in diesem Handbuch auf TOM-REF als `Special function` in TomDoors-M. Notwendige Änderungen und Anpassungen sind leicht erkennbar, falls TOM-REF als BS2000-Prozedur aufgerufen wird. Die Bedienung von TOM-REF selbst ist einfach und erfolgt überwiegend maskengesteuert.

Um Ihnen den Umgang mit TOM-REF zu erleichtern, finden Sie zu Beginn des Handbuchs eine Einführung in die Begriffswelt und die Funktionsweise von TOM-REF und den Einsatz von TOM-REF in den Projektphasen. Daran schließt sich in Kapitel 4 eine Beschreibung der Grundfunktionen an zusammen mit einer Erläuterung, wie die Bildschirmmasken bedient werden. Die für die tägliche Arbeit mit TOM-REF wichtigen Service-Funktionen sind in Kapitel 5 beschrieben.

Eher zum Nachschlagen dient der zweite Teil dieses Handbuchs. In den Kapiteln 6 bis 10 sind die einzelnen produktspezifischen Informationsmodelle, ihre Objekte und die möglichen Beziehungen zusammengefaßt.

Kapitel 11 beschreibt einige Sonderfälle der Datenmodellierung. Kapitel 12 enthält die Beschreibung der TOM-REF-Kommandos. In Kapitel 13 sind die Meldungen von TOM-REF mit der jeweiligen Bedeutung aufgeführt.

Am Ende des Handbuchs finden Sie eine Tabelle aller TOM-REF-Objekttypen und ihrer Texttypen und folgende Verzeichnisse:

- Glossar
- Abkürzungsverzeichnis
- Stichwortverzeichnis

1.4 Änderungen gegenüber dem Handbuch TOM-REF V2.2

SESAM V1, SESAM V14 und älter sowie der Anschluß zu DRIVE/QUERY werden nicht mehr unterstützt.

Wegen der Umstellung von SESAM/SQL-Server auf dem SQL2-Standard wurde der Teilstrukturkatalog zu SESAM/SQL V2 völlig neu implementiert. Die Teilstrukturkataloge zu SESAM und SQL entfallen. Eine direkte Verbindung zwischen den Programmen TOM-REF und SESAM/SQL-DDA besteht nicht mehr.

Dadurch bot sich eine Umstrukturierung des Handbuchs an, in einen einführenden beschreibenden Teil (Kapitel 1-5) und einen Nachschlageteil (Kapitel 6-13).

Die Informationsmodelle zu DVS-Leasy, SESAM/SQL V2, Process/JCL, UTM und UDS sind jetzt jeweils in einem eigenen Kapitel beschrieben. Damit wurde die Beschreibung der typspezifischen Objektmasken, ergänzt um eine Übersicht der Objekte und Beziehungen sowie die jeweiligen Teilstrukturkataloge thematisch geordnet.

Die Kommandobeschreibung wurde aus den einzelnen Kapiteln herausgelöst und ebenfalls in einem eigenen Kapitel zusammengefaßt.

Ergänzt wurde das Handbuch durch ein ausführliches Glossar und ein Abkürzungsverzeichnis.

1.5 Darstellungsmittel

Für die Darstellung der Kommandos gelten die folgenden Syntaxregeln:

GROSSBUCHSTABEN	bezeichnen Schlüsselwörter.
GROSSBUCHSTABEN	in halbfett stehen für die garantierte bzw. vorgeschlagene Abkürzung eines Kommandos oder eines Operanden.
<u>Unterstreichung</u>	gibt den von TOM-REF verwendeten Standardwert eines Operanden an.
kleinbuchstaben	bezeichnen Variable, die durch aktuelle Angaben ersetzt werden müssen.
{ }	Geschweifte Klammern schließen verschiedene Möglichkeiten ein, von denen eine ausgewählt werden muß.
[]	Eckige Klammern schließen Wahlangaben ein; d.h. der gesamte Inhalt kann weggelassen werden.
dicktengleich	im Fließtext bezeichnet Bildschirmzitate und Eigennamen von TOM-REF, z.B. PFL.
Rasterung	beispielhafte Eingaben in Auswahlmasken
Siehe auch	am Ende der Kommandobeschreibung gibt Hinweise auf verwandte Kommandos.

Zusätzlich werden folgende Piktogramme verwendet:



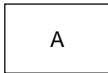
für wichtige Hinweise und weiterführende Informationen.

► wenn Sie etwas tun sollen.

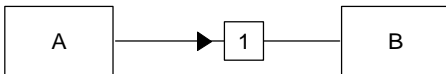
Notation der Entity-Relationship-Diagramme

Der Strukturkatalog des Data Dictionary wird mit Entity-Relationship-Diagrammen dargestellt. Die verwendete Notation ist eng angelehnt an den Vorschlag, den James Martin und Carma Mc Clure in ihrem Buch "Diagramming Techniques for Analysts and Programmers" gemacht haben.

Basis-Symbole



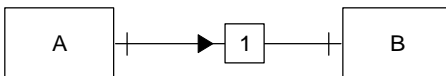
Ein rechteckiger Kasten stellt einen **Objekttyp** dar, hier in diesem Beispiel den Objekttyp A.



Eine von einem Objekttyp ausgehende Linie mit einem Pfeil stellt einen **Beziehungstyp** dar, in diesem Beispiel der Beziehungstyp 'A nach B'. Ein Konnektor, hier z.B. 1, verweist auf die Liste der Elementaraussagen. In dieser Liste wird auch der inverse Beziehungstyp 'B nach A' beschrieben.

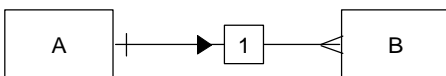
Wird ein Objekt vom Typ 'A' in das Data Dictionary eingetragen, kann bei diesem Objekt auch die Beziehung vom Typ 'A nach B' eingetragen werden. Die inverse Beziehung vom Typ 'B nach A' wird im Data Dictionary nicht eingetragen, sondern auf Anforderung von der Beziehung des Typs 'A nach B' abgeleitet.

Wertigkeiten von Beziehungstypen



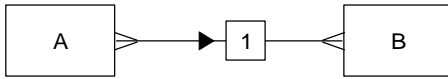
Der Beziehungstyp mit der **Wertigkeit 1:1** wird durch einen Strich senkrecht zur Beziehungslinie an beiden Enden angegeben.

Das Beispiel sagt aus, daß von einem Objekt vom Typ A jeweils eine Beziehung zu einem Objekt vom Typ B besteht und umgekehrt.



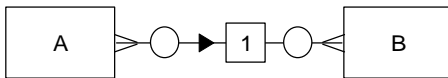
Der Beziehungstyp mit der **Wertigkeit 1:n** wird durch einen Strich senkrecht zur Beziehungslinie an einem Ende und einer Verzweigung zu drei Linien am anderen Ende angegeben.

Das Beispiel sagt aus, daß von einem Objekt vom Typ A jeweils eine Beziehung zu einem oder mehreren Objekten vom Typ B besteht und von einem Objekt des Typs B zu jeweils einem Objekt des Typs A.



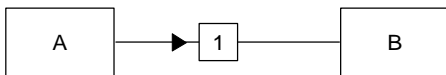
Der Beziehungstyp mit der **Wertigkeit n:m** wird durch eine Verzweigung zu drei Linien an beiden Enden angegeben.

Das Beispiel sagt aus, daß von einem Objekt vom Typ A jeweils eine Beziehung zu einem oder mehreren Objekten vom Typ B besteht und von einem Objekt des Typs B zu jeweils einem oder mehreren Objekten des Typs A.

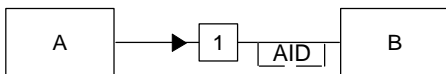


Kann in der Wertigkeit auch der **Wert Null** auftreten, so wird zusätzlich zu den bereits erläuterten Symbolen ein kleiner Kreis eingezeichnet. Das Beispiel sagt aus: Von einem Objekt des Typs A besteht eine Beziehung zu keinem, einem oder mehreren Objekten des Typs B. Von einem Objekt des Typs B besteht eine Beziehung zu keinem oder einem Objekt des Typs A.

Arten von Beziehungstypen

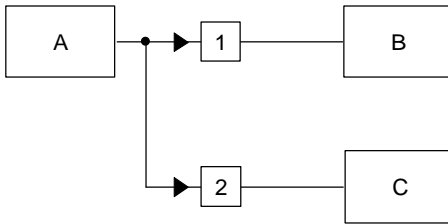


Dieser bereits beschriebene Beziehungstyp ist ein **binärer Beziehungstyp**.

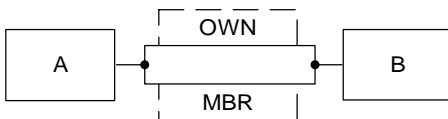


Gehören zu einem Beziehungstyp nicht nur Typ-neutrale Attributtypen, sondern auch **Typ-spezifische** und werden diese **mit einem Kurznamen** versehen, wird dieser **Beziehungstyp** durch eine unterbrochene Linie dargestellt, die mit der Linie verbunden ist, die den Beziehungstyp angibt. Der Kurzname wird dort eingetragen.

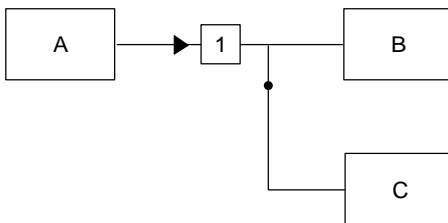
Im Beispiel gehören zum Beziehungstyp 'A nach B' Typ-spezifische Attributtypen, die durch 'AID' gekennzeichnet sind.



Beziehungstypen, die sich gegenseitig **ausschließen**, werden so dargestellt, daß die Abzweigungsstelle der Beziehungslinien mit einem dicken Punkt markiert wird. Im Beispiel sind die Beziehungstypen 'A nach B' oder ausschließlich 'A nach C' dargestellt.

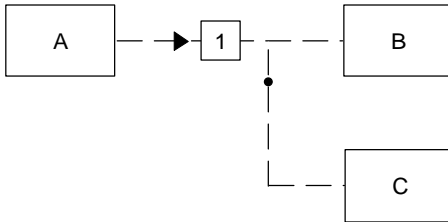


Ein solcher Beziehungstyp kann auch zwischen Objekttypen bestehen und zwar dann, wenn **Typ-spezifische Attributtypen** sich gegenseitig **ausschließen**. Im Beispiel: 'A nach B als Owner (OWN)' oder 'A nach B als Member (MBR)'.



Hat bei den sich ausschließenden Beziehungstypen **ein Beziehungstyp Vorrang** vor dem anderen, so wird der dicke Punkt an die Abzweigung zu dem Objekttyp gezeichnet, der nachrangig ist.

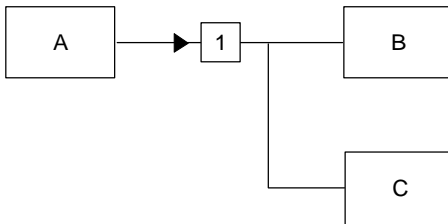
In dem Beispiel hat der Beziehungstyp 'A nach B' Vorrang vor dem Beziehungstyp 'A nach C', d.h. für den konkreten Fall, wenn eine Beziehung von A nach B aufgebaut werden kann, dann wird keine Beziehung von A nach C aufgebaut.



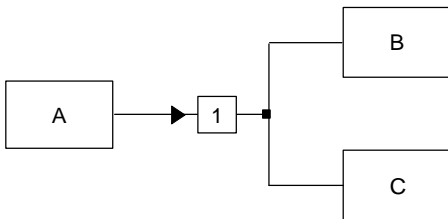
Soll ein Beziehungstyp so sein, daß die **Beziehung jeweils über den Namen** hergestellt wird und nicht explizit als Eintrag im Data Dictionary, wird die Beziehungslinie unterbrochen dargestellt. Hierfür ist allerdings eine bestimmte Namenskonvention erforderlich: Zielobjekt(Ausgangsobjekt)Qualifizierung.

Im Beispiel wird die Beziehung 'A vorrangig nach B' oder 'A nach C' aus dem Namens-eintrag des Objekts A abgeleitet.

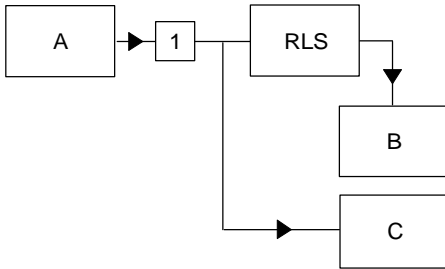
Ein konkreter Fall: Es existiere ein Objekt B mit Namen "Rose" und ein Objekt C mit Namen "Rose". Die Namenskonvention für ein Objekt A lautet: "Rose(Nelke)5". Die Beziehung zum Objekt vom Typ B wird aufgebaut, da vorrangig.



Beziehungstypen, die sich gegenseitig einschließen, werden so dargestellt, daß die Beziehungslinien sich an einer Stelle verzweigen. Im Beispiel sind die Beziehungstypen 'A nach B' und einschließlic 'A nach C' dargestellt. Demnach gilt, daß 'A nach B' immer vorhanden sein muß, wenn 'A nach C' existiert und umgekehrt.



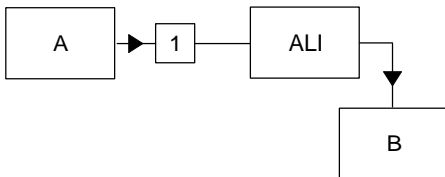
Die **Kombination** von sich ausschließenden mit sich einschließenden Beziehungstypen wird so dargestellt, daß statt des Punktes ein schwarzes Viereck eingezeichnet wird. Im Beispiel sind die Beziehungstypen 'A nach B und/oder C' und 'A nach C und/ oder B' dargestellt.



Mehrstellige Beziehungstypen, d.h. Beziehungstypen, bei denen mehr als zwei Objekttypen in einer Beziehung beteiligt sind, benutzen den Hilfsobjekttyp "RLS". Zwar besteht hier die Beziehung zwischen zwei Objekttypen, aber ein dritter oder weitere sind ebenfalls beteiligt.

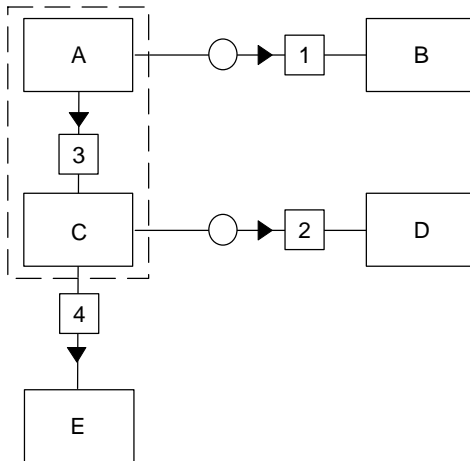
Im Beispiel ist der Beziehungstyp 'A nach B mit C' dargestellt.

Wenn die Wertigkeit der Ziel-Objekttypen Null sein kann, ist dies immer so zu interpretieren, daß mindestens drei Objekte vorhanden sein müssen oder keine. Dies gilt auch, wenn mehr als drei Objekttypen beteiligt sind.



Beziehungstypen, bei denen **Synonyme** auftreten, benutzen den Hilfsobjekttyp "ALI". Im Beispiel ist ausgedrückt, daß ein Beziehungstyp 'A nach B' besteht, bei dem Objekte vom Typ B untereinander als Synonyme auftreten. Im einfachsten Fall kann auch im Objekttyp ALI ein synonymer Name zu B gespeichert sein.

Besonderheiten der Darstellung



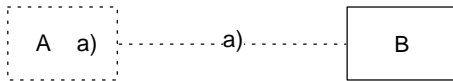
Gleichartige Beziehungstypen werden zusammengefaßt dargestellt, um die Übersichtlichkeit zu erhöhen und möglichst wenig sich kreuzende Linien zu erzeugen. Gleichartig sind Beziehungstypen dann, wenn sie die gleiche Bedeutung haben (z.B. enthält/ ist enthalten) und die gleichen Wertigkeiten.

Die betroffenen Objekttypen werden mit einer unterbrochenen Linie umschlossen. Die Linie mit dem Pfeil, die den Beziehungstyp darstellt, geht in diesem Fall von der Zusammenfassung aus, bzw. endet an der Zusammenfassung. Zusammenfassungen können auch geschachtelt auftreten.

Solche Beziehungstypen schließen sich gegenseitig aus und gleichzeitig ein (und/oder). Im Beispiel werden die Objekttypen A und C zusammengefaßt, weil sie gleichartige Beziehungstypen zu B haben. Die Beziehungstypen sind (1) 'A nach B' und/oder 'C nach B' und (2) 'A nach D' und/oder 'C nach D'.

Daneben können weitere Beziehungstypen bestehen, die an der Zusammenfassung nicht beteiligt sind:

- Innerhalb der Zusammenfassung sind Beziehungstypen möglich, in dem Beispiel der Beziehungstyp (3) 'A nach C'.
- Ein Beziehungstyp, der nur zu einzelnen Objekttypen der Zusammenfassung gehört, ist in dem Beispiel der Beziehungstyp (4) 'C nach E'.



a) siehe JCL-Teilstrukturkatalog

Objekttypen und Beziehungstypen die in einem **anderen Teilstrukturkatalog** enthalten sind, werden gepunktet eingezeichnet und sind mit einem Querverweis versehen. Auf diese Weise werden die Übergänge von einem Teilstrukturkatalog zu einem anderen dargestellt. Das Beispiel zeigt den Objekttyp B zu dem vom Objekttyp A ausgehend der Beziehungstyp 'A nach B' besteht. Objekttyp A und Beziehungstyp 'A nach B' liegen in einem anderen Teilstrukturkatalog, in dem Beispiel im JCL-Teilstrukturkatalog.

2 Data Dictionary: Bedeutung, Nutzen und Einsatz

Nachdem heute immer deutlicher wird, daß ein Data Dictionary für die rationelle Software-Entwicklung, den Betrieb und die Wartung unbedingt notwendig ist, bleiben doch oft die Fragen im Raum stehen, was denn ein Data Dictionary eigentlich ist und wie der Nutzen aussieht, den man sich von ihm verspricht. Diese Einführung möchte, ausgehend von einem konkreten Fall, diese Fragen beantworten.

2.1 Begriffsklärung

Einfach übersetzt bedeutet Dictionary soviel wie Wörterbuch und Data Dictionary wäre somit ein Daten-Wörterbuch, ein Nachschlagewerk, in dem man alle Daten findet. Da nun ein solches Data Dictionary bei der elektronischen Datenverarbeitung (EDV) verwendet wird, sind dort insbesondere die Daten gespeichert, die in der EDV verarbeitet werden.

Doch hier gilt es ein Mißverständnis auszuräumen. Nicht alle Daten sind im Data Dictionary gespeichert. Die Daten, die verarbeitet werden, sind nach wie vor in Datenbanken und Dateien abgelegt, wie z.B. eine Kunden-Anschrift, die aus Name, Wohnort und Straße besteht. Im Data Dictionary aber ist gespeichert, daß Daten dieser Art und Struktur bei der Verarbeitung benutzt werden. Im genannten Beispiel wäre dies die Tatsache, daß es eine Datenbank gibt, in der Sätze vorhanden sind, die den Kunden-Namen, seinen Wohnort und seine Straße enthalten. Genauer müßte man also sagen, daß ein Data Dictionary nicht die Daten enthält, die von der EDV verarbeitet werden, sondern die Informationen, wie diese Daten strukturiert sind (Meta-Daten). Bild 2 soll dies verdeutlichen.

In einem Data Dictionary sind allerdings nicht alleine die eine Datenbank oder Datei beschreibenden Daten enthalten, sondern prinzipiell alle Daten, die ein Anwendungssystem als Ganzes betreffen, also auch Beschreibungen von Modulen, Masken usw..

Das Data Dictionary läßt sich auch in anderen Bereichen einsetzen. Jede Art von Daten, seien es solche, die eine beliebige Struktur in einem Unternehmen oder die Struktur beliebiger Informationen beschreiben, läßt sich dort ablegen.

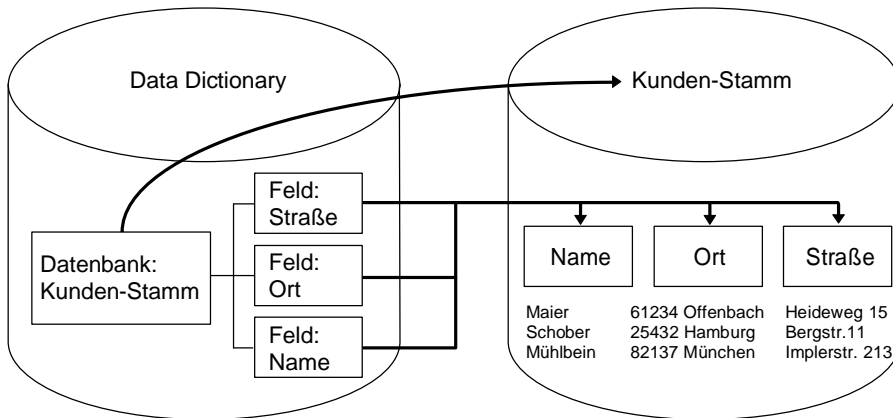


Bild 2 Beziehung zwischen Meta-Daten im Data Dictionary und Anwendungsdaten

2.2 Nutzen eines Data Dictionary

Ganz allgemein hat ein Data Dictionary die Aufgabe, zur Übersichtlichkeit beizutragen und zwar dann, wenn DV-Systeme einen Komplexitätsgrad erreicht haben, der mit anderen Mitteln kaum noch durchschaut werden kann. An einem zum Zweck der Demonstration bewußt einfach gehaltenen Beispiel soll dies klar werden.

2.2.1 Problemstellung: Komplexität der Systeme

Das Anwendungssystem 'Kunden-Bestellung' hat die Aufgabe, eingehende Kundenbestellungen zu speichern und für die Lieferung der Artikel Rechnungen zu stellen. Dabei soll der Artikelbestand jeweils überprüft bzw. um die zu liefernden Artikel vermindert werden. Kunden, die das erste Mal bestellen, sollen in den Kundenbestand aufgenommen werden.

Das Anwendungssystem 'Kunden-Bestellung' besteht aus zwei Programmen, die wiederum aus vier Modulen gebildet werden. Jedes dieser Module gibt eine Maske aus. In diesen Modulen werden zwei Dateien verarbeitet, die eine enthält 3 Felder, die andere 4. In einem Diagramm dargestellt, ergibt sich Bild 3.

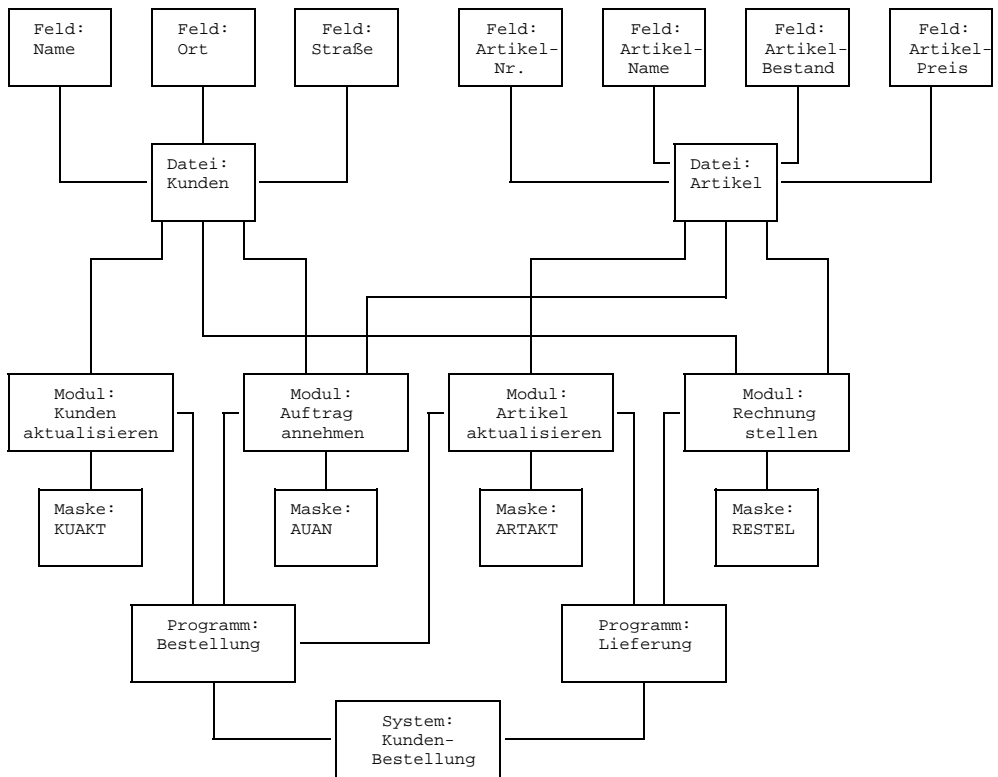


Bild 3 Diagramm des Anwendungssystems 'Kundenbestellung'

Wie schon dieses kleine Beispiel zeigt, wird es bereits bei einem relativ geringen Komplexitätsgrad schwierig, den Überblick zu behalten - wieviel mehr noch bei größeren Anwendungen, die in der Praxis die Regel sind.

2.2.2 Lösungsschritt 1: Systemstruktur modellieren

Für den Einsatz eines Data Dictionary ist es unerlässlich, sich zunächst einmal ein klares Bild von der gegebenen Situation zu verschaffen. Dabei ist gleichgültig, ob ein bestehendes System betrachtet werden soll oder eines, das zu entwerfen ist.

Klarheit kann man sich verschaffen, indem man ein Modell der zu beschreibenden Wirklichkeit erstellt. Für dies Beispiel könnte ein Modell so aussehen, wie es Bild 4 zeigt.

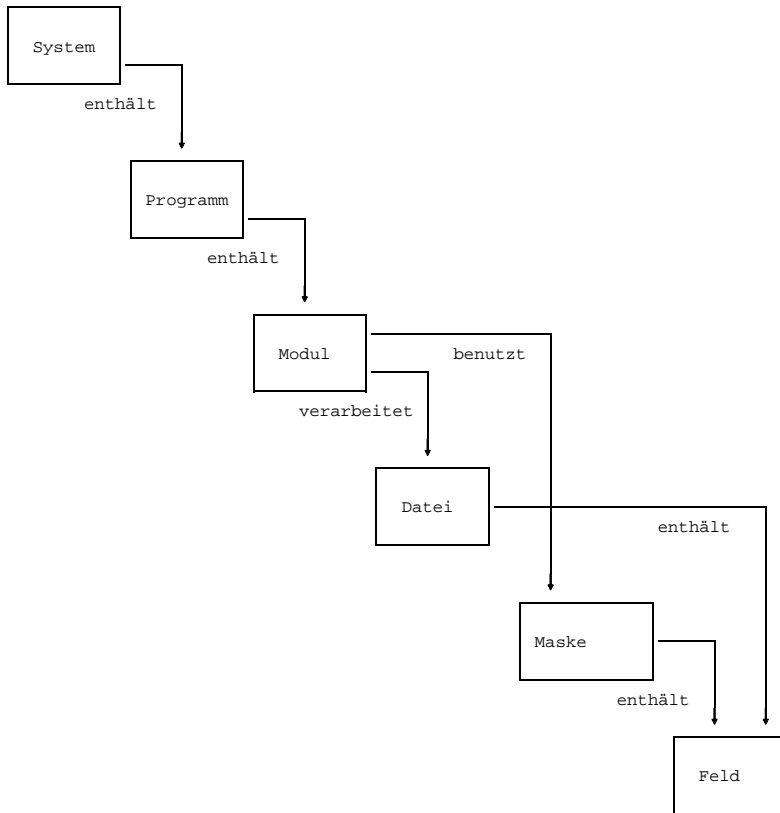


Bild 4 Modell einer einfachen Systemstruktur

Das Modell enthält von jedem Typ eines in dem Diagramm (Bild 3) vorkommenden Objekts je ein Exemplar und stellt somit eine höhere Abstraktionsstufe dar. Dies sind die Objekttypen System, Programm, Modul, Datei, Maske und Feld. Ferner enthält das Modell die Typen von Beziehungen, die zwischen den Objekten bestehen. Dies sind: System enthält Programm, Programm enthält Modul, Modul verarbeitet Datei, Modul benutzt Maske, Datei enthält Feld, Maske enthält Feld. Der zuletzt genannte Beziehungstyp Maske enthält Feld ist in dem Diagramm (Bild 3) nicht enthalten. Würde man ihn dort mit seinen Ausprägungen abbilden, ergäbe sich ein Gewirr von Verbindungen zwischen den Masken bzw. den Dateien und den Feldern. Somit sind anhand des abstrakten Modells mehr Zusammenhänge darstellbar, als dies in Bildern oder Beschreibungen möglich wäre, die alle Ausprägungen bestimmter Objekttypen enthalten.

Um das Modell zu vervollständigen, könnte man noch eintragen, welche Wertigkeiten der Beziehungstypen vorliegen, z.B. wieviele Programme zu einem System gehören können: mindestens 1, beliebig viele. Hierauf wird in diesem Beispiel verzichtet.

2.2.3 Lösungsschritt 2: Modell speichern

Soll im Data Dictionary unsere Anwendung abgespeichert werden, so muß sie die gleiche Struktur wie unser Modell haben. Lassen wir nun zunächst unberücksichtigt, wie wir das Modell in das Data Dictionary hineinbringen. Nehmen wir an, daß es bereits der Hersteller dort hinterlegt hat.

Die Strukturen des Data Dictionary nach diesem Modell zeigt Bild 5.

System					
Name	Erstel- lungs- datum	Verant- wort- licher	Beschrei- bender Text		Beziehung zu Programm

Programm					
Name	Erstel- lungs- datum	Verant- wort- licher	Beschrei- bender Text		Beziehung zu Modul

Modul					
Name	Erstel- lungs- datum	Verant- wort- licher	Beschrei- bender Text	Beziehung zu Datei	Beziehung zu Maske

Datei					
Name	Erstel- lungs- datum	Verant- wort- licher	Beschrei- bender Text		Beziehung zu Feld

Maske					
Name	Erstel- lungs- datum	Verant- wort- licher	Beschrei- bender Text		Beziehung zu Feld

Feld					
Name	Erstel- lungs- datum	Verant- wort- licher	Beschrei- bender Text		

Bild 5 Einfache Data Dictionary-Struktur

Für jeden Objekttyp des Modells wurde ein Behälter angelegt. Diese Behälter sind strukturiert, so daß einzelne Attribute der Objekte eingetragen werden können. Mit anderen Worten: Es werden nunmehr auch noch Attributtypen aufgenommen, die bisher in dem Modell nicht erwähnt wurden. Die ersten vier Attributtypen sind bei jedem Objekttyp gleich, man kann sie als typ-neutrale Attributtypen bezeichnen. Dies sind: Der Name

des Objekts, sein Erstellungsdatum, der Name des dafür Verantwortlichen, sowie ein beliebiger beschreibender Text. Zu jedem einzelnen dieser Objekttypen kann man sich weitere, typ-spezifische Attributtypen vorstellen, wie z.B. bei einem Feld seine Länge. Der Einfachheit halber sind aber hier solche Attributtypen weggelassen. Was außerdem vorgesehen ist, sind die Beziehungstypen. Sie werden in dem Beispiel in Bild 5 wie Attributtypen dargestellt.

An diesem Punkt der Überlegungen erkennen wir bereits, daß im Data Dictionary eine Struktur vorbereitet ist, die erlaubt, eine Fülle von Informationen über unser Anwendungssystem aufzunehmen.

2.2.4 Lösungsschritt 3: System speichern

Der Weg, die Daten über das Anwendungssystem in das Data Dictionary einzubringen, wird unterschiedlich sein, je nachdem wie sie vorliegen:

- als Diagramm oder in ähnlicher Form, nachdem die Designphase bereits abgeschlossen ist, aber das System noch nicht weiter spezifiziert und realisiert wurde;
- als Quellcode, erstellt zu einem DV-System, das bereits produktiv eingesetzt wird;
- als Designdaten, die bereits im Data Dictionary vorliegen, weil in der Designphase mit Hilfe des Data Dictionary gearbeitet wurde.

Nehmen wir den ersten Fall an, so haben wir zwar auf dem Papier bereits alle wichtigen Daten gesammelt, die wir nunmehr auf einmal oder Schritt für Schritt während des fortschreitenden Designs einzelner Komponenten und deren Realisierung ins Data Dictionary einbringen. Wir benutzen dazu die Menü- oder Kommandooberfläche des Data Dictionary-Systems, das unser Data Dictionary verwaltet.

Der zweite Fall betrifft ein bereits existierendes System, das nachdokumentiert werden soll, damit für Wartung und weitere Entwicklung mit dem Data Dictionary gearbeitet werden kann. Geeignete Analyse-Programme erleichtern hier das Finden der Data Dictionary-relevanten Daten und ihre Übernahme ins Data Dictionary. Man läßt sie die Quellprogramme des Anwendungssystems analysieren und erwartet, daß sie die Data Dictionary-Daten erkennen und ins Data Dictionary speichern.

Im dritten Fall sind womöglich alle Objekte bereits im Data Dictionary erfaßt, aber ggf. müssen noch zusätzliche Attribute und Beziehungen eingetragen werden. Dieser Fall tritt dann ein, wenn wir in der Designphase ein Werkzeug benutzt haben, das seine Daten im Data Dictionary ablegt.

Am Ende, auf welchem der genannten Wege auch immer, befinden sich die Daten, die das Anwendungssystem beschreiben, im Data Dictionary, wie Bild 6 zeigt.

System					
Name	Erstellungsdatum	Verantwortlicher	Beschreibender Text	Beziehung zu Programm	Beziehung zu Bildschirm
S-KUBEST	1.8.94	Hiller	Kunden-Bestellung	BESTELL, LIEFG	KUBEST

Programm				
Name	Erstellungsdatum	Verantwortlicher	Beschreibender Text	Beziehung zu Modul
BESTELL	1.7.94	Müller	Bestellungsabwicklung	M-KUAKT, M-AUAN
LIEFG	15.6.94	Marx	Liefervorbereitung	M-ARTAKT, M-RESEL

Modul					
Name	Erstellungsdatum	Verantwortlicher	Beschreibender Text	Beziehung zu Datei	Beziehung zu Bildschirm
M-KUAKT	1.6.94	Müller	Kunden aktualisieren	Kunden	KUAKT
M-AUAN	15.6.94	Müller	Auftrag annehmen	Kunden, Artikel	AUAN
M-ARTAKT	25.4.94	Marx	Artikel aktualisieren	Artikel	ARTAKT
M-RESEL	27.4.94	Marx	Rechnung stellen	Artikel	RESEL

Datei				
Name	Erstellungsdatum	Verantwortlicher	Beschreibender Text	Beziehung zu Feld
Kunden	18.4.94	Müller	Kunden-Stammdatei	K-Name, K-Ort, K-Straße
Artikel	18.4.94	Müller	Artikel-Stammdatei	A-Nummer, A-Name A-Bestand, A-Preis

Bildschirm				
Name	Erstellungsdatum	Verantwortlicher	Beschreibender Text	Beziehung zu Feld
KUAKT	18.4.94	Müller	Kundenformular	K-Name, K-Ort, K-Straße
AUAN	18.4.94	Müller	Auftragsformular	A-Name, A-Nummer, A-Name
ARTAKT	20.4.94	Marx	Artikelformular	A-Nummer, A-Name, A-Bestand, A-Preis
RESEL	20.4.94	Marx	Rechnungsformular	K-Name, K-Ort, K-Straße, A-Nummer, A-Preis, A-Name
KUBEST	23.5.94	Hiller	Hauptmenu	

Feld			
Name	Erstellungsdatum	Verantwortlicher	Beschreibender Text
K-Name	15.3.94	Huber	Namensfeld
K-Ort	15.3.94	Huber	Ortsbezeichnung
K-Straße	15.3.94	Huber	Straßenname und Hausnr.
A-Nummer	20.5.94	Maier	Artikelnummer
A-Name	20.5.94	Maier	Artikelbezeichnung
A-Bestand	20.5.94	Maier	Artikel-Bestand
A-Preis	20.5.94	Maier	Einzelpreis

Bild 6 Einfache Data Dictionary-Struktur mit Dateninhalt

2.2.5 Lösungsschritt 4: Gespeicherte Daten auswerten

Das Data Dictionary-System kann auf Anfrage über die gespeicherten Daten verschiedene Auskünfte liefern. So wird z.B. interessieren, welche Felder in einer bestimmten Maske zu ändern sind, wenn in einer Datei die Felder geändert werden.

Generelle Abfragen können sein:

- Welche Module und Programme gehören zu dem Anwendungssystem Kundenbestellung?
- In welchen Modulen wird eine Datei verarbeitet?
- Wo wird ein Feld verwendet?
- Welche Felder gehören zu welchen Dateien? usw.

Generelle Listen über den gesamten Datenbestand zeigen außerdem alle Objekte und ihre Beziehungen in beiden Richtungen, also nicht nur in der einen Richtung, die in Bild 4 durch den Pfeil gekennzeichnet ist, sondern auch entgegen der Pfeilrichtung. Außerdem werden alle Attribute und, besonders wichtig für die Dokumentation, die beschreibenden Texte auf Wunsch ausgegeben.

2.2.6 Problemlösung: Der erkennbare Nutzen

Der Nutzen eines Data Dictionary ist demnach vielfältig: Ein Data Dictionary dokumentiert ein Anwendungssystem lückenlos und aktuell, so daß es nach verschiedenen Kriterien ausgewertet werden kann.

- In den Entwicklungsphasen kann das Team, das entwickelt, nicht nur durch ständige Fortschreibung das Data Dictionary füllen, sondern auch benötigte Informationen daraus gewinnen. Damit dient das Data Dictionary der Abstimmung und Koordinierung und verhindert, daß z.B. ein Datenfeld mit unterschiedlichen Namen mehrfach angelegt wird.
- Den Entwicklungsprozeß unterstützende Werkzeuge, wie z.B. Generatoren können auf den Datenbestand des Data Dictionary zugreifen und Datenlayouts, Modulrahmen, Testrahmen usw. generieren.
- Bei der Nachdokumentation kann nach dem Einsatz eines Analyseprogramms festgestellt werden, ob Doppeldeutigkeiten im System vorhanden sind. In diesem Fall sollte die Anwendung angepaßt werden oder unterschiedliche Data Dictionary-Daten werden bewußt geführt.

Dieses Beispiel zeigt, daß beliebige Systemstrukturen im Data Dictionary abgebildet werden können. Beispielsweise könnten alle Daten für den Betrieb eines Rechenzentrums in einem Data Dictionary abgelegt sein oder eine Zusammenfassung aller Daten, die heute in verschiedenen Dateien, Tabellen usw. teilweise doppelt gehalten werden. Damit ist eine rationellere Abwicklung des Betriebs möglich. Oder beim Ablauf von Anwendungen kann das Data Dictionary aktiv zur Koordinierung und Kontrolle genutzt werden.

2.3 TOM-REF in den Projektphasen einsetzen

Bei einer ingenieurmäßigen Erstellung von Software wird das Gesamtprojekt in einzelne überschaubare Schritte, die Phasen, eingeteilt. Für jede Phase werden die erforderlichen Tätigkeiten mit ihren Voraussetzungen und Ergebnissen festgelegt. Damit wird die Basis für eine produktivitätssteigernde und qualitätssichernde Vorgehensweise geschaffen.

2.3.1 Projektorganisation in Grundzügen

Die Einteilung eines Gesamtprojektes in Phasen kann nach unterschiedlichen Gesichtspunkten festgelegt werden.

Beispiel einer Phaseneinteilung

In der Praxis hat sich folgende, noch nicht verfeinerte Einteilung bewährt:

Problemanalyse:

Voruntersuchungen zur Aufgabenstellung mit IST-Aufnahmen, Formulieren von Anforderungen und Ausarbeitung von (ersten) Arbeitsplatz-Beschreibungen;

System-Entwurf:

Festlegen der fachlichen System-Struktur (Fach-Konzept) und der Benutzeroberfläche sowie Herleiten der erforderlichen dv-technischen System-Struktur (Dv-Konzept);

Realisierung:

Für jedes Objekt der dv-technischen Systemstruktur Entwurf des Codes, Code-Erstellung und Test;

Integration:

Stufenweises Testen von Funktionen, Systemverhalten und Sicherstellen der gewünschten Software-Eigenschaften;

Einsatz:

Einführung der produktiven Anwendung der Software sowie Produktiv-Einsatz und Beseitigung von Fehlern bzw. Anpassungen an kleinere Änderungen in der Software-Umgebung.

Für größere Projekte ist diese Einteilung im Sinne eines transparenten Projektmanagement jedoch noch zu grob und muß entsprechend verfeinert werden. Je nachdem, in welcher Situation sich ein Anwender befindet, kann die Phasendefinition auch davon abweichend gewählt werden.

Im folgenden werden die Einsatzmöglichkeiten von TOM-REF in den einzelnen Phasen überblicksweise beschrieben.

Data Dictionary-Einsatz organisieren

Im Data Dictionary von TOM-REF können projektbegleitend Informationen hinterlegt werden, wie sie in den einzelnen Entwicklungsphasen benötigt werden. In den einzelnen Projektphasen übernimmt TOM-REF also recht unterschiedliche Aufgaben, die von dem Verwalten eines System-Entwurfs bis hin zum Verwalten von Testdaten reichen. TOM-REF kann diese Aufgaben jedoch nur dann vernünftig wahrnehmen, wenn der Inhalt des Data Dictionary korrekt, vollständig und möglichst redundanzfrei ist.

Der Einsatz des Data Dictionary muß deshalb in geeigneter Weise organisiert werden. In der Praxis hat sich gezeigt, daß

- zentrale Data Dictionary-Informationen am besten von wenigen Personen verwaltet werden, etwa von einem Data Dictionary-Administrator und seinem Vertreter,
- eigentlich unterschiedliche Arten von Data Dictionary zu betrachten sind.

Verschiedene Data Dictionary organisieren

Der Administrator kümmert sich u.a. um die Installation und den laufenden Betrieb der Data Dictionary, die in dem Projekt zum Einsatz kommen. Dabei lassen sich folgende Konstellationen von Data Dictionary einsetzen:

Zentrales Data Dictionary

In diesem Data Dictionary sind Informationen über die aktuell produktiv eingesetzten Anwendungen abgespeichert.

Private Data Dictionary

In den privaten Data Dictionary hat der einzelne Entwickler Informationen für seine Entwicklungsaufgaben, die er isoliert von anderen Entwicklern bearbeitet.

Projekt-Data Dictionary

Das Projekt-Data Dictionary sollte aktuelle Informationen enthalten, die sich während der Arbeit am Projekt schon konsolidiert haben. Die Entwickler können damit die aktuellen Informationen austauschen, bevor sie vom Administrator in das zentrale Data Dictionary eingetragen wurden. Je nach dem Stand des Projekts in den einzelnen Phasen hat es einen unterschiedlichen Reifegrad.

Das Bild 7 stellt das Zusammenspiel der einzelnen Data Dictionary im Projektverlauf dar:

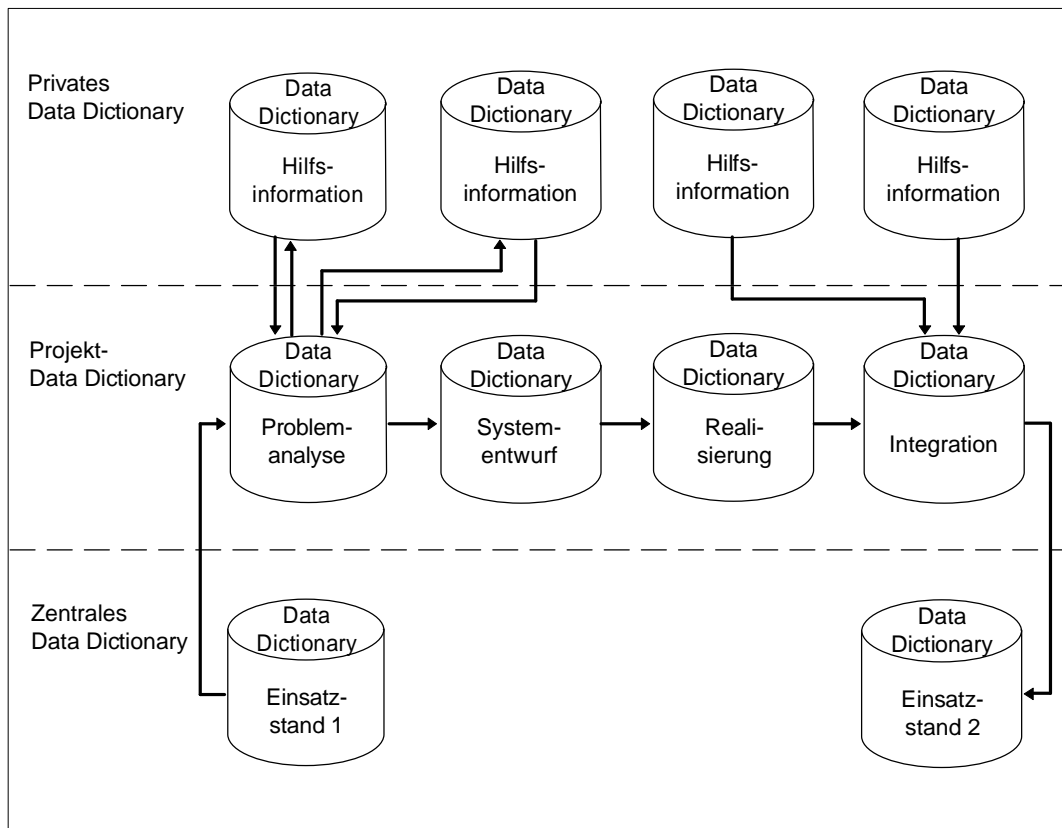


Bild 7 Data Dictionary im Projektverlauf

2.3.2 Problemanalyse

Aufgaben der Phase

Nachdem ein Projekt angestoßen wurde, muß als nächstes die Aufgabe aus fachlicher Sicht möglichst gründlich analysiert werden. Dazu wird zuerst der Ist-Zustand ermittelt. Dieser bildet die Ausgangsbasis und den Orientierungsrahmen für das Ausarbeiten der Soll-Vorstellungen.

Nun werden die Sollvorstellungen genau analysiert. Die Fachebene sammelt alle relevanten Anforderungen an die Software und gewichtet sie nach ihrer Bedeutung.

Das Ergebnis ist eine fachliche Problemanalyse, in der auch erste Beschreibungen zukünftig entstehender Arbeitsplätze enthalten sind. Die Beschreibungen zeigen, in welchem organisatorischen Einsatzbereich die Software verwendet werden soll. Dieser erste Entwurf wird in den folgenden Phasen weiter präzisiert.

Existierendes DV-System nachdokumentieren

In der Phase der Problemanalyse erfolgt eventuell die Nachdokumentation des bestehenden DV-Systems. Für diesen Zweck werden die Code- und Text-Analyseprogrammen eingesetzt, die innerhalb des gesamten Data Dictionary-Systems zur Verfügung stehen. Mit TOM-CA werden COBOL- und COLUMBUS-COBOL-Programme analysiert; TOM-JA dient zur Analyse von BS2000-Prozeduren und TOM-TA zur Analyse von Texten. Mit dem Analysebaustein aus UDS-DDA können DDL-Sourcen aus UDS-Datenbanken und COBOL-DML-Aufrufe analysiert werden.

Die Analyseprogramme hinterlegen ihr Ergebnis in Data Dictionary, die mit TOM-REF weiter bearbeitet werden können. Mit der "Incorporate"-Funktion werden die einzelnen Analyseergebnisse in das zentrale Data Dictionary zusammengeführt.

Istzustand analysieren

Der Ist-Zustand wird mit den Grund- und Servicefunktionen von TOM-REF analysiert. Die Auswertung kann entweder am Bildschirm oder über Druckerlisten vorgenommen werden. So kann sich der Entwickler einen Überblick über die vorhandenen Objekte eines bestimmten Typs verschaffen oder auch Verwendungsnachweise in mehrstufiger Form erhalten. Mit TOM-REFG lassen sich Informationsflüsse analysieren und übersichtlich darstellen.

Die Informationen, die mit den Analyse-Läufen gewonnen wurden, erlauben Rückschlüsse auf die Einsatzbereiche und die fachlichen Funktionen der Anwendungen. Damit lassen sich Reichweite und Aufwand von neuen Projekten besser abschätzen und konkretere Vorstellungen über den Soll-Zustand von zukünftigen DV-Systemen entwickelt werden.

2.3.3 Systementwurf

Aufgaben der Phase

In der Phase des Systementwurfs werden zuerst alle fachlichen Eigenschaften und die erforderliche logische Datenorganisation festgelegt. Anschließend sind folgende Aufgaben zu erledigen:

- Die erforderlichen Software-Funktionen und deren Aufteilung auf Systemkomponenten muß festgelegt werden.
- Das System muß durch geeignete Anordnung von Systemkomponenten in Hierarchien strukturiert werden.
- Der Funktions- und Leistungsumfang der Komponenten muß festgelegt werden.
- Die Schnittstellen und die Wechselwirkung zwischen den Komponenten müssen festgelegt werden.
- Die physikalische Datenorganisation (Internal Schema) muß passend zur logischen Datenorganisation entwickelt werden.
- Der Leistungsumfang der zu erstellenden Software muß verabschiedet werden.

Diese Arbeitsschritte sind sowohl für die zu erstellende Software als auch für die erforderliche Test-Software durchzuführen.

Das Data Dictionary wird schrittweise während der Entwurfsarbeiten mit neuen Beschreibungen von Objekten versorgt, sobald einzelne Informationen vernünftigerweise abgespeichert werden können.

Fachliche Eigenschaften dokumentieren

Standardobjekttypen zur Dokumentation der fachlichen Systemeigenschaften sind zur Zeit noch nicht definiert. Über benutzereigene Objekttypen können Sie jedoch Ihr eigenes Modell von der fachlichen Systemstruktur entwerfen. Dabei sind es die Entwurfsmethoden, die die Definition von Objekttypen bestimmen.

Von den Standardobjekttypen erlauben `RECORD`, `FIELDGROUP` und `FIELD` das Festschreiben von grundlegenden Datenorganisationseigenschaften. Dialogabläufe lassen sich mit den Standardobjekttypen `FORMAT (MSK)`, `RELATIONSHIP (RLS)` und `RETURN-INFO (RET)` modellieren.

DV-technische Systemstruktur beschreiben

Für die Beschreibung der dv-technischen Systemeigenschaften stehen eine Reihe von Standardobjekttypen zur Verfügung. Besonders mächtig unterstützt werden die modul-spezifischen Beschreibungen. Diese werden, solange die Zusammenhänge zwischen den einzelnen Modulen nicht in den Vordergrund treten, über die Objektmasken vorge-

nommen. Mit den N2Charts von TOM-REFG lassen sich jedoch besonders einfach Entwürfe für die Modulverständigung über `CAL`- und `ENP`-Objekte erstellen.

Auf die Beschreibung der Modulschnittstellen im Data Dictionary sollte großen Wert gelegt werden, da sich diese Informationen in der Realisierungsphase weiter verwenden lassen.

Ausgehend von den Beschreibungen der Module sind dann Programme und BS2000-Prozedurinhalte zu beschreiben.

Mit den Servicefunktionen läßt sich überprüfen, ob es Inkonsistenzen zwischen den Objektbeschreibungen und Beziehungsbeschreibungen gibt.

Neben den Software-Eigenschaften ist auch die physikalische Datenorganisation zu beschreiben. Dies geschieht mit Hilfe der Objekttypen für Dateien und Datenbanken.

Dabei werden während der Entwurfsphase stetig und nicht erst am Phasenende Informationen in das Data Dictionary eingetragen und fortgeschrieben.

Data Dictionary-gestützter Modulentwurf

Die Data Dictionary-Informationen über Dateien und Module können in Modulrahmensourcen umgesetzt werden. Rufen Sie dazu unter TOM-TI die Generierungsfunktion `SSGEN` auf. `SSGEN` wertet die Informationen, die zu einem Modul im Data Dictionary vorliegen, soweit wie möglich aus und erzeugt daraus einen (COLUMBUS-)COBOL-Modulrahmen. Dadurch ersparen Sie sich manuellen Übertragungsaufwand.

Der Source-Vorschlag kann dann mittels TOM-TI ergänzt werden. Dazu gehört beispielsweise auch die Beschreibung der modulinternen Abläufe in Pseudocode, der der COLUMBUS-Syntax folgt.

2.3.4 Realisierung

Aufgaben der Phase

In diesem Prozeßschritt erfolgt die Code-Erstellung für die Komponenten, die vorher entworfen wurden, zusammen mit den Testmitteln für den zugehörigen Komponententest. Diese Arbeiten lassen sich mit Code-Generatoren beschleunigen.

Mit Betriebssystem- oder Datenbanksystemfunktionen sind die Daten physikalisch zumindest in einer Testversion zu installieren. Nachdem eine einzelne Komponente zumindest syntaxmäßig fehlerfrei implementiert wurde, wird diese einem Komponententest unterworfen. Dabei werden die syntaktisch fehlerfreien Software-Komponenten in einer zumindest simulierten Systemumgebung hinsichtlich ihrer internen Ablauflogik einzeln geprüft.

Standard-Sourceteile generieren

Außer dem bereits erwähnten Schnittstellen-Generator SSGEN gibt es standardmäßig in TOM-REF die Möglichkeit, aus COPY-Beschreibungen vom Typ `INL` COBOL-COPY-Strecken zu generieren. Dabei werden je nach Einsatzzweck COPY-Strecken aus den Data Dictionary-Einträgen generiert für

- Dateibeschreibungen
- Datenbereiche
- MOVE- und Initialisierungsanweisungen

Der generierte Code wird in einer PLAM-Bibliothek, meistens die TOMLIB, abgelegt.

DDL generieren

Wenn im Systementwurf UDS- oder SESAM/SQL V2-Datenbanken im Data Dictionary beschrieben wurden, kann mit dem Generatorbaustein von UDS-DDA bzw. SESAM-DDA eine DDL-Source aus diesen Beschreibungen generiert werden. Die so erzeugte Source muß nach den üblichen Schritten übersetzt werden. Sie wird vom Datenbank-System zum Einrichten einer neuen oder zum Restrukturieren einer bereits existierenden Datenbank verwendet.

Modul testen

Mit dem Werkzeug TOM-MTEST können Sie einzelne Komponenten testen, bevor die Umfeldkomponenten realisiert sind. Dazu werden aus den Objektbeschreibungen im Data Dictionary Hilfsmodule generiert, welche die fehlenden Umfeldkomponenten simulieren. Das genaue Verfahren ersehen Sie aus Bild 8.

Tools durch Komponentenlisten steuern

Komponentenlisten sind Elemente in PLAM-Bibliotheken, die beispielsweise eine Namensliste von Modulen enthalten. Sie können diese Komponentenlisten zum Steuern des TomDoors-Monitor-Ablaufs verwenden.

Komponentenlisten lassen sich durch die TOM-REF-Servicefunktionen nach verschiedenen Gesichtspunkten erstellen. Typische Anwendungsbeispiele sind Komponentenlisten für Übersetzungs- und Binderabläufe nach Veränderungen von Feld, Satz- und COPY-Beschreibungen.

Verfahren TOM-MTEST

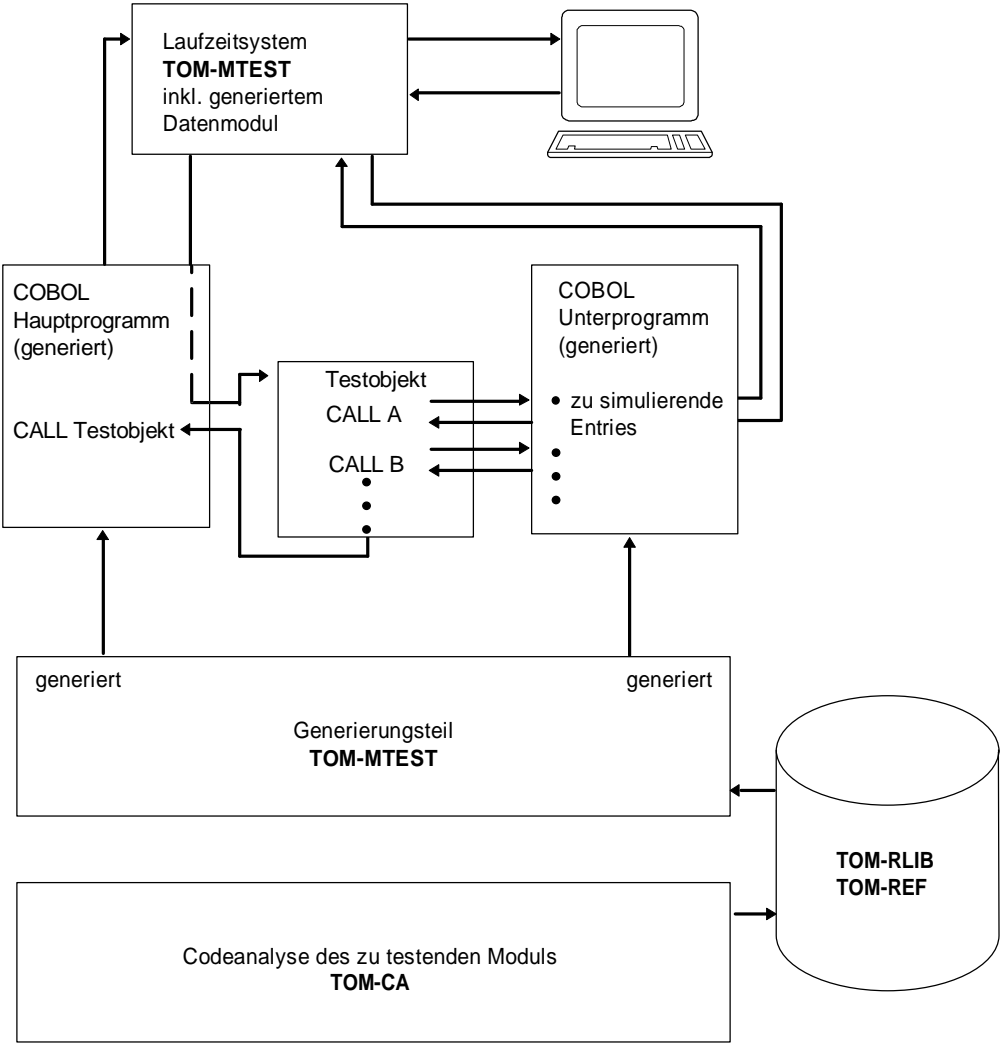


Bild 8 Verbindung von TOM-REF zu TOM-MTEST

2.3.5 Integration

Aufgaben der Phase

Nach dem Abschluß der Realisierungsarbeiten sind die einzelnen Anwendungskomponenten sukzessive zu einem vollfunktionsfähigen Gesamtsystem zu integrieren. Zuerst werden Funktionskomplexe, dann das Systemverhalten und schließlich das Verhalten in der Einsatzumgebung getestet, und eventuell auftretende Fehler werden korrigiert.

Nach Abschluß aller Testarbeiten beginnt der Produktivbetrieb.

Nachdokumentation

Nach dem Abschluß der letzten Korrekturen an einzelnen Komponenten ist mit Hilfe der TOM-Analyseprogrammen nachzuprüfen, ob die Entwurfsbeschreibungen eingehalten wurden. Bei Abweichungen ist gegebenenfalls das Data Dictionary zu korrigieren. Für die Kontrolle bieten sich Servicefunktionen und N2Charts an.

Data Dictionary-Informationen konsolidieren

Bei Abschluß der Integrationsarbeiten ist dafür Sorge zu tragen, daß die privaten Data Dictionary in das Projekt-Data Dictionary eingearbeitet werden. Anschließend ist das Projekt-Data Dictionary in das zentrale Data Dictionary einzuspielen, welches die produktiv ablaufenden DV-Anwendungen beschreibt. Dies sollte von dem Data-Dictionary-Administrator mit Hilfe der Servicefunktion `Incorporate` vorgenommen werden.

2.3.6 Einsatz

Aufgabe der Phase

In dieser Phase wird die Anwendung produktiv eingesetzt. Auftretende Fehler sind zu korrigieren und Anpassungen an geänderte Hardware- und Basis-Software-Eigenschaften durchzuführen.

Wartung mit dem Data Dictionary unterstützen

Die Informationen des TOM-REF Data Dictionary über Objekte und ihre Verwendung können nach unterschiedlichen Gesichtspunkten online abgefragt werden. Dadurch wird die Tragweite einzelner Korrekturen transparenter. Dies erleichtert die Auswahl von Korrekturen mit geringstem Aufwand.

3 Konzept von TOM-REF

Nach der Einführung in das Thema Data Dictionary folgt in diesem Kapitel eine Einführung in das Data Dictionary-System TOM-REF. Zur Verwendung des Begriffspaares Typ-Ausprägung ist noch eine Vorbemerkung notwendig: Dieses Begriffspaar stammt aus dem Bereich der Datenbanken (UDS: Type - Occurrence). Ein Data Dictionary ist auch nichts anderes als eine spezialisierte Datenbank. Daher hat dort dieses Begriffspaar die gleiche Bedeutung wie bei TOM-REF. Bildlich gesprochen kann ein Typ wie ein Behälter aufgefaßt werden und eine Ausprägung als ein zum jeweiligen Behälter passender Inhalt.

Für den speziellen Fall des Data Dictionary wird allerdings eine weitere Begriffsebene eingeführt, weil ein Data Dictionary eine höhere Abstraktionsstufe darstellt. Es enthält Daten über Daten, also lediglich sogenannte Meta-Daten. Das folgende Bild 9 soll dies veranschaulichen:

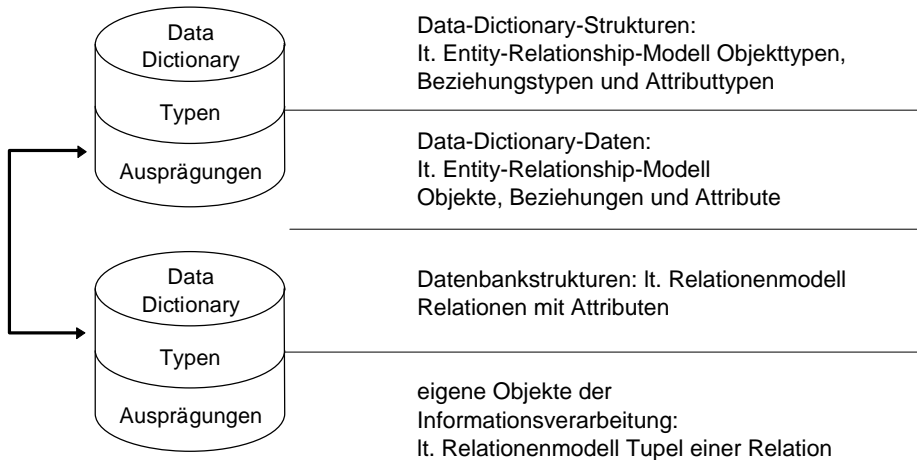


Bild 9 Begriffspaar "Typ-Ausprägung" in verschiedenen Ebenen

Die Typen einer Datenbank sind identisch mit den Ausprägungen im Data Dictionary. Um ganz genau zu sein, muß im Fall des Data Dictionary von Meta-Typen und Meta-Ausprägungen gesprochen werden. Im folgenden wird bewußt auf diese Unterscheidung verzichtet, da aus dem jeweiligen Kontext die Bedeutung hervorgeht.

3.1 Der Strukturkatalog

Die Teilstrukturkataloge bilden gemeinsam den Strukturkatalog von TOM-REF. Die jetzt definierten Teilstrukturkataloge für JCL, UTM, Programmdesign und Programmrealisierung auf der Funktionsseite und UDS, SESAM/SQL V2, DVS/LEASY auf der Informationsseite werden bevorzugt in den Entwicklungsphasen 3 und 4 des Phasenplans verwendet.

Für die Verwendung des Data Dictionary in den frühen Phasen (1 und 2) werden nur für den Funktionsentwurf Standard-Strukturen angeboten. Für den Datenentwurf liegt ein Vorschlag vor, der als Grundlage einer eigenen Modellierung dienen kann (siehe Seiten 255 und 258).

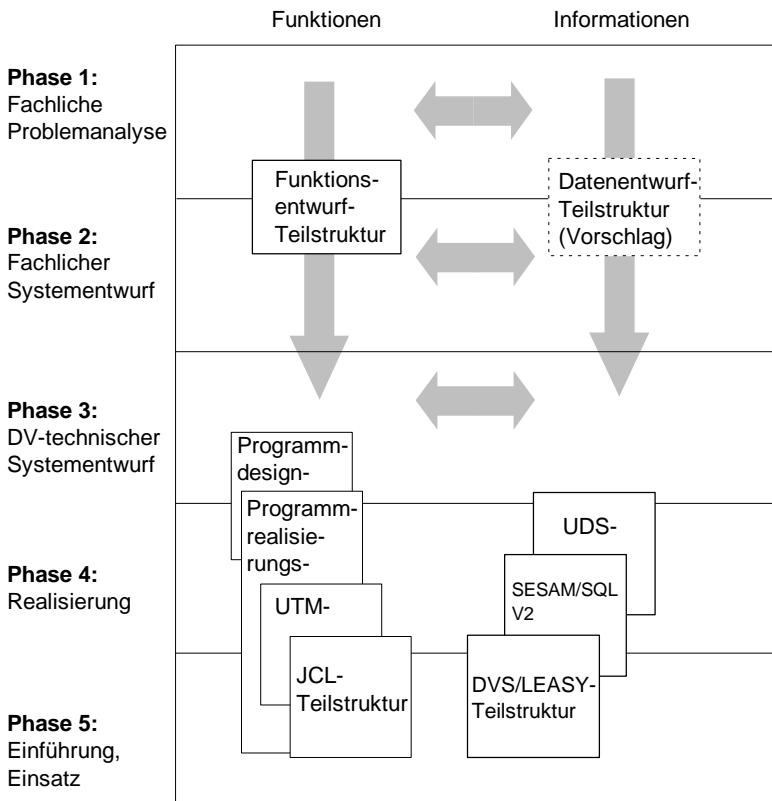


Bild 10 Teilstrukturkataloge in den einzelnen Entwicklungsphasen

Die Phase der Einführung und des Einsatzes (5) wird in Anfängen bereits durch die JCL-, die DVS/LEASY- und die Programmrealisierungs-Teilstruktur unterstützt.

Der Strukturkatalog von TOM-REF enthält eine ganze Reihe von Objekt-, Beziehungs- und Attributtypen, die benötigt werden, damit Werkzeuge mit dem Data Dictionary arbeiten können. Die folgende Tabelle gibt einen Überblick.

Teilstrukturen	Werkzeuge, die sie hauptsächlich nutzen
JCL-Teilstruktur	TOM-JA : Analyseprogramm für BS2000-DO- und ENTER-Prozeduren TOM-REFG : Grafische Ein-/Ausgabe
Programmdesign-Teilstruktur	TOM-CA : Analyseprogramm für COBOL-Programm-Module TOM-MTEST: Testrahmen für Modultest TOM-REF : Copy-Strecken-Generator TOM-REFG : Grafische Ein-/Ausgabe SSGEN : Schnittstellengenerator (TOM-TI-Toolbörse) TOM-TA : Text-Analyseprogramm
Programmrealisierung-Teilstruktur	TOM-REFG : Grafische Ein-/Ausgabe
UTM-Teilstruktur	TOM-CA : Analyseprogramm für UTM-Teilprogramm-Module (COBOL) TOM-REFG : Grafische Ein-/Ausgabe
DVS/LEASY-Teilstruktur	TOM-REFG : Grafische Ein-/Ausgabe TOM-MP : Maskenprototyping TOM-CA : Analyse von COBOL-Programmen
UDS-Teilstruktur	TOM-CA : Hier speziell Analyse der DML-Anweisungen zusammen mit UDS-DDA TOM-REFG : Grafische Ein-/Ausgabe UDS-DDA : DDL-Analyse, DML-Analyse und DDL-Generierung
SESAM/SQL V2-Teilstruktur	SESAM/SQL-DDA: Datenbank-Analyse, SQL-Transformation und SQL-Generierung TOM-REFG : Grafische Ein-/Ausgabe TOM-CA : Analyse von ESQ-LOBOL-Programmen
Funktionsentwurf-Teilstruktur	TOM-REFG : Grafische Ein-/Ausgabe und Unterstützung der strukturierten Analyse

Tabelle 1 Teilstrukturen und ihre Verwendung

In den Kapitel 6 bis 10 sind die einzelnen Teilstrukturen in Form von Entity-Relationship-Diagrammen mit den zugehörigen Elementaraussagen enthalten.

Der gesamte Strukturkatalog läßt sich auch folgendermaßen gliedern:

- Vom Hersteller vordefinierte Objekttypen, Beziehungstypen und Attributtypen, die den Standard darstellen. Ab Seite 36 ist der Standard beschrieben.
- Vom Benutzer eingebrachte Objekttypen und Beziehungstypen (Benutzereigene Erweiterung). Bestimmte Attributtypen sind mit fester Länge vorgegeben. Sie können ihre Bedeutung festlegen. Die benutzereigenen Erweiterung sind ab Seite 40 beschrieben.

Zur Behandlung von Objekten benutzereigener Typen steht ein einheitliches Masken-Layout zur Verfügung.

Einen Sonderfall bilden die Standard-Objekttypen `RLS` (Relationship) und `RET` (Return-Info). Ihre Objekte werden ebenfalls in diesem Masken-Layout bearbeitet. Besonders wichtig ist der Beziehungstyp `RLS` zur Modellierung mehrstelliger Beziehungen.

Ein eigener Unterabschnitt ist dem Objekttyp `ALI` (Alias) zur Behandlung von Synonymen gewidmet.

3.1.1 Standard-Einträge

Die im Strukturkatalog enthaltenen Objekt-, Beziehungs- und Attributtypen sind die herstellerseitig vorgegebenen Standard-Typen. Dieser Standard garantiert, daß keine falschen oder unnötigen Daten in TOM-REF eingetragen werden können. TOM-REF prüft jede Eingabe gegen den Strukturkatalog und weist sie zurück, wenn sie nicht in die definierte Struktur paßt.

Objekttypen

Für jeden Standard-Objekttyp in TOM-REF existiert mindestens eine Maske für die Anzeige, Eingabe usw.. Der Langname des Objekttyps ist im Kopfteil der Maske angegeben. Daneben existiert zu jedem Objekttyp ein Kurzname. Der Kurzname kann bei Eingaben verwendet werden, so daß man sich die langen Namen nicht merken muß und der Schreibaufwand geringer wird.

Die Objekttypen der einzelnen Teilstrukturen sind in den Kapiteln 6 bis 10 beschrieben. Dort finden Sie auch die Einbindung der Objekte in den Teilstrukturkatalog und eine Beschreibung, wie Sie die einzelnen Objekte modellieren. Im Anhang finden Sie eine Tabelle aller Objekttypen in TOM-REF V3.0.

Beziehungstypen

Die Beziehungen, die laut Strukturkatalog von TOM-REF möglich sind, müssen bei demjenigen Objekt erstellt werden, von dem die Beziehung ausgeht. Man spricht in diesem Zusammenhang von der Beziehungsliste des Ausgangsobjekts. Zur Erstellung der Beziehungsliste eines Objekts steht die typspezifische Objektmaske und, falls der Platz nicht ausreicht, eine für alle Objekttypen einheitliche Beziehungslistenmaske zur Verfügung.

So wird z.B. die Beziehung vom Objekt Modul A zu Objekt Format B im Sinne von Modul A verwendet. Format B beim Modul A eingetragen. Die inverse Beziehung im Sinne von Format B wird in Modul A verwendet. In TOM-REF nicht explizit eingetragen. Die Servicefunktionen erzeugen auf Wunsch Verwendungsnachweise, die auch die inversen Beziehungen ausgeben.

Eine Übersicht der möglichen Beziehungen finden Sie bei der Objektbeschreibung in den Kapiteln 6 bis 10. Die Bedeutung der einzelnen Beziehungen ist dem zugehörigen Teilstrukturkatalog zu entnehmen, in dem zu jedem Beziehungstyp eine Elementar-Aussage gemacht wird.

Attributtypen

Die Standard-Attributtypen, die für Objekt- und Beziehungstypen vorgesehen sind, können in zwei Gruppen unterschieden werden:

- Typ-neutrale Attributtypen
- Typ-spezifische Attributtypen

Typ-neutrale Attributtypen sind für alle Objekt- und Beziehungstypen gleich. Typ-spezifische Attributtypen hingegen kommen nur bei bestimmten Objekt- bzw. Beziehungstypen vor.

Alle Attributtypen sehen Sie in den Masken zu den jeweiligen Objekttypen. Die typ-neutralen Attributtypen sind im oberen Teil der Maske enthalten. Die zu den Beziehungstypen gehörigen sind als Spalten der Tabelle vorgesehen, in die Beziehungen eingetragen werden. Die folgenden Tabellen 2 und 3 zeigen die typ-neutralen Attributtypen und geben eine kurze Erläuterung.

Attributtypen lt. Masken	Länge in Bytes	Erläuterung
Type	3 bzw. 15	Der Name des Objekttyps steht in der Überschrift. Der Kurzname wird bei Eingaben verwendet. Zusammen mit dem Namen dient der Typ zur Identifizierung eines Objekts (3 Bytes für den Kurznamen, 15 für den Langnamen).
Name	variabel	Zusammen mit dem Typ dient der Name zur Identifizierung eines Objekts. Die Namenslänge hängt vom einzelnen Objekttyp ab.
Creation Date	8	Hier wird von TOM-REF automatisch das Datum der ersten Eingabe eingetragen und zwar in der Form: tt.mm.jj.
Last Change Date	8	Hier wird von TOM-REF automatisch das Datum der letzten Änderung eingetragen und zwar in der Form: tt.mm.jj.
Responsible	3	Sie können für jedes Objekt das Kurzzeichen eines dafür Verantwortlichen eintragen.
Short Description	80	Sie können eine kurze Beschreibung des Objekts eintragen.
Text	1	Hier trägt TOM-REF automatisch ein, ob zum Objekt ein TOM-REF-Text (Siehe Seite 89) vorhanden ist. In einer eigenen Maske kann der Benutzer einen beliebigen Text bis zur Länge von 800 Zeichen ins Data Dictionary eintragen (Texte beliebiger Länge können in die angeschlossene TOMLIB oder PLAM-Bibliothek ausgelagert werden.)
8 einstellige Attributtypen	8x1	1-5 Für künftige TOM-REF-Erweiterungen reserviert 6-8 Frei benutzbare Attributtypen

Tabelle 2 Objekttyp-neutrale Attributtypen

Attributtypen lt. Masken	Länge in Bytes	Erläuterung
Type	3	Der Typ-Eintrag wird in einigen wenigen Fällen von TOM-REF Typ-spezifisch ausgewertet.
8 einstellige Attributtypen	8x1	1-5 Für künftige TOM-REF-Erweiterungen reserviert 6-8 Frei benutzbare Attributtypen
OCCURS-Klausel	5	Anzahl, wie oft das Zielobjekt vom Typ FLD oder FLG vorkommt. Dient der Abbildung der OCCURS-Klausel von COBOL.

Tabelle 3 Beziehungstyp-neutrale Attributtypen

Typ-spezifische Attributtypen sind bei einigen Objekttypen so zahlreich, daß ein Folge-
maske zur Darstellung notwendig ist. In den Kapiteln 6-10 sind die einzelnen Masken
beschrieben.

Frei benutzbare Attributtypen

Unter den typ-neutralen Standard-Attributtypen für Objekt- und Beziehungstypen befinden sich jeweils acht einstellige Attributtypen. Von diesen sind die ersten fünf für TOM-REF-Erweiterungen vorgesehen, die folgenden drei sind frei benutzbar. Diese drei können Sie mit eigenen Bedeutungen hinterlegen. Hier können Sie z.B. durch ein Kennzeichen den Entwicklungsstand eines Objekts wie Modul oder Programm führen. Anhand von Auswertungen läßt sich so verfolgen, welche Objekte in welcher Phase des Entwicklungsprozesses sind.

Synonym-Behandlung

Wie bereits erwähnt (Tabelle 2), identifiziert TOM-REF die einzelnen Objekte anhand des Typ- und des Namenseintrags. Für jeden Objekttyp dürfen demnach nur eindeutige Namen eingetragen werden. Will man zu einem Objekt mehrere Namen eintragen (Synonyme), kann man den Objekttyp *Alias* (ALI) benutzen (siehe Bild 11).

Synonyme sind Sonderfälle, da in einem Data Dictionary Doppeldeutigkeiten nicht sinnvoll sind. Wenn ein unternehmensweites Data Dictionary aufgebaut werden soll, müssen aber unter Umständen gewisse Doppeldeutigkeiten bewußt aufgenommen werden. So gibt es z.B. in älteren Anwendungen unterschiedliche Schreibweisen des gleichen Feldnamens oder Felder unterschiedlichen Namens und Länge aber mit gleichem Inhalt, die ins Data Dictionary aufgenommen werden sollen.

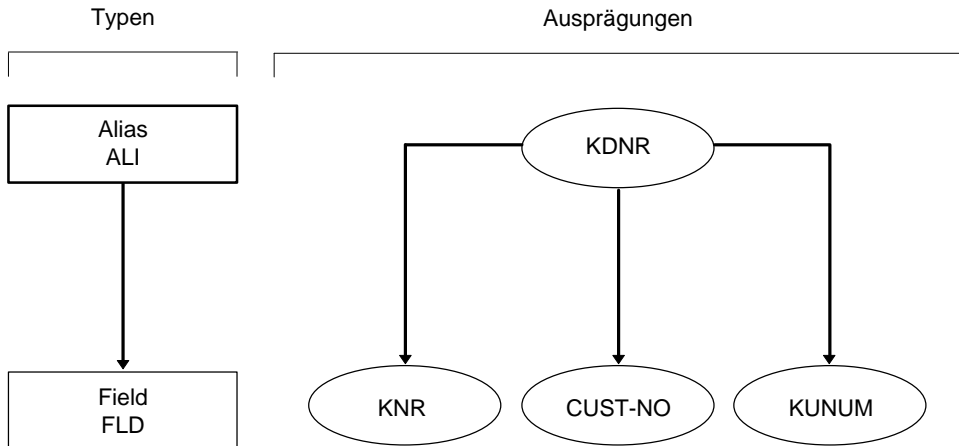


Bild 11 Beispiel: Synonyme für ein Feld

3.1.2 Benutzereigene Erweiterungen

Sie können eigene Entity-Relationship-Modelle in TOM-REF einbringen. Dafür können Sie eigene Objekttypen, Beziehungstypen und Attributtypen definieren. Diese eigenen Teilstrukturkataloge können Sie mit dem Standard verbinden (siehe Bild 12). Mit der Definition und dem Hinterlegen eigener Teilstrukturkataloge erreichen Sie, daß TOM-REF bei der Eingabe gegen diese Erweiterungen ebenfalls prüfen kann.

Benutzereigene Objekttypen

Zur Beschreibung dieser Objekttypen steht Ihnen eine Maske mit einheitlichem Layout zur Verfügung (siehe Seite 203). Freie Objekttypen müssen Sie allerdings im Unterschied zu den Standard-Objekttypen in allen Dialogen eingeben.

Wenn Sie Ihr Data Dictionary um benutzereigene Objekttypen erweitern wollen, müssen Sie diese Objekte TOM-REF bekanntgeben (siehe Seite 55).

Benutzereigene Beziehungstypen

Von den benutzereigenen Objekttypen können beliebige Beziehungstypen ausgehen. Ziel-Objekttypen können wieder benutzereigene oder auch Standard-Objekttypen sein. Umgekehrt können Sie die Standard-Objekttypen zum Ausgang von Beziehungstypen machen, die als Ziel-Objekttyp einen benutzereigenen Objekttyp haben. Bild 12 zeigt an einem einfachen Beispiel diesen Sachverhalt.

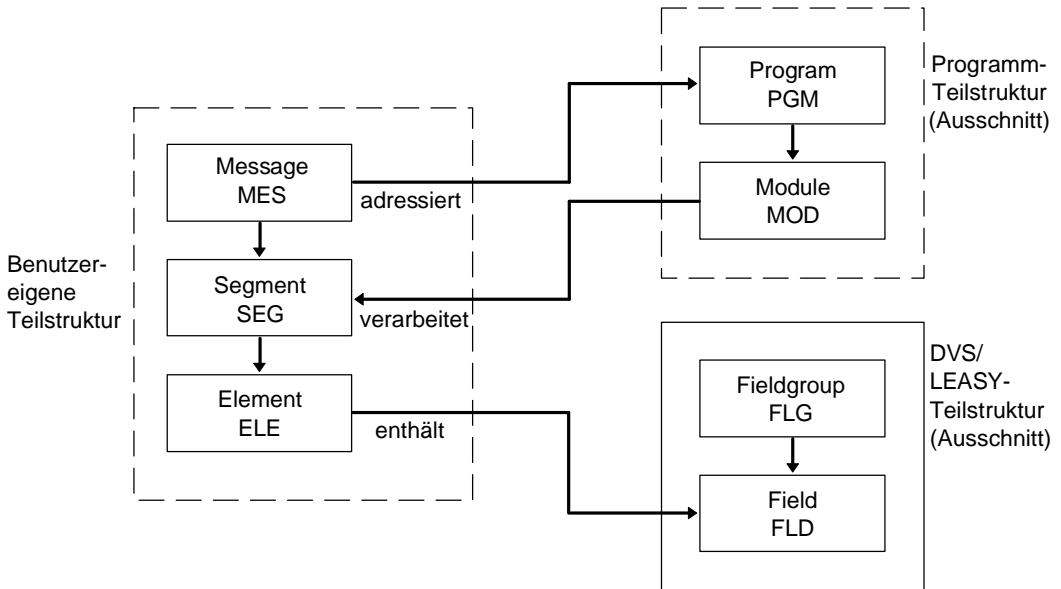


Bild 12 Benutzereigene und vordefinierte Teilstrukturkataloge

In den Standard-Beziehungstypen sind auch rekursive Beziehungstypen wie z.B. Modul (MOD) ruft Modul (MOD) oder Fieldgroup (FLG) enthält Fieldgroup (FLG) enthalten.

Mehrstellige Beziehungstypen mit einem Objekttyp als Ausgang und mehreren Objekttypen als Ziel können Sie bilden, indem Sie sich der Hilfs-Objekttypen *RLS* und ggf. *RET* bedienen. Das folgende Bild 13 zeigt das prinzipielle Vorgehen an einem einfachen Beispiel: Dargestellt wird der rekursive Beziehungstyp Format (MSK) adressiert Format (MSK) und benutzt dazu Function-Key (CMD) und Field (FLD). Der Objekttyp Function-Key (CMD) ist kein Standard-Objekttyp. Auf Seite 55 ist zu dieser Thematik ein weiteres, ausführliches Beispiel enthalten.

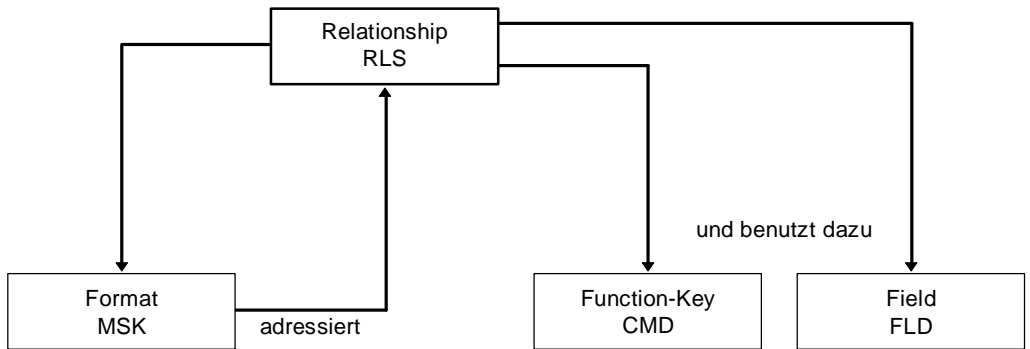


Bild 13 Modellierung mehrstelliger Beziehungstypen

Benutzereigene Attributtypen

Für benutzereigene Objekt- und Beziehungstypen sind die Typ-neutralen Attributtypen vorgesehen. Darunter gibt es die frei benutzbaren Attributtypen, denen Sie eine eigene Bedeutung geben können. Desweiteren sind bei den benutzereigenen Objekttypen vier frei benutzbare Attributtypen mit der jeweiligen festen Länge von 32 Bytes vorgesehen.

In den Text-Attributtypen (TOM-REF-Text mit max. 800 Zeichen und TOM-TI-Longtext beliebiger Länge, der in der TOMLIB oder einer PLAM-Bibliothek abgelegt wird) können Sie beliebige eigene Informationen hinterlegen. Die Auswertung dieser Texte bleibt in der aktuellen Version von TOM-REF Ihnen selbst überlassen.

3.2 Datenhaltung

Die Datenhaltung von TOM-REF wird in diesem Abschnitt in logischer (Konfigurierbarkeit) und physikalischer Hinsicht (verwendetes Datenhaltungssystem und Namenskonventionen) betrachtet.

3.2.1 Konfigurationen

Gemäß seinen Erfordernissen können Sie unterschiedliche Konfigurationen des TOM-REF-Data Dictionary-Systems bilden. Als Basis dafür gibt es die Möglichkeit, mit zwei Data Dictionaries im Verbund zu arbeiten. Hierbei wird davon ausgegangen, daß eines mehr eine Funktion im Sinne von zentralem Data Dictionary hat und das zweite das private Data Dictionary ist.

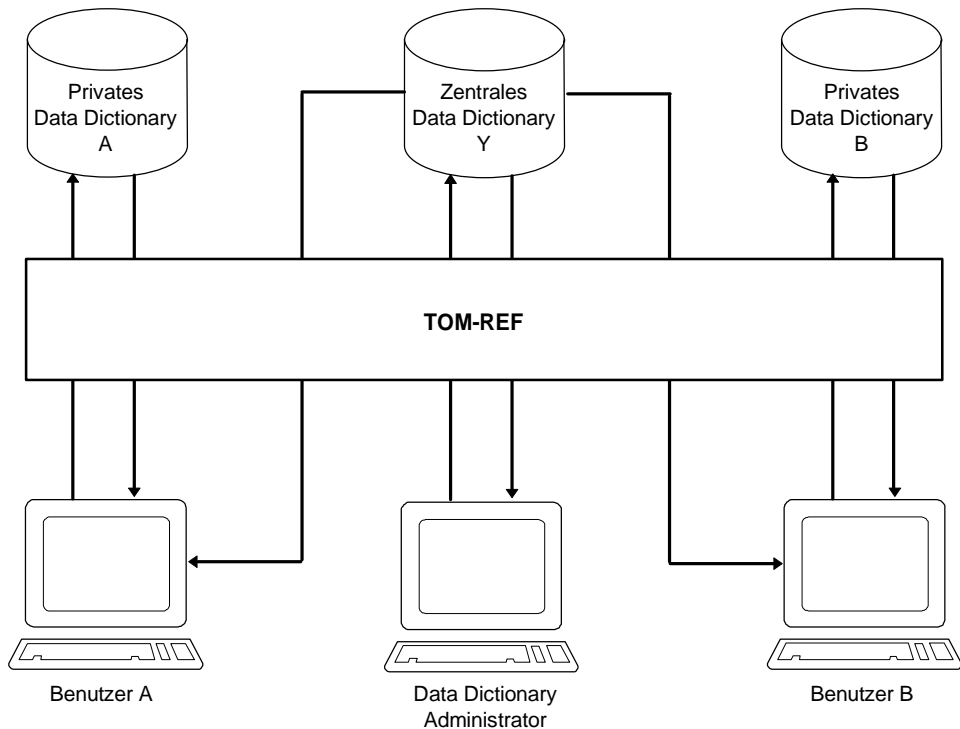


Bild 14 Zentrales und private Data Dictionaries im Verbund

Im Bild 14 gibt es einen Benutzer A der ein privates Data Dictionary A lesend und schreibend im Zugriff hat und außerdem auf das zentrale Data Dictionary Y lesend zugreift. Ein zweiter Benutzer B schreibt in sein privates Data Dictionary B und liest aus seinem Data Dictionary B und ebenfalls aus dem zentralen Data Dictionary Y. Damit ist gewährleistet, daß sich beide Benutzer nicht behindern, indem sie vorläufige Data Dictionary-Stände in eigener Regie führen.

Andererseits bewirkt aber der gemeinsame Zugriff auf das zentrale Data Dictionary, daß sie sich koordinieren können über die bereits auf gemeinsamer Abstimmung beruhenden Data Dictionary-Einträge. Diese Konfiguration erleichtert auch die unabhängige Entwicklung unterschiedlicher Anwendungen, ggf. in verschiedenen Projekten mit unterschiedlichen Teams bei gleichzeitiger Koordinierung. Ein Data Dictionary-Administrator sorgt für die Verwaltung des zentralen Data Dictionary. Er hat auch die Aufgabe, private Data Dictionary-Stände in das zentrale Data Dictionary zu übernehmen, wenn der Zeitpunkt dafür gekommen ist, z.B. nach Abschluß eines Projekts. Diese Übernahme geschieht kontrolliert, so daß keine unerwünschten Daten in das zentrale Data Dictionary gelangen.

Auf dieser Basis sind weitere Konfigurationen denkbar. So kann das zweite Data Dictionary, auf das lesend zugegriffen werden kann, ein weiteres privates Data Dictionary sein. Dies z.B. dann, wenn noch kein eigentliches zentrales Data Dictionary geführt wird, aber verschiedene Data Dictionary-Stände koordiniert werden sollen.

Auch sollten Analyseergebnisse, wie sie z.B. TOM-CA liefert, nicht sofort in ein bestehendes Data Dictionary übernommen werden, sondern zunächst in ein eigenes, für die Analyse eingerichtetes privates Data Dictionary. So können die Analysedaten in einem zweiten Schritt kontrolliert in einen umfassenderen Data Dictionary-Datenbestand eingegliedert werden.

3.2.2 Datenhaltungssystem

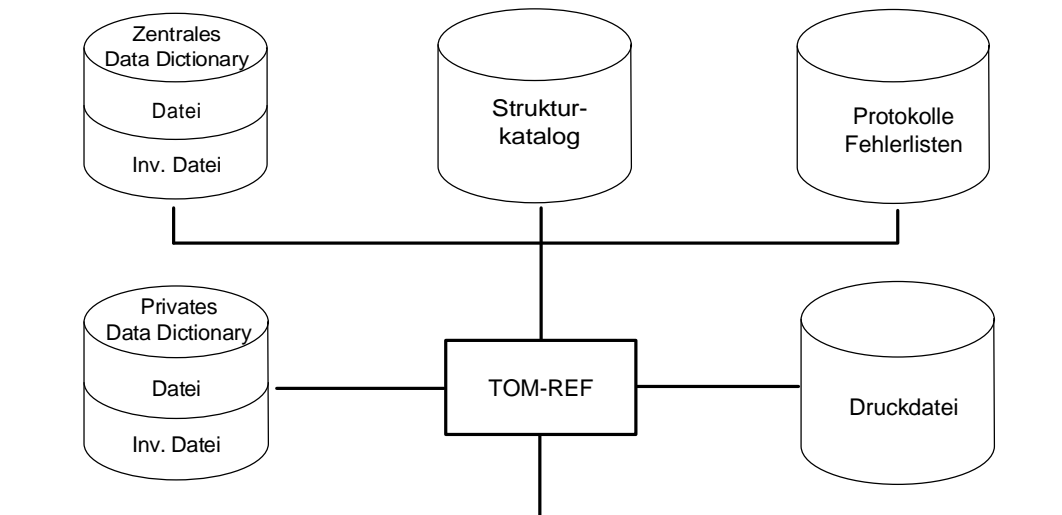
Als Datenhaltungssystem benutzt TOM-REF die DVS-Zugriffsmethode ISAM. Die Data Dictionary-Daten werden in jeweils zwei ISAM-Dateien geführt. In einer dieser beiden Dateien (TOMRFI-Datei) werden inverse Beziehungen gespeichert, das sind Beziehungen vom Zielobjekt zum Ausgangsobjekt zurück.

Jede TOM-REF-Installation enthält außerdem eine Modulbibliothek, in der unter anderem der Strukturkatalog abgelegt ist. Erweiterungen des Strukturkatalogs müssen jeweils als neues Modul in diese Bibliothek eingetragen werden. Data Dictionary-Konfigurationen benutzen immer einen gemeinsamen Strukturkatalog (Bild 15). Damit ist die Voraussetzung geschaffen, daß private Data Dictionary-Daten in ein zentrales Data Dictionary übernommen werden können.

Änderungen am Strukturkatalog sollte der Data Dictionary-Administrator vornehmen.

Bei Auswertungen, die gedruckt werden sollen, legt TOM-REF eine Druckdatei an, ferner legt er bei Bedarf Protokoll- und Fehlerdateien an. In TOM-Umgebung können Sie alle diese Dateien in einer gemeinsamen TOM-Bibliothek halten.

BS2000-Umgebung:



TOM-Umgebung:

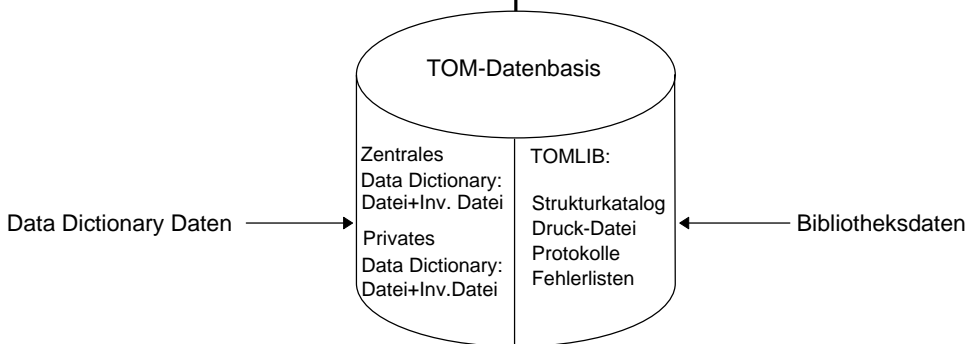
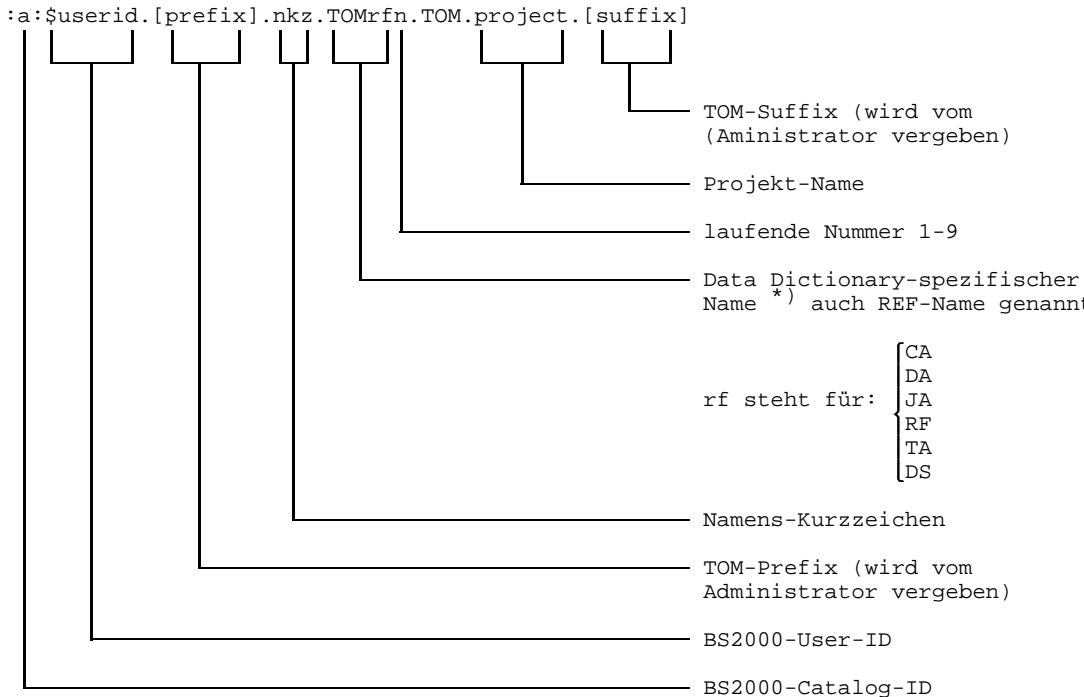


Bild 15 TOM-REF in BS2000- und TOM-Umgebung

3.2.3 Namenskonventionen

Die Namen der beiden Data Dictionary-Dateien, die TOM-REF verwendet, müssen nach den Konventionen gebildet werden, die für TOM gelten. Dadurch ist gewährleistet, daß die Schutzfunktionen von TOM wirksam sind und TOM-REF auch in TOM-Umgebung ablaufen kann.



*) Die zugehörige invertierte Datei heißt TOMrfIn.

Bild 16 Namenskonventionen von TOM-REF

Beispiel

Der Name eines Data Dictionary kann bei Nichtverwendung von Prefix und Suffix z.B. folgendermaßen lauten:

```
:A:$A1234567.LEO.TOMRF1.TOM.BEISPIEL
```

Die Datei mit den inversen Beziehungen hat dann folgenden Namen:

```
:A:$A1234567.LEO.TOMRFI1.TOM.BEISPIEL
```

Eine andere Möglichkeit ist z.B. bei Verwendung von Prefix und Suffix:

```
:C:$B8910112.V11A.LEO.TOMCA9.TOM.BEISPIEL.INST1
```

3.3 Funktionen

Die Funktionen von TOM-REF können in Grund- und Servicefunktionen unterteilt werden. Diese Funktionen werden über Menüs oder Kommandos aufgerufen. Das Zusatzprodukt TOM-REFG bietet eine alternative Masken-Oberfläche und weiteren Service.

3.3.1 Grundfunktionen

Die Grundfunktionen von TOM-REF wirken auf Objekte, Beziehungen und deren Attribute. Dies gilt sowohl für Standardtypen als auch für benutzereigene Erweiterungen. Im einzelnen stehen folgende Grundfunktionen zur Verfügung:

- Einträge anzeigen
- Einträge neu aufnehmen
- Bestehende Einträge ändern
- Einzelne Einträge löschen

Layout der Masken

Die TOM-REF-Masken (Menüs zur Funktionsauswahl und Masken für Objekte, Beziehungen, Attribute) werden in den folgenden Kapiteln im Detail dargestellt. Hier soll lediglich das Grundprinzip gezeigt werden, nach dem die Masken aufgebaut sind (siehe Bild 17).

Das Beispiel zeigt die Maske für den Objekttyp Modul (MOD). Das Modul, das gerade angezeigt wird (Funktion Change), heißt AVANZ und ist im Data Dictionary mit dem TOM-REF-Namen TOMRF1 eingetragen. Standardmäßig sind folgende Beziehungstypen vorhanden: Modul (MOD) bearbeitet Record (REC) und Modul(MOD) bearbeitet Format (MSK). Es wurden Beziehungen zu den Objekten vom Typ Record mit Namen STEUDA, ASATZ und RETCODE, sowie zu dem Objekt AV02 vom Typ Format aufgebaut.

TR0320	Change	MODULE	[1]					\$A1234567.LEO.TOMRFL.TOM.BEISPIEL										
Name	:	AVANZ	[2]					Creation date: 16.01.95										
								Last change : 18.02.95										
Description	:	Modul zur Auftragsbearbeitung und Anzeige von Auftraegen				Responsible : Leo												
								Text : Y										
Attributes	:	1:	2:	3:	4:	5:	6:	7:	8:									
		Relationships to succeeding objects [3]						Attributes for relationships										
	Type	Object						Type	Occurs	1	2	3	4	5	6	7	8	
.	RECORD	STEUDA						.										
.	RECORD	ASATZ						.										
.	RECORD	RETCODE						.										
.	FORMAT	AV02						.										
.										
.										
.										
.										
Command:	[4]										
[5]																		

Bild 17 Beispiel für eine ausgefüllte TOM-REF-Maske

Erläuterung

Nr.	Bezeichnung	enthält
1	Kopfzeile	<ul style="list-style-type: none"> - Maskennummer - Funktion - Objekttyp-Name - TOM-REF-Dateiname (ohne TOM-Prefix bzw. TOM-Suffix)
2	typ-neutrale Standardattribute	<ul style="list-style-type: none"> - Objektname - Erstellungsdatum - Änderungsdatum - Kurzbeschreibung - Verantwortlicher - Hinweis auf Langtext - Attribute
3	Tabelle der ausgehenden Beziehungen	<ul style="list-style-type: none"> - Spalte zum Editieren der Beziehungen - Objekttyp-Name - Objektname - Beziehungstyp - Occurs-Angabe - Attribute der Beziehung
4	Kommandozeile	
5	Meldungszeile	

Dialog steuern

Mit Kommandos können Sie den Dialog mit TOM-REF beliebig steuern. Wenn Sie keine Kommandos benutzen, wird der Masken-Dialog in einer bestimmten Weise geführt: zur Funktionsauswahl kehrt TOM-REF jedesmal in das Hauptmenü zurück. Die folgenden Tabellen enthalten eine Übersicht der Kommandos mit Kurzbeschreibung ihrer Wirkung.

Kommando	Kurzbeschreibung
CHANGE	Ändern eines Objekts oder einer Beziehung.
CREATE	Erstellen eines Objekts oder einer Beziehung.
DELETE	Löschen eines Objekts oder einer Beziehung.
DISPLAY	Anzeigen eines Objekts oder einer Beziehung.

Tabelle 4 Kommandos für Grundfunktionen

Kommando	Kurzbeschreibung
SERVICES	Anzeigen der Auswahlmaske der Servicefunktionen.
CHECK	Überprüfen von Objekten und Beziehungen auf Verträglichkeit mit dem Strukturkatalog.
COPY	Ausgabe der Maske zur Servicefunktion Rename/copy.
DELSTRUCT	Ausgabe der Maske zur Servicefunktion Delete structure.
DISDIR	Anzeigen von Inhaltsverzeichnissen. Entweder alle Objekte eines Typs bzw. einer Gruppe von Typen oder eine Selektion von Objekten eines oder mehrerer Typen.
DISP-USES	Ausgabe der Maske zur Servicefunktion Display uses.
DISP-WHERE-USED	Ausgabe der Maske zur Servicefunktion Display where-used.
GENERATE COPY	Ausgabe der Maske zur Servicefunktion Generate functions (Generierung von Copy-Elementen).
INCORPORATE	Ausgabe der Maske zur Servicefunktion Incorporate from another Data Dictionary.
INITIATE	Ausgabe der Maske zur Servicefunktion Initiate TOM-REF Data Dictionary (Neues Data Dictionary anlegen).

Tabelle 5 Kommandos für Servicefunktionen

Kommando	Kurzbeschreibung
PRDIR	Drucken eines Inhaltsverzeichnisses aller Objekte oder der Objekte eines bestimmten Typs.
PRINT	Ausgabe der Auswahlmaske für die Servicefunktion Print functions.
REESTABLISH	Servicefunktion Reestablish TOMRFI (Data Dictionary aus TOMLIB holen).
RENAME	Ausgabe der Maske zur Servicefunktion Rename/copy.
RESTORE	Ausgabe der Maske zur Servicefunktion Restore from TOMLIB (Data Dictionary aus TOMLIB holen).
SAVE	Ausgabe der Maske zur Servicefunktion Save into TOMLIB (Data Dictionary in TOMLIB sichern).
UDS	Ausgabe der Maske zum Aufruf des UDS-DDA

Tabelle 5 Kommandos für Servicefunktionen (Fortsetzung)

Kommando	Kurzbeschreibung
EDITOR	Wechseln in den TOM-TI, ggf. mit gleichzeitigem Holen eines Members der TOM-Bibliothek
END/RETURN	Beenden von TOM-REF
HELP	Ausgeben einer Kommandosyntax oder aller gültigen Objekttypen oder der Beziehungen, die von einem Objekt eines bestimmten Typs ausgehen dürfen.
N2CHARTS	Verzweigen nach TOM-REFG (wenn installiert)
PROT ON/OFF	Ein- bzw. Ausschalten der Protokollierung des Maskeninhalts
TEXT DELETE	Text eines Objekts löschen (TOM-REF-Text und TOM-TI-Langtext)
DDTEXT DELETE	Löschen des TOM-REF-Textes eines Objekts.
LIBTEXT DELETE	Löschen des TOM-TI-Langtextes eines Objekts.
TEXT OFF	Ausschalten der Textbearbeitung.
TEXT ON	Einschalten der Textbearbeitung des TOM-REF-Textes.
TEXT ON, LIB=ON	Einschalten der Textbearbeitung des TOM-TI-Langtextes.

Tabelle 6 Kommandos zur Verarbeitung

Die Grundfunktionen sind besonders effektiv mit der alternativen Bildschirm-Oberfläche des Zusatzprodukts TOM-REFG bedienbar, die, wenn vorhanden, von TOM-REF aus mit dem Kommando `N2CHARTS` unmittelbar angesteuert werden kann.

TOM-REFG einsetzen

Mit TOM-REFG können Sie sich am BS2000-Terminal jederzeit einen Überblick über den dokumentierten strukturellen Aufbau von Anwendungssystemen im Data Dictionary verschaffen, z.B. hinsichtlich der Modulhierarchie, einschließlich ihrer Schnittstellen zu verwendeten Masken, Dateien oder sonstigen Systemkomponenten. Darüberhinaus können Sie Strukturen gleich in dieser N²Chart-Oberfläche ändern und neu ins Data Dictionary eingeben. Die für die N²Chart-Anzeige selbst nicht wichtigen Objektattribute lassen sich durch sofortigen Wechsel in die Data Dictionary-Oberfläche von TOM-REF zusätzlich anzeigen bzw. editieren.

TOM-REFG benutzt zur Strukturdarstellung eine N²Chart-Technik (Siehe Bild 18), die keine spezielle grafische Eigenschaft eines Terminals erfordert, sondern auf Basis von Alpha-Terminals realisiert ist.

```

Type: MSK Sel:A-I          Columns: 06  Max. Lines/Field: 02/02/02
+-----+-----+-----+-----+-----+-----+
!AV01      !          !AV04      !AV06      !          !          !
+-----+-----+-----+-----+-----+-----+

Type: MOD Sel: -          Ref: I-To:REC Sel:A- O          I-From:REC Sel:A- R
+-----+-----+-----+-----+-----+-----+
!AVAUS     !STEUDA    !STEUDA    !STEUDA    !          !          !
+-----+-----+-----+-----+-----+-----+
!RETCODE   !AVANZ     !          !          !          !IOFUNC   !
+-----+-----+-----+-----+-----+-----+
!RETCODE   !          !AVUPD     !          !          !IOFUNC   !
+-----+-----+-----+-----+-----+-----+
!RETCODE   !          !          !AVSER     !STEUDA    !          !
+-----+-----+-----+-----+-----+-----+
!          !          !          !RETCODE   !AVINI     !IOFUNC   !
+-----+-----+-----+-----+-----+-----+
!          !ASATZ     !ASATZ     !          !ASATZ     !AVIO     !
!          !RETCODE   !RETCODE   !          !RETCODE   !          !
+-----+-----+-----+-----+-----+-----+

Type: MSK Sel:A- O
+-----+-----+-----+-----+-----+-----+
!          !AV02      !AV04      !          !          !          !
+-----+-----+-----+-----+-----+-----+

C-Object:AVAUS   At Col: 01 Page: ... Trace: ... Store: .. In Mbr: N2CHARTS.NC1

```

Bild 18 Beispiel für N²Chart-Anzeige

Der Mittelteil des Beispiels zeigt die Module eines Programmsystems AVAUS, AVANZ usw., die nach ihrer Aufruffolge auf der Diagonalen von links nach rechts angeordnet sind. Die Felder oberhalb der Diagonalen enthalten die übergebenen, die Felder unter-

halb der Diagonalen die zurückgegebenen Daten. Das Modul AVAUS z.B. ruft das Modul AVANZ und übergibt den Datensatz STEUDA. AVANZ wiederum kehrt zurück zu AVAUS mit dem Datensatz RETCODE. Zusammen bilden die Modulschnittstellen. Zu jedem Modul werden zusätzlich die verwendeten Ein-/Ausgabemasken AV01, AV02, AV04 und AV06 angezeigt (Eingaben im Kopfteil, Ausgaben im Fußteil). Das Modul AVAUS z.B. benutzt die Bildschirm-Maske AV01 für die Eingabe.

Ein N²Chart zeigt in der Regel einen Ausschnitt aus einer Gesamtstruktur. Durch geeignete Einstellungen (Columns und Max.Lines/filed) läßt sich erreichen, daß entweder ein kleiner Ausschnitt mit sehr detaillierten Informationen (z.B. alle zwischen Modulen übergebenen Daten) oder ein größerer Ausschnitt mit entsprechend eingeschränkten Informationen sichtbar wird.

Die anzuzeigenden Informationen werden durch die Angabe von Objekttypen gemäß TOM-REF-Systematik für den Kopf-, Mittel- und Fußteil definiert.

Innerhalb von Objekttypen sind weitere Filterungen möglich (*SEL*). So werden im Beispiel die Eingabemasken im Kopfteil durch ein *I* (für Input) im ersten Attributbyte des *SEL*-Parameters und die Ausgabemasken im Fußteil durch ein *O* (für Output) im zweiten Attributbyte des *SEL*-Parameters identifiziert.

Die Navigation mit der N²Chart-Anzeige durch die Systemstruktur wird durch eine Blätter- und Trace-Funktion unterstützt (Page- und Trace-Eintrag in der Fußzeile).

Die N²Charts können zur weiteren dokumentationstechnischen Bearbeitung mit TOM-DOC in eine TOM-Bibliothek übertragen werden (Store- und Member-Eintrag in der Fußzeile).

3.3.2 Servicefunktionen

Unter dem Namen Servicefunktionen sind die Funktionen zusammengefaßt, mit denen Sie den Bestand in einem Data Dictionary auswerten und verwalten können.

Auswerten

Ein wichtiger Faktor beim Einsatz eines Data Dictionary-Systems ist seine Fähigkeit, seinen Datenbestand nach verschiedenen Kriterien auswerten zu können. TOM-REF kann folgende Aufstellungen liefern:

- Alle Objekte eines bestimmten Typs oder einer Gruppe von Typen, wobei sich die Menge der Objekte noch nach Auswahlkriterien bzgl. des Objektnamens und der einstelligen Attribute eingrenzen läßt.

- Zu einem bestimmten Objekt entweder alle Beziehungen, die von ihm ausgehen oder nur diejenigen Beziehungen zu Objekten eines bestimmten Typs, wobei der gesamte Pfad innerhalb einer Struktur untersucht wird.
- Zu einem bestimmten Objekt entweder alle Beziehungen, die auf dieses Objekt verweisen (inverse Beziehungen) oder nur diejenigen Beziehungen zu diesem Objekt, die von Objekten eines bestimmten Typs ausgehen.
- Allen Objekte, die nicht Ziel einer Beziehung sind.
- Alle Objekte, die zwar Ziel einer Beziehung sind, aber selbst noch nicht erstellt wurden.

Die Auswertungen werden am Bildschirm und/oder in eine Druckdatei ausgegeben. In den beiden letzten Fällen ist auch eine Ausgabe in eine Komponentenliste der angeschlossenen TOMLIB oder PLAM-Bibliothek möglich.

Generieren

Mit TOM-REF können Sie aus Data Dictionary-Einträgen

- verschiedenartige Copy-Strecken zur Verwendung in COBOL-Programmen generieren.
- im Zusammenspiel mit dem Zusatzprodukt UDS-DDA und DDL-Sourcen zur Erstellung eines UDS-Datenbank-Schemas generieren.

Die Copy-Strecken oder DDL-Sourcen erzeugt TOM-REF und legt sie in der angeschlossenen TOMLIB oder PLAM-Bibliothek ab.

Eine Copy-Strecke wird als Objekt vom Typ `COPY (INL)` in das Data Dictionary eingetragen, von dem auch Beziehungen zu Feldern, Feldgruppen usw., die dazu verwendet werden sollen, aufgebaut werden müssen. An Copy-Strecken können generiert werden:

- Datensatz-Beschreibungen
- MOVE-Anweisungen
- Initialisierende MOVE-Anweisungen
- FILE-DESCRIPTION-Anweisungen.

Zum Generieren von SQL- und DDL-Sourcen stehen die Produkte UDS-DDA und SESAM/SQL-DDA zur Verfügung, für die Bearbeitung von SESAM/SQL V2-Datenbanken das Produkt SESAM/SQL-DDA. Beide Produkte sind jeweils in eigenen Handbüchern beschrieben.

Auf die Datenhaltung von TOM-REF greifen auch der Schnittstellengenerator SSGEN und der Teststrahlengenerator TOM-MTEST zu.

Verwalten

Neben einigen einfachen Funktionen wie

- Data Dictionary in einer TOMLIB sichern
- Data Dictionary aus einer TOMLIB holen
- Invertierten Datei wiederherstellen
- Data Dictionary neu anlegen

spielt besonders das Mischen von Data Dictionaries und die Fortschreibung und Pflege des Strukturkatalogs bei der Administration von TOM-REF eine wichtige Rolle. Außerdem können Sie Diagnose-Unterlagen erstellen.

Verschiedene Datenbestände mischen

Das Mischen ist immer dann erforderlich, wenn ein Datenbestand eines Data Dictionary in ein anderes Data Dictionary eingegliedert werden soll, z.B. ein privates Data Dictionary in ein zentrales Data Dictionary oder in ein anderes privates. Folgende Vorgehensweise wird empfohlen:

1. Schritt: **Einzumischenden Datenbestand prüfen**
Der einzumischende Datenbestand wird gegen den Strukturkatalog des Data Dictionary geprüft, in das inkorporiert werden soll. Zu diesem Zweck wird das einzumischende Data Dictionary als privates Data Dictionary eröffnet. Die Eröffnung muß unter der REF-Installation des Data Dictionary erfolgen, in das eingeholt werden soll. Dann wird die Servicefunktion `Check objects and relationships` aufgerufen.
2. Schritt: **Sich widersprechende Objekte umbenennen**
Wenn Verstöße gegen den Strukturkatalog erkannt wurden, dann können die entsprechenden Objekte mit der Servicefunktion `Rename` umbenannt werden.
3. Schritt: **Datenbestände mischen**
Bei der eigentlichen Mischfunktion (`Incorporate`) können Sie entweder dialoggesteuert arbeiten oder aber generell festlegen, wie doppelte Einträge zu behandeln sind: Ob fehlende Einträge bei gleichen Objekten zum Löschen führen sollen, ob ungleiche Beziehungen gelöscht werden sollen, und ob gleiche Objekte älteren Datums generell überschrieben werden sollen. Beim dialoggesteuerten Arbeiten werden jedesmal, wenn gleiche Objekte gefunden werden, beide am Bildschirm angezeigt und Sie können entscheiden, welchen Eintrag mit welchem Objekten und Beziehungen Sie behalten möchten.

3.4 Strukturkatalog erweitern

Zur Fortschreibung und Pflege des Strukturkatalogs gibt es in TOM-REF Assembler-Makros, mit denen Sie Objekttypen und Beziehungstypen definieren können. Dies benötigen Sie, um benutzereigene Objekttypen und Beziehungstypen, auch solche, die von Standard-Objekttypen ausgehen sollen, aufzunehmen (siehe auch Seite 40). Das aus den Makros erzeugte Modul wird in die Nachladebibliothek von TOM-REF eingetragen. TOM-REF greift immer auf dieses Modul zu, so daß Sie sich z. B. mit dem Kommando `HELP` jederzeit die benutzereigenen Erweiterungen anzeigen lassen können.

TOM-REF führt bei der Eingabe von neuen Objektdefinitionen und -beziehungen Prüfungen durch, ob beispielsweise benutzereigene Objekttypen eingegeben werden dürfen oder Beziehungen zwischen Objekttypen zulässig sind. Diese Prüfungen werden anhand der Standardvorgaben in einem Prüfmodul durchgeführt. Die Standardverknüpfungen (Standard-Objektkatalog) können installationsspezifisch erweitert werden.

Erweiterungen des Strukturkataloges werden TOM-REF über eine Prozedur mit dem Namen `TOM-REF.REFVAL.PRC` bekannt gegeben, die ein neues Modul `REFVAL` erzeugt. Gegen dieses Modul werden Definitionen und Beziehungen geprüft.

Beispiel

Am folgenden Beispiel wird eine mögliche Erweiterung des Strukturkataloges erläutert:

```

/PROC N, (&TOMOML=TOM.OML, &INSTUID=TOMREFV3,           -
/          &ASSUID=ASSEMBH, &ASSEMBH=ASSEMBH)
/SYSFILE SYSDTA=(SYSCMD)
/REMARK PLEASE INPUT &INSTUID (INSTALLATION USERID OF TOM-REF WITHOUT $)
/ERASE *
/SYSFILE SYSLST=TOM-REF.REFVAL.ASSEMBLIST
/CAT $&INSTUID..&TOMOML, STATE=U, ACCESS=WRITE, SHARE=NO
/EXEC $&ASSUID..&ASSEMBH
//COMPILE SOURCE = *SYSDTA           -
//, SOURCE-PROPERTIES=STD           -
//, COMPILER-ACTION=MODULE-GENERATION(MODE=STD)           -
//, COMPILER-TERMINATION=STD           -
//, MODULE-LIBRARY = $&INSTUID..&TOMOML.(           -
//ELEMENT = REFVAL (VERSION = *UPPER-LIMIT))           -
//, LISTING=(           -
//MACRO-PRINT=(MACRO-ORIGIN-INFO=INSERTED),           -
//, CROSS-REFERENCE=NO,           -
//, LAYOUT=(LINES-PER-PAGE=50))           -
//, MACRO-LIBRARY=( $&INSTUID..TOM.REF.MACLIB)           -
//, COPY-LIBRARY=*NONE           -
//, TEST-SUPPORT=NO           -
          REFVPG1
*****
* DESCRIPTION OF OTHER TOMREF TYPES AND / OR RELATIONS
* AS DEMONSTRED IN THE FOLLOWING EXAMPLE;
*****
          REFOTH PSE, TPS, 'Pseudocode', 0, -, PSE
          REFREL PSE, (MOD, REC, FLG, FLD)
          REFREL MOD, (PSE)
          REFREL *M, (MOD, PGM, PSE)
          REFGRP *X
          REFREL *X, (CAL, ENP, RLS, RET)
*****
* END OF MACRO DEFINITIONS
*****
          REFVPG2
          END
//END
/STEP
/SYSFILE SYSLST=(PRIMARY)
/CAT $&INSTUID..&TOMOML, STATE=U, ACCESS=READ, SHARE=YES
/ENDP

```


Erläuterung:

- `REFOTH` und `REFREL` sind Schlüsselworte, die bekanntgeben, ob ein neuer, benutzereigener Objekttyp oder neue Beziehung erlaubt werden soll.
- `PSE` ist in diesem Beispiel der neu zuzulassende Objekttyp. Der zugehörige Langtext soll unter dem Typ `TPS` im Data Dictionary geführt werden, die Langbezeichnung für den Objekttyp `PSE` ist "Pseudocode".
- Die weiteren Parameter der `REFOTH`-Anweisung sind für Erweiterungen vorgesehen. 0 und - müssen übernommen werden, als letzter Parameter ist der neu aufzunehmende Objekttyp noch einmal anzugeben.
- Mit der `REFREL`-Anweisung wird festgelegt, welche Beziehungen für den jeweiligen Objekttyp zu anderen Objekttypen erlaubt sein sollen. Die Beziehungen werden additiv zu eventuell bereits bestehenden abgelegt. Es gilt also:
 1. Für einen benutzereigenen Objekttyp werden nur die in der Anweisung `REFREL` angegebenen Beziehungen zugelassen.
 2. Dagegen sind die für einen Standard-Objekttyp in der Anweisung `REFREL` gemachten Angaben zusätzlich zu den Standardbeziehungen erlaubt. Von Objekten der Typen `DAR`, `DSP`, `FKY`, `FLD`, `LIB`, `LTR`, `SDY`, `UTP` und `VOL` dürfen grundsätzlich keine Beziehungen ausgehen. Eine für einen dieser Typen gegebene `REFREL`-Anweisung ist wirkungslos.

Auf das Beispiel bezogen heißt das: Ein Objekt des neu definierten Objekttyps `PSE` darf nur Beziehungen zu den Standardtypen `MOD`, `REC`, `FLG` und `FLD` haben, während vom Objekttyp `MOD` ausgehend neben den erlaubten Standardbeziehungen (`FIL`, `CAL`, `ENP` usw.) zusätzlich Beziehungen zu Objekttypen des Typs `PSE` erlaubt sind.

- Mit der folgenden `REFREL`-Anweisung wird der neu definierte Objekttyp `PSE` in die Standard-Typgruppe `*M` aufgenommen. Die Objekttypen `MOD` und `PGM` sind in der Typgruppe `*M` standardmäßig vorhanden. Sie könnten deshalb auch weggelassen werden.
- Die `REFGRP`-Anweisung definiert eine neue Typgruppe `*X`. In der folgenden `REFREL`-Anweisung werden diejenigen Objekttypen angegeben, die zur Typgruppe `*X` gehören, also `CAL`, `ENP`, `RLS`, `RET`.

4 Mit TOM-REF arbeiten

Dies Kapitel beschreibt, wie die Masken unter TOM-REF aufgebaut sind und wie Sie über die Masken und mit Kommandos die Arbeit in TOM-REF steuern können. Sie erfahren auch, wie Sie Objekte und Beziehungen anlegen und bearbeiten.

4.1 TOM-REF starten und beenden

TOM-REF können Sie auf zwei verschiedene Arten aufrufen:

- als `Special function` aus TomDoors-M
- als Prozedur aus der BS2000-Kommandoebene

Um TOM-REF mit der `Special function` aufzurufen, haben Sie zwei Möglichkeiten:

- Sie kreuzen das Feld `Special function TOMREF` in der Auswahlmaske `02 Selection` von TomDoors-M an.
- Sie geben `SPECIALFUNCTION_(TOMREF)` in der Kommandozeile einer Maske von TomDoors-M ein, falls die `Special function` in der zentralen oder eigenen TOMLIB installiert ist.

Sie können TOM-REF auch aus der BS2000-Kommandoebene als Prozedur aufrufen:

```
/CALL-PROCEDURE $USERID.TOMREF.SPF
```

TOM-REF erwartet dabei folgende Angaben:

- die Kennung, unter der TOM-REF installiert ist (`&INSTUID=`)
- Ihre Benutzerkennung in der Länge von maximal 8 Zeichen (`&USERID=`)
- ein Namenskürzel mit der maximalen Länge von drei Zeichen (`&NKZ=`)
- einen Projekt-Namen mit der maximalen Länge von acht Zeichen (`&PROJECT=`)

Aus diesen Angaben leitet TOM-REF den Namen der TOMLIB und des Data Dictionary ab. Wenn das Data Dictionary noch nicht existiert, wird es angelegt. Eine Beschreibung der Namenskonvention finden Sie auf Seite 46.

Sie beenden TOM-REF mit

- dem Kommando `END` in der Kommandozeile
- dem Kommando `RETURN` in der Kommandozeile
- der Taste `[K]` in der Identifikationsmaske TR0000

4.2 Allgemeiner Maskenaufbau

Die TOM-REF-Masken haben entsprechend den TOM-Konventionen generell folgenden Aufbau:

- Statuszeile [1]
mit Maskennummer, Funktion sowie Kennung und Dateiname des aktuellen Data Dictionary.
- Arbeitsbereich [2]
mit Ein- und Ausgabefeldern zur Abwicklung des Benutzerdialogs für jede Funktion.
- Kommandozeile [3]
für die Eingabe von TOM-REF-Kommandos
- Meldungsfenster [4]
für Vollzugsmeldungen, Fehlermeldungen und Rückfragen von TOM-REF.

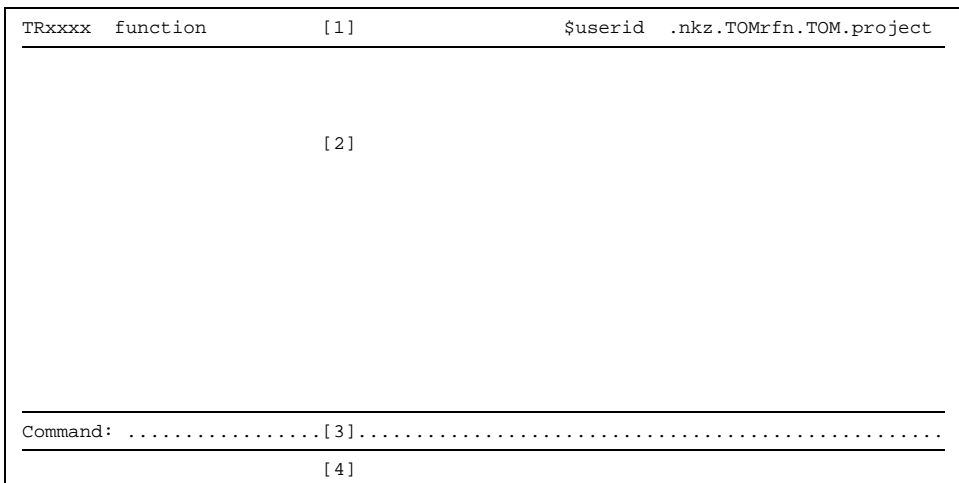


Bild 19 Allgemeiner Aufbau der TOM-REF-Masken

4.3 Data Dictionary auswählen

Wenn Sie TOM-REF zum erstenmal aufgerufen haben, oder wenn das Data Dictionary neu erstellt wurde, kann TOM-REF noch keine Strukturen anzeigen. Es bestehen nun prinzipiell drei Möglichkeiten, das Data Dictionary mit Einträgen zu versorgen:

- durch die Angabe eines Data Dictionary, das von einem Analyse-Programm erzeugt wurde;
- durch das Mischen verschiedener Data Dictionaries mit der Servicefunktion `Incorporate` (siehe Seite 151).
- durch Dateneingabe im Dialog.

Die Dialogeingabe dient vor allem zum Entwurf eines Systems. Sie müssen in diesem Fall alle Objekte und deren Beziehungen explizit in das Data Dictionary eintragen. Dazu verwenden Sie die Masken, die in den Kapitel 6-10 beschrieben sind. TOM-REF überprüft bei der Dialogeingabe, ob die angegebenen Objekte und Beziehungen erlaubt sind.

Das Data Dictionary, das von einem der Analyse-Programme TOM-CA, TOM-JA, TOM-TA oder den Analysebausteinen von UDS-DDA bzw. SESAM-DDA erzeugt wurde, dient zur Nachdokumentation eines bestehenden Systems. Die dabei erzeugten Objekte enthalten die Standardwerte. Erläuternde Texte und benutzereigene Attribute müssen Sie noch selbst hinzufügen. Um mit diesem Data Dictionary zu arbeiten, müssen Sie in der Identifikationsmaske TR0000 den Namen der Datei angeben.

In der Eingangsmaske TR0000 legen Sie ein privates und ein zentrales Data Dictionary fest. Sie müssen immer ein privates Data Dictionary angeben; dagegen ist die Angabe des zentralen optional.

Die Felder unter `Private` sind mit Werten vorbelegt, die Sie beim Aufruf des TOM-REF angegeben haben oder die in der Aufruf-Prozedur standardmäßig vorgegeben sind. Sie können die Werte, mit Ausnahme von `Catid`, überschreiben.

Die Felder `Catid` sind schreibgeschützt. Sie enthalten die jeweilige Katalogkennung, falls der Administrator diese zu Dokumentationszwecken in die Aufruf-Prozedur geschrieben hat. TOM-REF arbeitet grundsätzlich mit der Default-Katalogkennung.

```

TR0000  Identification                               $userid  .nkz.TOMrfl.TOM.project
-----
TOM-REF V3.0A activated

TOM-REF Data Dictionaries:

  Private
  - Catid:           :....:
  - Userid:          $userid
  - Identification:  nkz
  - REF-name:       TOMrfl
  - Project:        project

  Central
  - Catid:           :....:
  - Userid:          $.....
  - Identification:  ...
  - REF-name:       TOM...
  - Project:        .....

-----
Command: .....

```

Bild 20 Identifikationsmaske TR0000 zur Auswahl der Data Dictionaries

Für die Angabe von Data Dictionaries sind folgende Konventionen zu beachten:

- Der REF-Name muß aus den drei Buchstaben TOM und einer ergänzenden Kennung bestehen. Die ergänzende Kennung erhält eine Kennziffer aus dem Bereich 1 bis 9. Es gelten standardmäßig folgende Ergänzungszeichen:
 - CAn (TOM-CA)
 - DAn (UDS-DDA)
 - DSn (SESAM/SQL-DDA und TOM-REF)
 - JAn (TOM-JA)
 - RFn (TOM-REF)
 - TAn (TOM-TA)

Weitere Informationen zur Namenskonvention finden Sie auf Seite 46. Wenn Sie ein Data Dictionary angegeben haben, das nicht existiert, dann wechselt TOM-REF zur Servicefunktion `Initiate TOM-REF Data Dictionary` (siehe auch Seite 167). Sie können dann ein neues Data Dictionary einrichten.

- Die Existenz der TOMRFI-Datei (Inverse Beziehungen) zu einem Data Dictionary ist Voraussetzung, damit Sie mit dem Data Dictionary arbeiten können. Gibt es zu einem Data Dictionary keine TOMRFI-Datei oder ist sie leer, erfolgt eine Abfrage, ob die Datei neu angelegt werden soll (Meldung TR-179). Nach der Bestätigung der Vorbelegung `Y` mit der Taste **(DÜ)** wird die zugehörige TOMRFI-Datei angelegt und in der Folgemaske bestätigt (Meldung TR-198). Die TOMRFI-Datei wird nicht angelegt, wenn Sie vor dem Betätigen der Taste **(DÜ)** die Vorbelegung `Y` mit `N` überschreiben oder wenn Sie die Taste **(K1)** betätigen. In beiden Fällen wird das nicht Anlegen der TOMRFI-Datei durch die Meldung TR-80 in der Folgemaske bestätigt.

- Wenn sich ein bestehendes System nur mit mehreren Analyseläufen erfassen läßt, dann können mit der Servicefunktion `Incorporate` die verschiedenen Datenbestände der Data Dictionaries in einem einzigen Data Dictionary zusammengefaßt werden. Sie erhalten damit eine umfassende Nachdokumentation eines bestehenden Systems.

Wenn das angegebene Data Dictionary noch nicht existiert, dann verzweigt TOM-REF zur Servicefunktion `Initiate TOM-REF Data Dictionary` und erwartet von Ihnen die Bestätigung, daß das Data Dictionary angelegt werden soll. Ansonsten wird die Auswahlmaske TR0001 aufgerufen.

Wenn Sie ein zentrales Data Dictionary angegeben haben, dann überprüft TOM-REF vor der Ausführung jeder Funktion, die ein Objekt verändert oder neu in das private Data Dictionary schreibt, ob das Objekt bereits in einer der beiden Data Dictionaries vorhanden ist. Ansonsten wird nur Ihr eigenes Data Dictionary überprüft.



Sie können nur dann ein Objekt im privaten Data Dictionary beschreiben oder ändern, wenn es nicht im zentralen Data Dictionary steht.

TOMLIB-Behandlung

TOM-REF benötigt zum Ablauf grundsätzlich eine dem Data Dictionary zugeordnete TOMLIB oder eine entsprechende PLAM-Bibliothek. Ist keine von beiden Dateien vorhanden, versucht TOM-REF, eine TOMLIB anzulegen. Sie müssen folgende Vorkehrungen treffen, damit TOM-REF eine TOMLIB anlegen kann:

- Sie müssen eine Basis-TOMLIB (evtl. auf einer anderen Kennung) einrichten.
- In der Aufruf-Prozedur müssen Sie den Parameter `BASETOMLIB=` mit dem Dateinamen der Basis-TOMLIB versorgen. Falls die Basis-TOMLIB unter einer anderen Kennung katalogisiert ist, ist der Pfadname (`$userid.dateiname`) anzugeben.

TOM-REF erzeugt die TOMLIB durch Kopieren der Basis-TOMLIB. Wenn die TOMLIB nicht angelegt werden kann, können Sie das Data Dictionary nicht mit TOM-REF bearbeiten und Sie erhalten folgende Fehlermeldung:

```
TR-186      Creation of TOMLIB unsuccessful - give new parameters
```



Die TOMLIB darf nicht durch ein Paßwort geschützt sein, da sonst bestimmte Funktionen nicht ausgeführt werden.

4.4 Funktionen auswählen

Nach der Identifikationsmaske TR0000 wird die Auswahlmaske TR0001 ausgegeben.

Über die Auswahlmaske TR0001 können Sie Funktionen auswählen oder Voreinstellungen tätigen. Nach der Bearbeitung der ausgewählten Funktionen gibt TOM-REF automatisch wieder die Auswahlmaske TR0001 aus. Dann können Sie weitere Funktionen auswählen oder TOM-REF beenden.

```

TR0001 Selection Structure-catalog                               $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Please select object-,rel.ship-types/Sub-menus and functions (e.g. al,f2,r,5)

Object-types:
a . Field (FLD)                i . Function-key (FKY)      Sub-menus (S):
b . Fieldgroup (FLG)          j . Command (CMD)         q . SESAM/SQL (S)
c . Record (REC)             k . Volume (VOL)         r . Process/JCL (S)
d . File (FIL)               l . Sequence (SEQ)       s . UTM (S)
e . Primary-key (PKY)        m . Physical-file (PFL)  t . UDS (S)
f . Secondary-key (SKY)      n . Alias (ALI)
g . Format (MSK)             p . Other                  w . Copy (S)

Relationship-types:
x . Relationship              y . Relationship extended

Functions for object-types/relationship-types:
1 . Display                  2 . Create                  3 . Change                  4 . Delete
Functions without types:
5 . Services                 6 . N2Charts

Command: .....

```

Bild 21 Die Auswahlmaske TR0001

In Maske TR0001 können Sie alle Funktionen von TOM-REF auswählen, Grundfunktionen und Servicefunktionen:

- Einzelne Grundfunktionen (DISPLAY, CREATE, CHANGE, DELETE) zur Bearbeitung von Objekten und Beziehungen
- Auswahlmaske TR9000 für die Servicefunktionen oder direkt einzelne Servicefunktionen aufrufen.
- Bestimmte Funktionen anderer TOM-Produkte, die beim Arbeiten mit TOM-REF nützlich sind:
 - N2Charts/TOM-REFG
 - Editor TOM-TI über Kommando aufrufen

Desweiteren können Sie in der Auswahlmaske TR0001 mit Kommandos folgende Voreinstellungen setzen:

- Protokollierung des Dialogs einschalten / ausschalten.
- Textmaske bei der Bearbeitung von Objekten ausgeben / nicht ausgeben.

Mit dem Kommando `HELP` können Sie Hilfe-Masken ansteuern. Sie erhalten wahlweise entweder eine Beschreibung der Syntax aller Kommandos oder eine Übersicht, welche Objekttypen der Strukturkatalog enthält, bzw. welche Beziehungen zwischen den Objekttypen möglich sind.

Es gibt drei unterschiedliche Methoden der Eingabe in der Auswahlmaske TR0001:

- Ankreuzen mit `x` oder einem beliebigen Zeichen im Arbeitsbereich.
- Eingeben eines Kurzkommandos in der Kommandozeile. Ein Kurzkommando ist eine max. zweistellige Zeichenkombination aus den Kennbuchstaben und Nummern des Arbeitsbereichs (siehe auch Seite 68).
- Eingeben eines Kommandos in der Kommandozeile (siehe auch Kapitel 12).

Mit den beiden ersten Methoden können Sie Funktionen verketten. Dies bedeutet, daß Sie in einer Auswahlmaske mehrere Funktionen ankreuzen bzw. durch Kurzkommandos eingeben können. Bei der Entscheidung für eine der beiden Methoden, müssen Sie sich im Arbeitsbereich der Auswahlmaske TR0001 auskennen. Der Arbeitsbereich ist auf Seite 60 dargestellt.

Das Anstoßen einer Funktion ist mit jeder der drei Eingabemethoden möglich. (Ausnahme: Das direkte Aufrufen einer Servicefunktion oder des Editors TOM-TI ist nur über Kommando möglich.) Voreinstellungen können Sie nur über die entsprechenden Kommandos festlegen. Dasselbe gilt für die Ausgabe der Hilfe-Masken. Wie Sie eine bestimmte Funktion mit der gewählten Eingabemethode aufrufen, erfahren Sie ab Seite 73.

Für die Eingabemethoden Ankreuzen und Kurzkommando gilt: Falls ein Objekttyp nicht explizit in der Auswahlmaske TR0001 aufgeführt ist, so ist er in einem Sub-Menü enthalten. Deshalb muß zur Festlegung dieses Objekttyps zunächst die Auswahlmaske des betreffenden Sub-Menüs angesteuert werden (siehe Seite 69).

Die Objekttypen des Teilstrukturkatalogs für Funktionsentwurf und die benutzereigenen Objekttypen rufen Sie über den Kennbuchstaben `p` (`other`) auf.

4.4.1 Der Arbeitsbereich der Auswahlmaske TR0001

Im Arbeitsbereich der Auswahlmaske TR0001 können Sie durch Ankreuzen mit x oder einem beliebigen Zeichen Funktionen starten. Es können Kennbuchstaben und Nummern mit den jeweils danebenstehenden Bedeutungen angekreuzt werden. Ferner lassen sich aus den Kennbuchstaben und Nummern Kurzkommandos aufbauen und in der Kommandozeile eingeben.

Die Nummern 1 bis 4 bezeichnen die Grundfunktionen `DISPLAY`, `CREATE`, `CHANGE` und `DELETE`. Falls eine dieser Nummern gewählt wird, muß zusätzlich noch ein Kennbuchstabe angekreuzt werden. Die Kennbuchstaben a bis p bezeichnen einzelne Objekttypen, die Kennbuchstaben q bis w Sub-Menüs. Die Sub-Menüs q bis t enthalten Objekt- oder Beziehungstypen, die einem bestimmten Bereich (z.B. UTM, SESAM/SQL) zuzuordnen sind. Jeder Objekttyp, der nicht explizit unter einem Kennbuchstaben a bis p genannt ist, kann über sein Sub-Menü angesprochen werden. Im Sub-Menü des Objekttyps `COPY` (Kennbuchstabe w) legen Sie die Art des Copy-Elements fest (siehe auch Seite 205).

1

Der Kennbuchstabe x steht für Beziehung, der Kennbuchstabe y für die Erweiterung einer Beziehung. Die Auswahlmaske TR9000 für Servicefunktionen erreichen Sie durch Ankreuzen der Nummer 5. Die N2Charts rufen Sie durch Ankreuzen der Nummer 6 auf.

4.4.2 Eingaben in die Auswahlmaske TR0001

Dieser Abschnitt informiert in Abhängigkeit von der Eingabemethode, welche Funktionen ausgewählt werden können und welche Eingaben hierzu nötig sind. Voreinstellungen zur Textbearbeitung oder Protokollierung können über die Kommandos `TEXT` bzw. `PROT` getätigt werden. Die Eingaben in die Auswahlmaske TR0001 müssen Sie mit der Taste abschliessen.

Im Arbeitsbereich ankreuzen

Zunächst wird vorausgesetzt, daß nur eine Funktion ausgewählt wird. In Abhängigkeit von der gewünschten Funktion muß angekreuzt werden:

- Grundfunktion:
 - Ein Kennbuchstabe a - p bzw. w und eine Nummer 1-4, falls der Typ des Objekts im Arbeitsbereich aufgeführt ist.
Zum Ablauf beim Kennbuchstaben w (Objekttyp `COPY`) siehe unten. Nach dem Abschicken der Maske mit der Taste wird eine leere, typspezifische Objektmaske ausgegeben, in dessen Statuszeile die gewählte Funktion übernommen wurde. Wie eine solche Maske zu bearbeiten ist, ist ab Seite 71 beschrieben.

- Ein Kennbuchstabe für ein Sub-Menü (q-t), falls der Typ des Objekts nicht im Arbeitsbereich aufgeführt ist.
Eine Nummer kann angekreuzt werden. Sie wird dann überschreibbar in das folgende Sub-Menü übernommen.
Nach dem Abschicken der Masken mit der Taste **[DÜ]** wird die Maske des Sub-Menüs ausgegeben. Nun ist der gewünschte Objekttyp und, falls noch nicht geschehen, die Grundfunktion anzukreuzen. Nach dem Abschicken dieser Maske mit der Taste **[DÜ]** wird ebenfalls eine leere, typspezifische Objektmaske ausgegeben. Der weitere Ablauf ist identisch mit dem der Kennbuchstaben a-p.
- Ein Kennbuchstabe x bzw. y und eine Nummer 1-4, falls eine Beziehung bzw. die Erweiterung einer Beziehung bearbeitet werden soll.
Nach dem Abschicken der Maske mit der Taste **[DÜ]** wird die Auswahlmaske für Beziehungstypen TR0008 ausgegeben. Der weitere Ablauf ist ab Seite 100 (Beziehungen bearbeiten) beschrieben.
- Auswahlmaske TR9000 der Servicefunktionen: Nummer 5.
Nach dem Abschicken der Maske mit der Taste **[DÜ]** wird die Auswahlmaske TR9000 der Servicefunktionen ausgegeben, den Sie nach den Angaben im Kapitel 5 bearbeiten können.
- N2Charts: Nummer 6 (siehe Handbuch "TOM-REFG" [5])

Sie können auch mehrere Funktionen verketteten, d.h. mehrere Kennbuchstaben oder Nummern ankreuzen. Auch in einem Sub-Menü ist mehrfaches Ankreuzen möglich. In welcher Reihenfolge die Funktionen bearbeitet werden und wie der Ablauf durch die Funktionstasten gesteuert wird, ist ab Seite 91 beschrieben.

Copy-Elemente auswählen

Beim Objekttyp `COPY` (Kennbuchstabe w) kann bereits durch Ankreuzen mit einem der folgenden Zeichen die Art des Copy-Elements festgelegt und somit die Maske für das Sub-Menü übersprungen werden:

- D – DATA-Area
- M – MOVE-Statements
- I – INIT-Statements
- F – FILE-Description

Die Beschreibung der Copy-Masken finden Sie ab Seite 205.

Auswahl zurücksetzen

Wenn Sie irrtümlich eine Funktion angekreuzt haben, setzen Sie diese Auswahl zurück, indem Sie das Kreuz einfach mit einem Leerzeichen überschreiben.

Kurzkommando in der Kommandozeile eingeben

Aus den Kennbuchstaben und den Nummern des Arbeitsbereichs lassen sich für Grundfunktionen zweistellige Kurzkommandos aufbauen. Dies soll an einem Beispiel gezeigt werden:

Beispiel

Ein Objekt vom Typ Feld soll erstellt werden:
 Der Kennbuchstabe für Feld ist a.
 Die Nummer für die Grundfunktion `Erstellen` ist 2.
 Dann lautet das entsprechende Kurzkommando `a2`.

Ferner können Sie folgende einstellige Kurzkommandos eingeben:

Kurzkommando	Bedeutung
5	Auswahlmaske TR9000 der Servicefunktionen ausgeben
6	N2Charts aufrufen
q-t,w	Sub-Menü ausgeben

Nach dem Abschicken der Maske mit der Taste **[DÜ]** ist der weitere Ablauf identisch mit dem bei der Eingabemethode `Ankreuzen` aufgezeigten.

Sie können Kurzkommandos auch verketteten. Dazu müssen Sie die einzelnen Kurzkommandos durch Komma oder Semikolon getrennt in die Kommandozeile eingeben. Die Kurzkommandos werden in der Reihenfolge der Eingabe abgearbeitet.

Kommando in der Kommandozeile eingeben

- Kommando für eine Grundfunktion.
 Zu jeder der Grundfunktionen `DISPLAY`, `CREATE`, `CHANGE`, `DELETE` gibt es ein gleichlautendes Kommando. Syntax und Wirkungsweise dieser Kommandos sind in Kapitel 12 beschrieben.
- Kommando für eine Servicefunktion.
 Das Kommando `SERVICES` führt in die Auswahlmaske TR9000 für Servicefunktionen. In dieser Maske können gezielt eine oder mehrere Servicefunktionen ausgewählt werden. Ferner gibt es für jede Servicefunktion ein eigenes Kommando, mit dem man die Maske TR9000 überspringt und direkt in die Maske für die Servicefunktion gelangt. Syntax und Wirkungsweise dieser Kommandos sind im Kapitel 12 beschrieben.



Wird die Auswahlmaske TR0001 sowohl mit Markierungen im Arbeitsbereich als auch mit Eingaben in der Kommandozeile abgeschickt, so wird das Kommando ausgeführt. Die Markierungen im Arbeitsbereich werden ignoriert.

4.4.3 Typ im Sub-Menü auswählen

Durch Ankreuzen oder Eingeben eines Kurzkommandos in der Auswahlmaske TR0001 können Sie nur diejenigen typspezifischen Objektmasken direkt ansteuern, die explizit im Arbeitsbereich aufgeführt sind (Kennbuchstaben $a-p$ bzw. w). Alle anderen typspezifischen Objektmasken können über eine eigene Auswahlmaske, ihr Sub-Menü, angesteuert werden (Kennbuchstaben $q-t$).

Im Sub-Menü können Sie dann den gewünschte Objekttyp explizit angeben. Falls von Objekttypen des Sub-Menüs erweiterte Beziehungen ausgehen können, so sind die erweiterten Beziehungstypen ebenfalls im Sub-Menü vermerkt. Durch Ankreuzen eines Beziehungstyps gelangen Sie in die typspezifische Beziehungsmaske. In den Teilstrukturkatalogen (siehe die Kapitel 6-10) ist jeweils vermerkt, zu welchem Sub-Menü ein bestimmter Objekttyp oder Beziehungstyp gehört. Sämtliche Sub-Menüs sind in den Kapiteln 6-10 beschrieben.

Zunächst wird gezeigt, wie Sie von der Auswahlmaske TR0001 in ein Sub-Menü gelangen:

- Sub-Menü durch Ankreuzen ansteuern:
In der Auswahlmaske TR0001 muß nur der Kennbuchstabe des Sub-Menüs angekreuzt werden, das den Objekttyp bzw. den Beziehungstyp enthält. Eine Nummer für die Grundfunktion kann angekreuzt werden. Sie wird in das Sub-Menü überschreibbar übernommen.
Nach dem Abschicken der Masken mit der Taste **[DÜ]** gibt TOM-REF das gewünschte Sub-Menü aus.
- Sub-Menü durch Eingeben eines Kurzkommandos ansteuern:
In der Auswahlmaske TR0001 muß ein ein- oder zweistelliges Kurzkommando eingegeben werden. Ein einstelliges Kurzkommando besteht aus dem Kennbuchstaben des Sub-Menüs, ein zweistelliges aus dem Kennbuchstaben des Sub-Menüs und der Nummer der Grundfunktion.
Nach dem Abschicken der Masken mit der Taste **[DÜ]** gibt TOM-REF das gewünschte Sub-Menü aus. War das Kurzkommando zweistellig, ist die Grundfunktion im Sub-Menü bereits überschreibbar markiert.

Ein Sub-Menü können Sie durch Ankreuzen, Kurzkommando oder Kommando nach denselben Regeln bearbeiten, die für die Auswahlmaske TR0001 gelten. Außer den N2Charts können Sie alle Funktionen auswählen und Voreinstellungen festlegen.

Wenn alle Funktionen, die Sie in einem Sub-Menü ausgewählt haben, vollständig bearbeitet sind, gibt TOM-REF automatisch wieder das Sub-Menü aus.

Beispiel

Wenn Sie in der Auswahlmaske TR0001 das Kurzkommando `q` mit der Taste `[DÜ]` abschicken, erhalten Sie das Sub-Menü TR2000 für SESAM/SQL am Bildschirm.

```

TR2000 Selection SESAM/SQL-Types                               $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Please select object-types/relationship-types and functions (e.g. e2,a3,5)

Object-types:
a X SQL-Catalog (RCT)                                         g . SQL-Referential-Constraint (RFC)
b . SQL-Schema (RSM)                                         h . SQL-Storage-Group (RSG)
c . SQL-Base-Table (RBT)                                     i . SQL-User (RUS)
d . SQL-Index (RIX)                                         j . Field (FLD)
e . SQL-Uniqueness-Constraint (RUC)                         k . Physical-File (PFL)
f . SQL-Check-Constraint (RCC)                              l . Volume (VOL)

Relationship-types:
o . Base-Table-Info (XBT)                                    q . Index-Info (XIX)
p . Base-Table-Info ext.                                    r . Index-Info ext.

Functions for object-types:
1 . Display          2 X Create                               3 . Change          4 . Delete
Functions without object-types:
5 . Services
-----
Command: .....

```

Bild 22 Maske TR2000 mit Eingabebeispiel

Mit den Markierungen `a` und `2` wird das Erstellen von `RCT`-Objekten eingeleitet. Dieselbe Wirkung hätte das Kurzkommando `a2`.

4.5 Objekte bearbeiten

Dieser Abschnitt beschreibt den allgemeinen Aufbau der Objektmasken und die vordefinierten, typneutralen Attribute der Objekte. Sie erfahren, wie Sie Objekte erstellen, ändern, löschen oder anzeigen und welche Wirkung Funktionstasten und die Funktionskettung haben.

4.5.1 Die Objektmaske

Die typspezifischen Masken, in denen die einzelnen ausgewählten Objekte bearbeitet werden, sind gleichartig aufgebaut. Sie unterscheiden sich darin, daß für bestimmte Objekttypen neben den neutralen Attributtypen auch noch typspezifische Attributtypen vorhanden sind.

Eine Objektmaske ist folgendermaßen aufgebaut:

TRxxxx	function type	\$userid	.nkz.TOMrfn.TOM.project
Name	:[1].....	Creation date:	..[3]...
Description	:[2].....	Last change	: ..[4]...
	Responsible	: ...[5]
		Text	: . [6]
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.	[7]	
	[15]		
	Relationships to succeeding objects	Attributes for relationships	
Type	Name	Type Occurs	1 2 3 4 5 6 7 8
[8].....[9]....[10].....	[11][12].[13]	[14]
.
.
.
.
<hr/>			
Command:		

Bild 23 Aufbau einer Objektmaske

Die Statuszeile enthält neben der Maskennummer die gewählte Grundfunktion (function) und den gewählten Objekttyp (type). Ansonsten ist der Aufbau der Statuszeile so, wie auf Seite 60 beschrieben.

Erläuterung

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Objektname	Zeichenfolge; Länge und Wertebereich abhängig vom Objekttyp	Name des Objekts, das bearbeitet werden soll.
2	Kurzbeschreibung	beliebiger Text	Kurzbeschreibung des Objekts
3	Erstellungsdatum	tt.mm.jj	Das Erstellungsdatum wird bei der Funktion CREATE automatisch eingetragen.
4	letztes Änderungsdatum	tt.mm.jj	Das Änderungsdatum wird bei der Funktion CHANGE automatisch eingetragen.
5	Verantwortlicher	beliebige Zeichenfolge	Ein Namenskurzzeichen kann bei den Funktionen CREATE oder CHANGE eingetragen werden.
6	Textfeld	Y/N	Automatische Anzeige, ob im Data Dictionary ein TOM-REF-Text vorhanden ist.
7	Einstellige Attribute des Objekts	beliebige Zeichen	Attribute dienen der genaueren Spezifizierung von Objekten. Die ersten fünf sind für künftige Versionen reserviert.
8	Markierspalte	x,i,d,e,m,a,b r,*	Markierzeichen bei den Funktionen CREATE und CHANGE zum Bearbeiten von Beziehungen (Abschnitt 4.5.3.2) und Markierzeichen für erweiterte Beziehungen.
9	Objekttyp	Kurzname oder Langname (z.B. FLG, FIELDGROUP)	Typ des Objekts, zu dem eine Beziehung besteht oder erstellt werden soll.
10	Objektname	beliebige Zeichenfolge; Länge hängt ab vom Typ	Name des Objekts, zu dem eine Beziehung besteht oder erstellt werden soll.

Nr.	Bezeichnung	Wertebereich	Bedeutung
11	Kennzeichen der Beziehung	_ oder *	Das Kennzeichen der einfachen (_) oder erweiterten (*) Beziehung wird automatisch eingetragen.
12	Typ der Beziehung	Einfache B.: Beliebig Erweiterte B.: Standardnamen	Bei einfachen Bez. können Kurznamen für die Bedeutung der Bez. eingetragen werden. (Z.B. MBR für Member oder OWN für Owner)
13	Occurs	Zahl	Wertigkeit der Beziehung: Es kann angegeben werden, wie oft das Zielobjekt in der Beziehung enthalten ist. (Z.B.: Wie oft ist ein Feld in einer Feldgruppe enthalten.)
14	Einstellige Attribute der Beziehung	beliebige Zeichen	Hier kann die Beziehung genauer spezifiziert werden. Die ersten fünf Attribute sind für künftige Versionen reserviert.
15	Typspezifische Attribute	(in Kap.6 beschrieben)	Nach den neutralen Attributen sind, sofern vorhanden, die typspezifischen Attribute aufgeführt.

Die über `Other` ausgewählten Objektmasken (Benutzereigene Objekttypen und Objekttypen des Teilstrukturkatalogs für Funktionsentwurf) unterscheiden sich von der hier dargestellten Objektmaske in folgender Weise: Der Objekttyp steht nicht in der Statuszeile sondern in der ersten Zeile des Arbeitsbereichs vor dem Objektamen, und zwar in der dreistelligen Kurzform. (siehe Seite 203).

4.5.2 Grundfunktionen

Mit den Grundfunktionen `DISPLAY`, `CREATE`, `CHANGE` und `DELETE` können einzelne Objekte einschließlich ihrer Beziehungsliste und ihrer Langtexte bearbeitet werden.

Die Beziehungsliste umfaßt alle Beziehungen, die vom Objekt ausgehen. Einem Objekt können zwei Texte zugeordnet werden. Der Text im Data-Dictionary kann max. 800 Zeichen lang sein und wird als TOM-REF-Text bezeichnet. Der Text in der TOMLIB, der beliebig lang sein kann, wird TOMLIB-Langtext genannt. Zur Bearbeitung eines TOMLIB-Langtextes ruft TOM-REF den Editor TOM-TI auf.

Zur Bearbeitung eines Objekts stehen die folgenden drei Masken zur Verfügung:

- typspezifischer Objektmaske,
zur Bearbeitung der Attribute des Objekts und der Beziehungsliste, soweit diese bereits in der typspezifischen Objektmaske ausgegeben werden kann.
- Beziehungslisten-Maske,
zur Bearbeitung der Fortsetzung der Beziehungsliste. Je nach Länge der Beziehungsliste kann diese Maske mehrmals ausgegeben werden.
- Textmaske,
zur Bearbeitung von Texten. Voraussetzung für die Ausgabe dieser Maske ist, daß die Textbearbeitung eingeschaltet ist. (Kommando `TEXT`) Die Textmaske ist je nach Voreinstellung der Textbearbeitung entweder die TOM-REF-Maske TR0002 (`TEXT ON`) oder die Maske für TOM-TI (`TEXT ON, LIB=ON`).

Ausgangspunkt der Bearbeitung eines Objekts ist die jeweilige typspezifische Objektmaske. Sämtliche typspezifischen Objektmasken sind ab Kapitel 6 dargestellt und beschrieben.

Eine typspezifische Objektmaske steuern Sie entweder über die Auswahlmaske TR0001 oder ein Sub-Menü an (siehe auch Seite 64).

Wenn Sie eine typspezifische Objektmaske über Ankreuzen oder über Kurzkommando aufrufen, ist die Angabe des Objekttyps und der Grundfunktion erforderlich. TOM-REF gibt die typspezifische Objektmaske entweder leer oder mit einem vorbelegten, überschreibbaren Objektnamen aus. Letzteres ist der Fall, wenn Sie während des laufenden TOM-REF-Betriebs bereits ein Objekt des angegebenen Objekttyps mit einer der Grundfunktionen bearbeitet haben. Ausgegeben wird der Name des zuletzt mit einer Grundfunktion bearbeiteten typgleichen Objekts.

Beim Aufrufen einer typspezifischen Objektmaske über Kommando muß TOM-REF (neben dem Objekttyp und der Grundfunktion) der Objektname bekannt sein. Deshalb wird die typspezifische Objektmaske beim Aufruf über Kommando, soweit möglich, bereits mit Inhalt ausgegeben.

Die Beziehungsliste wird, soweit das nicht bereits in der Objektmaske geschehen ist, in der Beziehungslisten-Maske bearbeitet. Diese Maske wird automatisch nach der typspezifischen Objektmaske ausgegeben. Der Aufbau dieser Maske ist ab Seite 84 erläutert.

Die notwendigen Informationen für das Bearbeiten von Texten in der Textmaske finden Sie auf Seite 89.

Das Bearbeiten mehrerer Objekte mit derselben Grundfunktion oder das Ketten von Grundfunktionen erfordert eine Steuerung des Ablaufs durch die Funktionstasten. Die Wirkung der Funktionstasten ist ab Seite 91 beschrieben.

4.5.2.1 Objekt anzeigen [DISPLAY]

Zunächst muß, wie auf Seite 64 gezeigt, die typspezifische Objektmaske angesteuert werden. Bei Ansteuerung über Kommando entfällt der erste Schritt der folgenden Ablaufbeschreibung. In der Statuszeile der Objektmaske muß als Funktion `DISPLAY` stehen.

- **Leere (oder mit einem Objektnamen vorbelegte) Objektmaske:**
Identifizieren des Objekts.

Zur Identifizierung muß der Name des Objekts angegeben werden. Die Maske wird mit der Taste `[DÜ]` abgeschickt. TOM-REF gibt dann die Objektmaske mit Inhalt aus.

- **Objektmaske:** Attribute und Beziehungsliste anzeigen lassen.

Die Objektmaske enthält die Attribute des Objekts. Die Beziehungsliste wird ebenfalls angezeigt, soweit auf der Maske Platz ist. Erweiterte Beziehungen (Markierung * vor dem Typ der Beziehung) können durch Markieren mit `x` oder * in der Markierspalte der betreffenden Beziehungszeile angezeigt werden. (Siehe Absatz "Erweiterte Beziehungen anzeigen".)

1. Fall:

Die Beziehungsliste kann vollständig angezeigt werden, d.h. die Kommandozeile ist leer.

Nach dem Betätigen der Taste `[DÜ]` ist entweder die Anzeige für dieses Objekt beendet oder es wird ein Langtext dieses Objekts ausgegeben. Letzteres ist nur dann der Fall, wenn die Textbearbeitung eingeschaltet ist.

2. Fall:

Die Beziehungsliste kann nicht vollständig angezeigt werden, d.h. in der Kommandozeile ist ein + vorgegeben.

Abschicken der Maske mit der Taste `[DÜ]` führt in die Beziehungslisten-Maske. Falls auf die Anzeige der restlichen Beziehungen verzichtet werden soll, muß das + in der Kommandozeile mit `_` überschrieben werden. Nach dem Abschicken der Maske mit der Taste `[DÜ]` ist dann, wie oben beschrieben, entweder die Anzeige für dieses Objekt beendet oder es wird ein Text dieses Objekts ausgegeben.

- **Beziehungslisten-Maske:** Fortsetzung der Beziehungsliste anzeigen lassen.

Zur Anzeige aller Beziehungen muß die Beziehungslisten-Maske evtl. mehrmals ausgegeben werden. In den Masken kann geblättert werden (siehe Seite 89). Falls ein Vorwärtsblättern möglich ist, ist die Kommandozeile mit + vorbelegt. Das Blättern kann beendet werden durch Überschreiben des + mit `_` nebst Abschicken der Maske mit der Taste `[DÜ]`.

1. Fall:

Alle restlichen Beziehungen können in der Maske angezeigt werden, d.h. die Kommandozeile ist leer.

Mit der Taste **[DÜ]** wird die Anzeige der Beziehungen beendet.

2. Fall:

Es ist mindestens noch eine weitere Folgemaske vorhanden, d.h. in der Kommandozeile ist ein + vorgegeben.

Jeweils mit der Taste **[DÜ]** können alle Folgemasken der Reihe nach abgerufen werden.

– **Textmaske:** Text anzeigen lassen.

Diese Maske wird nur ausgegeben, wenn die Textbearbeitung eingeschaltet ist und wirklich ein Text existiert. Mit der Taste **[DÜ]** wird die Anzeige des Textes beendet, falls es sich um die TOM-REF-Textmaske TR0002 handelt. Andernfalls ist nach der Rückkehr aus dem TOM-TI die Anzeige beendet.

Falls die Grundfunktion über Kommando angesteuert wurde, so wird nach der Anzeige des Objekts wieder die Maske ausgegeben, in die das Kommando gegeben wurde.

Ansonsten ist mit der Ausgabe der leeren typspezifischen Objektmaske die Anzeige des Objekts, samt Beziehungsliste und Langtext, abgeschlossen. Abschicken der leeren Masken mit der Taste **[DÜ]** führt in die jeweiligen Auswahlmaske zurück (TR0001 oder Sub-Menü).

Erweiterte Beziehungen anzeigen

Erweiterte Beziehungen sind durch ein * in der Spalte vor dem Typ der Beziehung gekennzeichnet. Mit dem Anzeigen der erweiterten Beziehung ist gemeint, daß die typspezifische Beziehungsmaske ausgegeben werden soll, der die typspezifischen Attribute der Beziehung enthält.

Erweiterte Beziehungen, die angezeigt werden sollen, müssen in der Markierspalte der betreffenden Zeile mit r oder * gekennzeichnet werden. Jeweils mit der Taste **[DÜ]** werden alle typspezifischen Beziehungsmasken in der durch die Markierung festgelegten Reihenfolge abgerufen. Nach der Ausgabe der letzten markierten Maske erscheint wieder die Maske, in der markiert wurde.



Mit der Grundfunktion `DISPLAY` können auch Objekte des zentralen Data-Dictionary angezeigt werden.

4.5.2.2 Objekt erstellen [CREATE]

Zunächst muß, wie auf Seite 64 gezeigt, die typspezifische Objektmaske angesteuert werden. In deren Statuszeile muß als Funktion `CREATE` stehen. Bei Ansteuerung über Kommando ist bereits der Name des Objekts eingetragen.

- **Leere (oder mit Objektnamen vorbelegte) Objektmaske:** Objekt identifizieren und Attribute erstellen.

Der Name des Objekts muß angegeben werden (Identifizierung).

Falls im privaten oder zentralen Data Dictionary bereits ein Objekt mit diesem Namen vorhanden ist, weist TOM-REF die Eingaben nach dem Abschicken der Masken mit der Taste `[DÜ]` zurück.

Attribute können eingegeben werden. Bei einigen Objekttypen gibt es Attribute, die versorgt werden müssen (Pflicht-Attribute).

Beziehungen können hier oder erst in der Beziehungslisten-Maske erstellt werden. In der Regel lassen sich aus Platzgründen nicht alle Beziehungen bereits in der Objektmaske erstellen. In diesem Fall muß die Beziehungslisten-Maske (evtl. mehrmals) zum Erstellen der restlichen Beziehungen herangezogen werden.

Bei der Bearbeitung der Beziehungsliste ist das Eintragen von Markierzeichen in die Markierspalte möglich (siehe Seite 84).

Falls erweiterte Beziehungen erstellt werden sollen: Siehe Absatz "Erweiterte Beziehungen erstellen".

Nach dem Abschicken der Maske mit der Taste `[DÜ]` gibt TOM-REF die Beziehungslisten-Maske zum Erstellen von Beziehungen aus. Falls bereits in der Objektmaske Beziehungen erstellt wurden, wird dieser mit den bereits erstellten Beziehungen nicht überschreibbar ausgegeben. Andernfalls ist die Beziehungslisten-Maske leer.

- **Beziehungslisten-Maske:** Beziehungsliste erstellen.

Beim Erstellen von Beziehungen in dieser Maske lassen sich zwei Fälle unterscheiden, je nachdem in welcher Form die Beziehungslisten-Maske ausgegeben wird:

1. TOM-REF hat eine leere Maske mit der Aufforderung zur Eingabe von Beziehungen ausgegeben. Im Meldungsfenster steht:

```
TR-24   Add relationships
```

1. Möglichkeit:

Beenden der Erstellung von Beziehungen mit der Taste `[DÜ]`.

2. Möglichkeit:

Es werden Beziehungen erstellt.

Nach dem Betätigen der Taste **[DÜ]** wird die Maske entweder nicht überschreibbar ausgegeben (Weiter ab 2.) oder leer. Letzteres ist der Fall, wenn die Maske vollständig mit Beziehungen ausgefüllt wurde (Weiter ab 1.).

TOM-REF weist Beziehungen zurück, die lt. Strukturkatalog nicht zulässig sind.

Die Beziehungen werden in der Reihenfolge der Erstellung in die Beziehungsliste übernommen (Siehe auch Hinweis).

2. TOM-REF hat eine nicht überschreibbare Maske mit Auflistung der gesamten Beziehungsliste oder eines Teils der Beziehungsliste ausgegeben.

In den nicht überschreibbaren Masken kann geblättert werden. In der Kommandozeile ist stets + vorgegeben (Vorwärtsblättern). Das Blättern kann beendet werden durch Überschreiben des + mit _ nebst Abschicken der Maske mit der Taste **[DÜ]**. Bei der Ausgabe der letzten Folgemasken der Beziehungsliste genügt das Betätigen der Taste **[DÜ]** zum Beenden des Blätterns. In beiden Fällen wird wieder eine leere Maske ausgegeben (weiter ab 1).

(Bei der Anzeige einer erweiterten Beziehung (Kennzeichen * vor dem Typ) gilt: Die Eingabe von r oder * in der betreffenden Markierspalte bewirkt die Ausgabe der typspezifischen Beziehungsmaske. Dieser kann, wie im Absatz "Erweiterte Beziehungen erstellen" beschrieben, bearbeitet werden. Danach erfolgt wieder die Rückkehr in die nicht überschreibbare Beziehungslisten-Maske.)

– **Textmaske:** Text erstellen.

Diese Maske wird nur ausgegeben, wenn die Textbearbeitung eingeschaltet ist. Falls es sich um die TOM-REF-Textmaske TR0002 handelt, so wird der Text eingegeben und die Maske mit der Taste **[DÜ]** abgeschickt. Im Fall der TOM-TI-Maske ist die Erstellung des Langtextes nach der Rückkehr aus dem TOM-TI beendet.

Falls die Grundfunktion über Kommando angesteuert wurde, wird nach dem Erstellen des Objekts diejenige Maske ausgegeben, in dem das Kommando gegeben wurde.

Ansonsten ist mit der Ausgabe der leeren typspezifischen Objektmaske die Erstellung des Objekts, samt Beziehungsliste und Langtext, abgeschlossen. Abschicken der leeren Maske mit der Taste **[DÜ]** führt in die jeweiligen Auswahlmaske zurück (TR0001 oder Sub-Menü).

Erweiterte Beziehungen erstellen

Zwischen Objekten bestimmter Typen können erweiterte Beziehungen erstellt werden. Im Gegensatz zu den bisher beschriebenen einfachen Beziehungen, können den erweiterten Beziehungen zusätzlich typspezifische Attribute zugeordnet werden. Der Typ einer erweiterten Beziehung ist durch einen Standardnamen fest vorgegeben.

Dem jeweiligen Teilstrukturkatalog (siehe Kapitel 6-10) ist zu entnehmen, ob der Typ des Objekts das Erstellen von erweiterten Beziehungen erlaubt. Beispielsweise kann von einem Objekt vom Typ `RBT` eine erweiterte Beziehung zu einem Objekt vom Typ `FLD` aufgebaut werden. Der Standardtyp der Beziehung ist `XBT`.

Zu jedem Standardtyp gibt es eine typspezifische Beziehungsmaske für die typspezifischen Attribute. In den Kapiteln 6-10 sind sämtliche typspezifischen Beziehungsmasken dargestellt und beschrieben.

Vorgehen

1. Beziehungszeile in der Beziehungslisten-Maske oder Objektmaske:

- ▶ Geben Sie den Typ und den Namen des Objekts an, zu dem die erweiterte Beziehung aufgebaut werden soll.
- ▶ Legen Sie den Standardtyp der Beziehung fest.

Es können angegeben werden: `OCCURS`, einstellige Attribute und das Zeichen `*` vor dem Standardtyp zur Kennzeichnung der erweiterten Beziehung. Das Zeichen `*` wird jedoch von TOM-REF automatisch übernommen.

Falls als nächste Maske die typspezifische Beziehungsmaske zur Festlegung der typspezifischen Attribute der Beziehung ausgegeben werden soll, muß in die Markierspalte `*` oder `x` eingetragen werden. Ansonsten wird die Erstellung der erweiterten Beziehung beendet.

- ▶ Schicken Sie die Maske mit der Taste `[DÜ]` ab.

2. Typspezifischer Beziehungsmaske:

Diejenigen typneutralen Attribute, die bereits in der Beziehungslisten-Maske oder in der Objektmaske festgelegt wurden, werden in der typspezifischen Beziehungsmaske übernommen. Die Maske ist überschreibbar.

- ▶ Geben Sie die typspezifischen Attribute ein.
- ▶ Schicken Sie auch diese Maske mit der Taste `[DÜ]` ab. Damit ist die Erstellung der erweiterten Beziehung beendet.

Eine erweiterte Beziehung können Sie in der Markierspalte der Beziehungslisten-Maske oder der Objektmaske auch mehrfach markieren. Nach der letzten typspezifischen Maske gibt TOM-REF wieder die Maske aus, in dem markiert wurde.

i Für Generierungsfunktionen, wie z.B. den Schnittstellengenerator SSGEN (TOM-TI-Toolbörse) oder den COPY-Strecken-Generator, müssen die Beziehungen in einer bestimmten Reihenfolge eingegeben werden. Nachträgliche Änderungen in der Reihenfolge müssen über die Grundfunktion `CHANGE` vorgenommen werden.

4.5.2.3 Objekt ändern [CHANGE]

Zunächst muß, wie auf Seite 64 gezeigt, die typspezifische Objektmaske angesteuert werden. Bei Ansteuerung über Kommando entfällt der erste Schritt der folgenden Ablaufbeschreibung. In der Statuszeile der Objektmaske muß als Funktion `CHANGE` stehen.

- **Leere (oder mit Objektnamen vorbelegte) Objektmaske:** Objekt identifizieren.

Zur Identifizierung muß der Name des Objekts angegeben werden.

Die Maske wird mit der Taste abgeschickt.

TOM-REF gibt dann die Objektmaske mit Inhalt zum Ändern aus.

- **Objektmaske:** Attribute ändern und evtl. in Beziehungsliste markieren.

Die Objektmaske enthält die Attribute des Objekts. Sie können geändert werden. Die Beziehungsliste ist ausgegeben soweit in der Maske Platz vorhanden ist. Sie kann durch das Eintragen von Markierzeichen in der Markierspalte bearbeitet werden (siehe Seite 84).

Falls nur Attribute des Objekts geändert werden sollen, läßt sich die Ausgabe der Beziehungslisten-Masken überspringen, indem in der Kommandozeile das vorgegebene + mit - überschrieben und die Maske mit der Taste abgeschickt wird.

Falls die Beziehungsliste vollständig ausgegeben werden kann und keine neuen Beziehungen ans Ende der Liste angefügt werden sollen, so müssen zur Bearbeitung der Liste bereits jetzt die nötigen Markierungen in der Markierspalte eingetragen werden.

Falls erweiterte Beziehungen erstellt, geändert oder einfache Beziehungen erweitert werden sollen: Siehe Absatz "Erweiterte Beziehungen erstellen oder ändern".

Falls markiert wird, bearbeitet TOM-REF nach dem Abschicken der Maske die Markierung und gibt danach die Beziehungslisten-Maske mit dem Beginn der Beziehungsliste aus. Ansonsten wird die Fortsetzung der Beziehungsliste ausgegeben.

- **Beziehungslisten-Maske:** Beziehungsliste bearbeiten.

Das Bearbeiten der Beziehungsliste umfaßt folgende Tätigkeiten:

- Bearbeiten von Beziehungen durch Markieren in der Markierspalte (siehe Seite 84 und Absatz "Erweiterte Beziehungen erstellen oder ändern")
- Anfügen von Beziehungen ans Ende der Beziehungsliste (wie bei der Grundfunktion `CREATE` beschrieben)

Beim Bearbeiten der Beziehungsliste in dieser Maske lassen sich zwei Fälle unterscheiden, je nachdem in welcher Form die Beziehungslisten-Maske ausgegeben wird:

1. TOM-REF hat eine leere Maske mit der Aufforderung zur Eingabe von Beziehungen ausgegeben.

Falls Beziehungen ans Ende der Beziehungsliste angefügt werden können, steht im Meldungsfenster:

```
TR-24      Add relationships
```

Falls Beziehungen eingefügt werden können (Markieren mit `i`), steht im Meldungsfenster:

```
TR-03      Insert relationships
```

Es können Beziehungen erstellt werden, die TOM-REF in derselben Reihenfolge entweder ans Ende der Beziehungsliste anfügt oder in die Beziehungsliste einfügt.

1. Möglichkeit:

Es werden keine Beziehungen erstellt.

Nach dem Abschicken der Maske mit der Taste `[DÜ]` ist dann entweder das Bearbeiten der Beziehungsliste beendet oder, beim Markieren mit `i`, das Einfügen von Beziehungen.

2. Möglichkeit:

Es werden Beziehungen erstellt.

Nach dem Betätigen der Taste `[DÜ]` wird dieselbe Maske entweder mit Inhalt ausgegeben (weiter ab 2.) oder leer. Letzteres ist der Fall, wenn die zuvor ausgegebene Maske vollständig mit Beziehungen ausgefüllt wurde (weiter ab 1.).

TOM-REF weist Beziehungen zurück, die lt. Strukturkatalog nicht zulässig sind.

2. TOM-REF hat eine Maske mit Inhalt ausgegeben.
Dabei handelt es sich entweder um die vollständige Beziehungsliste oder um eine Folgemaske der Beziehungsliste.
In dieser Maske kann entweder markiert oder (bei Folgemasken) geblättert werden. In der Kommandozeile ist stets `+` vorgegeben (Vorwärtsblättern). Das Blättern kann beendet werden durch Überschreiben des `+` mit `_` nebst Abschicken der Maske mit der Taste `[DÜ]`. Bei der Ausgabe der letzten Folgemaske der Beziehungsliste genügt das Betätigen der Taste `[DÜ]` zum Beenden des Blätterns. In beiden Fällen wird wieder eine leere Maske ausgegeben (weiter ab 1).

– **Textmaske:** Text bearbeiten.

Diese Maske wird nur ausgegeben, wenn die Textbearbeitung eingeschaltet ist. Der Textinhalt kann gelöscht, d.h. auf Blank gesetzt werden, der Texteintrag (im Data Dictionary bzw. in der TOMLIB) bleibt jedoch bestehen. Der Texteintrag kann nur über das Kommando `TEXT DELETE` gelöscht werden.

Mit der Taste `[DÜ]` wird die Bearbeitung des Langtextes beendet.

Falls die Grundfunktion über Kommando angesteuert wurde, wird nach dem Ändern des Objekts diejenige Maske ausgegeben, in dem das Kommando gegeben wurde.

Ansonsten ist mit der Ausgabe der leeren typspezifischen Objektmaske die Änderung des Objekts, samt Beziehungsliste und Text, abgeschlossen. Abschicken der leeren Maske mit der Taste `[DÜ]` führt in die jeweiligen Auswahlmaske zurück (TR0001 oder Sub-Menü).

Erweiterte Beziehung erstellen oder ändern

Hinsichtlich der Bearbeitung einer erweiterten Beziehung sind folgende Fälle von Bedeutung:

1. Erweiterte Beziehung erstellen.

Erweiterte Beziehungen können Sie genauso erstellen, wie ab Seite 77 beschrieben.

2. Erweiterte Beziehung ändern.

Die Änderung einer erweiterten Beziehung wird eingeleitet durch die Eingabe von `x` oder `*` in der Markierspalte. Nach dem Abschicken der Maske mit der Taste `[DÜ]` wird die typspezifische Beziehungsmaske ausgegeben, in dem geändert werden kann. Die Änderungen sind mit der Taste `[DÜ]` abzuschließen. Es erfolgt die Rückkehr in die Maske, in der markiert wurde.

3. Erweiterte Beziehung in einfache Beziehung überführen.

Durch Markieren mit `e` in der Markierspalte wird eine erweiterte Beziehung nach dem Betätigen der Taste `[DÜ]` in eine einfache Beziehung überführt.

4. Einfache Beziehung erweitern.

Eine bereits existierende einfache Beziehung wird unter der Voraussetzung, daß das Beziehungsattribut `TYPE` leer ist (`Type=-`), in folgenden Schritten erweitert und mit den typspezifischen Attributwerten versorgt:

- Markierzeichen `x` in die Markierspalte eintragen. Maske mit der Taste `[DÜ]` abschicken. Die Maske wird erneut, jedoch überschreibbar, ausgegeben.
- Den Standard-Beziehungstyp in das Beziehungsattribut `TYPE` eintragen. Maske mit der Taste `[DÜ]` abschicken. Die Maske wird erneut ausgegeben. Die Beziehung gilt nun als erweitert. Die beiden folgenden Schritte sind zu durchlaufen, falls typspezifische Attribute versorgt werden sollen.

- Markierzeichen \times (oder $*$) in die Markierspalte eintragen. Maske mit der Taste **[DÜ]** abschicken. Nun wird die typspezifische Beziehungs-Maske ausgegeben.
- Versorgen der typspezifischen Attribute. Nach dem Abschicken der Maske mit der Taste **[DÜ]** wird wieder die zuletzt angezeigte Beziehungslisten-Maske ausgegeben. Die Erweiterung der Beziehung ist abgeschlossen.

4.5.2.4 Objekte löschen [DELETE]

Zunächst muß, wie auf Seite 64 gezeigt, die typspezifische Objektmaske angesteuert werden. Bei Ansteuerung über Kommando entfällt der erste Schritt der folgenden Ablaufbeschreibung. In der Statuszeile der Objektmaske muß als Funktion `DELETE` stehen.

- **Leere (oder mit Objektnamen vorbelegte) Objektmaske:** Objekt identifizieren.

Zur Identifizierung muß der Name des Objekts angegeben werden. Die Maske wird mit der Taste **[DÜ]** abgeschickt.

TOM-REF gibt dann die Objektmaske mit Inhalt aus.

Das Objekt kann nicht gelöscht werden, falls es zur Beziehungsliste eines anderen Objekts gehört. TOM-REF gibt dann nur den Namen des Objekts mit folgender Meldung aus:

```
TR-104   This object is part of another und must not be deleted
```

Nach dieser Meldung ist die Löschfunktion für dieses Objekt beendet. Die Löschung eines anderen Objekt dieses Typs kann durch Überschreiben des Objektnamens eingeleitet werden. Wird die Maske ohne Änderung mit der Taste **[DÜ]** abgeschickt, ist die Löschfunktion für den Objekttyp beendet.

- **Objektmaske und ggf. Beziehungslisten-Maske:** Attribute und Beziehungsliste anzeigen lassen.

Vor der Löschung des Objekts werden nach evtl. mehrmaligem Betätigen der Taste **[DÜ]** die Attribute und die Beziehungsliste ausgegeben. Dabei ist jeweils ein + in der Kommandozeile vorgegeben, falls noch eine Folgemaske existiert. In den Folgemasken kann geblättert werden (siehe auch Seite 89). Nach der Anzeige der Objektmaske bzw. der letzten Folgemaske gibt TOM-REF nach dem Betätigen der Taste **[DÜ]** wieder die Objektmaske aus.

- **Objektmaske:** Entscheiden, ob Objekt gelöscht werden soll.

Im Meldungsbereich der Objektmaske steht:

```
TR-01   Do you want to delete object ? (y/n)
```

Falls das Objekt gelöscht werden soll, muß die Maske mit der Taste **[DÜ]** abgeschickt werden.

Falls das Objekt nicht gelöscht werden soll, muß das vorgegebene γ in der Kommandozeile mit N überschrieben und die Maske mit der Taste **[DÜ]** abgeschickt werden.

Falls die Grundfunktion über Kommando angesteuert wurde, wird nach dem Löschen des Objekts diejenige Maske ausgegeben, in dem das Kommando gegeben wurde.

Ansonsten ist mit der Ausgabe der leeren typspezifischen Objektmaske die Löschfunktion für dieses Objekt beendet. Abschicken der leeren Maske mit der Taste **[DÜ]** führt in die jeweilige Auswahlmaske zurück (TR0001 oder Sub-Menü).

Falls das Objekt gelöscht wurde, steht im Meldungsfenster der jeweiligen Masken:

```
TR-02      Object is deleted
```



Das Löschen von Objektstrukturen kann einfacher mit der Servicefunktion `Delete structure` vorgenommen werden (siehe Seite 148).

Die Texte des Objekts werden nicht angezeigt, aber ebenfalls gelöscht.

4.5.3 Beziehungsliste bearbeiten

Die Beziehungsliste eines Objekts umfaßt alle Beziehungen, die von diesem Objekt ausgehen. Zu ihrer Bearbeitung steht die Beziehungslisten-Maske zur Verfügung. Sofern in der typspezifischen Objektmaske Platz vorhanden ist, können auch dort bereits Beziehungen bearbeitet werden. Der Aufbau einer Beziehungszeile im Arbeitsbereich ist in beiden Masken identisch.

Der Aufbau der Beziehungslisten-Maske und die Bedeutung der Eingabefelder ist im nachfolgenden Abschnitt beschrieben.

Bei den Grundfunktionen `CREATE` und `CHANGE` kann die Beziehungsliste durch Markieren in der Markierspalte bearbeitet werden. Die Bedeutung der möglichen Markierzeichen ist ab Seite 86 beschrieben.

4.5.3.1 Die Beziehungslisten-Maske

Die Beziehungslisten-Maske hat für alle Objekttypen denselben Aufbau.

Die Statuszeile wird mit Ausnahme der letzten Ziffer der Maskennummer aus der typspezifischen Objektmaske übernommen. Die letzte Ziffer der Maskennummer der Beziehungslisten-Maske ist immer eine 1.

Nr.	Bezeichnung	Anmerkung zum Inhalt
4	Kennzeichen der Beziehung	Wird automatisch eingetragen. Einfache Beziehung: <code>_</code> . Erweiterte Beziehung: <code>*</code> .
5	Typ der Beziehung	Bei einfachen Beziehungen kann ein Kurzname für die Bedeutung der Beziehung angegeben werden. Dies kann als Kurzform einer Elementaraussage aufgefaßt werden. Die Bezeichnungen COL, FOR, UTR, GRT, VFB und STO sind Standardtypen erweiterter Beziehungen und dürfen auch nur als solche verwendet werden.
6	OCCURS	Anzahl, wie oft das Zielobjekt in der Beziehung vorkommt. Die OCCURS-Angabe kann für Felder und Feldgruppen auch beim Objekt erfolgen (Siehe Hinweis).
7	Einstellige Attribute der Beziehungen	Mit den einstelligen Attributen können Beziehungen näher spezifiziert werden, z.B. für Phasenüberwachungen. Die ersten fünf sind für TOM-REF-Erweiterungen reserviert.

FILLER

In den Beziehungen können auch Beziehungen zu `FILLER` angegeben werden. Dabei ist es nicht erforderlich, diese Felder zu definieren. Es müssen folgende Angaben gemacht werden:

Objekttyp: `FLD`
 Objektname: `FILLER(99)`

99 gibt in COBOL-Notation die Anzahl der Zeichen des `FILLERS` an.



Hinsichtlich der `OCCURS`-Attribute bei Feldern und Feldgruppen gilt: Das `OCCURS`-Attribut bei der Beziehung hat Vorrang vor dem `OCCURS`-Attribut beim Objekt.

4.5.3.2 Beziehungen markieren

Die Beziehungsliste zum Objekt kann bei der Grundfunktion `CREATE` oder `CHANGE` über die Beziehungslisten-Maske (oder soweit möglich über die Objektmaske) durch Markieren bearbeitet werden. Es können die in der folgenden Tabelle beschriebenen Zeichen in der Markierspalte des Arbeitsbereichs eingegeben werden.

Zeichen	Funktion	Bedeutung
x	change	Ändern der Beziehung
i	insert	Einfügen von neuen Beziehungen
d	delete	Löschen der angegebenen Beziehung
e	delete extension	Erweiterung einer Beziehung löschen. Die einfache Beziehung bleibt erhalten.
m	move	Markieren der Beziehung für weitere Bearbeitung (Siehe a, b).
a	after	Die mit d markierte Beziehung wird nach der mit a markierten Beziehung eingefügt.
b	before	Die mit m markierte Beziehung wird vor der mit b markierten Beziehung eingefügt.

Tabelle 7 Markierfunktionen in der Beziehungslisten-Maske

Ablauf der einzelnen Markierfunktionen:

- x Die mit x markierten Beziehungen sind nach dem Abschicken der Maske mit der Taste **DÜ** zur Änderung freigegeben. Die Änderungen werden wirksam, wenn die Maske erneut mit der Taste **DÜ** abgeschickt wird.
- i Zunächst werden diejenigen Beziehungen, vor die neue Beziehungen eingefügt werden sollen, mit i markiert. Nach dem Abschicken der Maske mit der Taste **DÜ** wird eine leere Beziehungslisten-Maske ausgegeben. In diesem können, wie bei der Grundfunktion `CREATE` beschrieben (siehe Seite 77), Beziehungen erstellt werden. Das Erstellen der Beziehungen für *eine* Markierung i wird beendet, wenn die Beziehungslisten-Maske nicht vollständig mit Beziehungen gefüllt abgeschickt wird. Die Beziehungen werden in der erstellten Reihenfolge eingefügt.
- d Die mit d markierten Beziehungen werden nach dem Abschicken der Maske mit der Taste **DÜ** unmittelbar gelöscht.
- e Nach dem Abschicken der Maske mit der Taste **DÜ** wird für die erste mit e markierte Beziehung die typspezifische Beziehungsmaske mit Inhalt ausgegeben. Im Meldungsfenster steht:
- TR-412 Do you want to delete extension (Y/N)
- Falls die Erweiterung gelöscht werden soll, muß die Maske mit der Taste **DÜ** abgeschickt werden.

Falls die Erweiterung nicht gelöscht werden soll, so ist zunächst das in der Kommandozeile vorgegebene γ mit \mathbb{N} zu überschreiben. Danach ist die Maske mit der Taste $\overline{\text{DÜ}}$ abzuschicken.

Der Vorgang wiederholt sich, falls noch weitere Beziehungen mit e markiert wurden.

m, a, b :

Es können mehrere Beziehungen mit m markiert werden. Falls die Beziehungen nicht in einer Maske stehen, kann in die entsprechenden Masken geblättert werden. Schließlich muß noch das Ziel der mit m markierten Beziehungen mit a bzw. b markiert werden. Nach dem Abschicken der Maske mit der Taste $\overline{\text{DÜ}}$ werden die mit m markierten Beziehungen in der Reihenfolge der Markierung an das mit a bzw. b markierte Ziel gebracht.

Diese Abläufe können auch gekettet werden, d.h. mehrfaches Markieren in einer Maske ist auch mit verschiedenen Markierzeichen möglich. Die Markierungen werden dann in folgender Reihenfolge abgearbeitet:

1. x
2. e
3. i
4. $d; m, a, b$

Innerhalb jedes der vier Schritte ist der Ablauf so, wie oben gezeigt.

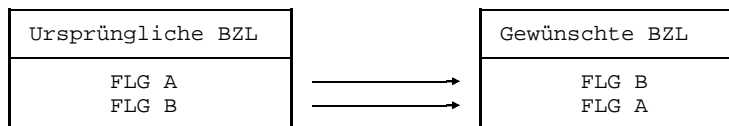
Falls zusätzlich noch mit den Zeichen $*$ zw. r für erweiterte Beziehungen markiert wird, so werden diese in der Rangfolge im 2. Schritt abgearbeitet.

Sonderfall

Für bestimmte Änderungen an einzelnen Beziehungen ist die Markierfunktion x nicht ausreichend, was anhand des folgenden Beispiels demonstriert wird.

Beispiel

Die Beziehungsliste (BZL) eines Objekts vom Typ Record soll geändert werden, und zwar so, wie es die Pfeile in der folgenden Abbildung andeuten.



Werden die beiden Änderungen durch zweifaches Markieren mit x eingeleitet, so führt dies schließlich zu einer Fehlermeldung (TR-51). Die gewünschten Änderungen werden nicht wirksam. Der Grund ist, daß TOM-REF die Eingaben einer Maske auf Duplikate überprüft. In diesem Beispiel wären die Zielobjekte

FLG A und FLG B für TOM-REF doppelt vorhanden. Deshalb werden die Änderungen abgewiesen. Ein korrekter Weg, die gewünschten Änderungen einzubringen, wäre:

1. Zielobjekt FLG B löschen
2. Zielobjekt FLG A in FLG B ändern
3. Zielobjekt FLG A erstellen

4.5.3.3 In der Beziehungsliste blättern

Falls zur Beziehungsliste Folgemasken existieren, so kann in diesen vorwärts, rückwärts, an den Anfang oder an das Ende der Beziehungsliste geblättert werden.

Im einzelnen gibt es folgende Kommandos zum Blättern:

- + : Vorwärts um eine Maske.
- : Rückwärts um eine Maske.
- ++ : Ende der Beziehungsliste.
- : Anfang der Beziehungsliste.
- ⌵ : Beenden des Blätterns.

4.5.4 Texte bearbeiten

Jedem Objekt können zwei Arten Text zugeordnet werden:

- TOM-REF-Text: er kann maximal 800 Zeichen lang sein und wird in der TOM-REF-Textmaske TR0002 bearbeitet.
- TOM-TI-Langtext: er kann beliebig lang sein und wird in der TOM-TI-Maske bearbeitet.

Bei der Bearbeitung eines Objekts wird entweder einer der beiden genannten Textmasken ausgegeben oder keiner. Welcher Fall eintritt ist zunächst von der Voreinstellung in der Aufruf-Prozedur abhängig. Falls dort der Parameter `TEXT=N` (Standard) angegeben ist, wird keine Textmaske ausgegeben, andernfalls (`TEXT=Y`) wird die TOM-REF-Textmaske ausgegeben. Diese Voreinstellung können Sie jedoch mit dem Kommando `TEXT` ändern. Eine Beschreibung dieses Kommandos finden Sie in Kapitel 12.

4.5.4.1 Die TOM-REF-Textmaske

Die TOM-REF-Textmaske ist in Bild 25 dargestellt.

Die Statuszeile ist mit Ausnahme der Maskennummer identisch mit der Statuszeile der zugehörigen Objektmaske. Die Maskennummer der Textmaske ist stets `TR0002`. Im Arbeitsbereich kann der Text gemäß der gewählten Grundfunktion bearbeitet werden.

Mit dem Kommando `EDIT C=name1,T=type1` kann ein TOMLIB-Element generiert werden, das keinem Objekt zugeordnet ist. Ein Löschen dieses TOMLIB-Elements kann damit nicht über TOM-REF-Kommandos erfolgen. Es besteht jedoch folgende Möglichkeit zum Löschen: Zunächst wird das Kommando `EDIT C=name2,T=type2` für ein beliebiges anderes TOMLIB-Element abgesetzt. In der TOM-TI-Maske können Sie dann mit folgendem Kommando das TOMLIB-Element (name1,type1) löschen:

```
DE L=tomlib(E=name1.type1)
```

4.5.5 Funktionstasten und Funktionskettung

Bei der Auswahl einer Funktion in der Auswahlmaske TR0001 oder in einem Sub-Menü gilt im Bezug auf die Eingabemethode:

Mit den Kommandos `DISPLAY`, `CREATE`, `CHANGE`, `DELETE` können Sie nur eine Funktion für ein Objekt bearbeiten. Anders verhält es sich beim Ankreuzen und den Kurzkommandos: Durch Kettung können mehrere Objekte unterschiedlicher Typen mit verschiedenen Funktionen bearbeitet werden. Die Ablaufsteuerung für diese Fälle ist auf den Seiten 92 und 96 beschrieben.

4.5.5.1 Wirkung der Funktionstasten

K3 Maskeninhalte wiederherstellen

Der ursprüngliche Zustand der Maske wird wiederhergestellt, z.B. wenn die Maske durch eine Meldung verschoben wurde.

K2 TOM-REF unterbrechen

Die Wirkung der Taste **K2** hängt davon ab, wie Sie TOM-REF gestartet haben:

1. TOM-REF wurde in einer BS2000-Prozedur gestartet.
Mit der Taste **K2** wird in den BS2000-Kommandomodus verzweigt. Mit dem Systemkommando `/RESUME-PROCEDURE` wird wieder an der Stelle aufgesetzt, an der die Funktion innerhalb von TOM-REF unterbrochen wurde.
2. TOM-REF wurde im TomDoors-M gestartet:
Mit der Taste **K2** wird in den TomDoors-M verzweigt. Mit dem TOM-Kommando `RES[UME]` wird wieder an der Stelle aufgesetzt, an der die Funktion innerhalb von TOM-REF unterbrochen wurde.

Wenn nach der Unterbrechung nicht der gesamte ursprüngliche Maskeninhalte ausgegeben wird: Taste **K3** drücken (s.o.).

K1 Funktion abbrechen

Die Wirkung der Taste **K1** hängt von Maske ab, in dem sie gedrückt wird:

Objektmaske: Abbruch der Funktion.

Beziehungslisten-Maske: Beendigung der Bearbeitung der Beziehungsliste. Es wird die Bearbeitung der momentan anstehenden Markierung beendet.

Textmaske: Abbruch der Funktion.

Beim Ankreuzen und den Kurzkommandos in der Auswahl-Maske TR0001 oder in einem Sub-Menü hat die Taste **K1** zusätzlich noch eine andere Wirkung. Mehr dazu in den beiden folgenden Abschnitten.

4.5.5.2 Funktionen durch Ankreuzen verketteten

Im Arbeitsbereich der Auswahlmaske TR0001 können Sie mehrere Objekttypen, Sub-Menüs und Funktionen auf einmal ankreuzen. So legen Sie mehrere Funktionen fest, die dann in einer bestimmten Reihenfolge ablaufen. Die Reihenfolge der Abarbeitung wird bestimmt durch die Reihenfolge der Objekttypen, Sub-Menüs und Funktionen in der Auswahlmaske TR0001.

Dies läßt sich am einfachsten an einem Beispiel demonstrieren. In *Beispiel 1* wird zunächst der Fall betrachtet, daß Objekttypen, aber keine Sub-Menüs angekreuzt werden.

Beispiel 1

In der Auswahlmaske TR0001 werden die Objekttypen `Field` und `Fieldgroup`, sowie die Funktionen `CREATE` und `CHANGE` angekreuzt. Diese Eingabe bewirkt vier einzelne Funktionen mit fester Reihenfolge:

1. `Field CREATE` (Felder erstellen)
2. `Field CHANGE` (Felder ändern)
3. `Fieldgroup CREATE` (Feldgruppen erstellen)
4. `Fieldgroup CHANGE` (Feldgruppen ändern)

Innerhalb jeder Funktion können beliebig viele Objekte bearbeitet werden. Wurden alle gewünschten Objekte innerhalb der Funktion bearbeitet, erfolgt ein Wechsel der Funktion. Nach der zweiten Funktion erfolgt zusätzlich ein Wechsel des Objekttyps. Statt der Objektmaske für `Field` wird nun die Objektmaske für `Fieldgroup` ausgegeben. Ist die letzte Funktion beendet, wird wieder die Auswahlmaske TR0001 ausgegeben.

```

TR0001 Selection Structure-catalog          $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Please select object-,rel.ship-types/Sub-menus and functions (e.g. a1,f2,r,5)

Object-types:                               Sub-menus (S):
a x Field (FLD)                             i . Function-key (FKY)    q . SESAM/SQL (S)
b x Fieldgroup (FLG)                       j . Command (CMD)       r . Process/JCL (S)
c . Record (REC)                          k . Volume (VOL)       s . UTM (S)
d . File (FIL)                            l . Sequence (SEQ)     t . UDS (S)
e . Primary-key (PKY)                     m . Physical-file (PFL)
f . Secondary-key (SKY)                   o . Alias (ALI)
g . Format (MSK)                          p . Other                w . Copy (S)

Relationship-types:
x . Relationship                            y . Relationship extended

Functions for object-types/relationship-types:
1 . Display                               2 x Create                3 x Change                4 . Delete
Functions without types:
5 . Services                             6 . N2Charts
-----
Command: .....

```

Bild 26 Maske TR0001 mit Eingabebeispiel

Wie die einzelnen Wechsel einzuleiten sind, wird im folgenden allgemeingültig aufgezeigt.

Weiteres Objekt bearbeiten:

Ist die Bearbeitung eines Objekts abgeschlossen, wird wieder die leere typspezifische Objektmaske ausgegeben. Es können Einträge für ein weiteres Objekt vorgenommen werden. Das hierzu notwendige Vorgehen wurde auf Seite 73 für alle Grundfunktionen beschrieben.

Funktion wechseln:

Das Drücken der Taste **[DÜ]** in einer leeren Objektmaske bewirkt den Wechsel der Funktion.

Von der letzten Funktion eines Objekttyps erfolgt der Wechsel zur ersten Funktion des nächsten Objekttyps. Nach der letzten Funktion des letzten Objekttyps ist die Bearbeitung beendet.

Objekttyp wechseln:

Das Drücken der Taste **[KI]** in einer leeren Objektmaske bewirkt den Wechsel des Objekttyps.

Es geht weiter mit der ersten Funktion des nächsten Objekttyps. Wurde kein weiterer Objekttyp angekreuzt, ist die Bearbeitung beendet.

Die Wirkung der Funktionstasten für das oben genannte Beispiel zeigt nochmals das folgende N²Chart.

TR0001	DÜ1			
	CREATE FLD	DÜ1 leer	K1	
		CHANGE FLD	DÜ1 leer	
K1			CREATE FLG	DÜ1 leer
K1 DÜ1 leer				CHANGE FLG

Bild 27 Wirkung der Funktionstasten

Es bleibt noch die Frage zu klären, wie der Ablauf ist, wenn in der Auswahlmaske TR0001 die gewünschten Objekttypen nicht explizit aufgeführt sind und deshalb Sub-Menüs angekreuzt werden müssen. Dies soll im *Beispiel 2* demonstriert werden.

Beispiel 2

In der Auswahlmaske TR0001 werden die Sub-Menüs UTM und SESAM/SQL V2 und die Funktionen CREATE und CHANGE angekreuzt.

```

TR0001 Selection Structure-catalog                $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Please select object-,rel.ship-types/Sub-menus and functions (e.g. al,f2,r,5)

Object-types:                                     Sub-menus (S):
a . Field (FLD)                                  i . Function-key (FKY)      q X SESAM/SQL (S)
b . Fieldgroup (FLG)                             j . Command (CMD)         r . Process/JCL (S)
c . Record (REC)                                 k . Volume (VOL)          s X UTM (S)
d . File (FIL)                                   l . Sequence (SEQ)        t . UDS (S)
e . Primary-key (PKY)                            m . Physical-file (PFL)
f . Secondary-key (SKY)                          o . Alias (ALI)
g . Format (MSK)                                 p . Other                  w . Copy (S)

Relationship-types:
x . Relationship                                 y . Relationship extended

Functions for object-types/relationship-types:
1 . Display          2 X Create          3 X Change          4 . Delete
Functions without types:
5 . Services         6 . N2Charts

-----
Command: .....
-----

```

Bild 28 Maske TR0001 mit Eingabebeispiel

Nach dem Abschicken der Auswahlmaske wird das Sub-Menü TR2000 für SESAM/SQL V2-Objekttypen ausgegeben. Die in der Auswahlmaske TR0001 angekreuzten Funktionen werden automatisch in das Sub-Menü TR3000 übernommen.

```

TR2000  Selection SESAM/SQL-Types                               $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Please select object-types/relationship-types and functions (e.g. e2,a3,5)

Object-types:
a X SQL-Catalog (RCT)                                           g . SQL-Referential-Constraint (RFC)
b X SQL-Schema (RSM)                                           h . SQL-Storage-Group (RSG)
c . SQL-Base-Table (RBT)                                       i . SQL-User (RUS)
d . SQL-Index (RIX)                                           j . Field (FLD)
e . SQL-Uniqueness-Constraint (RUC)                            k . Physical-File (PFL)
f . SQL-Check-Constraint (RCC)                                 l . Volume (VOL)

Relationship-types:
o . Base-Table-Info (XBT)                                       q . Index-Info (XIX)
p . Base-Table-Info ext.                                       r . Index-Info ext.

Functions for object-types/relationship-types:
1 . Display                2 X Create                3 X Change                4 . Delete
Functions without types:
5 . Services

-----
Command: .....

```

Bild 29 Maske TR2000 mit Eingabebeispiel

Das Sub-Menü ist überschreibbar.

Es können die gewünschten SESAM/SQL V2-Objekttypen angekreuzt werden. Werden z.B. die Objekttypen SQL-Catalog und SQL-Schema angekreuzt, so führt das zu folgenden Funktionen:

1. SQL-Catalog CREATE (SESAM/SQL-Datenbank erstellen)
2. SQL-Catalog CHANGE (SESAM/SQL-Datenbank ändern)
3. SQL-Schema CREATE (SESAM/SQL-Schema erstellen)
4. SQL-Schema CHANGE (SESAM/SQL-Schema ändern)

Nach dem Abschicken der Maske mit der Taste **[DÜ]** kann die Abarbeitung der Funktionen, wie im vorangegangenen Beispiel gezeigt, erfolgen. Nach dem Ende der letzten Funktion wird das Sub-Menü TR2000 leer ausgegeben. Die Bearbeitung von SESAM/SQL V2-Objekttypen könnte fortgesetzt werden. Ansonsten wird durch das Drücken der Taste **[KI]** die Bearbeitung der SESAM/SQL V2-Objekttypen beendet.

Nach dem Abschicken der Maske mit der Taste **[DÜ]** wird das Sub-Menü TR8000 für UTM-Objekttypen ausgegeben. Die in der Auswahlmaske TR0001 angekreuzten Funktionen werden automatisch übernommen.

```

TR8000 Selection UTM-Types                               $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Please select object-types/relationship-types and functions (e.g. c2 )

Object-types:
a X UTM-Program (UTP)           i . UTM-TLS (TLS)
b . UTM-TAC (TAC)              j . UTM-LTERM (LTR)
c . UTM-Application (UAP)
d . UTM-GSSB (GSS)
e . UTM-LSSB (LSS)

Relationship-types:
g . Storage-info (STO)         o . Format-info (FOR)
h . Storage-info ext.         p . Format-info ext.

Functions for object-types/relationship-types:
1 . Display                    2 X Create                3 X Change                4 . Delete
Functions without types:
5 . Services

-----
Command: .....
-----

```

Bild 30 Maske TR8000 mit Eingabebeispiel

Das Sub-Menü ist überschreibbar. Sie können die gewünschten UTM-Objekttypen ankreuzen. Wird z.B. der Objekttyp `UTM Program` angekreuzt, so führt das zu folgenden Funktionen:

1. UTM Program CREATE (UTM-Programme erstellen)
2. UTM Program CHANGE (UTM-Programme ändern)

Nach dem Abschicken der Maske mit der Taste `(DÜ)` kann die Abarbeitung der Funktionen, wie im vorangegangenen Beispiel gezeigt, erfolgen. Nach dem Ende der letzten Funktion wird das Sub-Menü TR8000 leer ausgegeben. Die Bearbeitung von UTM-Objekttypen könnte fortgesetzt werden. Ansonsten wird durch das Drücken der Taste `(K1)` die Bearbeitung der UTM-Objekttypen beendet und wieder die Auswahlmaske TR0001 ausgegeben.

4.5.5.3 Funktionen mit Kurzkommandos verketteten

In der Kommandozeile der Auswahlmaske TR0001 können Sie mehrere Kurzkommandos, getrennt durch Komma oder Semikolon eingeben. Die somit festgelegten Funktionen laufen in der Reihenfolge der Eingabe ab. Ein Wechsel der Funktion erfolgt durch Drücken der Taste `(DÜ)` in einer leeren Objektmaske.

Beispiel

```

TR0001 Selection Structure-catalog          $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Please select object-,rel.ship-types/Sub-menus and functions (e.g. a1,f2,r,5)

Object-types:
a . Field (FLD)           i . Function-key (FKY)      Sub-menus (S):
b . Fieldgroup (FLG)      j . Command (CMD)          q . SESAM/SQL (S)
c . Record (REC)         k . Volume (VOL)          r . Process/JCL (S)
d . File (FIL)           l . Sequence (SEQ)        s . UTM (S)
e . Primary-key (PKY)    m . Physical-file (PFL)    t . UDS (S)
f . Secondary-key (SKY)  o . Alias (ALI)
g . Format (MSK)         p . Other                  w . Copy (S)

Relationship-types:
x . Relationship          y . Relationship extended

Functions for object-types/relationship-types:
1 . Display              2 . Create                  3 . Change                  4 . Delete
Functions without types:
5 . Services             6 . N2Charts

-----
Command: d4,a3,a2,b3.....
-----

```

Bild 31 Maske TR0001 mit Eingabebeispiel

Diese Eingabe bewirkt vier Funktionen in fester Reihenfolge:

1. d4 File DELETE (Dateien löschen)
2. a3 Field CHANGE (Felder ändern)
3. a2 Field CREATE (Felder erstellen)
4. b3 Fieldgroup CHANGE (Feldgruppen ändern)

Innerhalb jeder Funktion können beliebig viele Objekte, wie auf Seite 73 beschrieben, bearbeitet werden.

Nach dem Abschicken der Auswahlmaske TR0001 mit der Taste **DU** wird die leere Objektmaske TR0040 für `File` ausgegeben. Es können Objekte des Typs `File` gelöscht werden. Das Drücken der Taste **DU** in einer leeren Objektmaske bewirkt den Übergang zur nächsten Funktion. Dann wird die Objektmaske TR0010 für `Field` ausgegeben und es können Felder geändert werden, usw. Wird schließlich bei der letzten Funktion in der leeren Objektmaske TR0020 für `Fieldgroup` die Taste **DU** gedrückt, so gelangt man wieder in die Auswahlmaske TR0001 zurück.

Wird im Kurzkommando ein Sub-Menü angesprochen, so wird vor dem Ablauf der betreffenden Funktion zunächst das Sub-Menü ausgegeben, in dem die gewünschten Objekttypen angekreuzt werden können.

Beispiel

```

TR0001 Selection Structure-catalog          $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Please select object-,rel.ship-types/Sub-menus and functions (e.g. a1,f2,r,5)

Object-types:
a . Field (FLD)                i . Function-key (FKY)      Sub-menus (S):
b . Fieldgroup (FLG)           j . Command (CMD)          q . SESAM/SQL (S)
c . Record (REC)               k . Volume (VOL)          r . Process/JCL (S)
d . File (FIL)                 l . Sequence (SEQ)        s . UTM (S)
e . Primary-key (PKY)          m . Physical-file (PFL)    t . UDS (S)
f . Secondary-key (SKY)        o . Alias (ALI)
g . Format (MSK)                p . Other                    w . Copy (S)

Relationship-types:
x . Relationship                y . Relationship extended

Functions for object-types/relationship-types:
1 . Display                    2 . Create                  3 . Change                4 . Delete
Functions without types:
5 . Services                    6 . N2Charts

-----
Command: q1,s1.....
-----

```

Bild 32 Maske TR0001 mit Eingabebeispiel

Nach dem Abschicken der Maske mit der Taste **[DÜ]** wird aufgrund des ersten Kurzkommandos `q1` das Sub-Menü TR2000 für SESAM/SQL V2-Objekttypen ausgegeben. Im Arbeitsbereich ist die Funktion 1 angekreuzt.

```

TR2000 Selection SESAM/SQL-Types          $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Please select object-types/relationship-types and functions (e.g. e2,a3,5)

Object-types:
a X SQL-Catalog (RCT)          g . SQL-Referential-Constraint (RFC)
b X SQL-Schema (RSM)           h . SQL-Storage-Group (RSG)
c . SQL-Base-Table (RBT)       i . SQL-User (RUS)
d . SQL-Index (RIX)            j . Field (FLD)
e . SQL-Uniqueness-Constraint (RUC) k . Physical-File (PFL)
f . SQL-Check-Constraint (RCC) l . Volume (VOL)

Relationship-types:
o . Base-Table-Info (XBT)      q . Index-Info (XIX)
p . Base-Table-Info ext.       r . Index-Info ext.

Functions for object-types/relationship-types:
1 X Display                    2 . Create                  3 . Change                4 . Delete
Functions without types:
5 . Services

-----
Command: b1.....
-----

```

Bild 33 Maske TR2000 mit Eingabebeispiel

Die Maske ist überschreibbar. Die gewünschten Objekttypen könnten nun im Arbeitsbereich angekreuzt werden. Dann wäre der weitere Ablauf so, wie im letzten Abschnitt beschrieben. Es kann aber auch mit Kurzkommandos auf der Basis der Bezeichnungen im Sub-Menü TR2000 weitergearbeitet werden. Sollen z.B. alle Schemata einer SESAM/SQL V2-Datenbank angezeigt werden, so wäre das Kurzkommando `b1` in der Kommandozeile einzugeben. Nach dem Abschicken der Maske mit der Taste `[DÜ]` würde die Funktion, wie im vorangegangenen Beispiel gezeigt, ablaufen. Nach dem Ende der Funktion wird das leere Sub-Menü TR2000 ausgegeben. Es können weitere SESAM/SQL V2-Objekttypen bearbeitet werden. Andernfalls wird mit der Taste `[DÜ]` aufgrund des zweiten Kurzkommandos `s1` zum Sub-Menü TR8000 für UTM-Objekttypen gewechselt. Das weitere Vorgehen ist genau so, wie eben für die SESAM/SQL V2-Objekttypen beschrieben. Nach dem Beenden der Funktionen im Sub-Menue TR8000 wird mit der Taste `[DÜ]` in die Auswahlmaske TR0001 zurückgekehrt.

4.6 Beziehungen bearbeiten

Beziehungen können, wie im letzten Abschnitt gezeigt wurde, innerhalb der Beziehungsliste des Ausgangsobjekts bearbeitet werden. Es besteht jedoch auch die Möglichkeit Beziehungen außerhalb der Beziehungsliste zu bearbeiten. Das hierzu notwendige Vorgehen ist Thema dieses Abschnitts. Mit Bearbeiten ist wiederum das Anzeigen, Erstellen, Ändern und Löschen von Beziehungen gemeint. Beziehungen zwischen bestimmten Objekttypen können erweitert sein. In diesem Fall ist das Attribut `TYPE` der Beziehung mit einem der Standardnamen `XBT`, `XIX`, `STO` oder `FOR` belegt. Einfache Beziehungen sind solche, die nicht erweitert oder nicht erweiterbar sind.

Die Bearbeitung wird in der Auswahlmaske TR0001 eingeleitet. Dies kann, wie auf Seite 64 beschrieben, durch Kommando, Kurzkommando oder Ankreuzen geschehen. Die folgende Tabelle zeigt die möglichen Funktionen und ihre Ansteuerung in der Auswahlmaske TR0001.

Funktion	Kommando	Kurzko.	Ankreuzen
Einfache Bez. anzeigen	DISP REL[,T=SIR]	x1	x und 1
Einfache Bez. erstellen	CR REL[,T=SIR]	x2	x und 2
Einfache Bez. ändern	CH REL[,T=SIR]	x3	x und 3
Einfache Bez. löschen	DE REL[,T=SIR]	x4	x und 4
Erweiterte Bez. anzeigen (Zwei Alternativen mit identischer Wirkung)	DISP REL[,T=xxx]	x1	x und 1
	DISP REL EXT[,T=xxx]	y1	y und 1
Erweiterte Bez. erstellen	CR REL[,T=xxx]	x2	x und 2
Erweiterung der Bez. erstellen (Voraussetzung: Type=_)	CR REL EXT[,T=xxx]	y2	y und 2
Erweiterte Bez. ändern	CH REL[,T=xxx]	x3	x und 3
Erweiterung der Bez. ändern	CH REL EXT[,T=xxx]	y3	y und 3
Erweiterte Bez. löschen	DE REL[,T=xxx]	x4	x und 4
Erweiterung der Beziehung löschen (einfache Bez. bleibt)	DE REL EXT[,T=xxx]	y4	y und 4

Die ersten vier Funktionen der Tabelle sind auf Seite 101 beschrieben. Sie beziehen sich ausnahmslos auf einfache Beziehungen (`SIR` bedeutet Simple Relationship). Dabei ist es gleichgültig, ob die Beziehungen erweiterbar sind oder nicht.

Alle anderen Funktionen betreffen erweiterte bzw. erweiterbare Beziehungen. Sie sind ab Seite 104 beschrieben.

Die Kommandos, mit denen Sie Beziehungen ebenfalls bearbeiten können, sind in Kapitel 12 beschrieben.

Die Masken werden in folgender Hierarchie ausgegeben:

1. Auswahlmaske TR0001
2. Auswahlmaske für Beziehungstypen TR0008
3. Beziehungsmaske (TR0200 oder typspezifische Maske)

Mit der Taste **[K1]** kann in die nächst höhere Hierarchiestufe zurückgekehrt werden.

4.6.1 Einfache Beziehungen

Sie können eine einfache Beziehung nur modellieren, wenn das Ausgangsobjekt der Beziehung existiert. Anhand der ausgegebenen Masken läßt sich der Ablauf für alle Funktionen in mehrere Schritte untergliedern. Ausgangspunkt der Bearbeitung ist jeweils die Auswahlmaske TR0001.

Vorgehen:

- ▶ Wählen Sie in der Auswahlmaske TR0001 die entsprechende Funktion aus. Die Wahlmöglichkeiten entnehmen Sie der Tabelle auf Seite 100.

In der hier dargestellten Auswahlmaske wird als Beispiel eine einfache Beziehung über Kommando erstellt.

```

TR0001 Selection Structure-catalog                $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Please select object-,rel.ship-types/Sub-menus and functions (e.g. a1,f2,r,5)

Object-types:                                     Sub-menus (S):
a . Field (FLD)                                   i . Function-key (FKY)   q . SESAM/SQL (S)
b . Fieldgroup (FLG)                              j . Command (CMD)      r . Process/JCL (S)
c . Record (REC)                                  k . Volume (VOL)       s . UTM (S)
d . File (FIL)                                    l . Sequence (SEQ)     t . UDS (S)
e . Primary-key (PKY)                             m . Physical-file (PFL)
f . Secondary-key (SKY)                           o . Alias (ALI)
g . Format (MSK)                                  p . Other               w . Copy (S)

Relationship-types:
x . Relationship                                  y . Relationship extended

Functions for object-types/relationship-types:
1 . Display                                     2 . Create               3 . Change               4 . Delete
Functions without types:
5 . Services                                    6 . N2Charts

-----
Command: CR REL.....

```

Bild 34 Maske TR0001 mit Eingabebeispiel

Mit dem Kommando `CR REL,T=SIR` würde die folgende Maske TR0008 übersprungen.

- ▶ Schicken Sie die Maske mit der Taste **[DÜ]** ab.
- ▶ Kreuzen Sie in der Auswahlmaske TR0008 das Feld `simple relationship` an Die ausgewählte Grundfunktion (`CREATE`) wurde nicht überschreibbar in diese Maske übernommen.

```

TR0008 Selection Relationship-types          $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Please select relationship-types

. Storage-info (STO)
. Format-info (FOR)
. Table-column (XBT)
. Index-column (XIX)

X simple relationship

Functions for relationship-types:
. Display          X Create          . Change          . Delete
-----
Command: .....

```

Bild 35 Maske TR0008 mit Eingabebeispiel

Die Maske TR0008 bietet allgemein folgende Möglichkeiten:

- Beenden der Funktion:
Mit der Taste **[K1]** erfolgt die Rückkehr in die Auswahlmaske TR0001, sofern keine Funktion mehr zu bearbeiten ist. Falls infolge Kettung noch eine weitere Funktion zu bearbeiten ist, wird die erste Maske zu dieser Funktion ausgegeben.
- Bearbeiten einer weiteren Beziehung mit derselben Grundfunktion:
- Ferner kann durch Ankreuzen von erweiterten Beziehungstypen die Bearbeitung von erweiterten Beziehungen mit der aktuellen Grundfunktion eingeleitet werden (siehe auch Seite 104).
- ▶ Schicken Sie die Maske mit **[DÜ]** ab.
- ▶ Geben Sie nun in der Maske TR0200 für einfache Beziehungen `Type` und `Name` für Ausgangs- und Zielobjekt der Beziehung ein. Die Typen können, wie es im Eingabebeispiel der Fall ist, auch in der Kurzform eingegeben werden.

```

TR0200  Create relationship                                $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----

Relationship from Object
Type      Name
FLG..... JAHR.....

to succeeding object                                Attributes for relationships
Type      Name                                     Type Occurs 1 2 3 4 5 6 7 8
FLD.....  MONAT.....                               . . . . .

-----
Command: .....
-----

```

Bild 36 Maske TR0200 mit Eingabebeispiel

- ▶ Schicken Sie die Maske mit der Taste **(DÜ)** ab.
Die Maske wird mit den identifizierenden Angaben erneut ausgegeben. Die Bedeutung der Attributfelder finden Sie in der Tabelle auf Seite 86.
- ▶ Sie können nun in der Maske TR0200 die Beziehung bearbeiten. Die Bearbeitung ist abhängig von der ausgewählten Funktion:
 - Anzeigen:
Die Attribute der Beziehung werden angezeigt.
Die Maske wird mit der Taste **(DÜ)** abgeschickt.
 - Erstellen:
Es können (wie im Eingabebeispiel) Attribute erstellt werden.
Die Maske wird mit der Taste **(DÜ)** abgeschickt.
TOM-REF fügt dann die Beziehung ans Ende der Beziehungsliste des Ausgangsobjekts an.
 - Ändern:
Die Attribute der Beziehung werden angezeigt und können geändert werden.
Typ und Name des Zielobjekts sind ebenfalls überschreibbar. Die Maske wird mit der Taste **(DÜ)** abgeschickt.
 - Löschen:
Die Attribute werden angezeigt. Im Meldungsfenster steht:

TR-411 Do you want to delete relationship (Y/N)

Falls die Beziehung gelöscht werden soll, ist die Maske mit der Taste **[DÜ]** abzuschicken.

Falls die Beziehung nicht gelöscht werden soll, ist zunächst das in der Kommandozeile vorgegebene **Y** mit **N** zu überschreiben. Danach wird die Maske mit der Taste **[DÜ]** abgeschickt.

```

TR0200  Create relationship                               $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Relationship from Object
Type      Name
FLG      JAHR

to succeeding object
Type      Name
FLD      MONAT

Attributes for relationships
Type Occurs 1 2 3 4 5 6 7 8
CON ..12. . . . . . . . . .
-----
Command: .....
    
```

Bild 37 Maske TR0200 mit Eingabebeispiel

4.6.2 Erweiterte Beziehungen bearbeiten

Erweiterte Beziehungen sind nur zwischen Objekten bestimmter Typen möglich. Sie besitzen im Unterschied zu den einfachen Beziehungen noch zusätzlich typspezifische Attribute, welche in der jeweiligen typspezifischen Beziehungsmaske (siehe Kapitel 6-10) bearbeitet werden. So kann z.B. zwischen einem Objekt vom Typ **RIX** und einem Objekt vom Typ **FLD** eine erweiterte Beziehung (**Type=XIX**) erstellt werden. Eine einfache Beziehung zwischen einem Objekt vom Typ **RIX** und einem Objekt vom Typ **FLD** ist erweiterbar, falls das Beziehungsattribut **Type** leer ist (**Type=_**). Umgekehrt lässt sich eine erweiterte Beziehung (**Type=XIX**) in eine einfache Beziehung (**Type=_**) überführen. Den jeweiligen Teilstrukturkatalogen (siehe Kapitel 6-10) ist zu entnehmen, ob eine Beziehung erweitert sein kann oder nicht.

Erweiterte Beziehungen können Sie nur dann erstellen, wenn das Ausgangsobjekt der Beziehung existiert. Anhand der ausgegebenen Masken lässt sich der Ablauf für alle Funktionen in fünf Schritte untergliedern. Ausgangspunkt der Bearbeitung ist jeweils die Auswahlmaske TR0001. Der Ablauf könnte jedoch auch aus einem Sub-Menü oder einer leeren Objektmaske (durch Kommando) eingeleitet werden.

Vorgehen:

- ▶ Wählen Sie in der Auswahlmaske TR0001 die entsprechende Funktion aus. Die Wahlmöglichkeiten entnehmen Sie der Tabelle auf Seite 100.

```

TR0001 Selection Structure-catalog                $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Please select object-,rel.ship-types/Sub-menus and functions (e.g. a1,f2,r,5)

Object-types:
a . Field (FLD)                i . Function-key (FKY)      Sub-menus (S):
b . Fieldgroup (FLG)          j . Command (CMD)          q . SESAM/SQL (S)
c . Record (REC)             k . Volume (VOL)           r . Process/JCL (S)
d . File (FIL)               l . Sequence (SEQ)         s . UTM (S)
e . Primary-key (PKY)        m . Physical-file (PFL)    t . UDS (S)
f . Secondary-key (SKY)      o . Alias (ALI)
g . Format (MSK)             p . Other                  w . Copy (S)

Relationship-types:
x . Relationship              y . Relationship extended

Functions for object-types/relationship-types:
1 . Display                  2 . Create                  3 . Change                  4 . Delete
Functions without types:
5 . Services                 6 . N2Charts
-----
Command: CR REL.....

```

Bild 38 Maske TR0001 mit Eingabebeispiel

In der hier dargestellten Auswahlmaske wird als Beispiel eine erweiterte Beziehung erstellt und über Kommando ausgewählt. Es soll eine erweiterte Beziehung zwischen einem Objekt vom Typ `RIX` und einem Objekt vom Typ `FLD` erstellt werden. Bei Eingabe des Kommandos mit dem Operanden `T=XIX` würde die Maske TR0008 überspringen.

- ▶ Schicken Sie die Maske mit der Taste `[DÜ]` ab.
- ▶ Kreuzen Sie in der Auswahlmaske TR0008 das Feld der erweiterten Beziehung (hier `XIX`) an. Die ausgewählte Grundfunktion (`CREATE`) wurde nicht überschreibbar in diese Maske übernommen. Mehrfaches Ankreuzen ist ebenfalls möglich. Dann werden die folgenden Beziehungsmasken in der markierten Reihenfolge ausgegeben. Das Feld `Simple relationship` kann im Falle einer `EXTENDED`-Funktion nicht angekreuzt werden.

```

TR0008 Selection Relationship-types          $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Please select relationship-types

. Storage-info (STO)
. Format-info  (FOR)
. Table-column (XBT)
X Index-column (XIX)

. simple relationship

Functions for relationship-types:
. Display          X Create          . Change          . Delete

-----
Command: .....

```

Bild 39 Maske TR0008 mit Eingabebeispiel

Die Maske TR0008 bietet allgemein folgende Möglichkeiten:

- Beenden der Funktion:
Mit der Taste **(K)** erfolgt die Rückkehr in die Auswahlmaske TR0001, sofern keine Funktion mehr zu bearbeiten ist. Falls infolge Kettung noch eine weitere Funktion zu bearbeiten ist, wird die erste Maske zu dieser Funktion ausgegeben.
 - Bearbeiten einer weiteren Beziehung mit derselben Grundfunktion:
 - Ferner kann durch Ankreuzen von erweiterten Beziehungstypen die Bearbeitung von erweiterten Beziehungen mit der aktuellen Grundfunktion eingeleitet werden.
- Schicken Sie die Maske mit **(DÜ)** ab.
- Geben Sie nun in der typspezifischen Maske (hier TR2040) *Type* und *Name* für Ausgangs- und Zielobjekt der Beziehung ein. Diese Maske wird leer ausgegeben. Die Typen können, wie es im Eingabebeispiel der Fall ist, auch in der Kurzform eingegeben werden.

```

TR2040 function relationship extended          $userid .nkz.TOMrfn.TOM.project
-----
Description : ..... Creation date: .....
              ..... Last change : .....
              ..... Responsible : ...
              ..... Text : .

Relationship from Object
Type          Name
RIX..... Indl.....

to succeeding objects
Type          Name          Attributes for relationships
FLD..... kdnr..... * XIX .....

Index-Column-Length : 45.

-----
Command: .....
-----

```

Bild 40 Maske TR2040 mit Eingabebeispiel

- ▶ Schicken Sie die Maske mit der Taste **[DÜ]** ab.
Die Maske wird mit den identifizierenden Angaben erneut ausgegeben. Die Bedeutung der typspezifischen Attributfelder ist auf Seite 237 beschrieben.
- ▶ Sie können nun in der Maske TR2040 die Beziehung bearbeiten. Die Bearbeitung ist abhängig von der ausgewählten Funktion, die in der Statuszeile vermerkt ist. Für jede Funktion wird im folgenden der Ablauf skizziert. Die Bearbeitung ist mit der Taste **[DÜ]** abzuschließen.
 - DIS REL und DIS REL EXT
Alle Attribute der Beziehung werden angezeigt.
 - CR REL
Es können (wie im Eingabebeispiel) Attribute erstellt werden.
TOM-REF fügt die Beziehung ans Ende der Beziehungsliste des Ausgangsobjekts an.
 - CR REL EXT
Die Attribute der einfachen Beziehung werden nicht überschreibbar angezeigt.
Die Attribute der Erweiterung können erstellt werden.
 - CH REL
Alle Attribute können geändert werden.
War das Attribut des Beziehungstyps bisher leer (`Type=.`), so wird die Beziehung zuvor automatisch erweitert (`Type=xxx`).
 - CH REL EXT
Alle Attribute der Erweiterung können geändert werden.

— DEL REL

Im Meldungsfenster steht:

```
TR-411    Do you want to delete relationship (Y/N)
```

Falls die Beziehung nicht gelöscht werden soll, ist das in der Kommandozeile vorgegebene `Y` mit `N` zu überschreiben.

— DEL REL EXT

Im Meldungsfenster steht:

```
TR-412    Do you want to delete extension (Y/N)
```

Falls die erweiterte Beziehung nicht in eine einfache überführt werden soll, ist das in der Kommandozeile vorgegebene `Y` mit `N` zu überschreiben.

5 Servicefunktionen

Die Funktionen in der Auswahlmaske TR9000 werden auch Servicefunktionen genannt.

Die Maske TR9000 erreichen Sie, wenn Sie in Maske TR0001 (siehe Seite 64) die Funktion SERVICES ankreuzen oder mit dem Kommando SER.

```
TR9000  Service selection                               $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Please select function(s):                               Component list
. Display directory for type : ...
. Display where-used
. Display uses
. Display objects which are not part of another object .....CLS
. Display referenced objects which are not defined .....CLS
. Print functions

Generate functions :   . Copy           . UDS

. Rename / copy object
. Delete structure
. Incorporate from another Data Dictionary
. Check objects and relationships
. Administration (ISAM)

-----
Command: .....
-----
```

Bild 41 Auswahlmaske TR9000 für Servicefunktionen

Die folgende Tabelle informiert über die Bedeutung der einzelnen Servicefunktionen. Ferner ist zu jeder Servicefunktion der Abschnitt angegeben, in dem der Ablauf beschrieben ist.

Servicefunktion	Bedeutung der Servicefunktion	Abschnitt
Display directory for type	Anzeigen von typbezogenen Objektverzeichnissen.	5.1
Display where-used	Anzeigen der Beziehungen, in denen ein bestimmtes Objekt Zielobjekt ist.	5.2
Display uses	Anzeigen der Beziehungen, in denen ein bestimmtes Objekt Ausgangsobjekt ist.	5.3
Display objects which are not part of another object	Anzeigen der Objekte, die in keiner Beziehung Zielobjekt sind.	5.4
Display referenced objects which are not defined	Anzeigen der Objekte, die in einer Beziehung Zielobjekt sind, aber noch nicht erstellt wurden.	5.5
Print functions	Drucken von Objektverzeichnissen oder Objekteinträgen.	5.6
Generate functions	Generieren von Copy-Elementen und UDS-DDL-Sourcen.	5.7
Rename / copy object	Umbenennen / Kopieren von Objekten.	5.8
Delete structure	Löschen von Objektstrukturen.	5.9
Incorporate from another Data Dictionary	Übertragen von Einträgen eines anderen Data Dictionary ins aktuelle.	5.10
Check objects and relationships	Überprüfen von Objekten und Beziehungen auf Verträglichkeit mit dem Strukturkatalog.	5.11
Administration (ISAM)	Data Dictionary in TOMLIB sichern. Data Dictionary aus TOMLIB holen TOMRFI-Datei neu aufbauen. Neues Data Dictionary anlegen.	5.12

Servicefunktionen können Sie in den folgenden Masken aufrufen:

- Auswahlmaske TR0001
- Folgemasken (einschließlich der Auswahlmaske TR0008 für Beziehungstypen)
- leere typspezifische Objektmaske (evtl. mit Objektnamen vorbelegt)

Eine Servicefunktion können Sie auf zwei Arten aufrufen:

- über Kommando:
Im Kapitel 12 TOM-REF-Kommandos ist beschrieben, für welche Servicefunktionen es Kommandos gibt. Nach dem Abschicken der Maske mit der Taste **[DÜ]** wird die Maske für die gewählte Servicefunktion ausgegeben. Der weitere Ablauf ist im Abschnitt für die betreffende Servicefunktion beschrieben.
- über die Auswahlmaske TR9000 in zwei Schritten:
 - 1. Schritt:* Auswahlmaske TR9000 ansteuern.
Aus einer der eben genannten Masken wird die Auswahlmaske TR9000 aufgerufen. In der Auswahlmaske TR0001 oder einem Sub-Menü ist dies durch Ankreuzen, Kurzkommando oder das Kommando `SERVICES`, wie auf Seite 64 beschrieben, möglich. Aus einer typspezifischen Objektmaske kann die Auswahlmaske TR9000 nur mit dem Kommando `SERVICES` aufgerufen werden.
 - 2. Schritt:* Eine oder mehrere Servicefunktionen auswählen.
In der Auswahlmaske TR9000 können eine oder mehrere Servicefunktionen angekreuzt werden. Die Abarbeitung der gewählten Servicefunktionen erfolgt in der markierten Reihenfolge. Nach dem Abschicken der Maske mit der Taste **[DÜ]** wird die Maske für die erste markierte Servicefunktion ausgegeben. Der weitere Ablauf ist im Abschnitt für die betreffende Servicefunktion beschrieben.

Ausgangspunkt der Beschreibung einer Servicefunktion ist im folgenden stets die Auswahlmaske TR9000.

5.1 Display directory for type

Funktionen:

1. Alle Objekte eines Objekttyps oder einer Objekttyp-Gruppe anzeigen (ab Seite 112).
2. Auswahl von Objekten eines oder mehrerer Objekttypen anzeigen (ab Seite 114).

Pro Objekt werden folgende Attribute angezeigt: Typ, Name, Verantwortlicher, Erstellungsdatum, letztes Änderungsdatum.

Die Funktionen können auch über das Kommando `DISDIR` angesteuert werden (siehe Kapitel 12).

5.1.1 Alle Objekte eines Objekttyps oder einer Objekttyp-Gruppe anzeigen

Es gibt standardmäßig folgende Gruppen von Objekttypen:

Bezeichnung	Bedeutung
*T	Alle Objekttypen
*D	Die Objekttypen FLD, FLG, REC
*M	Die Objekttypen MOD, PRG

Vorgehen:

- ▶ Legen Sie in der Auswahlmaske TR9000 Funktion und Typ(gruppe) fest.
Kreuzen Sie dazu die Servicefunktion `Display directory for type an`: In das dreistellige Feld derselben Zeile muß entweder die Bezeichnung einer Gruppe oder eines Objekttyps eingetragen werden.
- ▶ Schicken Sie die Maske mit der Taste `[DÜ]` ab.

Die Objekte werden in der Maske TR9019 angezeigt. Die Anzeige für den Objekttyp FLD könnte folgendermaßen aussehen:

Type	Name	Resp.	Creation	Changed
FIELD	ABTEILUNG	LEO	20.05.95	. .
FIELD	NACHNAME	EDI	09.05.95	20.05.95
FIELD	VORNAME	LEO	15.11.94	. .
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Command:

TR-10 No more objects in list

Bild 42 Maske TR9019 mit Ausgabebeispiel

Die Maske enthält in jeder Zeile des Arbeitsbereichs die Anzeige für ein Objekt. Das Ergebnis konnte in einer Maske ausgegeben werden. Mit der Taste **DU** erfolgt die Rückkehr in den Auswahlmaske TR9000.

i Wenn die ausgegebene Liste länger als eine Maske ist, können die Folgemasken mit der Taste **DU** der Reihe nach abgerufen werden. In der Kommandozeile ist dann ein + vorgegeben. In den Folgemasken kann geblättert werden.

5.1.2 Auswahl von Objekten eines oder mehrerer Objekttypen anzeigen

Als Auswahlkriterien für ein Objekt kommen der Objektname und die einstelligen Attribute in Frage.

Vorgehen:

- ▶ Kreuzen Sie in der Auswahlmaske TR9000 die Funktion `Display directory for type an.`
- ▶ Schicken Sie die Maske mit der Taste `[DÜ]` ab.
- ▶ Kreuzen Sie in der Auswahlmaske TR9010 das Sub-Menü an, das den oder die gewünschten Objekttypen enthält.
Mehrfaches Ankreuzen ist möglich. Die Sub-Menüs werden dann in der ausgewählten Reihenfolge ausgegeben.

```

TR9010  Directory selection                               $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Please select sub-menu(s):

. JCL
. Process
. UTM
. DVS/LEASY
. UDS
X SESAM/SQL

-----
Command: .....

```

Bild 43 Maske TR9010 mit Eingabebeispiel

- ▶ Schicken Sie Ihre Auswahl (hier: `SESAM/SQL`) mit der Taste `[DÜ]` ab. Aufgrund der Auswahl wird die Maske TR9084 mit den `SESAM/SQL`-Objekttypen ausgegeben.
- ▶ Geben Sie dann in der nachfolgenden Auswahlmaske TR90xx den Objekttyp und wahlweise Auswahlkriterien an. Hinter einem angekreuzten Objekttyp kann ein Startname oder eine Auswahlinformation angegeben werden, um die Menge der anzuzeigenden Objekte dieses Typs einzugrenzen.
Bei benutzereigenen Objekttypen (`Other`) muß die dreistellige Typbezeichnung (anstelle der Markierung) explizit angegeben werden.

```

TR9084  Directory selection (SESAM/SQL)                $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Please mark type and insert (if wanted) startname or select-information

X  SQL-Base-Table   (RBT) .....
.  SQL-Chck-Constr (RCC) .....
.  SQL-Catalog     (RCT) .....
.  SQL-Ref-Constr  (RFC) .....
.  SQL-Index       (RIX) .....
.  SQL-Store-Group (RSG) .....
X  SQL-Schema      (RSM) .....
.  SQL-Uniq-Constr (RUC) .....
.  SQL-User        (RUS) .....
.  Field           (FLD) .....
.  Physical-File   (PFL) .....
.  Volume          (VOL) .....

... Other .....

-----
Command: .....
    
```

Bild 44 Maske TR9084 mit Eingabebeispiel

Der Startname legt einen Objektnamen fest, ab dem die Objekte in alphabetischer Reihenfolge angezeigt werden. Ferner sind folgende Eingaben möglich:

A*	Alle Objekte, deren Name mit dem Buchstaben A beginnt.
*A	Alle Objekte, deren Name mit dem Buchstaben A endet.
A	Alle Objekte, deren Name den Buchstaben A enthält.

Die Auswahlinformation stellt Bedingungen an die einstelligen Attribute der anzuzeigenden Objekte. Die möglichen Einträge haben folgende Wirkung:

␣	Leerzeichen - bei der Zusammenstellung des Inhaltsverzeichnisses wird dieses Attribut nicht überprüft.
-	Bei diesem Attribut darf kein Eintrag gemacht worden sein.
+	Bei diesem Attribut muß ein Eintrag stehen.
x	'beliebiges Zeichen' - Das Attribut muß den Wert 'x' haben.

Es werden nur solche Objekte angezeigt, die beiden Auswahlkriterien genügen.

5.2 Display where-used

Mit der Servicefunktion `Display where-used` können Sie sich zu einem ausgewählten Objekt die inversen Beziehungen anzeigen lassen. Das sind diejenigen Beziehungen, in denen das ausgewählte Objekt Zielobjekt ist. Angezeigt werden direkte und indirekte Beziehungen.

- Direkte Beziehungen können Sie (oder ein Tool) explizit zwischen Objekten erstellen. Bisher wurde ausschließlich dieser Fall betrachtet. Deshalb konnte das Beiwort direkt entfallen.
- Indirekte Beziehungen leitet TOM-REF automatisch aus den direkten Beziehungen ab, indem es Pfade durch das Netz der direkten Beziehungen verfolgt. TOM-REF zeigt indirekte Beziehungen bis zur Ebenentiefe 20 an. (Die Ebenentiefe einer direkten Beziehung ist 2.)

Im einzelnen handelt es sich um folgende Funktionen:

1. Alle inversen Beziehungen anzeigen (ab Seite 118).
2. Inverse Beziehungen anzeigen, die von Objekten eines bestimmten Typs ausgehen (ab Seite 120).

Bei beiden Funktionen werden pro Beziehung Typ und Name des Ausgangsobjekts angezeigt.

Für die Fallbeispiele dieses Abschnitts wird folgende Struktur als gegeben vorausgesetzt:

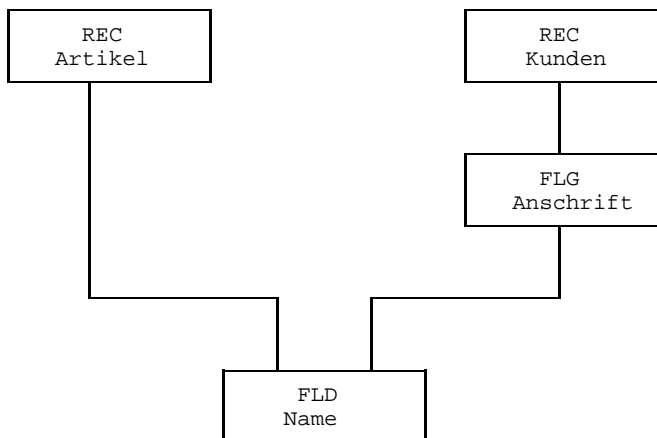


Bild 46 Beispielstruktur

Es besteht demnach eine indirekte Beziehung zwischen dem Record 'Kunden' und dem Feld 'Name'. Die restlichen Beziehungen sind direkt.

5.2.1 Alle inversen Beziehungen anzeigen

Ausgangspunkt der folgenden Ablaufbeschreibung ist die Auswahlmaske TR9000. Wie in der Einleitung zu diesem Kapitel beschrieben, wird die Auswahlmaske TR9000 übersprungen, wenn die Servicefunktion `Display where-used` mit dem Kommando `DISP-WHERE-USED` aufgerufen wird. In diesem Fall entfällt Punkt 1 der Ablaufbeschreibung.

Vorgehen:

- ▶ Kreuzen Sie in Auswahlmaske TR9000 die Servicefunktion `Display where-used` an.
- ▶ Schicken Sie die Maske mit der Taste `(DÜ)` ab.
- ▶ Tragen Sie in der Maske TR9020 Typ und Name des Zielobjekts ein. In der letzten Zeile des Arbeitsbereichs kann der Name einer Komponentenliste angegeben werden. In diesem Fall wird das Ergebnis der folgenden Anzeige als Element in die TOMLIB aufgenommen (Elementtyp: `CLS`).

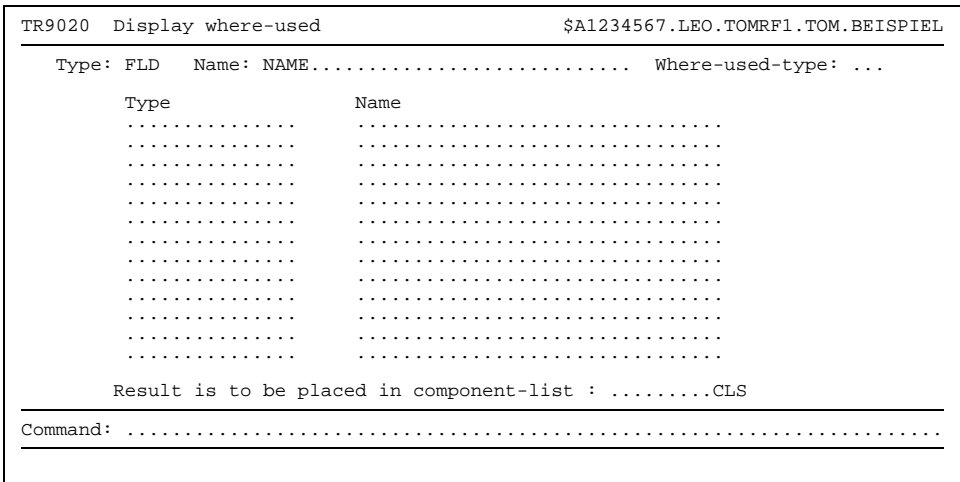


Bild 47 Maske TR9020 mit Eingabebeispiel

- ▶ Schicken Sie die Maske mit der Taste `(DÜ)` ab.

5.2.2 Inverse Beziehungen anzeigen, die von Objekten eines Typs ausgehen

Ausgangspunkt der folgenden Ablaufbeschreibung ist die Auswahlmaske TR9000. Wie in der Einleitung zu diesem Kapitel beschrieben, wird die Auswahlmaske TR9000 übersprungen, wenn die Servicefunktion `Display where-used` mit dem Kommando `DISP-WHERE-USED` aufgerufen wird. In diesem Fall entfällt Punkt 1 der Ablaufbeschreibung.

Vorgehen:

- ▶ Kreuzen Sie in der Auswahlmaske TR9000 die Servicefunktion `Display where-used an`.
- ▶ Schicken Sie die Maske mit der Taste `DÜ` ab.
- ▶ Legen Sie in der nachfolgenden Maske TR9020 Typ und Name des Zielobjekts sowie den Typ der Ausgangsobjekte (Where-used-type) fest. In der letzten Zeile des Arbeitsbereichs kann der Name einer Komponentenliste angegeben werden. In diesem Fall wird das Ergebnis der folgenden Anzeige als Element in die TOMLIB aufgenommen (Elementtyp: `CLS`).

TR9020 Display where-used		\$A1234567.LEO.TOMRFL.TOM.BEISPIEL
Type: FLD Name: NAME..... Where-used-type: REC		
Type	Name	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
.....	
Result is to be placed in component-list :CLS		
Command:		

Bild 49 Maske TR9020 mit Eingabebeispiel

- ▶ Schicken Sie die Maske mit der Taste `DÜ` ab.

5.3 Display uses

Mit der Servicefunktion `Display uses` können Sie sich zu einem ausgewählten Objekt die Beziehungen anzeigen lassen, die von diesem Objekt ausgehen. Angezeigt werden direkte und indirekte Beziehungen.

- Direkte Beziehungen können Sie (oder ein Tool) explizit zwischen Objekten erstellen. Bisher wurde ausschließlich dieser Fall betrachtet. Deshalb konnte das Beiwort direkt entfallen.
- Indirekte Beziehungen leitet TOM-REF automatisch aus den direkten Beziehungen ab, indem er Pfade durch das Netz der direkten Beziehungen verfolgt. TOM-REF zeigt indirekte Beziehungen bis zur Ebenentiefe 20 an. (Die Ebenentiefe einer direkten Beziehung ist 2.)

Im einzelnen handelt es sich um folgende Funktionen:

1. Alle Beziehungen anzeigen (ab Seite 123).
2. Beziehungen zu Objekten eines bestimmten Typs anzeigen (ab Seite 125)

Bei beiden Funktionen werden pro Beziehung Typ und Name des Zielobjekts angezeigt. Ferner wird, falls vorhanden, der Typ der direkten Beziehung zum angezeigten Zielobjekt ausgegeben, ggf. mit Kennzeichen * für Erweiterung.

Für die Fallbeispiele dieses Abschnitts wird die folgende Struktur als gegeben vorausgesetzt:

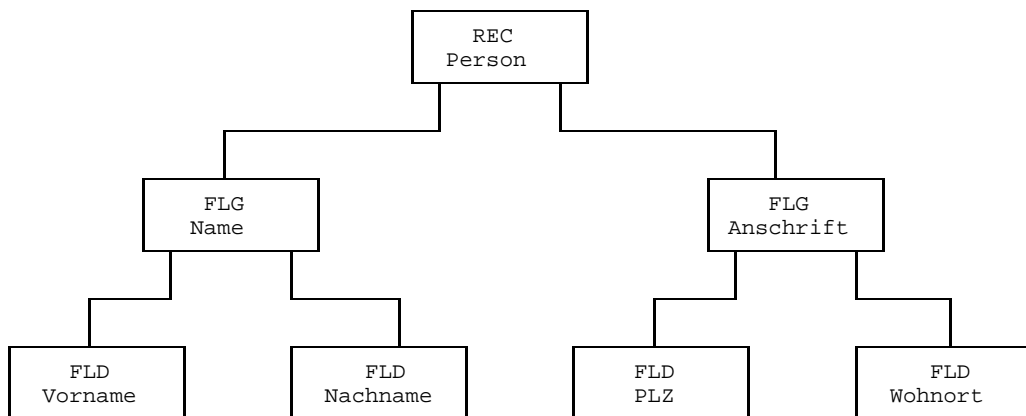


Bild 51 Beispielstruktur

Vorausgesetzt wird, daß die direkten Beziehungen beim Ausgangsobjekt in der Reihenfolge des Bildes (von links nach rechts) erstellt wurden. Indirekte Beziehungen bestehen zwischen dem Record Person und allen vier Feldern.

5.3.1 Alle Beziehungen anzeigen

Ausgangspunkt der folgenden Ablaufbeschreibung ist die Auswahlmaske TR9000. Wie in der Einleitung zu diesem Kapitel beschrieben, wird die Auswahlmaske TR9000 übersprungen, wenn die Servicefunktion `Display uses` mit dem Kommando `DISP-USSES` aufgerufen wird. In diesem Fall entfällt der erste Punkt der Ablaufbeschreibung.

Vorgehen:

- ▶ Kreuzen Sie in der Auswahlmaske TR9000 die Servicefunktion `Display uses` an.
- ▶ Schicken Sie die Maske mit der Taste **[DÜ]** ab.
- ▶ Tragen Sie in der Maske TR9030 Typ und Name des Ausgangsobjekts ein. In der letzten Zeile des Arbeitsbereichs kann der Name einer Komponentenliste angegeben werden. In diesem Fall wird das Ergebnis der folgenden Anzeige als Element in die TOMLIB aufgenommen (Elementtyp: `CLS`).

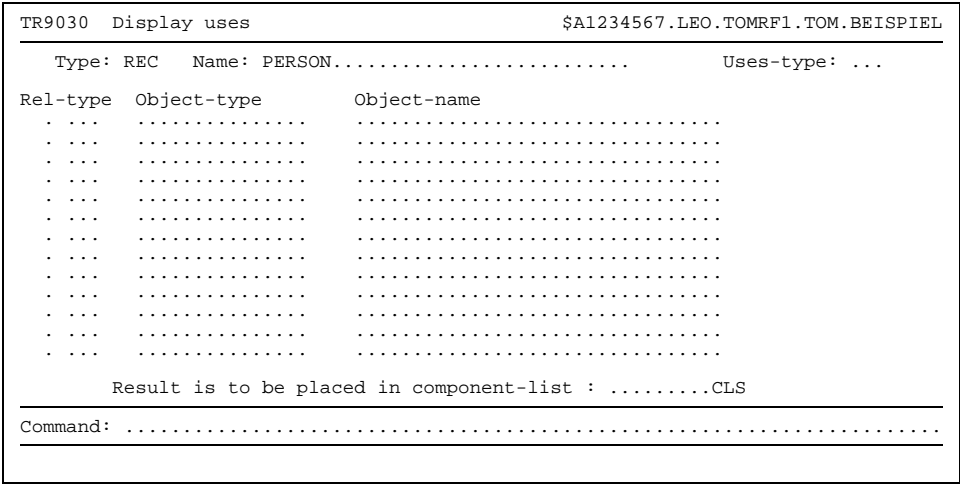


Bild 52 Maske TR9030 mit Eingabebeispiel

- ▶ Schicken Sie die Maske mit der Taste **[DÜ]** ab.

Die Maske TR9030 enthält das vollständige Ergebnis.
 Mit der Taste **[K1]** kann die Funktion beendet werden.
 Die Taste **[DÜ]** bewirkt die Ausgabe einer leeren Maske TR9030, in dem die Servicefunktion für ein weiteres Objekt fortgesetzt werden kann.

```

TR9030  Display uses                               $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Type: REC   Name: PERSON                           Uses-type:
-----
Rel-type  Object-type      Object-name
FIELDGROUP  NAME
FIELD       VORNAME
FIELD       NACHNAME
FIELDGROUP  ANSCHRIFT
FIELD       PLZ
FIELD       WOHNORT
. . .      . . . . .
. . .      . . . . .
. . .      . . . . .
. . .      . . . . .
. . .      . . . . .
. . .      . . . . .
. . .      . . . . .
. . .      . . . . .
. . .      . . . . .
. . .      . . . . .
-----
Result is to be placed in component-list :      .CLS
-----
Command: . . . . .
-----
TR-10  No more objects in list
    
```

Bild 53 Maske TR9030 mit Ausgabebeispiel

i Wenn die ausgegebene Liste länger als eine Maske ist, können die Folgemasken mit der Taste **[DÜ]** der Reihe nach abgerufen werden. In der Kommandozeile ist dann ein + vorgegeben. In den Folgemasken kann geblättert werden.

5.3.2 Beziehungen zu Objekten eines bestimmten Typs anzeigen

Ausgangspunkt der folgenden Ablaufbeschreibung ist die Auswahlmaske TR9000. Wie in der Einleitung zu diesem Kapitel beschrieben, wird die Auswahlmaske TR9000 übersprungen, wenn die Servicefunktion `Display uses` mit dem Kommando `DISP-USES` aufgerufen wird. In diesem Fall entfällt der erste Punkt der Ablaufbeschreibung.

Vorgehen:

- ▶ Kreuzen Sie in der Auswahlmaske TR9000 die Servicefunktion `Display uses` an.
- ▶ Schicken Sie die Maske mit der Taste **[DÜ]** ab.
- ▶ Legen Sie in der Maske TR9030 Typ und Name des Ausgangsobjekts einzutragen. Ferner muß der Typ der Zielobjekte (`uses-type`) eingegeben werden. In der letzten Zeile des Arbeitsbereichs kann der Name einer Komponentenliste angegeben werden. In diesem Fall wird das Ergebnis der folgenden Anzeige als Element in der TOMLIB abgespeichert (Elementtyp: `CLS`).

```

TR9030  Display uses                               $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
        Type: REC      Name: PERSON.....          Uses-type: FLD

Rel-type  Object-type          Object-name
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....

        Result is to be placed in component-list : .....CLS
-----
Command: .....
-----
  
```

Bild 54 Maske TR9030 mit Eingabebeispiel

- ▶ Schicken Sie die Maske mit der Taste **[DÜ]** ab.

Die Maske TR9030 enthält das vollständige Ergebnis.

Mit der Taste **[K1]** kann die Funktion beendet werden.

Die Taste **[DÜ]** bewirkt die Ausgabe einer leeren Maske TR9030, in dem die Servicefunktion für ein weiteres Objekt fortgesetzt werden kann.

TR9030 Display uses		\$A1234567.LEO.TOMRFL.TOM.BEISPIEL	
Type: REC	Name: PERSON	Uses-type: FLD	
Rel-type	Object-type	Object-name	
	FIELD	VORNAME	
	FIELD	NACHNAME	
	FIELD	PLZ	
	FIELD	WOHNORT	
.	
.	
.	
.	
.	
.	
.	
.	
.	
Result is to be placed in component-list :		.CLS	
Command:			
TR-10 No more objects in list			

Bild 55 Maske TR9030 mit Ausgabebeispiel



Wenn die ausgegebene Liste länger als eine Maske ist, können die Folgemasken mit der Taste **[DÜ]** der Reihe nach abgerufen werden. In der Kommandozeile ist dann ein + vorgegeben. In den Folgemasken kann geblättert werden.

Bei der Ausgabe erscheinen alle Objekte nur einmal, auch wenn sie in der Struktur mehrmals enthalten sind. Es ist daher bei der Ausgabe zu `Display uses` kein Rückschluß auf die Struktur möglich.

Indirekte Beziehungen mit Ebenentiefe größer als 20 werden, wie bereits erwähnt, nicht angezeigt. In diesem Fall erscheint die Meldung:

```
TR-35 Level greater than 20 present
```

5.4 Display objects which are not part of another object

Mit der Servicefunktion `Display objects which are not part of another object` können Sie sich alle Objekte anzeigen lassen, die keine inversen Beziehungen besitzen. Diese Objekte sind also in keiner Beziehung das Zielobjekt.

Diese Funktion unterstützt Sie bei der Fehlersuche. Wurde z.B ein Datensatz erstellt, der in keinem Programm verwendet wird, können Sie nachträglich noch Beziehungen aufbauen.

Vorgehen:

- ▶ Kreuzen Sie in Auswahlmaske TR9000 die Servicefunktion `Display objects which are not part of another object` an. In derselben Zeile des Arbeitsbereichs kann der Name einer Komponentenliste angegeben werden. In diesem Fall wird das Ergebnis der folgenden Anzeige als Element in der TOMLIB abgespeichert (Elementtyp: CLS).
- ▶ Schicken Sie die Maske mit der Taste **[DÜ]** ab.
Die Maske TR9040 enthält das vollständige Ergebnis.
Mit der Taste **[K1]** kann die Funktion beendet werden.

TR9040 Display not referenced objects		\$A1234567.EBE.TOMRF1.TOM.BEISPIEL		
Type	Name	Resp.	Creation	Changed
FIELD	JAHR	LEO	23.02.95	. .
FIELD	MONAT	LEO	22.05.95	22.05.95
FIELD	TAG	EDI	16.02.95	16.02.95
FIELDGROUP	ANSCHRIFT	EDI	21.02.95	. .
FIELDGROUP	NAME	LEO	15.02.95	. .
MODULE	STEUER	ULI	06.03.95	16.05.95
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
Command:				
TR-10 No more objects in list				

Bild 56 Maske TR9040 mit Ausgabebeispiel



Wenn die ausgegebene Liste länger als eine Maske ist, können die Folgemasken mit der Taste **[DÜ]** der Reihe nach abgerufen werden. In der Kommandozeile ist dann ein + vorgegeben. In den Folgemasken kann geblättert werden.

Die Maske TR9040 wird nur dann ausgegeben, wenn mindestens ein Objekt existiert, das keine inverse Beziehung besitzt. Ansonsten wird die Maske TR9000 erneut mit folgender Endmeldung ausgegeben:

```
TR-124      No not-referenced objects found
```


5.5 Display referenced objects which are not defined

Mit der Servicefunktion `Display referenced objects which are not defined` können Sie sich alle Objekte anzeigen lassen, die zwar Zielobjekt in einer Beziehung sind, aber noch nicht erstellt wurden.

Vorgehen:

- ▶ Kreuzen Sie in der Auswahlmaske TR9000 die Servicefunktion `Display referenced objects which are not defined` an. In derselben Zeile des Arbeitsbereichs kann der Name einer Komponentenliste angegeben werden. In diesem Fall wird das Ergebnis der folgenden Anzeige als Element in der TOMLIB abgespeichert (Elementtyp: `CLS`).
- ▶ Schicken Sie die Maske mit der Taste `[DÜ]` ab.

Die Maske TR9050 enthält das vollständige Ergebnis.

Links steht jeweils das nicht erstellte Zielobjekt, rechts das zugehörige Ausgangsobjekt. Das Ergebnis ist alphabetisch nach den Ausgangsobjekten (`Referenced from`) geordnet. Das erste Ordnungskriterium ist der Typ, das zweite der Name.

Mit der Taste `[K1]` kann die Funktion beendet werden.

```

TR9050  Display not defined objects                                $A1234567.EBE.TOMRF1.TOM.BEISPIEL
-----
Object                                     Referenced from
Type Name                                  Type Name
FLD  VORNAME                               FLG  ADRESSE
FLD  NACHNAME                              FLG  ADRESSE
FLD  ORT                                    FLG  ADRESSE
REC  KUNDEN                                 MOD  A
REC  MITARBEITER                           MOD  B
FLG  NUMMER                                 REC  AUFTRAG
...
...
...
...
...
...
...
...
...
Command: .....
-----
TR-10  No more objects in list
    
```

Bild 57 Maske TR9050 mit Ausgabebeispiel



Wenn die ausgegebene Liste länger als eine Maske ist, können die Folgemasken mit der Taste **[DÜ]** der Reihe nach abgerufen werden. In der Kommandozeile ist dann ein + vorgegeben. In den Folgemasken kann geblättert werden.

Die Maske TR9050 wird nur dann ausgegeben, wenn mindestens ein noch nicht erstelltes Zielobjekt existiert. Ansonsten wird die Maske TR9000 erneut mit folgender Endmeldung ausgegeben:

```
TR-125      No undefined objects referenced
```

5.6 Print functions

Mit der Servicefunktion `Print functions` können Sie den Druck verschiedener Listen anstoßen. Der Inhalt der Listen wird in die Maske TR9070 festgelegt. Die Maske TR9070 läßt sich über die Auswahlmaske TR9000 durch Ankreuzen von `Print functions` ansteuern. Mit dem Kommando `PRINT`, das in Kapitel 12 beschrieben ist, kann die Maske TR9070 ebenfalls erreicht werden.

TOM-REF schreibt die Listen standardmäßig in die logische Systemdatei `SYSLST`. Sie können die Listenausgabe auch in eine Datei umlenken. Dazu müssen Sie in der Maske TR9070 folgende Eingaben machen:

- ▶ Überschreiben Sie das Feld `Output-file is SYSLST` mit einem Leerzeichen (`_`).
- ▶ Tragen Sie einen Datei-Teilnamen in das Feld `Output-file is` ein. TOM-REF akzeptiert alle vom BS2000 erlaubten Zeichen und bildet den Dateinamen `nkz.teilname.LST.project`.

Nach dem Beenden der Servicefunktion `Print functions` wird wieder die Standardeinstellung für `SYSLST` wirksam.

```

TR9070  Print functions                               $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Output-file is SYSLST : X
Output-file is      : LEO.....LST.BEISPIEL

Print dir/obj for   Object group   : . (a:JCL,b:Process,c:UTM,d:DVS/LEASY,
                               e:UDS,h:SESAM/SQL)
                   Object         : ... .....
                   Component-list  : .....CLS
                   Total directory : .
                   All objects     : .                               Text : N

Print uses for      Object         : ... .....
                   Component-list  : .....CLS
                   All objects     : .

Print where-used for Object         : ... .....
                   Component-list  : .....CLS
                   All objects     : .
-----
Command: .....
-----

```

Bild 58 Maske TR9070

In der Maske TR9070 werden folgende Funktionen angeboten:

`Print dir/obj for`

Drucken von Objektverzeichnissen / Objekteinträgen.

Ein Objektverzeichnis ist eine Inhaltsangabe, die sich entweder auf alle Objekte oder auf die Objekte eines bestimmten Typs bezieht. Pro Objekt werden in einer Zeile Typ, Name und die Attribute `Verantwortlicher`, `Erstellungsdatum` und `letztes Änderungsdatum` gedruckt. Die Ausgabe ist alphabetisch nach Objekttypen geordnet, innerhalb eines Typs nach Objektnamen.

Ein Objekteintrag besteht aus allen Attributen, Beziehungen und Texten des betreffenden Objekts. Nach jedem Objekt erfolgt ein Seitenvorschub.

Das Verzeichnis aller Objekte wird gedruckt, wenn Sie `Total directory` ankreuzen.

Der Druck der Verzeichnisse und Einträge von Objekten eines bestimmten Typs wird durch Eingabe des Kennbuchstabens für die Typgruppe im Feld `object group` eingeleitet. In diesem Fall muß die Eingabe in einem typgruppen-spezifischen Folgebildschirm fortgesetzt werden. Der genaue Ablauf ist auf Seite 135 beschrieben.

Die restlichen drei Felder dieser Funktion dienen dem Druck von Objekteinträgen, wobei die Menge der zu druckenden Objekte jeweils unterschiedlich ist:

- Ein bestimmtes Objekt (`object`):
Typ (dreistellige Kurzbezeichnung) und Name des Objekts sind anzugeben.
- Die Objekte einer Komponentenliste (`Component-list`):
Der Name der Komponentenliste ist anzugeben.
- Alle Objekte (`All objects`):
Das Feld ist mit `x` anzukreuzen.

Durch Belegung des Feldes `Text` können Sie bestimmen, ob zu einem Objekteintrag auch ein dem Objekt zugeordneter Text gedruckt werden soll. Auf Seite 89 wurde dargestellt, daß jedem Objekt bis zu zwei Texte zugeordnet sein können: Der TOM-REF-Text und / oder der TOM-TI-Langtext. Im einzelnen bestehen folgende Möglichkeiten, das Feld `Text` zu belegen:

`N` : Es wird kein Text gedruckt.

`S` : Der TOM-REF-Text wird gedruckt.

`L` : Der TOM-TI-Langtext wird gedruckt.

`B` : Beide Texte, TOM-REF-Text und TOM-TI-Langtext, werden gedruckt.

Die Vorbelegung des Feldes `Text` richtet sich danach, welches Kommando aktiv ist:

`TEXT OFF`: Vorbelegung `N`

`TEXT ON`: Vorbelegung `S`

`TEXT ON,LIB=ON`: Vorbelegung `L`

`Print uses`

Drucken der Beziehungen eines Objekts.

Mit Beziehungen sind hier sowohl die direkten als auch die indirekten gemeint.

Die Menge der Objekte, deren Beziehungen gedruckt werden sollen, ist festzulegen:

- Ein bestimmtes Objekt (`Object`):
Typ (dreistellige Kurzbezeichnung) und Name des Objekts sind anzugeben.
- Die Objekte einer Komponentenliste (`Component-list`):
Der Name der Komponentenliste ist anzugeben.
- Alle Objekte (`All objects`):
Das Feld ist anzukreuzen.

Pro Beziehung werden in einer Zeile Typ und Name des Zielobjekts und die Ebenentiefe gedruckt. Falls vorhanden wird auch der Typ der direkten Beziehung zum Zielobjekt, ggf. mit Kennzeichen * der Erweiterung, ausgegeben. An der Reihenfolge der gedruckten Zielobjekte lassen sich anhand der Ebenentiefe Strukturen im Beziehungsnetz nachvollziehen. Die Zielobjekte werden bis zur Ebenentiefe 20 ausgegeben. Falls deshalb ein Abbruch in einem Beziehungspfad nötig ist, ist das letzte Zielobjekt mit zwei Leerzeichen (`_ _`) gekennzeichnet. Kennzeichnung ** bedeutet, daß das Zielobjekt bereits bei einem anderen Beziehungspfad ausgegeben wurde. In diesem Fall ist zusätzlich die Ebenentiefe des aktuellen Beziehungspfades angegeben.

`Print where used`

Drucken der inversen Beziehungen eines Objekts.


Inverse Beziehungen sind diejenigen, in denen das angegebene Objekt Zielobjekt ist.

Mit Beziehungen sind hier sowohl die direkten als auch die indirekten gemeint.

Die Menge der Objekte, deren inverse Beziehungen gedruckt werden sollen, ist festzulegen:

- Ein bestimmtes Objekt (`Object`):
Typ (dreistellige Kurzbezeichnung) und Name des Objekts sind anzugeben.
- Die Objekte einer Komponentenliste (`Component-list`):
Der Name der Komponentenliste ist anzugeben.
- Alle Objekte (`All objects`):
Das Feld ist anzukreuzen.

Pro inverser Beziehung werden in einer Zeile Typ und Name des Ausgangsobjekts sowie die Ebenentiefe der Beziehung ausgedruckt. An der Reihenfolge der gedruckten Ausgangsobjekte lassen sich anhand der Ebenentiefe Strukturen im Beziehungsnetz nachvollziehen. Die Ausgangsobjekte werden bis zur Ebenentiefe 20 ausgegeben. Falls deshalb ein Abbruch in einem Beziehungspfad nötig ist, ist das letzte Zielobjekt mit zwei Leerzeichen (_) gekennzeichnet. Kennzeichnung ** bedeutet, daß das Zielobjekt bereits bei einem anderen Beziehungspfad ausgegeben wurde. In diesem Fall ist zusätzlich die Ebenentiefe des aktuellen Beziehungspfades angegeben.

In der Maske TR9070 können Sie mehrere Funktionen auf einmal anstoßen. Die Maske ist mit der Taste  abzuschicken.

Die Maske TR9070 wird nochmals mit Meldung ausgegeben. Der Meldungstext hängt davon ab, ob die Listen zusätzlich Hinweise für den Anwender enthalten. Ist dies der Fall, erscheint folgende Meldung:

```
TR-2001    Additional messages are written in SYSLST
```

Andernfalls wird folgende Meldung ausgegeben:

```
TR-2002    Wanted objects and directories written
```

Ab Seite 140 ist beschrieben, unter welchen Umständen Hinweise in den Listen enthalten sind und wie diese Hinweise lauten.

Die gedruckten Listen sind mit einer doppelten Seitennumerierung versehen, um die Orientierung beim Durchblättern zu erleichtern. Die Seitennumerierung hat die Form xxxx/yyyy.

xxxx ist die übergeordnete Nummer. Sie wird bei jeder neuen Seite weiter gezählt.

yyyy ist die untergeordnete Nummer und beginnt bei eins. Sie wird weitergezählt, falls sich die Angaben zu einem Objekt über mehrere Seiten erstrecken,

5.6.1 Objektverzeichnisse und Objekteinträge drucken

Wird in der Maske TR9070 `object group` mit einem der dort angegebenen Kennbuchstaben `a` bis `h` belegt, so gibt TOM-REF eine typgruppen-spezifischen Maske aus (Nummern TR9073 bis TR9079, TR9083).

Beim Belegen mit einem anderen als den angegebenen Zeichen wird zunächst in die Auswahlmaske TR9072 verzweigt, in dem eine oder mehrere Typgruppen angekreuzt werden können. Nach dem Abschicken der Maske TR9072 mit der Taste `[DÜ]` werden dann die typgruppen-spezifischen Masken in der Reihenfolge der Markierung ausgegeben.

Alle Masken TR9072 bis TR9079 und TR9083 sind am Ende dieses Abschnitts zusammengestellt.

Die Eingaben in eine typgruppen-spezifische Maske werden anhand des SESAM/SQL-Maske TR9083 erläutert.

Dir		Type	Startname	Endname
	<input checked="" type="checkbox"/>	RBT
	<input type="checkbox"/>	RCC
X	<input type="checkbox"/>	RCT
	<input type="checkbox"/>	RFC
X	<input type="checkbox"/>	RIX	AUFTRAG.....	KUNDEN.....
	<input type="checkbox"/>	RSG
	<input type="checkbox"/>	RSM
	<input type="checkbox"/>	RUC
	<input type="checkbox"/>	RUS
X	<input checked="" type="checkbox"/>	FLD
	<input checked="" type="checkbox"/>	PFL	*EINGABE.....
X	<input type="checkbox"/>	VOL	*125*.....
...	<input type="checkbox"/>	Other
Command:				

Bild 59 Maske TR9083

Vor jedem der im Arbeitsbereich aufgeführten Objekttypen der SESAM/SQL-Gruppe befinden sich zwei Markierspalten:

- Die erste Spalte kreuzen Sie an, wenn die Einträge von Objekten des Typs gedruckt werden sollen.
- Die zweite Spalte kreuzen Sie an, wenn ein Verzeichnis der Objekte des Typs gedruckt werden soll.

Durch Eingaben in die Felder `Startname` und `Endname` können Sie die Menge der zu druckenden Objekte eingrenzen. Die Angaben für `Startname` und `Endname` wertet TOM-REF sowohl bei der Markierung Objektverzeichnis als auch bei der Markierung Objekteinträge aus. Falls für beide Fälle verschiedene Start- bzw. Endnamen gelten sollen, müssen die beiden Funktionen nacheinander, also nicht in derselben Maske angestoßen werden.

Eine weitere Eingrenzung der Objekte kann durch die Verwendung des Zeichens `*` bei der Angabe von `Startname` getroffen werden. Im einzelnen gibt es folgende Möglichkeiten für die Verwendung von `*`:

- `*<STRING>*`
Es werden alle Objekte angesprochen, die im Objektname an einer beliebigen Stelle die Zeichenfolge `<STRING>` enthalten.
- `*<STRING>`
Es werden alle Objekte angesprochen, deren Objektname mit der Zeichenfolge `<STRING>` endet.
- `<STRING>*`
Es werden alle Objekte angesprochen, deren Objektname mit der Zeichenfolge `<STRING>` beginnt.

Eine Eingabe bei `Endname` wird ignoriert, wenn `Startname` mit Verwendung von `*` eingegeben wird.

Einträge der Objekte eines `OTHER`-Typs, ggf. auch mit Verzeichnis, können ebenfalls gedruckt werden. Hierbei ist der `OTHER`-Typ im ersten Feld (dreistellig) anzugeben.

Aufgrund des Eingabebeispiels in Bild 59 würden nach dem Abschicken der Maske mit der Taste **DÜ** folgende Listen gedruckt:

RBT	Verzeichnis aller RBT-Objekte.
RCT	Einträge aller RCT-Objekte.
RIX	Einträge der RIX-Objekte von Auftrag bis Kunden.
FLD	Verzeichnis und Einträge aller FLD-Objekte.
PFL	Verzeichnis aller PFL-Objekte, deren Name mit EINGABE endet.
VOL	Einträge der VOL-Objekte, deren Name die Zeichenfolge 125 enthält.

Print-Masken:

TR9072	Print selection	\$A1234567.LEO.TOMRF1.TOM.BEISPIEL
<p>Please select sub-menu(s):</p> <ul style="list-style-type: none"> . JCL . Process . UTM . DVS/LEASY . UDS . SESAM/SQL 		
<hr/> <p>Command:</p> <hr/>		

Bild 60 Maske TR9072

TR9079	Print selection (JCL)	\$A1234567.LEO.TOMRF1.TOM.BEISPIEL																								
<table border="0"> <thead> <tr> <th>Dir</th> <th>Type</th> <th>Startname</th> <th>Endname</th> </tr> </thead> <tbody> <tr> <td>. .</td> <td>PRC</td> <td>.....</td> <td>.....</td> </tr> <tr> <td>. .</td> <td>ENT</td> <td>.....</td> <td>.....</td> </tr> <tr> <td>. .</td> <td>JV</td> <td>.....</td> <td>.....</td> </tr> <tr> <td>. .</td> <td>PJV</td> <td>.....</td> <td>.....</td> </tr> <tr> <td>... .</td> <td>Other</td> <td>.....</td> <td>.....</td> </tr> </tbody> </table>			Dir	Type	Startname	Endname	. .	PRC	ENT	JV	PJV	Other
Dir	Type	Startname	Endname																							
. .	PRC																							
. .	ENT																							
. .	JV																							
. .	PJV																							
... .	Other																							
<hr/> <p>Command:</p> <hr/>																										

Bild 61 Maske TR9079

TR9074		Print selection (Process)	\$A1234567.LEO.TOMRFL.TOM.BEISPIEL	
Dir	Type	Startname	Endname	
.	.	SYS	
.	.	PGM	
.	.	MOD	
.	.	MGR	
.	.	CAL	
.	.	ENP	
.	.	CBL	
.	.	LIB	
.	.	LMO	
.	.	NOM	
.	.	SOM	
...	.	Other	
Command:				

Bild 62 Maske TR9074

TR9078		Print selection (UTM)	\$A1234567.LEO.TOMRFL.TOM.BEISPIEL	
Dir	Type	Startname	Endname	
.	.	UAP	
.	.	GSS	
.	.	LSS	
.	.	LTR	
.	.	UTP	
.	.	TAC	
.	.	TLS	
...	.	Other	
Command:				

Bild 63 Maske TR9078

TR9075 Print selection (DVS/LEASY)		\$A1234567.LEO.TOMRFL.TOM.BEISPIEL	
Dir	Type	Startname	Endname
.	.	FLD
.	.	FLG
.	.	REC
.	.	FIL
.	.	PKY
.	.	SKY
.	.	MSK
.	.	CMD
.	.	FKY
.	.	INL
.	.	PFL
.	.	VOL
.	.	SEQ
...	.	Other
Command:			

Bild 64 Maske TR9075

TR9073 Print selection (UDS)		\$A1234567.LEO.TOMRFL.TOM.BEISPIEL	
Dir	Type	Startname	Endname
.	.	UDS
.	.	DBS
.	.	SSL
.	.	SUB
.	.	ARE
.	.	DAR
.	.	DBR
.	.	SBR
.	.	LMD
.	.	LMC
.	.	SEA
.	.	SET
.	.	SSY
.	.	SDY
.	.	DBK
.	.	ALI
...	.	Other
Command:			

Bild 65 Maske TR9073

TR9083 Print selection (SESAM/SQL)		\$A1234567.LEO.TOMRFL.TOM.BEISPIEL	
Dir	Type	Startname	Endname
.	.	RBT
.	.	RCC
.	.	RCT
.	.	RFC
.	.	RIX
.	.	RSG
.	.	RSM
.	.	RUC
.	.	RUS
.	.	FLD
.	.	PFL
.	.	VOL
...	.	Other
Command:			

Bild 66 Maske TR9083

5.6.2 Hinweise in den Listen

Neben den ausgewählten Objektverzeichnissen und Objekteinträgen kann eine Liste noch zusätzlich Hinweise enthalten. Ein Hinweis ist am Zeilenanfang durch drei Sterne (***) gekennzeichnet.

In welchen Fällen ein bestimmter Hinweis in die Liste geschrieben wird, ist im folgenden zusammengestellt:

Leeres Objektverzeichnis

- *** Directory is empty
Es wurde Print dir/obj for (Total directory) angekreuzt. Das zu druckende Objektverzeichnis ist jedoch leer.
- *** Directory for type <TYPE> is empty in specified range
In eine PRINT-Maske (TR9073 - TR9079, TR9083) wurde für einen der aufgeführten Typen die Spalte Dir angekreuzt. Das zu druckende Objektverzeichnis ist jedoch für den gewünschten Bereich leer.
- *** No object in specified range for type <TYPE>
In eine PRINT-Maske wurde für einen der aufgeführten Typen die erste Spalte angekreuzt. Zu dem Typ gibt es jedoch keine Objekteinträge in dem gewünschten Bereich.

Nicht existierende Objekte

*** Object is not defined: type: <TYPE> , name: <NAME>
In der Maske TR9070 wurde bei den Funktionen `Print dir/obj for` oder `Print uses for` ein Objekt angegeben. Das Objekt existiert jedoch nicht.

Unstimmigkeiten bei Komponentenlisten

*** Object is not defined: type: <TYPE> , name: <NAME>
In der Maske TR9070 wurde eine Komponentenliste eingetragen, in der ein Objekt aufgeführt ist, das nicht existiert.

*** Objecttype <TYPE> is not valid
In der Maske TR9070 wurde eine Komponentenliste eingetragen, die eine ungültige Typangabe enthält.

*** Invalid identifier in CLS
In der Maske TR9070 wurde eine Komponentenliste eingetragen, die eine Zeichenfolge enthält, die nicht aus <NAME>.<TYPE> interpretiert werden kann.

Anfang oder Ende einer Komponentenliste

Hinweise auf Anfang und Ende einer Komponentenliste beginnen nicht mit drei Sternen (***) , sondern mit drei Nummernzeichen (###). Damit soll zum Ausdruck kommen, daß kein fehlerhaftes Verhalten dokumentiert ist. Deshalb erscheint auch bei den beiden folgenden Hinweisen die Meldung TR-2002 (und nicht TR-2001), falls keine weiteren Hinweise vorhanden sind.

Start of componentlist '<CLS>'
Markiert den Anfang einer Reihe von Objekteinträgen, die aufgrund der Angabe der Komponentenliste <CLS> in der Maske TR9070 gedruckt werden.

End of componentlist '<CLS>'
Markiert das Ende einer Reihe von Objekteinträgen, die aufgrund der Angabe der Komponentenliste <CLS> in der Maske TR9070 gedruckt werden.

Nicht existierende Komponentenlisten

*** Componentlist '<CLS>' does not exist
Die in Maske TR9070 angegebene Komponentenliste existiert nicht.

Fehler bei TOMLIB-Zugriff

*** Error by access of TOMLIB
Ein Zugriff auf die TOMLIB ist nicht möglich.

5.7 Generate functions

Die Servicefunktion `Generate functions` bietet folgende Teilfunktionen:

- Copy: Generieren von Copy-Elementen (ab Seite 143).
- UDS: Generieren von DDL-Source und Analysieren bestehender DDL-Source oder COBOL-DML-Programme (ab Seite 145).

In der Auswahlmaske TR9000 ist die gewünschte Teilfunktion anzukreuzen. Nach dem Abschicken der Maske mit der Taste **DÜ** wird in die jeweilige Generierungsmaske verzweigt.

Es können auch beide Teilfunktionen auf einmal angekreuzt werden. Die Generierungsmasken werden dann nacheinander ausgegeben.

Jede Teilfunktion kann auch, wie in der Einleitung dieses Kapitels beschrieben, durch ein Kommando aufgerufen werden.

Die entsprechenden Kommandos lauten `GENERATE COPY` bzw. `UDS` und sind in Kapitel 12 beschrieben. Bei Aufruf über Kommando wird die Auswahlmaske TR9000 übersprungen.

5.7.1 Copy-Elemente erzeugen

TR9060	Generation selection	\$A1234567.LEO.TOMRF1.TOM.BEISPIEL
Generate COPY for		
Object	:INL
Component-list	:CLS
COPY-library	:	\$userid .nkz.TOMLIB.TOM.project
Command:		

Bild 67 Maske TR9060

In der Maske TR9060 wird die Generierung von Copy-Elementen angestoßen. Voraussetzung ist, daß die zugehörigen Copy-Objekte erstellt wurden (siehe Seite 205).

Um ein Copy-Element zu generieren, muß im Feld `Object` der Name des Copy-Objekts oder im Feld `Component-list` der Name einer Komponentenliste angegeben werden. Die Komponentenliste kann die Namen mehrerer Objekte des Typs `INL` enthalten, so daß mehrere Copy-Elemente auf einmal generiert werden können.

Das Feld `COPY-library` ist mit dem Namen der TOMLIB vorbelegt. Es kann überschrieben werden, falls das oder die Copy-Elemente in einer anderen Bibliothek abgelegt werden sollen.

Die Generierung wird mit der Taste **(DÜ)** gestartet.

Ein Copy-Element erhält denselben Namen wie das zugehörige Copy-Objekt. Der Typ lautet `INL`.

Beispiel

Das folgende Beispiel zeigt die Generierung von Copy-Elementen anhand einer COBOL-Source:

```

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
*****
*** FILE-C.INL ***
*****
        SELECT BEISPIEL
        ASSIGN TO DA-DISC-S-SYS011
        ORGANIZATION IS SEQUENTIAL
        ACCESS MODE IS SEQUENTIAL
        RESERVE NO ALTERNATE AREA
        FILE STATUS IS NACHNAME
        IN ADRESSE.
DATA DIVISION.
FILE SECTION.
*****
*** FILE-D.INL ***
*****
        FD BEISPIEL
        BLOCK CONTAINS 366 CHARACTERS
        RECORDING MODE IS F
        LABEL RECORD IS STANDARD.
PROCEDURE DIVISION.
*****
*** INIT.INL ***
*****
        MOVE SPACES TO VORNAME IN NAME IN ADRESSE
        MOVE SPACES TO NACHNAME IN NAME IN ADRESSE
        MOVE ZERO TO PLZ IN ADRESSE
        MOVE SPACES TO ORT IN ADRESSE
        MOVE SPACES TO VORNAME-DRUCK IN NAME-DRUCK IN
        ADRESSE-DRUCK
        MOVE SPACES TO NACHNAME-DRUCK IN NAME-DRUCK IN
        ADRESSE-DRUCK
        MOVE ZERO TO PLZ-DRUCK IN ADRESSE-DRUCK
        MOVE SPACES TO ORT-DRUCK IN ADRESSE-DRUCK
*****
*** MOVE.INL ***
*****
        AUSGABE-DATUM
        MOVE EINGABE-MONAT IN EINGABE-DATIN TO AUSGABE-MONAT
        IN AUSGABE-DATUM
        MOVE EINGABE-JAHRESZAHL IN EINGABE-DATIN TO
        MOVE EINGABE-JAHR IN EINGABE-FILLER-DATIN IN
        EINGABE-DATIN TO AUSGABE-JAHR IN
        AUSGABE-FILLER-DATUM IN AUSGABE-DATUM
        MOVE EINGABE-WOCHENTAG IN EINGABE-DATIN TO
        EINGABE-DATIN TO AUSGABE-ZEIT IN
        AUSGABE-FILLER-DATUM-01 IN AUSGABE-DATUM
        AUSGABE-FILLER-WOCHENTAG-01 IN
        AUSGABE-FILLER-DATUM-01 IN AUSGABE-DATUM

```

Beispiel für
FILE-COPY in
FILE-CONTROL

Beispiel für
FILE-COPY in
DATA DIVISION

Beispiel für
INIT-COPY

Beispiel
für
MOVE-COPY

5.7.2 UDS-DDA: Anschluß von UDS an TOM-REF

Das Produkt UDS-DDA (UDS-Data Dictionary Anschluß) verbindet das Datenbanksystem UDS mit TOM-REF. UDS-DDA unterstützt den Aufbau der Datenbank sowie die Nachdokumentation bestehender DDL-Sourcen und COBOL-DML-Programme.

Die Funktionen von UDS-DDA sind im Handbuch "UDS-DDA Benutzerhandbuch[13]" beschrieben.

TR9090	Generation selection	\$A1234567.LEO.TOMRF1.TOM.BEISPIEL
<hr/>		
Generate DDL Parameter :DG1		
<hr/>		
Command: <hr/>		

Bild 68 Maske TR9090

In der Maske TR9090 ist der Name eines Parametersatzes festzulegen, der im Folgebildschirm erstellt wird. Der weitere Ablauf ist im Handbuch [13] beschrieben.

5.8 Rename/copy object

Mit der Funktion `Rename` können Sie Objekte umbenennen, mit der Funktion `Copy` Objekte kopieren.

Beide Funktionen werden in der Auswahlmaske TR9000 durch Ankreuzen von `Rename / copy object` angesteuert. Es wird dann die Maske TR9200 ausgegeben.

Die Maske TR9200 kann auch mit den Kommandos `COPY` oder `RENAME` aufgerufen werden. In diesem Fall wird die Auswahlmaske TR9000 übersprungen. Die Kommandos sind ausführlich in Kapitel 12 beschrieben.

TR9200	Rename / Copy	\$A1234567.LEO.TOMRFL.TOM.BEISPIEL
Function : . (Rename, Copy) Object : to :		
Command:		

Bild 69 Maske TR9200

Im Feld `Function` müssen Sie angeben:

- R falls ein Objekt umbenannt werden soll oder
- C falls ein Objekt kopiert werden soll.

- Beim Umbenennen müssen bei `Object` die alten Bezeichnungen für Typ (dreistellige Kurzbezeichnung) und Name eingetragen werden, bei `to` die entsprechenden neuen Bezeichnungen.
 - Das Attribut `Letztes Änderungsdatum` wird automatisch mit dem aktuellen Datum überschrieben.
 - Falls zum umbenannten Objekt Beziehungen existieren, so werden diese automatisch an den neuen Namen angepaßt.
 - Während des Ablaufs der Servicefunktion `Rename` ist das Data Dictionary global gesperrt.
- Beim Kopieren müssen bei `Object` Typ (dreistellige Kurzbezeichnung) und Name des Originals, bei `to` die entsprechenden Angaben für die Kopie eingetragen werden.
 - `Copy` bietet sich an, wenn mehrere Objekte gleichen Typs erstellt werden, die sich nur gering unterscheiden. Mit der Grundfunktion `Change` lassen sich dann die objektspezifischen Angaben leicht nachtragen.
 - Kopiert werden sämtliche Attribute, Beziehungen und Texte. Als `Erstellungsdatum` wird automatisch das aktuelle Datum eingetragen. Bereits erstellte Objekte können durch `Copy` nicht überschrieben werden.
 - Während des Ablaufs der Servicefunktion `Copy` ist die Kopie gesperrt.

Nach dem Abschicken der Maske mit der Taste **[DÜ]** wird, sofern die Funktion unmittelbar ausgeführt werden kann, erneut die Maske TR9200 ausgegeben. In der Meldungszeile steht die Vollzugsmeldung:

```
TR-70      Copy/Rename completed
```

Der Arbeitsbereich ist leer, so daß ein weiteres Objekt, wie eben beschrieben, umbenannt oder kopiert werden könnte. Mit der Taste **[K1]** oder durch Abschicken einer leeren Maske TR9200 mit der Taste **[DÜ]** wird die Servicefunktion beendet und in die Auswahlmaske TR9000 zurückgekehrt.

Das Umbenennen bzw. Kopieren von Objekten ist nicht uneingeschränkt möglich. TOM-REF prüft zuvor, ob das neue Objekt in die Struktur des Data Dictionary paßt. Beispielsweise kann es zu Konflikten führen, wenn sich der Objekttyp ändert. In diesem Fall gibt TOM-REF stets einen entsprechenden Hinweis in der Meldungszeile aus und fragt, ob der Objekttyp auch wirklich geändert werden soll. Falls vor dem Abschicken der Maske mit der Taste **[DÜ]** das in der Kommandozeile vorgegebene `Y` mit `N` überschrieben wird, führt TOM-REF die Funktion nicht aus.

5.9 Delete structure

Mit der Servicefunktion `Delete structure` können Sie Strukturen bestimmter Objekte und ihrer Beziehungen ganz oder teilweise löschen.

Ausgehend vom Objekt, welches in der Hierarchie der Struktur an erster Stelle steht, werden der Reihe nach alle untergeordneten Objekte angezeigt. Für jedes einzelne Objekt ist zu entscheiden, ob es gelöscht werden soll oder nicht. Ein Objekt kann jedoch nur dann gelöscht werden, wenn es nicht Zielobjekt in einer Beziehung ist. Falls ein Objekt auch noch von Objekten außerhalb oder innerhalb der Struktur referenziert wird, kann es demzufolge nicht gelöscht werden.

Die Ausführung der Servicefunktion `Delete structure` kann zur Folge haben, daß danach Objekte im Data Dictionary vorhanden sind, die nicht mehr bezogen sind. Solche Objekte können Sie sich mit der Servicefunktion `Display objects which are not part of another object` (siehe Seite 127) anzeigen lassen.

Der genaue Ablauf der Servicefunktion `Delete structure` wird an folgendem Fallbeispiel erläutert:

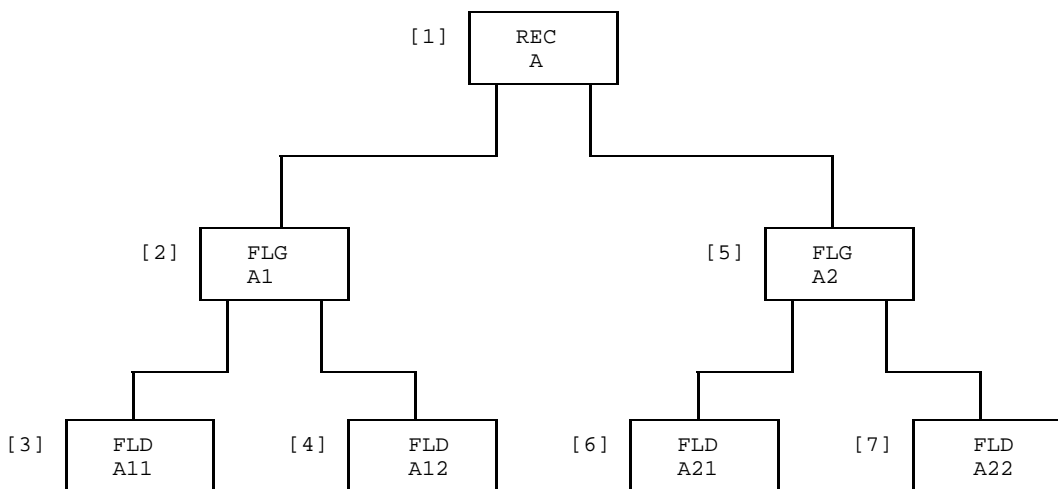


Bild 70 Beispielstruktur

Der Record A steht in der Hierarchie an erster Stelle. Die weitere Rangfolge ist in den Ziffern links neben den Objekten vermerkt. In dieser Reihenfolge werden die Objekte zum Löschen angezeigt. Vorausgesetzt wird, daß die Beziehungen bei den Ausgangsobjekten in der Reihenfolge des Bildes (von links rechts) erstellt wurden.

Ausgangspunkt der folgenden Ablaufbeschreibung ist die Auswahlmaske TR9000. Wie in der Einleitung zu diesem Kapitel beschrieben, wird die Auswahlmaske TR9000 übersprungen, wenn die Servicefunktion `Delete structure` mit dem Kommando `DELSTRUCT` aufgerufen wird. In diesem Fall entfällt Punkt 1 der Ablaufbeschreibung.

Vorgehen

- ▶ Kreuzen Sie in der Auswahlmaske TR9000 die Servicefunktion `Delete structure` an.
- ▶ Schicken Sie die Maske wird mit der Taste `[DÜ]` ab.
- ▶ Legen Sie dann in der Maske TR9210 Typ und Name des Objekts fest, das in der zu löschenden Struktur an erster Stelle steht.

TR9210 Delete structure	\$A1234567.LEO.TOMRFL.TOM.BEISPIEL
Delete all succeeding objects and relationships from Type: REC Name: A.....	
Type	Name
Resp	Creation
Change	
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
Command:	

Bild 71 Maske TR9210 mit Eingabebeispiel

- ▶ Schicken Sie die Maske mit der Taste `[DÜ]` ab.
Danach müssen Sie entscheiden, ob Objekt gelöscht werden soll. Im Arbeitsbereich sind in der zweiten Zeile Typ und Name des Objekts vermerkt, das gelöscht werden soll. In den folgenden Zeilen stehen die Objekte, zu denen direkte Beziehungen vorhanden sind.

5.10 Incorporate from another Data Dictionary

Mit der Servicefunktion `Incorporate from another Data Dictionary` können Sie Objekte (samt der Beziehungsliste und den Texten) von einem anderen Data Dictionary in das aktuelle Data Dictionary übertragen.

Dabei kann der Fall eintreten, daß zu einem zu übertragenden Objekt bereits ein Objekt desselben Typs und Namens im aktuellen Data Dictionary existiert. Bei solchen, sog. paarigen Objekten können Sie jeweils getrennt für Objektattribute, Beziehungsliste, TOM-REF-Text und TOM-TI-Langtext entscheiden, ob die Einträge des aktuellen Data Dictionary bestehen bleiben oder durch die entsprechenden Einträge des anderen Data Dictionary überschrieben werden.



Während der Übertragung ist ein Objekt im aktuellen Data Dictionary gesperrt.

Objekte, Beziehungen, Attribute und Texte werden in diesem Beispiel mit der Eigenschaft `alt` versehen, falls sie dem aktuellen Data Dictionary angehören. Analog bedeutet die Eigenschaft `neu` die Zugehörigkeit zum anderen Data Dictionary, also dem Data Dictionary, aus dem übertragen werden soll.

5.10.1 Allgemeiner Ablauf

Ausgangspunkt der folgenden Ablaufbeschreibung ist die Auswahlmaske TR9000. Wie in der Einleitung zu diesem Kapitel beschrieben, wird die Auswahlmaske TR9000 übersprungen, wenn die Servicefunktion `Incorporate from another Data Dictionary` mit dem Kommando `INCORPORATE` aufgerufen wird. In diesem Fall entfällt der erste Punkt der Ablaufbeschreibung.

- ▶ Kreuzen Sie in der Auswahlmaske TR9000 die Servicefunktion `Incorporate from another TOM-REF-library an`.
- ▶ Schicken Sie die Maske wird mit der Taste `[DÜ]` ab.
- ▶ Legen Sie dann in der Maske TR9100 den Umfang der Übertragung, den Modus und die Parameter fest.

```

TR9100  Incorporate                                     $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Incorporate from
  Catid      : .....:  Userid   : $A1234567
  Identification: LEO  REF-name : TOM...  Project : BEISPIEL
  Prefix     : .....
  Suffix     : .....
Incorporate
  Object-type : ...
  Start-name  : .....
  End-name    : .....
  Comp.-list  : .....CLS
Dialog Control : X
Immediate action : .
  Empty incoming attributes significant (Y/N)      : N
  Empty incoming text-elements significant (Y/N)   : N
  Delete unpaired old relationships (Y/N)         : Y

Delete all paired objects incl. relationships and text (Y/N) : N
-----
Command: .....
    
```

Bild 73 Maske TR9100 (mit Vorbelegung)

Der Ablauf der Funktion wird durch die Eingaben in der Maske TR9100 festgelegt. Die Eingaben in der Maske TR9100 können folgendermaßen untergliedert werden:

Umfang der Übertragung

Damit ist die Identifizierung des anderen Data Dictionary sowie die Eingrenzung der zur Übertragung in Frage kommenden Objekte gemeint.

Incorporate from

Hier geben Sie das Data Dictionary an, von dem Einträge übernommen werden sollen. Kennung, Namenskurzzeichen, Projektbezeichnung und ggf. Prefix und Suffix sind mit den aktuellen Parametern der *Specialfunction* TOMREF.SPF vorbelegt. Der Ref-Name ist anzugeben. Das Feld Catid ist schreibgeschützt. Es enthält die Katalogkennung, falls der Administrator diese zu Dokumentationszwecken in die Aufruf-Prozedur geschrieben hat. TOM-REF arbeitet grundsätzlich mit der Default-Katalogkennung.

Incorporate

Hier bestimmen Sie die Objekte, die ins aktuelle Data Dictionary übertragen werden sollen. Sie haben dazu drei Möglichkeiten:

- Kurzbezeichnung eines Objekttyps.
In diesem Fall sind alle Objekte dieses Typs gemeint. Diese Menge kann durch die Angabe eines Start-namens und/oder eines End-namens, bezogen auf die alphabetische Reihenfolge der Objekte, weiter eingegrenzt werden. Soll nur ein Objekt übertragen werden, so ist dessen Name sowohl als Start-name als auch als End-name einzugeben.

- Statt der Kurzbezeichnung eines Objekttyps kann auch *T eingegeben werden. Dann sind alle Objekte des Data Dictionary gemeint. In diesem Fall darf kein Start-name und/oder End-name angegeben werden.
- Name einer Komponentenliste. Das bedeutet, alle Objekte der Komponentenliste kommen für eine Übertragung ins aktuelle Data Dictionary in Frage.

Modus

Sie können entscheiden, ob die Funktion mit Sichtkontrolle (`Dialog Control`) oder automatisch (`Immediate action`) abläuft.

`Dialog Control`

In diesem Modus werden für alle paarigen Objekte jeweils Attribute, Beziehungsliste und TOM-REF-Text in der Maske TR9102 gegenübergestellt. Sie können entscheiden, welcher Eintrag jeweils übertragen werden soll. Die Übertragung ist abhängig von der Belegung der ersten drei der in der Maske TR9100 folgenden vier Parameter. Der vierte Parameter muß im Modus `Dialog Control` stets mit `N` belegt sein. Dieser Modus ist voreingestellt. Die genaue Ablaufbeschreibung in diesem Modus ist ab Seite 155 beschrieben.

`Immediate action`

In diesem Modus werden die Objekte ohne Sichtkontrolle unmittelbar übertragen. Die Übertragung ist ebenfalls abhängig von der Belegung der folgenden vier Parameter.

Parameter

Die Parameter gelten für beide Übertragungsmodi und legen weitere Modalitäten der Übertragung fest.

`Empty incoming attributes significant` (Y/N)

Dieser Parameter bezieht sich auf die acht einstelligen neutralen Objektattribute. Bei `Y` werden auch leere Attribute übertragen. Bei `N` werden nur die nicht leeren Attribute übertragen. Vorbelegung: `N`

Beispiel

Attribute vor dem Überschreiben:

Altes Objekt: A . B

Neues Objekt: . S T

Attribute nach dem Überschreiben:

Parameterwert Y: . S T

Parameterwert N: A S T

Empty incoming text-elements significant (Y/N)

Dieser Parameter bezieht sich auf den TOM-REF-Text. Bei Y ist auch Leertext für die Übernahme relevant. Bei N erfolgt die Übernahme nur, falls das neue Objekt nicht-leeren Text enthält. Vorbelegung: N

Delete unpaired old relationships (Y/N)

Dieser Parameter legt die Behandlung unpaariger Beziehungen fest. Damit sind Beziehungen gemeint die entweder nur beim alten Objekt oder nur beim neuen, zu übertragenden Objekt existieren.

Eine alte und eine neue Beziehung gelten als paarig, wenn ihre Zielobjekte in Typ und Namen übereinstimmen. Vorbelegung: Y

Bei Y werden alle unpaarigen alten Beziehungen gelöscht und durch die unpaarigen neuen ersetzt.

Ergebnis: Die Liste der neuen Beziehungen.

Bei N bleiben die unpaarigen alten Beziehungen erhalten, die paarigen alten werden gelöscht. Zusätzlich werden alle neuen Beziehungen aufgenommen.

Ergebnis: Alle unpaarigen alten Beziehungen plus sämtliche neuen Beziehungen.

Beispiel

Das Objekt Kunde vom Typ RECORD sei in beiden Data Dictionaries vorhanden.

Beziehungsliste des alten Objekts:

FIELD Name
FIELD Kunden-Nummer

Beziehungsliste des neuen Objekts:

FIELD Telefon-Nummer
FIELD Name

Beziehungsliste nach der Übertragung:

Parameterwert Y:

FIELD Telefon-Nummer
FIELD Name

Parameterwert N:

FIELD Kunden-Nummer
FIELD Telefon-Nummer
FIELD Name

Delete all paired old objects incl. relationships and text (Y/N)

Bei **Y** wird das Objekt vollständig gelöscht und dann neu aus dem anderen Data Dictionary aufgenommen. Die übrigen Parameter bleiben unberücksichtigt.

N bedeutet lediglich, daß die Übertragung unter Berücksichtigung der anderen Parameter stattfindet. Im Modus `Dialog Control` muß dieser Parameter mit **N** belegt sein.

Vorbelegung: **N**

5.10.2 Modus Dialog Control

Im Modus `Dialog Control` werden die Einträge zu zwei paarigen Objekten in der Maske TR9102 gegenübergestellt. Dies geschieht in drei getrennten Schritten:

1. Objektattribute
2. Beziehungsliste
3. TOM-REF-Text

Falls TOM-TI-Langtext zur Übertragung ansteht, wird zuerst der Text, der dem alten Objekt im aktuellen Data Dictionary zugeordnet ist, angezeigt. Anschließend wird der Text, der zum neuen Objekt gehört, angezeigt.

Bei jedem Schritt wird abgefragt, ob die angezeigten Einträge des alten Objekts mit denen des neuen Objekts überschrieben werden sollen. Wird ein Überschreiben gewünscht, werden die Einträge in Abhängigkeit der ersten drei Parameter übertragen. (Der vierte Parameter muß im Modus `Dialog Control` mit **N** belegt sein.)



Beachten Sie, daß der zweite Parameter nur die Übertragung des TOM-REF-Textes steuert. Für den TOM-TI-Langtext ist er ohne Bedeutung.

5.10.2.1 Behandlung der Objektattribute

In der Maske TR9102 werden auf der linken Seite die Attribute des Objekts im aktuellen Data Dictionary (old values), auf der rechten Seite die Attribute des paarigen Objekts im anderen Data Dictionary (new values) angezeigt. Es handelt sich um die neutralen Attribute `Verantwortlicher`, `Erstellungsdatum`, `Aenderungsdatum`, die acht einstelligen Attribute und die Kurzbeschreibung sowie um sämtliche objekttyp-spezifischen Attribute.

```

TR9102  Incorporate selection                               $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Type : RECORD                      Name : ADRESSE
-----
Old values:                                New-values:
RESP.                                     RESP.  LEO
CREAT. 23.02.95                          CREAT. 22.05.95
CHANGE 22.05.95                          CHANGE . .
ATT.   A C                               ATT.   XY
DESC.1 KUNDENANSCHRIFT                   DESC.1
-     2                                  -     2
-     3                                  -     3
-     4                                  -     4
REDEF.                                     REDEF.
COPY                                       COPY ANSCHRIFT
PREFIX                                    PREFIX
SUFFIX                                    SUFFIX
-----
Command: Y.....
-----
TR-06 Do you want to overwrite object ? (y/n)
    
```

Bild 74 Maske TR9102 mit Ausgabebeispiel



Wenn die ausgegebene Liste länger als eine Maske ist, können die Folgemasken mit der Taste **[DÜ]** der Reihe nach abgerufen werden. In der Kommandozeile ist dann ein + vorgegeben. In den Folgemasken kann geblättert werden.

Ansonsten (oder nach Beenden des Blätterns) ist in der Kommandozeile ein Y vorgegeben.

Falls die alten Attribute überschrieben werden sollen, ist die Maske mit der Taste **[DÜ]** abzuschicken. Dabei ist die Überschreibung der acht einstelligen Attribute davon abhängig, wie der Parameter *Empty incoming attributes significant (Y/N)* in der Maske TR9100 versorgt wurde. Bei Y werden alle acht einstelligen Attribute mit den neuen Attributen belegt. Bei N wird das neue Attribut nur dann übernommen, wenn es einen Wert besitzt, also nicht leer ist.

Falls die alten Attribute bleiben sollen, ist das Y mit N zu überschreiben und die Maske mit der Taste **[DÜ]** abzuschicken.

5.10.2.2 Behandlung der Beziehungslisten

Im Modus *Dialog Control* gilt:

Falls mindestens eines von zwei paarigen Objekten eine nicht leere Beziehungsliste besitzt, wird die Maske TR9102 mit der Gegenüberstellung beider Listen ausgegeben.

In der Maske TR9102 wird auf der linken Seite die Beziehungsliste des Objekts im aktuellen Data Dictionary (old values), auf der rechten Seite die Beziehungsliste des paarigen Objekts im anderen Data Dictionary (new values) angezeigt.

```

TR9102  Incorporate selection                                $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Type : RECORD                      Name : ADRESSE
-----
Old values:                          New-values:
FLD  NAME                            FLD  TEL.NR
FLD  WOHNORT                          FLD  NAME
-----
Command: Y.....
-----
TR-07  Do you want to overwrite relationships ? (y/n)

```

Bild 75 Maske TR9102 mit Ausgabebeispiel

Bei erweiterten Beziehungen wird hinter dem Objektnamen ein * ausgegeben.



Wenn die ausgegebene Liste länger als eine Maske ist, können die Folgemasken mit der Taste **(DÜ)** der Reihe nach abgerufen werden. In der Kommandozeile ist dann ein + vorgegeben. In den Folgemasken kann geblättert werden.

Ansonsten (oder nach Beenden des Blätterns) ist in der Kommandozeile ein Y vorgegeben.

Falls die alte Beziehungsliste erhalten bleiben soll, ist das Y mit N zu überschreiben und die Maske mit der Taste **(DÜ)** abzuschicken.

Falls die alte Beziehungsliste überschrieben werden soll, muß die Maske unmittelbar mit der Taste **(DÜ)** abgeschickt werden. Das Ergebnis der Überschreibung hängt davon ab, wie der Parameter `Delete unpaired old relationships (Y/N)` in der Maske TR9100 versorgt wurde. Bei Y ist die Beziehungsliste auf der rechten Seite der Maske, also die Beziehungsliste des paarigen Objekts im anderen Data Dictionary gültig. Bei N setzt sich die resultierende Beziehungsliste aus den unpaarigen alten Beziehungen auf der linken Seite und sämtlichen Beziehungen auf der rechten Seite zusammen.

5.10.2.3 Behandlung des TOM-REF-Textes

Hinsichtlich der Art des TOM-REF-Textes zu einem Objekt sind folgende Fälle zu unterscheiden:

1. Es ist Text vorhanden (TEXT=Y).
2. Es ist Leertext vorhanden (TEXT=Y).
3. Es ist kein Text vorhanden (TEXT=N).

In Klammern ist die entsprechende Belegung des neutralen Attributs TEXT angegeben.

Im Modus Dialog Control werden bei paarigen Objekten die TOM-REF-Texte stets in der Maske TR9102 gegenübergestellt, falls eine Übertragung ansteht.

i TOM-REF-Texte werden nur dann in der Maske TR9102 gegenübergestellt, wenn neuer Text vorhanden ist (TEXT=Y). Wenn neuer und alter Text leer oder nicht vorhanden sind, werden die Texte nicht gegenübergestellt und damit auch nicht übertragen.

```

TR9102 Incorporate selection                               $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Type : RECORD                      Name : ADRESSE
Old values:                        New-values:
Hier steht der TOM-REF-Text
des Objekts im aktuellen Data
Dictionary Das paarige Objekt
im anderen Data Dictionary
besitzt in diesem Beispiel
nur Leertext.

-----
Command: Y.....
-----
TR-08 Do you want to overwrite text ? (y/n)
    
```

Bild 76 Maske TR9102 mit Ausgabebeispiel

Die Maske TR9102 enthält auf der linken Seite (Old values) den TOM-REF-Text des Objekts im aktuellen Data Dictionary. Der TOM-REF-Text des paarigen Objekts im anderen Data Dictionary steht auf der rechten Seite (New values).

Die TOM-REF-Texte werden in der Maske TR9102 folgendermaßen angezeigt:

1. Es ist Text vorhanden:
Der Text wird, soweit in der Maske TR9102 Platz vorhanden ist, angezeigt. Falls eine Folgemaske existiert, ist in der Kommandozeile ein + vorgegeben. Mit der Taste **[DÜ]** wird die Maske TR9102 mit dem restlichen Text abgerufen. Blättern innerhalb der Masken ist möglich.
2. Es ist nur Leertext vorhanden:
In diesem Fall enthält die betreffende Seite Leerzeichen.
3. Es ist kein Text vorhanden (betrifft nur den alten Text):
Es werden ebenfalls Leerzeichen ausgegeben.

Übertragung steuern

Soll der alte TOM-REF-Text (*Old values*) erhalten bleiben, ist das in der Kommandozeile vorgegebene *Y* mit *N* zu überschreiben und die Maske mit der Taste **[DÜ]** abzuschicken. Die Funktion ist dann beendet.

Damit eine Übernahme des neuen TOM-REF-Textes (*New values*) überhaupt stattfinden kann, muß die Vorbelegung *Y* in der Kommandozeile erhalten bleiben. Ob der neue TOM-REF-Text nach dem Abschicken der Maske mit der Taste **[DÜ]** dann wirklich übernommen wird, hängt zum einen von der Art des Textes und zum anderen, wie der Textparameter *Empty incoming text-elements significant (Y/N)* in der Maske TR9100 versorgt wurde. Damit wird festgelegt, wie zu verfahren ist, wenn das zu übertragende Objekt nur Leertext enthält.

Die folgende Tabelle zeigt, welcher Text nach Beendigung der Funktion schließlich relevant ist. Die Angaben *Y* oder *N* stehen für die Versorgung des Textparameters.

Neuer Text Alter Text	Text vorhanden	Leertext vorhanden	Kein Text vorhanden
Text vorhanden	Y: Neuer Text N: Neuer Text	Y: Leertext N: Alter Text	
Leertext vorhanden	Y: Neuer Text N: Neuer Text	Keine Anzeige; d.h. keine Übertragung. Der alte Zustand des Textes bleibt unabhängig von der Versorgung des Textparameters erhalten.	
Kein Text vorhanden	Y: Neuer Text N: Neuer Text		

Bei der Übernahme eines unpaarigen Objekts gilt:
Enthält das zu übernehmende Objekt Leertext, wird kein Text übernommen. D.h. das neutrale Attribut *TEXT* wird mit *N* vorbelegt, falls der Parameter für den TOM-REF-Text mit *N* belegt ist.

5.10.2.4 Behandlung der TOM-TI-Langtexte

Ein TOM-TI-Langtext wird als Member der TOMLIB durch den Typ und den Namen des Objekts, zu dem er gehört, identifiziert. Da bei paarigen Objekten Typ und Name jeweils identisch sind, können nur dann verschiedene TOM-TI-Langtexte existieren, wenn sie in verschiedenen TOMLIB's abgelegt sind. Das bedeutet: Das Data Dictionary, aus dem übertragen wird, muß sich in `nkz` oder `project` vom aktuellen Data Dictionary unterscheiden. Dies wird im folgenden vorausgesetzt.

Es sind zwei Fälle zu unterscheiden, bei denen TOM-TI-Langtext vor der Übernahme angezeigt wird:

1. Fall: Es ist sowohl alter als auch neuer TOM-TI-Langtext vorhanden.

Der Ablauf läßt sich in drei Schritte gliedern:

- In der TOM-TI-Maske wird der alte Langtext angezeigt. Mit der Taste **K1** oder mit der Taste **DÜ** kann die Anzeige des alten Langtextes beendet werden.
- In der TOM-TI-Maske wird der neue Langtext angezeigt. Mit der Taste **K1** oder mit der Taste **DÜ** können Sie die Anzeige des neuen Langtextes beenden.
- In der Maske TR0005 entscheiden Sie, welcher Langtext relevant sein soll.

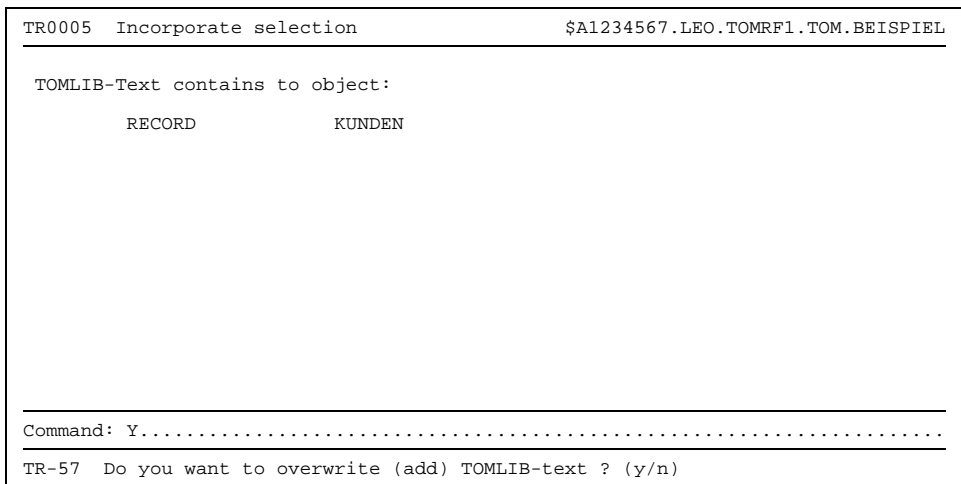


Bild 77 Maske TR0005 mit Ausgabebeispiel

Im Arbeitsbereich der Maske TR0005 sind Typ und Name der paarigen Objekte aufgeführt.

Falls der neue Langtext übernommen werden soll, ist die Maske TR0005 mit der Taste **[DÜ]** abzuschicken.

Falls der alte Langtext erhalten bleiben soll, ist entweder vor dem Betätigen der Taste **[DÜ]** das in der Kommandozeile vorgegebene γ mit N zu überschreiben oder unmittelbar die Taste **[K1]** zu betätigen.

2. Fall: Es ist nur neuer TOM-TI-Langtext vorhanden.

Der Ablauf läßt sich in drei Schritte gliedern:

- Die Maske TR0005 (Bild 77) wird mit einem Hinweis ausgegeben, daß kein alter Langtext vorhanden ist (Meldung TR-55). Die Maske ist mit der Taste **[DÜ]** abzuschicken.
- In der TOM-TI-Maske wird der neue Langtext angezeigt. Mit der Taste **[K1]** können Sie die Anzeige beenden.
- Die Maske TR0005 wird mit der Frage ausgegeben, ob der neue Langtext übernommen werden soll (Meldung TR-57). Soll der neue Langtext übernommen werden, ist die Maske TR0005 mit der Taste **[DÜ]** abzuschicken. Soll der alte Langtext erhalten bleiben, ist entweder vor dem Betätigen der Taste **[DÜ]** das in der Kommandozeile vorgegebene γ mit N zu überschreiben oder unmittelbar die Taste **[K1]** zu betätigen.

5.10.2.5 Ende der Übertragung

Nach der fehlerfreien Ausführung der Servicefunktion `Incorporate from another Data Dictionary` wird im Meldungsfenster der Maske TR9100 ausgegeben:

```
TR-58    Incorporate-function correct finished
```

Falls aufgrund der Eingaben in der Maske TR9100 keine Objekte zur Übertragung anstanden, wird folgende Meldung ausgegeben:

```
TR-59    No input-object existing
```

Die Anwendereingaben zur soeben beendeten Funktion bleiben in der Maske TR9100 bestehen. Es können Eingaben zu einer weiteren Funktion getätigt werden.

Mit der Taste **[K1]** wird die Servicefunktion `Incorporate from another Data Dictionary` beendet.

5.11 Check objects and relationships

Die Servicefunktion `Check objects and relationships` benötigen Sie, um

- Strukturen, die mit älteren TOM-REF-Versionen erstellt wurden, gegen den Strukturkatalog der aktuellen TOM-REF-Version zu prüfen;
- um die Übereinstimmung der Strukturkataloge verschiedener Installationen zu prüfen.

Wenn Sie in der Maske TR9000 das Feld `Check objects and relationships` ankreuzen und die Taste `[DÜ]` drücken, veranlassen Sie TOM-REF zum Überprüfen der Objekte und Beziehungen auf ihre Gültigkeit.

Wie in der Einleitung zu diesem Kapitel beschrieben, kann die Servicefunktion `Check objects and relationships` auch mit dem Kommando `CHECK` aufgerufen werden. Die Auswahlmaske TR9000 wird dabei übersprungen.

Konnte TOM-REF keine Inkonsistenzen feststellen, dann erhalten Sie folgende Meldung:

```
TR-166    No errors found
```

Ansonsten werden die fehlerhaften Objekte in die SYSLST-Datei eingetragen. Der Name der SYSLST-Datei ist standardmäßig: `nkz.SYSLST.LST.project`

Der Dateiname für SYSLST kann sich ändern, wenn das Kommando `CHECK` bei der Servicefunktion `Print functions` abgesetzt wird. Voraussetzung dafür ist, daß SYSLST innerhalb von `Print functions` in eine andere Datei umgelenkt wurde.

Die SYSLST-Datei wird in der Aufruf-Prozedur zugewiesen. Sie kann nach der Beendigung von TOM-REF im Editor gelesen werden.

5.12 Administration (ISAM)

Wenn Sie in der Auswahlmaske TR9000 für die Servicefunktionen das Feld `Administration (ISAM)` ankreuzen, wird nach dem Abschicken der Maske mit der Taste `(DÜ)` die Auswahlmaske TR9800 für die Administration ausgegeben.

In der Maske TR9800 werden folgende Servicefunktionen zur Verwaltung von Data Dictionaries angeboten:

- Data Dictionary in der TOMLIB sichern
- Data Dictionary aus der TOMLIB holen
- TOMRFI-Datei neu aufbauen
- Neues Data Dictionary anlegen

```
TR9800  ISAM administration selection                $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Please select function(s):
.  Save into TOMLIB
.  Restore from TOMLIB
.  Reestablish TOMREFI
.  Initiate TOM-REF Data Dictionary

-----
Command: .....
```

Bild 78 Auswahlmaske TR9800 für Administration

Die Auswahl erfolgt durch Ankreuzen der gewünschten Servicefunktion. Sie können auch mehrere Funktionen ankreuzen. Die Bearbeitung erfolgt dann in der Reihenfolge der Auswahlmaske TR9800. Nach der Bearbeitung der letzten markierten Funktion wird die Maske TR9800 erneut ausgegeben. Mit der Taste `(K1)` können sie dann in die Auswahlmaske TR9000 zurückkehren.

5.12.1 Save into TOMLIB - Data Dictionary in TOMLIB sichern

Mit der Servicefunktion `Save into TOMLIB` können Sie Ihr Data Dictionary als Member (Bibliothekselement) in der TOMLIB ablegen.

```

TR9900  Save into TOMLIB                               $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
                                         Element-name in TOMLIB: TOMRF1 .REF
                                         Erase TOMREF-library after save (Y/N): N
-----
Command: .....

```

Bild 79 Maske TR9900 (Sichern in TOMLIB)

Das Feld `Element-name in TOMLIB:` enthält den Namen und den Typ des Members, in dem das Data Dictionary in der TOMLIB gesichert werden soll. Der Name ist mit dem Teilnamen des aktuellen Data Dictionary (hier: TOMRF1) vorbelegt und überschreibbar. Der Typ `REF` ist nicht überschreibbar.

Das Feld für die Antwort auf die Frage, ob die Dateien des Data Dictionary nach dem Sichern in der TOMLIB gelöscht werden sollen, ist mit `N` (Nein) vorbelegt. Das Feld ist nur dann überschreibbar, wenn sich das Data Dictionary in der eigenen Kennung befindet. Bei einer Belegung mit `Y` (Ja) würden die Dateien nach dem Sichern gelöscht.

Der weitere Ablauf hängt von der Belegung des eben erläuterten Feldes ab:

1. Belegung `y`:

Die Maske TR9900 nach dem Betätigen der Taste `(DÜ)` mit folgender Meldung erneut ausgegeben:

```
TR-112  Data Dictionary has been erased after save
```

Dies bedeutet: Die Dateien des Data Dictionary wurden nach dem Sichern gelöscht. Mit der nächsten Eingabe (Taste `(DÜ)` oder Taste `(K1)`) wird TOM-REF beendet.

2. Belegung N:

Die Maske wird nach dem Betätigen der Taste **[DÜ]** mit folgender Meldung erneut ausgegeben, sofern die Sicherung erfolgreich abgeschlossen werden konnte:

```
TR-111      Data Dictionary has been saved
```

Dies bedeutet: Das Data Dictionary wurde gesichert, die Dateien sind weiterhin vorhanden.

Nach dem Betätigen der Taste **[K1]** verlassen Sie die Maske TR9900.

5.12.2 Restore from TOMLIB - Data Dictionary aus TOMLIB holen

Mit der Servicefunktion `Restore from TOMLIB` können Sie ein Data Dictionary, das Sie als Member in der TOMLIB gesichert haben, wieder ins BS2000 bringen. Dabei wird das aktuelle Data Dictionary mit dem Member aus der TOMLIB überschrieben. Das aktuelle Data Dictionary wird reorganisiert und eine neue TOMRFI-Datei angelegt.

Während des Ablaufs der Servicefunktion `Restore from TOMLIB` ist das Data Dictionary für andere Benutzer gesperrt.

```
TR9910  Restore from TOMLIB                               $A1234567.LEO.TOMRF1.TOM.BEISPIEL
-----
Element-name in TOMLIB : TOMRF1 .REF

Command: .....
```

Bild 80 Maske TR9910 (Data Dictionary aus TOMLIB holen)

Der Name des Members der TOMLIB ist mit dem Namen des aktuellen Data Dictionary vorbelegt (hier: TOMRF1). Der Name ist überschreibbar.

Der Typ des Members lautet `REF` und ist nicht überschreibbar.

Nach dem Abschicken der Maske TR9910 mit der Taste **[DÜ]** wird die Maske erneut mit der Frage ausgegeben, ob die Funktion ausgeführt werden soll:

```
TR-189    Do you want to restore Data Dictionary ?
```

Nach dem Abschicken der Maske TR9910 mit der Taste **[DÜ]** wird die Funktion ausgeführt. Der Abschluß der Funktion wird mit folgender Meldung in der Folgemaske TR9910 bestätigt:

```
TR-118 Data Dictionary is restored
```

Die Funktion wird nicht ausgeführt, wenn vor dem Abschicken der Maske mit der Taste **[DÜ]** das in der Kommandozeile vorgegebene **y** mit **n** überschrieben wird oder wenn die Taste **[KJ]** betätigt wird. In beiden Fällen wird das nicht Ausführen in der Folgemaske TR9910 bestätigt (Meldung TR-255).

Die Maske TR9910 verlassen Sie mit der Taste **[KJ]**.



Wenn Sie mit einem Data Dictionary gearbeitet haben und zu einem anderen über die Servicefunktion `Restore from TOMLIB` wechseln wollen, ohne daß das ursprüngliche überschrieben wird, empfiehlt sich folgendes Vorgehen: Anlegen eines neuen Data Dictionary mit der Servicefunktion `Initiate TOM-REF Data Dictionary` (siehe Seite 167). Danach kann das neue Data Dictionary über die Identifikationsmaske TR0000 zum aktuellen Data Dictionary erklärt und die Servicefunktion `Restore from TOMLIB` gestartet werden.

5.12.3

Reestablish TOMREFI - TOMRFI-Datei neu aufbauen

Mit der Servicefunktion `Reestablish TOMRFI` können Sie die TOMRFI-Datei (Inverse Beziehungen) wieder neu erstellen. Während des Ablaufs der Funktion ist das Data Dictionary für andere Benutzer gesperrt.

Wenn Sie diese Funktion in der Auswahlmaske TR9800 für die Administration ankreuzen, wird die Maske nach dem Abschicken mit der Taste **[DÜ]** erneut mit folgender Meldung ausgegeben:

```
TR-199    Do you really want to reestablish TOMREFI ? (y/n)
```

Wenn Sie die Maske mit der Taste **[DÜ]** abschicken, wird die Funktion ausgeführt.

Die Funktion wird nicht ausgeführt, wenn sie das in der Kommandozeile vorgegebene **y** mit **n** überschreiben und dann die Maske mit der Taste **[DÜ]** abschicken. In diesem Fall wird in der Folgemaske in der Meldungszeile der Hinweis ausgegeben, daß die Funktion nicht ausgeführt wurde (Meldung TR-180).

5.12.4 Initiate TOM-REF Data Dictionary - Neues Data Dictionary anlegen

Mit der Servicefunktion `Initiate TOM-REF Data Dictionary` können Sie ein neues Data Dictionary anlegen.

Falls zum neuen Data Dictionary noch keine TOMLIB existiert, wird diese ebenfalls automatisch angelegt.

TR9930	Initiate	\$A1234567.LEO.TOMRFL.TOM.BEISPIEL
TOMREF-Data Dictionary:		
Catid	:	:....:
Userid	:	: \$QM222TOM
Identification	:	: LEO
REF-name	:	: TOMRFL
Project	:	: BEISPIEL
Prefix	:	:
Suffix	:	:
Command:		

Bild 81 Maske TR9930 (Initialisierung eines Data Dictionary)

Die Felder in dieser Maske sind mit den Angaben aus der Statuszeile vorbelegt. Mit Ausnahme der Felder `Catid` (Katologkennung) und `Userid` sind alle Felder überschreibbar. Das Feld `Catid` dient lediglich zur Dokumentation und ist evtl. mit einem Wert aus der Aufruf-Prozedur vorbelegt.

Falls Sie die Felder `Prefix` bzw. `Suffix` versorgen, so werden die Namen der erstellten Dateien mit der angegebenen Vorsilbe bzw. Nachsilbe versehen.

Nach dem Ausfüllen der Maske TR9930 und dem Abschicken mit der Taste wird die Maske erneut mit einer Meldung ausgegeben, deren Inhalt davon abhängt, ob das spezifizierte Data Dictionary bereits existiert oder nicht.

1. Fall: Das Data Dictionary existiert noch nicht.

```
TR-200 Do you really want to initiate Data Dictionary ? (y/n)
```

2. Fall: Das Data Dictionary existiert bereits.

```
TR-201 Do you really want to initiate existing Data Dictionary ?
(y/n)
```

In beiden Fällen wird nach dem Abschicken der Maske mit der Taste **[DÜ]** die Funktion ausgeführt. In der Folgemaske TR9930 wird der Abschluß der Funktion durch folgende Meldung bestätigt:

```
TR-196    Data Dictionary is initiated
```

Soll die Funktion nicht ausgeführt werden so ist entweder vor dem Abschicken der Maske mit der Taste **[DÜ]** das in der Kommandozeile vorgegebene **Y** mit **N** zu überschreiben oder die Taste **[K1]** zu betätigen. In beiden Fällen wird das nicht Ausführen der Funktion in der Folgemaske TR9930 bestätigt (Meldung **TR-191**).

6 DVS/Leasy

Ein COBOL-Programm beschreiben Sie im Data Dictionary mit den Objekten und Beziehungen des DVS/Leasy-Teistrukturkatalogs. Ein COBOL-Programm beschreiben heißt, daß Sie die nötigen Objekte und Beziehungen erstellen. In Kapitel 4 ist beschrieben, wie Sie Objekte und Beziehungen erstellen können.

6.1 DVS/Leasy-Objekte

Die folgende Tabelle zeigt die Objekttypen der DVS/Leasy-Teilstruktur mit ihren Kurznamen, Namen und einer kurzen Erläuterung.

Kurzname	Langname des Objekttyps	Erläuterung
CMD	COMMAND	Kommando
FIL	FILE	Logische Dateibeschreibung in COBOL-Programm (File Description, LINK-Name)
FKY	FUNCTION-KEY	Funktionstaste
FLD	FIELD	Datenfeld eines COBOL-Programms oder einer Datenbank usw.
FLG	FIELDGROUP	Datenfeld-Gruppe eines COBOL-Programms oder einer UDS-Datenbank
INL	COPY	Copy-Element für ein COBOL-Programm (Data area ggf. mit Initialisierung, Move-Anweisungen oder File Description)
MSK	FORMAT	Masken-Format
PFL	PHYSICAL-FILE	Physikalische Datei

Tabelle 8 Objekttypen im DVS/LEASY-Teilstrukturkatalog

Kurzname	Langname des Objekttyps	Erläuterung
PKY	PRIMARY-KEY	Primärschlüssel einer ISAM-Datei, auch von LEASY verwendet
REC	RECORD	Datensatz
SKY	SECONDARY-KEY	Sekundärschlüssel einer Leasy-Anwendung
SEQ	SEQUENCE	Formatsequenz
VOL	VOLUME	Datenträger

Tabelle 8 Objekttypen im DVS/LEASY-Teilstrukturkatalog (Fortsetzung)

6.2 DVS-Leasy-Beziehungen

Zwischen welchen Objekten des DVS/Leasy-Teilstrukturkatalogs Sie Beziehungen erstellen können, zeigt die folgende Tabelle. Dabei bedeutet ein X in einem Feld, daß zwischen zwei Objekttypen eine Beziehung bestehen kann. Wie Sie diese Beziehungen erstellen, ist in Kapitel 4 beschrieben.

Objekttyp als Ausgang der Beziehung	Objekttyp als Ziel der Beziehung													
	CMD	FIL	FKY	FLD	FLG	MSK	PKY	REC	RET	RLS	SBR	SKY	TAC	VOL
CMD Command				X										
FIL File		X					X	X				X		
FLG Field Group				X	X									
INL Copy		X		X	X	X		X			X			
MSK Format				X	X	X		X	X	X				
PFL Physical File		X												X
PKY Primary Key				X	X									
REC Record				X	X		X					X		

Tabelle 10 Beziehungstypen des DVS/LEASY-Teilstrukturkatalogs

Objekttyp als Ausgang der Beziehung	Objekttyp als Ziel der Beziehung													
	CMD	FIL	FKY	FLD	FLG	MSK	PKY	REC	RET	RLS	SBR	SKY	TAC	VOL
RET Return Info	X		X	X		X							X	
RLS Relationship	X		X	X		X							X	
SKY Secondary Key				X	X		X							
SEQ Sequence						X								
TAC UTM TAC				X										

Tabelle 10 Beziehungstypen des DVS/LEASY-Teilstrukturkatalogs (Fortsetzung)

Mehrstellige Beziehungen können auch mit Hilfe der Typen `RLS` (vom Ausgangsobjekt zum Zielobjekt) und `RET` (vom Zielobjekt zum Ausgangsobjekt) gebildet werden. Hier ist das Beispiel 'Format (`MSK`) ruft Format (`MSK`) über Funktionstaste (`FKY`)' enthalten. Die direkte Beziehung geht von `MSK` zu `RLS` bzw. `RET`. Von `RLS` bzw. `RET` geht eine Beziehung zu `MSK` und `FKY`.

6.3 DVS/Leasy-Teilstrukturkatalog

Die Teilstrukturkataloge von DVS/Leasy, Funktionsentwurf und Datenentwurf sind Basis und Hilfe für die Beschreibung eines COBOL-Programms im Data Dictionary. Sie legen fest, welches Objekt das Ausgangsobjekt und welches das Zielobjekt einer Beziehung sein muß. Die Pfeile zeigen vom Typ des Ausgangsobjekts zum Typ des Zielobjekts. Eine Beziehung müssen Sie immer in die Beziehungsliste des Ausgangsobjekts eintragen. Die Elementaraussagen klassifizieren die Beziehungen näher.

Eine genaue Beschreibung der in diesem Modell verwendeten Darstellungsmittel finden Sie ab Seite 7.

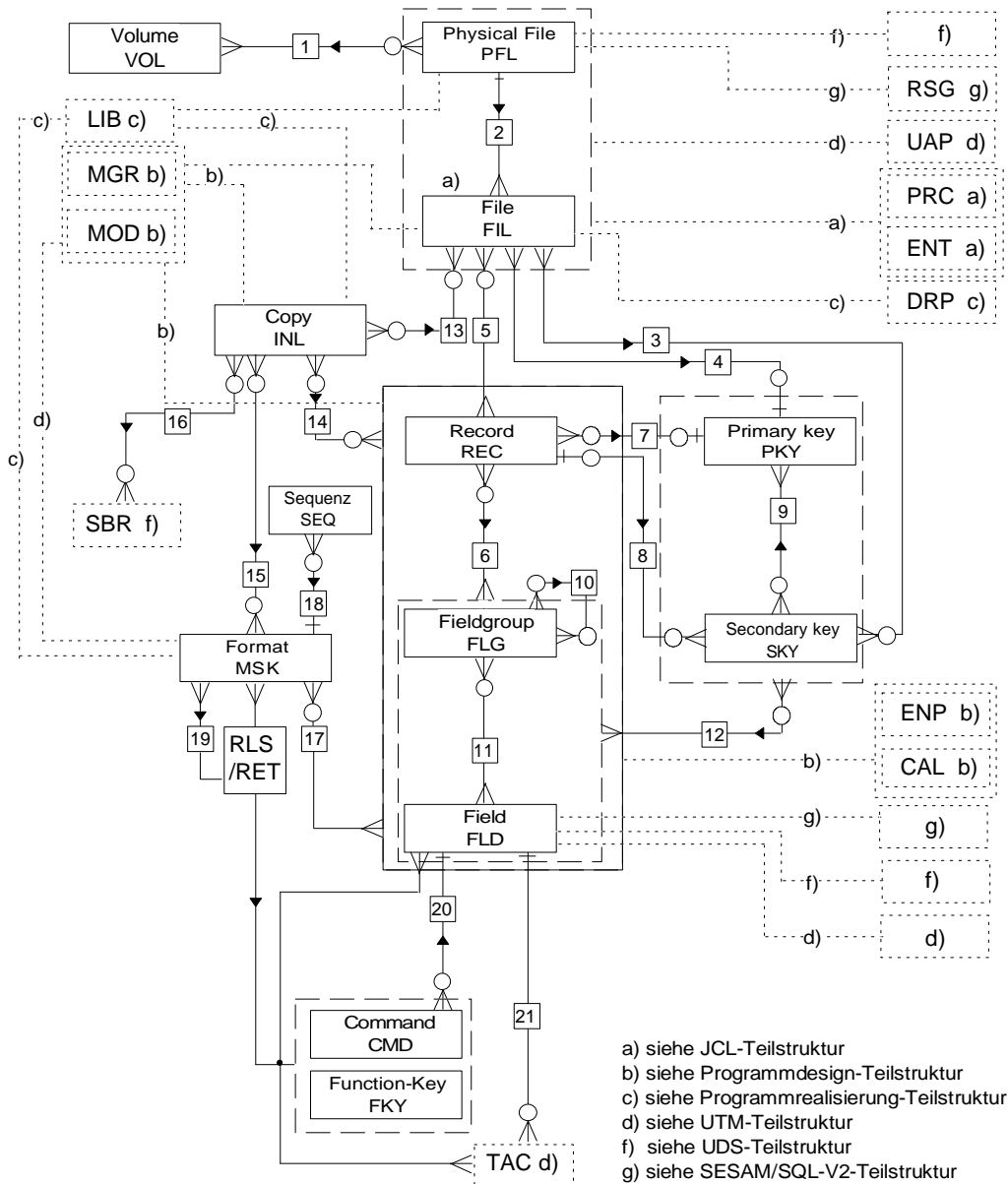


Bild 82 DVS/Leasy-Teilstrukturkatalog

Die Wertigkeit des Beziehungstyps (1:1, 1:n oder n:m) ist in diesem Entity-Relationship-Diagramm zwar enthalten, aber sie wird in der aktuellen Version von TOM-REF nicht unterstützt. Sie können allerdings anhand dieses Teilstrukturkatalogs für korrekte Wertigkeit sorgen.

Elementaraussagen

1	Eine physische Datei (PFL)	belegt	ein oder mehrere	VOL
	Ein Datenträger (VOL)	wird belegt	von keinem, einem oder mehreren	PFL
2	Eine physische Datei (PFL)	bildet ab	eine oder mehrere	FIL
	Eine logische Datei (FIL)	wird abgebildet	in einer	PFL
3	Eine logische Datei (FIL)	enthält	keinen, einen oder mehrere	SKY
	Ein Sekundärschlüssel (SKY)	ist enthalten	in einer oder mehreren	FIL
4	Eine logische Datei (FIL)	enthält	keinen oder einen	PKY
	Ein Primärschlüssel (PKY)	ist enthalten	in einer oder mehreren	FIL
5	Eine logische Datei (FIL)	enthält	eine oder mehrere	REC
	Ein Datensatz (REC)	ist enthalten	in keiner, einer oder mehreren	FIL
6	Ein Datensatz (REC)	enthält	ein oder mehrere	FLG und/oder FLD
	Eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist enthalten	in keinem, einem oder mehreren	REC

7	Ein Datensatz (REC)	enthält	keine oder einen	PKY
	Ein Primärschlüssel (PKY)	ist enthalten	in keinem, einem oder mehreren	REC
8	Ein Datensatz (REC)	enthält	keine, eine oder mehrere	SKY
	Ein Sekundärschlüssel (SKY)	ist enthalten	in keinem oder einem	REC
9	Ein Sekundärschlüssel (SKY)	referenziert	einen oder mehrere	PKY
	Ein Primärschlüssel (PKY)	wird referenziert	von keinem, einem oder mehreren	SKY
10	Eine Feldgruppe (FLG)	enthält	keine, eine oder mehrere	FLG
	Eine Feldgruppe (FLG)	ist enthalten	in keiner, einer oder mehreren	FLG
11	Eine Feldgruppe (FLG)	enthält	ein oder mehrere	FLD
	Ein Feld (FLD)	ist enthalten	in keiner, einer oder mehreren	FLG
12	Ein Primärschlüssel (PKY) und/oder ein Sekundärschlüssel (SKY)	enthält	ein oder mehrere	FLG und/oder FLD
	Eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist enthalten	in keinem, einem oder mehreren	PKY und/oder SKY
13	Eine Copy-Strecke (INL)	beschreibt	keine, eine oder mehrere	FIL
	Eine logische Datei (FIL)	wird beschrieben	von keiner, einer oder mehreren	INL
14	Eine Copy-Strecke (INL)	beschreibt	kein, ein oder mehrere	REC und/oder FLG und/oder FLD
	Ein Record (REC) und/oder eine Feldgruppe (FLG) und/oder ein Feld (FLD)	wird beschrieben	in keinem, einem oder mehreren	INL

15	Eine Copy-Strecke (INL)	beschreibt	kein, ein oder mehrere	MSK
	Ein Format (MSK)	wird beschrieben	in keiner, einer oder mehreren	INL
16	Eine Copy-Strecke (INL)	beschreibt	keine, eine oder mehrere	SBR
	Ein Subschema-Record (SBR)	wird beschrieben	in keiner, einer oder mehreren	INL
17	Ein Format (MSK)	enthält	ein oder mehrere	REC und/ oder FLG und/oder FLD
	Ein Datensatz (REC) und/oder eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist enthalten	in keinem, einem oder mehreren	MSK
18	Eine Format-Sequenz (SEQ)	enthält als erste der Sequenz	ein	MSK
	Ein Format (MSK)	ist enthalten als erste der Sequenz	in keiner, einer oder mehreren	SEQ
19	Ein Format (MSK)	ruft mit dem Inhalt eines oder mehrerer Feldes/r (FLD) und/oder einem oder mehreren UTM-TAC (TAC) und/oder einem oder mehreren Kommando/s (CMD) und/oder einer oder mehreren Funktionstaste/n (FKY)	ein oder mehrere	MSK

19	Ein Format (MSK)	wird gerufen mit dem Inhalt eines oder mehrerer Feldes/r (FLD) und/oder einem oder mehreren UTM-TAC (TAC) und/oder einem oder mehreren Kommando/s (CMD) und/oder einer oder mehreren Funktionstaste/n (FKY)	von einem oder mehreren	MSK
20	Ein Kommando (CMD)	wird eingegeben	in einem	FLD
	Ein Feld (FLD)	dient der Eingabe	von keinem, einem oder mehreren	CMD
21	Ein UTM-TAC (TAC)	wird eingegeben	in einem	FLD
	Ein Feld (FLD)	dient der Eingabe	von keinem, einem oder mehreren	TAC

6.4 Menü DVS/Leasy

Die Objekttypen des Teilstrukturkatalogs DVS/LEASY sind explizit in der Auswahlbildmaske TR0001 aufgeführt. Über die Auswahlmaske TR0001 können auch die Maske für den Objekttyp Alias (`ALI`) und die Maske für die `Other`-Objekttypen (incl. `RLS` und `RET`) aufgerufen werden.

Einen Sonderfall bildet der Objekttyp Copy (`INL`) des DVS/LEASY-Teilstrukturkatalogs. Für ihn existiert ein Sub-Menü zur Auswahl der Art des Copy-Elements. (siehe Seite 205)

```

TR0001 Selection Structure-catalog                $A1234567.LEO.TOMRFL.TOM.BEISPIEL
-----
Please select object-,rel.ship-types/Sub-menus and functions (e.g. al,f2,r,5)

Object-types:
a . Field (FLD)                i . Function-key (FKY)      Sub-menus (S):
b . Fieldgroup (FLG)           j . Command (CMD)          q . SESAM/SQL (S)
c . Record (REC)              k . Volume (VOL)           r . Process/JCL (S)
d . File (FIL)                l . Sequence (SEQ)         s . UTM (S)
e . Primary-key (PKY)          m . Physical-file (PFL)    t . UDS (S)
f . Secondary-key (SKY)        n . Alias (ALI)            w . Copy (S)
g . Format (MSK)               p . Other

Relationship-types:
x . Relationship                y . Relationship extended

Functions for object-types/relationship-types:
1 . Display                    2 . Create                  3 . Change                4 . Delete
Functions without types:
5 . Services                   6 . N2Charts

Command: .....

```

Bild 83 Objekttypen in der Auswahlmaske TR0001

In Auswahlmaske TR0001 können die Masken zu folgenden Objekttypen direkt ausgewählt werden:

- Objekttypen des DVS/LEASY-Teilstrukturkatalogs (Kennbuchstaben `a` bis `m`)
- Objekttyp Alias (`ALI`) (Kennbuchstabe `o`)
- `Other`-Objekttypen (Kennbuchstabe `p`)

In den folgenden Unterabschnitten sind die typspezifischen Objektmasken in der Reihenfolge der Kennbuchstaben beschrieben.

6.4.1 FLD - Feld

Die FLD-Informationen sind auf zwei Masken verteilt: Die Maske TR0010 enthält COBOL-spezifische Attribute, die Folgemaske TR0012 SESAM-SQL V2-spezifische Attribute.

Zuerst gibt TOM-REF die Maske TR0010 aus. In dieser Maske muß, falls noch nicht in der vorangegangenen Maske geschehen, der Objektname festgelegt werden. Objekttyp-neutrale Attribute (z. B. Description) müssen, sofern geplant, ebenfalls bereits in der Maske TR0010 eingetragen werden. Nach dem Abschicken der Maske TR0010 mit der Taste **[DÜ]** wird die Folgemaske TR0012 ausgegeben.

Die Folgemaske TR0012 kann auch übersprungen werden, wenn in der Maske TR0010 das in der Kommandozeile vorgegebene + mit _ überschrieben wird.

Der Objektname und die objekttyp-neutralen Attribute werden in die Folgemaske TR0012 (nicht überschreibbar) übernommen.

TR0010	function	FIELD	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
			Responsible	: ...
			Text	: .
Attributes	:	1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
COBOL-specific-attributes:				
Picture (int)	:	..[1].....	Blank when zero	: [11]
Picture (ext)	:	..[2].....	Justified right	: [12]
Picture (work)	:	..[3].....	Synchronized	: [13]
Usage	:	..[4].....	Is External (Y, N):	[14]
Value	:	..[5].....		
	:		
Occurs	:	..[6] Depending on :	[7].....	
		Indexed by :	[8].....	
Redefines	:	..[9].....		
Level	:	.. (only 77 or 88)	[10]	
Command: +.....				

Bild 84 Maske TR0010 für Objekttyp FLD

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Picture (int)	PICTURE-Anweisung in COBOL	Die PICTURE-Klausel legt die allgemeinen Eigenschaften eines Datenelementes fest und liefert außerdem die Angaben über die Druckaufbereitung. (int, ext, work sind drei vom Benutzer frei wählbare Möglichkeiten, sein Datenelement zu definieren.)
2	Picture (ext)	PICTURE-Anweisung in COBOL	
3	Picture (work)	PICTURE-Anweisung in COBOL	
4	Usage	DISPLAY INDEX BINARY PACKED DECIMAL COMP COMP-1 COMP-2 COMP-3 COMP-5	Die USAGE-Klausel legt fest, in welchem Datenformat ein Datenelement im Internspeicher der Datenverarbeitungsanlage abgespeichert wird.
5	Value	VALUE-Klausel in COBOL oder DB-KEY bei UDS	Die VALUE-Klausel bestimmt den Anfangswert eines Datenfeldes der WORKING-STORAGE SECTION oder den Wert, der einem Bedingungsnamen zugeordnet ist.
6	Occurs	Ganzzahl	Die OCCURS-Klausel wird benutzt, um Tabellen zu definieren. In der Klausel wird angegeben, wieviele Elemente die Tabelle haben soll, d.h. wie oft ein Feld wiederholt werden soll. Alle Elemente haben dasselbe Format. Die Größe der Tabelle ist variabel. Außerdem können Spezialindizes vereinbart werden.
7	Depending on	Datenname	
8	Indexed by	Spezialindex in COBOL	
9	Redefines	Datenname	Die REDEFINES-Klausel erlaubt es dem Programmierer, für einen Speicherbereich mehrere Beschreibungen anzugeben (COBOL).

Nr.	Bezeichnung	Wertebereich	Bedeutung
10	Level	77 <u>88</u>	Stufennummer 77 und 88, die in COBOL ein unabhängiges Feld oder einen Bedingungsnamen beschreiben.
11	Blank when zero	Alpha-Zeichen	Die BLANK-WHEN-ZERO-Klausel bewirkt in COBOL, daß ein Datenfeld immer mit Leerzeichen aufgefüllt ist, wenn sein Inhalt den Wert Null hat
12	Justified right	Alpha-Zeichen	Die JUSTIFIED-RIGHT-Klausel wird in COBOL verwendet, um nichtnumerische Daten innerhalb eines Datenfeldes nach rechts auszurichten.
13	Synchronized	Alpha-Zeichen	Die SYNCHRONIZED-Klausel dient in COBOL zur Ausrichtung eines Datenelements an einer vorgegebenen Grenze des Arbeitsspeichers.
14	Is External	Y(es) N(o) <u>Leerzeichen</u>	Bestimmt, ob ein Datenfeld mehreren Teilprogrammen zugeordnet ist.



Die Angaben 1 bis 14 entsprechen den Felddeklarationen in COBOL. Sie werden nur verwendet, wenn Sie ein Copy-Element generieren möchten. Wenn Sie ein Feld als UDS-Database-Key verwenden, dann muß bei VALUE DB-KEY angegeben werden.

Die OCCURS-Angabe in der Objekt-Definition wird als niederrangig betrachtet, so daß eine OCCURS-Angabe bei der Angabe von Beziehungen die OCCURS-Angabe der Objekt-Definition überschreiben kann.

Folgemaske für Objekttyp FLD

TR0012	function FIELD	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
		Responsible	:
		Text	: .
SQL-specific-attributes:			
Data Type	[1]: ...	(CHA, VAR, NUM, DEC, INT, SMI, FLO, REA, DBL, DAT, TIM, STA)	
Length/Precision	[2]:		
Decimal/Scale	[3]: ..		
Occurence	[4]: ...		
Default Option	[5]:		
Default-Literal	[6]:		
Null-Constraint	[7]:		
SESAM-specific-attributes			
Symbolic Name (SAN)	: ...	[8]	
Command:			

Bild 85 Maske TR0012 für Objekttyp FLD

Attribute in der Folgemaske TR0012

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Data Type	CHA, VAR, INT SMI, DEC, NUM FLO, REA, DBL, DAT, TIM, STA, OLD, VWC	Datentyp des Attributs.
2	Length/Precision	CHA: 1 bis 256 VAR: 1 bis 32000 INT: 4 oder ∞ SMI: 2 oder ∞ NUM: 1 bis 31 DEC: 1 bis 31 FLO: 1 bis 53 Rest: Blank	Länge des Attributs.
3	Decimal/Scale	0 bis 31	Anzahl der Nachkommastellen bei DEC oder NUM; muß kleiner oder gleich dem Wert bei Length/Precision sein
4	Occurence	1 bis 255 Voreinstellung: Leerzeichen	Gibt an, wie oft das Feld vorkommt.

Nr.	Bezeichnung	Wertebereich	Bedeutung
5	Default-Option	LITERAL DATETIME F-TIMESTAMP F-TIME F-DATE CURRENT-USER SYSTEM-USER <u>Leerzeichen</u>	gilt für alle Datentypen ausser DATETIME und kann eine SQL-Anweisung in Hochkomata enthalten für DATETIME-Datentypen } SQL-Funktion, die beim Erzeugen einer Datenbank ausgewertet und bei der ersten Dateneingabe ausgeführt werden. für CHARACTER-Datentypen ab 18 Zeichen für CHARACTER-Datentypen ab 24 Zeichen keine voreingestellte Option
6	Default-Literal	Zeichenkette <u>Leerzeichen</u>	wird ausgewertet, wenn die Default-Option auf LITERAL gesetzt ist; sind mehr als 59 Zeichen angegeben, wird die Zeichenkette als Kurztext gespeichert. Es dürfen max. 256 Zeichen angegeben werden.
7	Null-Constraint	<u>Leerzeichen</u> NOT NULL	Prüfung auf Nullwerte
8	Symbolic Name	Spaltenname <u>Leerzeichen</u>	SQLV1-spezifisch: symbolischer Spaltenname bei CALL-DML-Tabellen



Diese Attribute benötigen Sie, wenn Sie das Feld als Spaltenbeschreibung für eine SESAM-Datenbank nutzen wollen (siehe auch Seite 235).

6.4.2 FLG - Feldgruppe

```

TR0020  function FIELDGROUP                                $userid  .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
                                                Last change  : .....
Description : .....                               Responsible  : ...
                                                Text         : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Usage     : [1].....
Value     : [2].....
          : .....

Occurs    : [3].      Depending on : [4].....
                                                Indexed by  : [5].....

Redefines : [6].....

Synchronized : . [7]                                Is External (Y/N): . [8]

Relationships to succeeding objects          Attributes for relationships
Type      Name                                Type Occurs 1 2 3 4 5 6 7 8
. ....

Command: .....
-----

```

Bild 86 Maske TR0020 für Objekttyp FLG

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Usage	DISPLAY INDEX BINARY PACKED DECIMAL COMP COMP-1 COMP-2 COMP-3 COMP-5	Die USAGE-Klausel legt fest, in welchem Datenformat ein Datenelement im Internspeicher der Datenverarbeitungsanlage abgespeichert wird.
2	Value	VALUE-Klausel in COBOL oder DB-KEY bei UDS	Die VALUE-Klausel bestimmt den Anfangswert eines Datenfeldes der WORKING-STORAGE SECTION oder den Wert, der einem Bedingungsnamen zugeordnet ist.

Nr.	Bezeichnung	Wertebereich	Bedeutung
3	Occurs	Ganzzahl	Die OCCURS-Klausel wird benutzt, um Tabellen zu definieren. In der Klausel wird angegeben, wieviele Elemente die Tabelle haben soll, d.h. wie oft ein Feld wiederholt werden soll. Alle Elemente haben dasselbe Format. Die Größe der Tabelle ist variabel. Ausßerdem können Spezialindizes vereinbart werden.
4	Depending on	Datenname	
5	Indexed by	Spezialindex in COBOL COMP	
6	Redefines	Datenname	Die REDEFINES-Klausel erlaubt es dem Programmierer, für einen Speicherbereich mehrere Beschreibungen anzugeben (COBOL).
7	Synchronized	Alpha-Zeichen	Die SYNCHRONIZED-Klausel dient in COBOL zur Ausrichtung eines Datenelements an einer vorgegebenen Grenze des Arbeitsspeichers.
8	Is External	Y(es) N(o) <u>Leerzeichen</u>	Bestimmt, ob ein Datenfeld mehreren Teilprogrammen zugeordnet ist.

Beziehungen

Folgende Beziehungen können von einem FLG-Objekt ausgehen:

- zu einem oder zu mehreren Feldern (FLD)
- zu einer oder zu mehreren Feldgruppen (FLG)

FILLER

In den Beziehungen können auch Beziehungen zu `FILLER` angegeben werden. Dabei ist es nicht erforderlich, diese Felder zu definieren. Es können folgende Angaben gemacht werden:

Objekttyp: `FLD`
Objektname: `FILLER(99)`

"99" gibt in COBOL-Notation die Anzahl der Zeichen des `FILLERS` an.



Die COBOL-OCCURS-Angabe kann entweder beim Objekt selbst (für Felder und Feldgruppen) oder auch bei der Angabe der Beziehung erfolgen. Die `OCCURS`-Angabe in der Objekt-Definition wird als niederrangig betrachtet, sodaß eine `OCCURS`-Angabe bei der Angabe von Beziehungen die `OCCURS`-Angabe der Objekt-Definition überschreiben kann.

6.4.3 REC - Datensatz

TR0030	function RECORD	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	:	Responsible	: ...
		Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Usage	: .[1].....	Is external (Y, N):	[2]
Value	: .[3].....		
	:		
Redefines	: .[4].....		
Copy-name	: .[5].....		
Prefix	: .[6]....	Suffix	: .[7]....
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
:			
:			
Command:			

Bild 87 Maske TR0030 für Objekttyp REC

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Usage	DISPLAY INDEX BINARY PACKED DECIMAL COMP COMP-1 COMP-2 COMP-3 COMP-5	Die USAGE-Klausel legt fest, in welchem Datenformat ein Datenelement im Internspeicher der Datenverarbeitungsanlage abgespeichert wird.
2	Is External	Y(es) N(o) <u>Leerzeichen</u>	Bestimmt, ob ein Datenfeld mehreren Teilprogrammen zugeordnet ist.
3	Value	VALUE-Klausel in COBOL oder DB-KEY bei UDS	Die VALUE-Klausel bestimmt den Anfangswert eines Datenfeldes der WORKING-STORAGE SECTION oder den Wert, der einem Bedingungsnamen zugeordnet ist.

Nr.	Bezeichnung	Wertebereich	Bedeutung
4	Redefines	Datenname	Die REDEFINES-Klausel erlaubt es dem Programmierer, für einen Speicherbereich mehrere Beschreibungen anzugeben (COBOL).
5	Copy-name	alpha-zeichen	Name des Copy-Elements
6	Prefix	alphanumerische Zeichenkette mit Alphazeichen beginnend	PREFIX, das vor die Datennamen in der COBOL-Datenerklärung gehängt wird.
7	Suffix	alphanumerische Zeichenkette	SUFFIX, das an den Datennamen in der COBOL-Datenerklärung gehängt wird.

Geben Sie einen COPY-Namen an, dann erzeugt TOM-REF automatisch ein Objekt mit dem Typ `INL` und setzt es zu diesem Objekt in eine direkte Beziehung. Wenn das COPY-Objekt mit dem Typ `INL` bereits existiert, dann wird seine Beziehungsliste aktualisiert.

Beziehungen

Folgende Beziehungen können von einem `REC`-Objekt ausgehen:

- zu einem oder zu mehreren Feldern (`FLD`)
- zu einer oder zu mehreren Feldgruppen (`FLG`)
- zu einem Primärschlüssel (`PKY`)
- zu einem oder mehreren Sekundärschlüsseln (`SKY`)

6.4.4 FIL - Logische Datei

Wegen der vielen Attribute sind die FILE-Informationen auf zwei Masken, TR0040 und TR0042, verteilt. Die Folgemaske TR0041 zur Bearbeitung der Beziehungsliste wird im Anschluß an Maske TR0042 ausgegeben.

Die Angaben, die in diesen Masken gemacht werden, wertet der Generator für Copy-Elemente für die FILE-Description aus.

```

TR0040  function FILE                                     $userid  .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
Description : .....                               Last change  : .....
          : .....                               Responsible  : ...
          : .....                               Text         : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Access-mode : . (Sequential, Random, Dynamic) [1]

Record-key   : .....[2].....
Alt.-rec-key : .....[3].....

File-status-1 : .....[4].....
File-status-2 : .....[5].....
Assign to     : .....[6].....
Organization  : . (Sequential, Realtime, Indexed) [7]
Reserve areas : . [8]                               Is external (Y,N): . [9]
Select optional : . (Y,N) [10]

-----
Command: +.....
-----
    
```

Bild 88 Maske TR0040 für Objekttyp FIL

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Access-mode	S(EQUENTIAL) R(ANDOM) D(YNAMIC) <u>Leerzeichen</u>	Die ACCESS-MODE-Klausel bestimmt die Art des Zugriffes auf die Sätze einer Datei.
2	Record-key	datensname	Die RECORD-KEY-Klausel definiert den primären Satzschlüssel einer indizierten Datei.
3	Alt.-rec-key	datensname <u>NULL</u>	Die ALTERNATE-RECORD-KEY-Klausel definiert einen zum Primärschlüssel alternativen Satzschlüssel.

Nr.	Bezeichnung	Wertebereich	Bedeutung
4	File-status-1	datenname	Die FILE-STATUS-Klausel gibt ein Datenfeld an, in dem während der Verarbeitung der Zustand der Datei abgelegt ist.
5	File-status-2	datenname	Die FILE-STATUS-Klausel gibt ein Datenfeld an, in dem während der Verarbeitung der Zustand der Datei abgelegt ist.
6	Assign-to	Link-dateiname	Die ASSIGN-Klausel weist einer Datei des COBOL-Programms über eine Linkverbindung ein externes Gerät zu.
7	Organization	S(EQUENTIAL) R(ELATIVE) I(INDEXED)	Die ORGANIZATION-Klausel definiert den logischen Aufbau einer Datei.
8	Reserve areas	Ganzzahl; 0: NO(ALTERNATE) n: n (ALTERNATE)	Mit der RESERVE-Klausel bestimmt der Benutzer die Anzahl der Ein-/Ausgabebereiche
9	Is external	Y(es) N(o) <u>Leerzeichen</u>	Mit der IS EXTERNAL-Klausel bestimmt der Benutzer, ob die Datei mehreren Teilprogrammen zugeordnet ist.
10	Select optional	Y(es) N(o)	Der OPTIONAL-Zusatz bei der SELECT-Klausel besagt, daß die zugehörige Datei beim Programmablauf nicht vorhanden sein muß.

Folgemaske für den Objekttyp FIL

```

TR0042  function FILE                               $userid  .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
Description : .....                               Last change  : .....
          : .....                               Responsible  : .....
          : .....                               Text        : .....

Data record is : .[11].....
Record type [12]: . (1,2,3) - contains .... characters [13]
                  - size varying from: .[14] to ..... characters
                  depending on : .....
Block contains  : .[15] to ..... (Records, Characters)
Code-set       : .[16].....
Label-record   : . (Omitted, Standard, Non-standard) [17]
Linage        : .[18].....
  footer      : .[19].....
  top         : .[20].....
  bottom     : .[21].....
Value of id is : .[22].....
-----
Command: +.....
    
```

Bild 89 Maske TR0042 für den Objekttyp FIL

Die Angaben für Name und Description aus Maske TR0040 werden in die Maske TR0042 nicht überschreibbar übernommen.

Nr.	Bezeichnung	Wertebereich	Bedeutung
11	Data record is	datensatzname	Name des Datensatzes, in den geschrieben wird.
12	Record type	1 2 3	TOM-REF-interne Klausel (1 = normales Format, 2 = "varying" Format, 3 = Format mit "to" der RECORD-SIZE-Klausel).
13	- Contains - - - - - characters	satzlänge	Die RECORD-CONTAIN-Klausel bezeichnet die Länge eines logischen Datensatzes einer Datei.
14	size varying from - - - - - to charcaters - - - - - depending on	satzlänge <u>NULL</u> satzlänge <u>NULL</u> datensatzname <u>NULL</u>	Die RECORD SIZE-Klausel legt die Länge eines bestimmten Datensatzes fest.

Nr.	Bezeichnung	Wertebereich	Bedeutung
15	Block contains ----- to ----- (Records,...)	n n R(ECORDS) C(HARACT)	Die BLOCK-CONTAINS-Klausel gibt die maximale Größe eines physikalischen Blockes an.
16	Code-set	alphazeichen	Die CODE-SET-Klausel definiert die Zeichencodereinigbarungen, in denen Daten auf externen Geräten dargestellt werden.
17	Label-record	O(MITTED) S(TANDARD) N(ON-STANDARD)	Die LABEL-RECORD-Klausel gibt an, ob Kennsätze vorhanden sind und falls ja, bezeichnet sie diese.
18	Linage	datenname oder Ganzzahl	Die LINAGE-Klausel dient dazu, für eine Ausgabedatei die Länge einer logischen Seite in Form der Zeilenanzahl zu bestimmen. Außerdem kann der Abstand vom oberen bzw. unteren Rand der logischen Seite festgelegt sowie die Zeile innerhalb des Seitenrumpfes angegeben werden, in der der Fußteil beginnt.
19	footer		
20	top		
21	bottom		
22	Value of id is	Literal, datenname	Die VALUE-OF-Klausel dient zur Dokumentation von Benutzerkennsätzen. Im BS2000 wird sie nicht ausgewertet.

Beziehungen

Folgende Beziehungen können von einem `FIL`-Objekt ausgehen:

- zu einem oder zu mehreren Datensätzen (`REC`)
- zu einer oder zu mehreren logischen Dateien (`FIL`)
- zu einem Primärschlüssel (`PKY`)
- zu einem oder zu mehreren Sekundärschlüsseln (`SKY`)



Neben der logischen Datei mit dem Objekttyp `FIL` kennt TOM-REF auch die physikalische Datei mit dem Objekttyp `PFL`. Der Unterschied zeigt sich beim `FILE`-Kommando, das z.B. folgende Form hat:

```
/FILE dateiname, LINK=eingabe
```

dateiname hat in TOM-REF den Objekttyp `PFL` (Physikalische Datei)

eingabe hat in TOM-REF den Objekttyp `FIL` (logische Datei)

6.4.6 SKY - Sekundärschlüssel

TR0080	function SECONDARY-KEY	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	:	Responsible	: ...
	:	Text	: .
Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.			
Relationships to succeeding objects			Attributes for relationships
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
Command:			

Bild 91 Maske TR0080 für Objekttyp SKY

Beziehungen

Folgende Beziehungen können von einem SKY-Objekt ausgehen:

- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einem oder zu mehreren Primärschlüsseln (PKY)

6.4.7 MSK - Format

TR0050	function	FORMAT	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
		Responsible	: ...
		Text	: .
Attributes	:	1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects		Attributes for relationships		
Type	Name	Type	Occurs	1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
.
.
Command:				

Bild 92 Maske TR0050 für Objekttyp MSK

Beziehungen

Folgende Beziehungen können von einem MSK-Objekt ausgehen:

- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einem oder zu mehreren Records (REC)
- zu einer oder mehreren Bibliotheken (LIB)
- zu einer oder mehreren Relationships (RLS)
- zu einem oder mehreren Return-Infos (RET)

6.4.8 FKY - Funktionstasten

TR0100	function	FUNCTION-KEY	\$userid	.nkz.TOMrfn.TOM.project
Name	:	...	Creation date:
Description	:	Last change	:
		Responsible	: ...
		Text	: .
Attributes	:	1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects			Attributes for relationships	
Type	Name		Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
.
.
.
Command:				

Bild 93 Maske TR0100 für Objekttyp FKY

Der Wertebereich für den Objektnamen besteht aus folgenden Elementen:
 {K1, K2, ... K14, F1, ... F5, DUE}

Zum Objekttyp **FKY** gibt es keine spezifischen Attribute.

Beziehungen

Standardmäßig können keine Beziehungen von einem **FKY**-Objekt ausgehen.

6.4.9 CMD - Kommando

TR0090	function	COMMAND	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
		Responsible	: ...
		Text	: .
Attributes	:	1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects			Attributes for relationships	
Type	Name		Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
Command:				

Bild 94 Maske TR0090 für Objekttyp CMD

Beziehungen

Folgende Beziehungen können von einem CMD-Objekt ausgehen:

- zu einem Feld (FLD)

6.4.10 VOL - Datenträger

TR0120	function	VOLUME	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
		Responsible	: ...
		Text	: .
Attributes	:	1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Device	:		
Relationships to succeeding objects		Attributes for relationships		
Type	Name	Type	Occurs	1 2 3 4 5 6 7 8
.
.
.
.
.
.
Command:				

Bild 95 Maske TR0120 für Objekttyp VOL

Im spezifischen Attribut `Device` kann der Gerätetyp abgelegt werden.

Standardmäßig können keine Beziehungen von einem `VOL`-Objekt ausgehen.

6.4.12 PFL - Physikalische Datei

```

TR6010 function PHYSICAL-FILE                               $userid .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
                                                Last change  : .....
Description : .....                               Responsible  : ...
                                                Text        : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Standard-attributes:                                SQL-specific-attributes:
Primary Allocation actual : ....[1]..           Proof-State   : .....[6]...
Primary Allocation changed : ....[2]..           Proof-Time    : .....[7].....
Secondary Allocation actual : ....[3]..           Percent Free  : .. [8]
Secondary Allocation changed : ....[4]..           Destroy (Y/N) : . [9]
Share : . [5]                               Logging (Y/N) : . [10]

Relationships to succeeding objects                Attributes for relationships
Type      Name                                     Type Occurs 1 2 3 4 5 6 7 8
: .....
: .....

Command: .....
    
```

Bild 97 Maske TR6010 für Objekttyp PFL

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Primary Allocation actual	ganzzahl	Primärzuweisung
2	Primary Allocation changed	ganzzahl	Eintrag nur bei Änderung der bestehenden Primärzuweisung
3	Secondary Allocation actual	ganzzahl	Sekundärzuweisung
4	Secondary Allocation changed	ganzzahl	Eintrag nur bei Änderung der bestehenden Sekundärzuweisung

Nr.	Bezeichnung	Wertebereich	Bedeutung
5	Share	Y: Ja N: <u>Nein</u>	Mehrbenutzbarkeit
6	Proof-State	UNPROOFED NOT CREATED DIFFERENCES SYNCHRONOUS <u>Leerzeichen</u>	Dieses Feld wird von Sesam-DDA ausgefüllt, wenn Sie bei den Generierungsfunktionen einen Vergleich zwischen bestehender und abgelegter Datenbank auswählen.
6	Proof-Time	datum zeit Voreinstellung: Leerzeichen	Dieses Feld wird von Sesam-DDA ausgefüllt, wenn Sie bei den Generierungsfunktionen einen Vergleich zwischen Datenbank und Data Dictionary auswählen. Zeitstempel im Format: tt.mm.jj hh.mm.ss
8	Percent Free	0 bis 70 Voreinstellung: NIL-Zeichen	SQLV2-spezifisch: Mindestangabe des freien Plattenpeichers
9	Destroy	<u>Y: Ja</u> N: Nein	Nicht mehr benötigte Daten werden mit X'00' binär überschrieben.
10	Logging	Y: Ja N: <u>Nein</u>	Protokollieren

Beziehungen

Folgende Beziehungen können von einem PFL-Objekt ausgehen:

- zu einem oder zu mehreren Datenträgern (VOL), gilt nicht für SQL/V2.
- zu einer oder zu mehreren logischen Dateien (FIL), gilt nicht für SQL/V2.
- zu einer oder zu mehreren Bibliotheken (LIB), gilt nicht für SQL/V2.
- zu einer SQL/V2-Speicherguppe (RSG).

6.4.13 ALI - Alias

Mit dem Objekttyp `Alias` können Sie synonyme Objekte unter einem Oberbegriff zusammenfassen (siehe Seite 39 "Synonymbehandlung").

TR5010	function ALIAS	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change :
		Responsible :	...
		Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type Occurs	1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
Command:		

Bild 98 Maske TR5010 für Objekttyp ALI

Von `ALI`-Objekten sind Beziehungen zu Objekten beliebiger Typen möglich.

6.4.14 RET, RLS, Other - Benutzereigene Objekttypen

TOM-REF stellt für benutzereigene Objekttypen und solche, mit denen spezielle Modellierungen möglich sind (RET, RLS) ein Standard-Maskenlayout zur Verfügung. Damit können Sie auch Objekttypen, die nicht im Data Dictionary vorhanden sind, in Ihre Struktur einbeziehen. Sie können also den Standard-Strukturkatalog um eigene Objekttypen erweitern. Auf Seite 258 ist ein Vorschlag für eine Teilstruktur zum Datenentwurf enthalten. Diese Teilstruktur können Sie ebenfalls in Ihr Data Dictionary einbringen.

Benutzereigene Typen und ihre erlaubten Beziehungen müssen TOM-REF zuvor bekannt gegeben werden. Die Prozedur, mit der das geschieht, ist auf Seite 55 beschrieben. Benutzereigene Typen können nicht dieselbe Typ-Bezeichnung tragen wie die standardmäßigen Objekttypen.

Die Typbezeichnung muß aus zwei oder drei Zeichen bestehen. Dabei muß das erste Zeichen alphabetisch, das zweite und dritte Zeichen alphanumerisch sein. Sonderzeichen sind nicht zugelassen.

Im Gegensatz zu den anderen Objektmasken muß in diese Maske der Objekttyp explizit vor den Namen des Objekts eingetragen werden (das Feld ist grün unterlegt).

TR5000	function	\$userid	.nkz.TOMrfn.project
Object :		Creation date:	
Description :		Last change :	
.....		Responsible : ...	
.....		Text : .	
Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.			
9:		10:	
11:		12:	
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.....
.....
.....
.....
.....
.....
Command:			

Bild 99 Maske TR5000 für Other-Objekttypen

Neben den einstelligen Attributen (1 bis 8) stehen zur Beschreibung eines `other`-Objekts noch mehrstellige Attribute zur Verfügung (9 bis 12).

Für die Modellierung bestimmter Strukturen benötigen Sie eventuell die Objekttypen `RET` und `RLS`. Ihre Funktion ist auf Seite 55 näher beschrieben. Bearbeitet werden die Objekttypen mit dieser Maske, analog zu den benutzereigenen Objekttypen.

6.5 INL - Standard-Objektyp für Copy-Elemente

Mit der Servicefunktion `Generate functions` (siehe Seite 143) können Sie Copy-Elemente folgender Arten generieren:

- Data-area
(Datenbereich)
- MOVE-Statements
(zwischen Feldern, d.h. den kleinsten Einheiten von Datenbereichen)
- INIT-Statements
(Initialisierung von Datenfeldern)
- File-Description
(FILE-CONTROL-Paragraph oder FILE SECTION)

Voraussetzung für die Generierung eines Copy-Elements ist die Beschreibung desselben im Data Dictionary. Für jede der oben genannten Arten gibt es eine spezifische Copy-Objektmaske zur Beschreibung des Copy-Elements. Im folgenden Abschnitt erfahren Sie, wie ein Copy-Objektbildschirm einer bestimmten Art aufgerufen wird. Ab Seite 207 sind die einzelnen Copy-Objektmasken beschrieben.

6.5.1 Copy-Objektmasken aufrufen

Eine Copy-Objektmaske können Sie aus der Auswahlmaske TR0001 entweder indirekt über die Auswahlmaske TR0060 für Copy-Elemente oder direkt ansteuern.

- Copy-Maske indirekt ansteuern:
In der Auswahlmaske TR0001 ist der Kennbuchstabe `w` (für Copy) und die Kennziffer für die gewünschte Grundfunktion anzukreuzen. Nach dem Abschicken der Maske mit der Taste `[DÜ]` wird die Auswahlmaske TR0060 für Copy-Elemente ausgegeben.

```
TR0060 COPY-selection                               $userid .nkz.TOMrfn.TOM.BEISPIEL
-----
Please select COPY-type:
. Data-area
. MOVE-statements
. INIT-statements
. File-description

Command: .....
```

Bild 100 Auswahlmaske TR0060 für Copy-Elemente

In der Maske TR0060 können Sie die Art des Copy-Elements ankreuzen. Nach dem Abschicken der Maske mit der Taste (**DÜ**) wird die markierte Copy-Objektmaske ausgegeben.

Mehrfaches Ankreuzen in der Maske TR0060 ist ebenfalls möglich. Die einzelnen Copy-Objektmasken werden dann in der Reihenfolge der Markierung ausgegeben.

- Copy-Objektmaske *direkt* ansteuern:

Mit einer gezielten Eingabe in der Auswahlmaske TR0001 beim Kennbuchstaben *w* können Sie eine Copy-Objektmaske direkt ansteuern. Der folgenden Auflistung ist zu entnehmen, mit welchem Zeichen eine bestimmte Copy-Objektmaske identifiziert wird:

D	Data-Area
M	MOVE-Statement
I	INIT-Statements
F	FILE-Description

Die Kennziffer der gewünschten Grundfunktion ist ebenfalls anzukreuzen. Nach dem Abschicken der Maske mit der Taste (**DÜ**) wird die gewählte Copy-Objektmaske ausgegeben.

6.5.2 Copy-Element Data-Area

```

TR0062  Display COPY                                     $userid  .nkz.TOMrfn.TOM.BEISPIEL
-----
Name      : .....                               Creation date: .....
                                                Last change  : .....
Description : .....                               Responsible  : ...
                                                Text         : .

Attributes : 1:.  2:.  3:.  4:.  5:.  6:.  7:.  8:.

Generate Data-area

Prefix      : ..[1]...           Suffix : ..[2]...
Start-level : ..[3]
Level-index : ..[4]
Picture-use-priority : intern: .  extern: .  work: .  (1,2,3) [5]
Values is to be used : . (Y,N) [6]

Relationships to succeeding objects           Attributes for relationships
Type      Name                               Type Occurs 1 2 3 4 5 6 7 8
. ....

Command: .....

```

Bild 101 Maske TR0062 für Copy-Element Data-area

Die Struktur des Datenbereichs bildet TOM-REF anhand des Netzes der Beziehungen, ausgehend vom Copy-Element. Bevor Sie die Generierung des Datenbereichs anstoßen, muß dieser, bis hinunter zu den Datenelementen, vollständig im Data Dictionary beschrieben sein.

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Prefix	alphanumerische Zeichenkette mit Alphazeichen beginnend	PREFIX, das vor die Datennamen in der COBOL-Datenerklärung gehängt wird.
2	Suffix	alphanumerische Zeichenkette	SUFFIX, das an den Datennamen in der COBOL-Datenerklärung gehängt wird.
3	Start-level	COBOL-Stufennummer auf der Basis REC oder FLG	Höchste COBOL-Stufennummer des zu generierenden Copy-Elements.

Nr.	Bezeichnung	Wertebereich	Bedeutung
4	Level-index	numerische Zeichen	Schrittweite, mit der alle weiteren Stufennummern nach dem START-LEVEL gebildet werden.
5	Picture-use-priority	1 2 3	Reihenfolge, in der die PICTURE-Klauseln aus der Feld-Erklärung genommen werden (Siehe int, ext, work bei Field).
6	Values is to be used	Y N	Bei einer Belegung mit "Y" wird das Copy-Element für die WORKING-STORAGE SECTION generiert, bei einer Belegung mit "N" wird es für die LINKAGE-SECTION generiert.

Erläuterung

PICTURE-Priority gibt an, in welcher Reihenfolge die Picture-Angaben ausgewertet werden sollen, die für jedes Feld angegeben wurden. Die Zahlen von 1 bis 3 regeln die Prioritäts-Reihenfolge. Das heißt z.B., wenn Sie eine 1 bei `intern` angegeben haben, 2 bei `work` und 3 bei `extern`, dann wird zuerst die interne Angabe benutzt. Wenn die interne Angabe leer ist, dann wird die Work-Angabe benutzt und als letztes die externe.

Mit `VALUE` wird gesteuert, ob die COPY-Generierung des Datenelements mit `VALUE`-Angabe oder ohne `VALUE`-Angabe erfolgen soll. Bei Copy-Elementen, die in der Linkage-Section verwendet werden sollen, darf kein `VALUE` angegeben werden.

Beziehungen

Zu folgenden Objekttypen sind Beziehungen möglich:

- zu einer oder zu mehreren Feldgruppen (`FLG`)
- zu einem oder zu mehreren Feldern (`FLD`)
- zu einem oder zu mehreren Records (`REC`)
- zu einer oder zu mehreren logischen Dateien (`FIL`)
- zu einem oder zu mehreren Subschema-Records (`SBR`)
- zu einem oder zu mehreren Formaten (`MSK`)

6.5.3 Copy-Element MOVE-Statements

TR0063	Display COPY	\$userid	.nkz.TOMrfn.TOM.BEISPIEL
Name	:	Creation date:
Description	:	Last change	:
	Responsible	: ...
		Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Generate MOVE-Statements			
Move from			
Object	:		
Prefix	:	Suffix	:
Picture use-priority	: Intern: 1 Extern: 3 Work: 2		(1, 2, 3)
Move to			
Object	:		
Prefix	:	Suffix	:
Picture use-priority	: Intern: 1 Extern: 3 Work: 2		(1, 2, 3)
Command:		

Bild 102 Maske TR0063 für Copy-Element MOVE-Statements

MOVE-Statements können Sie zwischen den kleinsten Einheiten von Datenbereichen, also den Feldern, von TOM-REF generieren lassen. Vor der Generierung müssen Sie folgendes tun:

1. Zwei Datenbereiche, der Sendebereich und der Empfangsbereich, sind im Data Dictionary zu beschreiben. Das heißt, die notwendigen Objekte und Beziehungen sind, bis hinunter zu den FLD-Objekten, zu erstellen.

Beide Datenbereiche müssen jeweils mit einem bestimmten Objekt als Ausgangspunkt erstellt werden. Dieses Objekt ist in der Maske TR0063 für den Sendebereich mit *Move from* bezeichnet, für den Empfangsbereich mit *Move to*.

Wichtig für die spätere Generierung von MOVE-Statements ist, daß Felder, zwischen denen ein MOVE-Statement aufgebaut werden soll, denselben Objektnamen besitzen müssen.

2. Das Copy-Objekt der Art *MOVE-Statements* ist in der Maske TR0063 zu erstellen. Für die bereits erwähnten Objekte *Move from* und *Move to* müssen jeweils Typ und Name angegeben werden.

Prefix bzw. *Suffix* können angegeben werden. In diesem Fall werden die Angaben den entsprechenden Feldnamen in den MOVE-Statements vor- bzw. nachgestellt.

Beispiel

Im Data Dictionary seien folgende Strukturen vorhanden:

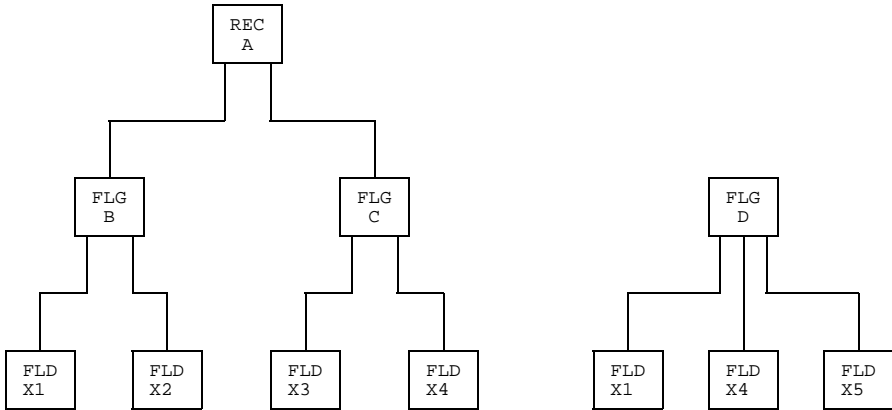


Bild 103 Beispielstrukturen für Copy-Element MOVE-Statements

Die Struktur links soll den Sendebereich darstellen, das Move from-Objekt ist REC A. Die Struktur rechts sei der Empfangsbereich mit dem Move to- Objekt FLG D. Es gibt in den Strukturen zwei FLD-Objekte mit gleichen Namen, FLD X1 und FLD X4. Somit würden zwei Move-Statements generiert. Mit der zusätzlichen Voraussetzung, daß beim Move from-Objekt das Feld Prefix mit dem Wert EIN versorgt wurde, sähen die Move-Statements folgendermaßen aus:

```

MOVE EIN-X1 IN EIN-B IN EIN-A TO X1 IN D
MOVE EIN-X4 IN EIN-C IN EIN-A TO X4 IN D
  
```

6.5.4 Copy-Element INIT-Statements

```

TR0064  Display COPY                                $userid  .nkz.TOMrfn.TOM.BEISPIEL
Name      : .....                                Creation date: .....
Description : .....                                Last change  : .....
                                                    Responsible  : ...
                                                    Text        : .
Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Prefix      : .....      Suffix : .....
Picture-use-priority : intern : .  extern : .  work : .  (1,2,3)
Values is to be used : . (Y,N)

Relationships to succeeding objects                Attributes for relationships
Type      Name                                     Type Occurs 1 2 3 4 5 6 7 8
.....
.....
.....
.....
.....
.....
.....
.....
.....

Command: .....
    
```

Bild 104 Maske TR0064 für Copy-Element INIT-Statements

Mit dieser Maske legen Sie die Initialisierung für Felder fest. Die Felder müssen vom Copy-Objekt ausgehend über das Netz der Beziehungen erreichbar sein.

Die Bedeutung von `Prefix`, `Suffix` und `Picture-use-priority` wurde bereits auf Seite 207 erläutert.

Der Wert (Y oder N), der im Feld `Values is to be used` einzugeben ist, hat folgende Wirkung:

- Y Das Feld wird mit dem beim Objekt angegebenen Wert initialisiert.
- N Character-Felder werden mit Leerzeichen initialisiert, numerische Felder mit 0.

Beziehungen

Zu folgenden Objekttypen sind Beziehungen möglich:

- zu einem oder zu mehreren Records (*REC*)
- zu einem oder zu mehreren Subschema-Records (*SBR*)
- zu einem oder zu mehreren Formaten (*MSK*)
- zu einer oder zu mehreren Feldgruppen (*FLG*)
- zu einem oder zu mehreren Feldern (*FLD*)
- zu einer oder zu mehreren logischen Dateien (*FIL*)

Beispiel

```
MOVE SPACES TO VORNAME IN NAME IN ADRESSE
MOVE SPACES TO NACHNAME IN NAME IN ADRESSE
MOVE ZERO TO PLZ IN ADRESSE
MOVE SPACES TO ORT IN ADRESSE
MOVE SPACES TO VORNAME-DRUCK IN NAME-DRUCK IN
  ADRESSE-DRUCK
MOVE SPACES TO NACHNAME-DRUCK IN NAME-DRUCK IN
  ADRESSE-DRUCK
MOVE ZERO TO PLZ-DRUCK IN ADRESSE-DRUCK
MOVE SPACES TO ORT-DRUCK IN ADRESSE-DRUCK
```

6.5.5 Copy-Element FILE-Description

TR0065	Display COPY	\$userid	.nkz.TOMrfn.TOM.BEISPIEL
Name	:	Creation date:
Description	:	Last change	:
	Responsible	: ...
		Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Copy section	: . (C, D)		
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type Occurs	1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
Command:			

Bild 105 Maske TR0065 für Copy-Element FILE-Description

Mit diese Maske definieren Sie ein Copy-Element für jedes in der Beziehungsliste angegebene FILE-Objekt.

Der Wert (C oder D), der im Feld `Copy section` einzugeben ist, hat folgende Wirkung:

- C Copy-Element für den FILE-CONTROL-Paragraphen.
- D Copy-Element für die FILE DESCRIPTION der FILE SECTION.

Beziehungen

Zu folgendem Objekttyp sind Beziehungen möglich:

- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einem oder zu mehreren Records (REC)
- zu einer oder zu mehreren logischen Dateien (FILL)
- zu einem oder zu mehreren Formaten (MSK)
- zu einem oder zu mehreren Subschema-Records (SBR)

Beispiel

```
ENVIRONMENT DIVISION.  
INPUT-OUTPUT SECTION.  
FILE-CONTROL.  
*****  
*** FILE-C.INL ***  
*****  
    SELECT BEISPIEL  
    ASSIGN TO DA-DISC-S-SYS011  
    ORGANIZATION IS SEQUENTIAL  
    ACCESS MODE IS SEQUENTIAL  
    RESERVE NO ALTERNATE AREA  
    FILE STATUS IS NACHNAME  
    IN ADRESSE.  
DATA DIVISION.  
FILE SECTION.  
*****  
*** FILE-D.INL ***  
*****  
    FD BEISPIEL  
    BLOCK CONTAINS 366 CHARACTERS  
    RECORDING MODE IS F  
    LABEL RECORD IS STANDARD.
```

7 SESAM/SQL V2

Eine SESAM/SQL V2-Datenbank beschreiben Sie im Data Dictionary mit den Objekten und Beziehungen des SESAM/SQL V2-Teilstrukturkatalogs. Eine SESAM/SQL V2-Datenbank beschreiben heißt, daß Sie die nötigen Objekte und Beziehungen erstellen. In Kapitel 4 ist beschrieben, wie Sie Objekte und Beziehungen erstellen können.

7.1 SESAM/SQL V2-Objekte

Die folgende Tabelle zeigt die Objekttypen der SESAM/SQL V2-Teilstruktur mit ihren Kurznamen, Namen und einer kurzen Erläuterung.

Kurzname	Langname des Objekttyps	Erläuterung
RBT	SQL-BASE-TABLE	Basistabelle/View
RCC	SQL-CHECK-CONSTRAINT	Check-Bedingung
RCT	SQL-CATALOG	Datenbank
RFC	SQL-REF-CONSTR	Referenzbedingung
RIX	SQL-INDEX	Index-Beschreibung
RSG	SQL-STORAGE-GROUP	Speichergruppe
RSM	SQL-SCHEMA	Datenbank-Schema
RUC	SQL-UNIQ-CONSTR	Eindeutigkeitsbedingung
RUS	SQL-USER	Eigentümer
FLD	FIELD	Tabellenspalte
PFL	PHYSICAL-FILE	Physikalische Datei
VOL	VOLUME	Datenträger

Tabelle 11 Objekttypen im SQL-Teilstrukturkatalog

7.2 SESAM/SQL V2-Beziehungen

Zwischen welchen Objekten des SESAM/SQL V2-Teilstrukturkatalogs Sie Beziehungen erstellen können, zeigt die folgende Tabelle. Dabei bedeutet ein X in einem Feld, daß zwischen zwei Objekttypen eine Beziehung bestehen kann; bei einem S handelt es sich um vordefinierte erweiterte Beziehungstypen, deren Namen im Data Dictionary hinterlegt sind.

Objekttyp als Ausgang der Beziehung	Objekttyp als Ziel der Beziehung											
	FLD	PFL	RBT	RCC	RCT	RFC	RIX	RSG	RSM	RUC	RUS	VOL
RBT Base Table	S	X	X	X		X	X			X		
RCC Check-Constr.	X											
RCT Catalog								X	X			
RFC Refer. Constr.	X									X		
RIX Index	S	X								X		
RSG Storage-Group												X
RSM Schema			X									
RUC Unique Constr.	X											
RUS User		X			X			X	X			

Tabelle 12 Beziehungstypen des SQL-Teilstrukturkatalogs

Die vordefinierten erweiterten Beziehungstypen für SESAM/SQL sind `XBT` und `XIX`. Den Beziehungstyp `XBT` erstellen Sie zwischen Objekten vom Typ `RBT` (View) und Objekten vom Typ `FLD`, eine `XIX`-Beziehung erstellen Sie zwischen Objekttypen `RIX` und `FLD`. Wie Sie diese Beziehungen erstellen, ist in Kapitel 4 beschrieben.

7.3 SESAM/SQL V2-Teilstrukturkatalog

Der SESAM/SQL V2-Teilstrukturkatalog ist die Basis für die Beschreibung einer SESAM/SQL V2-Datenbank im Data Dictionary. Er legt fest, welches Objekt das Ausgangsobjekt und welches das Zielobjekt einer Beziehung sein muß. Die Pfeile zeigen vom Typ des Ausgangsobjekts zum Typ des Zielobjekts. Eine Beziehung müssen Sie immer in die Beziehungsliste des Ausgangsobjekts eintragen. Die Elementaraussagen klassifizieren die Beziehungen näher.

Eine genaue Beschreibung der in diesem Modell verwendeten Darstellungsmittel finden Sie ab Seite 7.

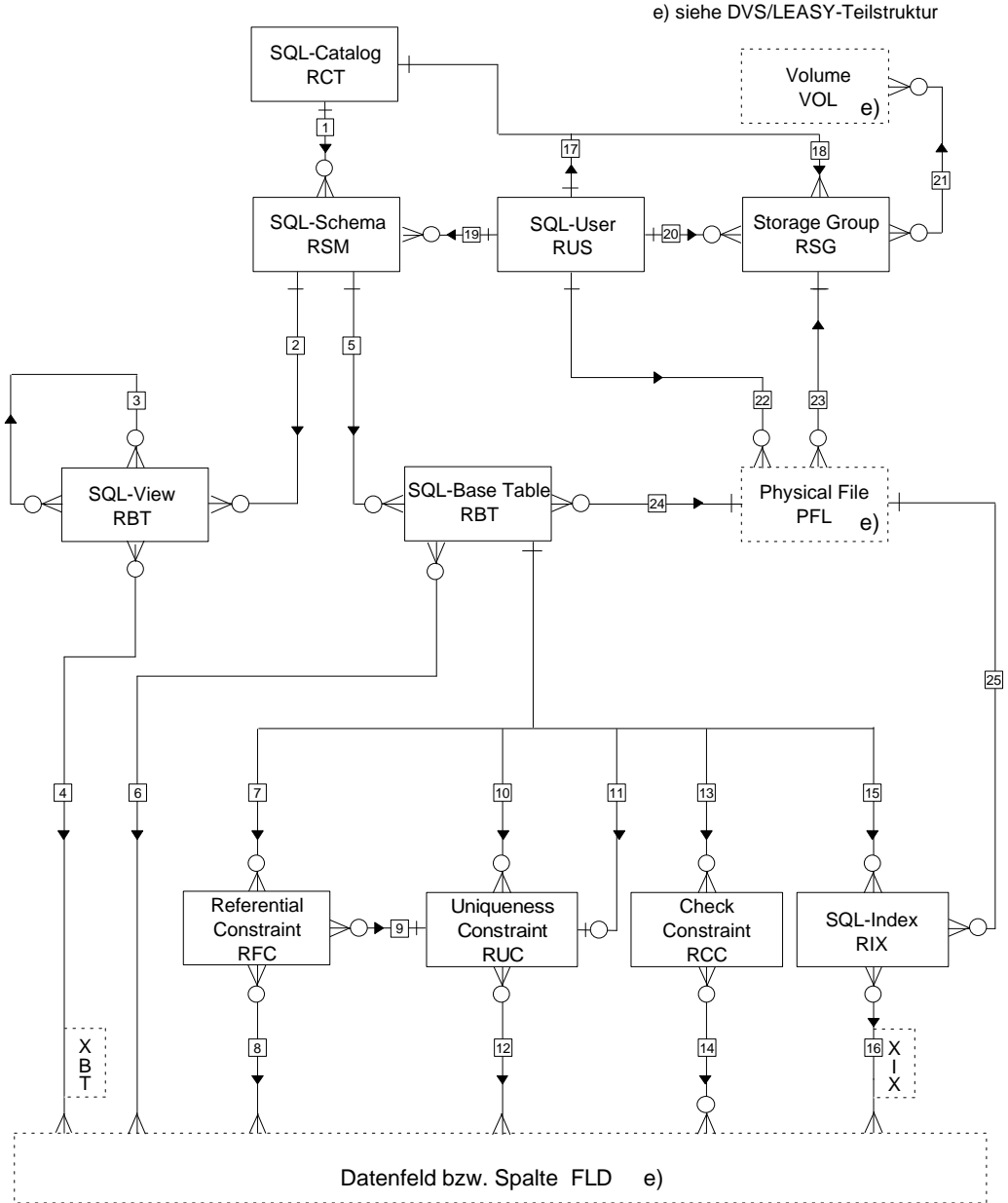


Bild 106 SESAM/SQL V2-Teilstrukturkatalog

Die Wertigkeit des Beziehungstyps (1:1, 1:n oder n:m) ist in diesem Entity-Relationship-Diagramm zwar enthalten, aber sie wird in der aktuellen Version von TOM-REF nicht unterstützt. Sie können allerdings anhand dieses Teilstrukturkatalogs für korrekte Wertigkeit sorgen.

Elementaraussagen

1	Eine SQL-Datenbank (RCT)	besteht aus	keinem, einem oder mehreren	RSM
	Ein Datenbank-Schema (RSM)	gehört zu	einem	RCT
2	Ein Datenbank-Schema (RSM)	besteht aus	keiner, einer oder mehreren	RBT
	Ein SQL-View (RBT)	gehört zu	einer	RSM
3	Ein SQL-View (RBT)	benutzt	keine, eine oder mehrere	RBT
4	Ein SQL-View (RBT)	benutzt	keine, eine oder mehrere	FLD
	Ein Datenfeld bzw. eine Spalte (FLD)	gehört zu	keiner, einer oder mehreren	RBT
5	Ein Datenbank-Schema (RSM)	besteht aus	keiner, einer oder mehreren	RBT
	Eine Basistabelle (RBT)	gehört zu	einer	RSM
6	Eine Basistabelle (RBT)	benutzt	eine oder mehrere	FLD
	Ein Datenfeld bzw. eine Spalte (FLD)	gehört zu	einer oder mehreren	RBT
7	Eine Basistabelle (RBT)	enthält	keinen, einen oder mehrere	RFC
	Eine Referenzbedingung (RFC)	gehört zu	einer	RBT
8	Ein Referenzbedingung (RFC)	besteht aus	einem oder mehreren	FLD
	Ein Datenfeld bzw. Spalte (FLD)	gehört zu	einem, keinem oder mehreren	RFC

9	Eine Referenzbedingung (RFC)	referenziert	einen	RUC
	Eine Eindeutigkeitsbedingung (RUC)	wird referenziert von	keinen, einen oder mehreren	RFC
10	Eine Basistabelle (RBT)	enthält	keine, eine oder mehrere	RUC
	Eine Eindeutigkeitsbedingung (RUC)	gehört zu	einer	RBT
11	Eine SQL-Basistabelle (RBT)	enthält	keine oder eine	RUC
	Ein Primärschlüssel (RUC)	ist definiert	für eine	RBT
12	Eine Eindeutigkeitsbedingung (RUC)	verweist auf	ein oder mehrere	FLD
	Ein Datenfeld bzw. eine Spalte (FLD)	gehört zu	einem	RUC
13	Eine Basistabelle (RBT)	enthält	keinen, einen oder mehrere	RCC
	Eine Check-Bedingung (RCC)	gehört zu	einer	RBT
14	Eine Check-Bedingung (RCC)	bezieht sich	auf keine, eine oder mehrere	FLD
	Ein Datenfeld oder Spalte (FLD)	wird benutzt	von keiner, einer oder mehreren	RCC
15	Eine Basistabelle (RBT)	enthält	keinen, einen oder mehrere	RIX
	Eine Index-Beschreibung (RIX)	ist definiert	für eine	RBT
16	Eine Index-Beschreibung (RIX)	besteht aus	einer oder mehreren	FLD
	Ein Datenfeld oder Spalte (FLD)	gehört zu	keinem, einem oder mehreren	RIX
17	Eine SQL-Datenbank (RCT)	gehört	einem	RUS
	Ein Eigentümer (RUS)	besitzt	eine	RCT

18	Eine SQL-Datenbank (RCT)	besteht aus	einer oder mehreren	RSG
	Eine SQL-Speichergruppe (RSG)	gehört zu	einem	RCT
19	Ein Eigentümer (RUS)	besitzt	kein, ein oder mehrere	RSM
	Ein Datenbank-Schema (RSM)	gehört	einem	RUS
20	Ein Eigentümer (RUS)	besitzt	keine, eine oder mehrere	RSG
	Eine SQL-Speichergruppe (RSG)	gehört	einem	RUS
21	Eine SQL-Speichergruppe (RSG)	verweist auf	keine, eine oder mehrere	VOL
	Ein privater Datenträger (VOL)	wird zugewiesen	keinem, einem oder mehreren	RSG
22	Einem Eigentümer (RUS)	gehören	keine, eine oder mehrere	PFL
	Eine physikalische Datei (PFL)	gehört	einem	RUS
23	Eine physikalische Datei (PFL)	ist zugewiesen	einer	RSG
	Eine SQL-Speichergruppe (RSG)	nimmt auf	keine, eine oder mehrere	PFL
24	Eine Basistabelle (RBT)	ist gespeichert in	einer	PFL
	Eine physikalische Datei (PFL)	nimmt auf	keine, eine oder mehrere	RBT
25	Eine Index-Beschreibung (RIX)	ist gespeichert in	einer	PFL
	Eine physikalische Datei (PFL)	nimmt auf	keine, eine oder mehrere	RIX

7.4 Sub-Menü SESAM/SQL

Die Objekttypen des SESAM/SQL V2-Teilstrukturkatalogs erstellen und bearbeiten Sie in den objekttyp- und beziehungstyp-spezifischen Masken. Diese Masken erreichen Sie über das Sub-Menü SESAM/SQL in der Auswahlmaske TR2000.

Das Sub-Menü SESAM/SQL erhalten Sie, wenn Sie in der Maske TR0001 (siehe Seite 177) den Kennbuchstaben **q** auswählen.

TR2000	Selection SESAM/SQL-Types	\$userid	.nkz.TOMrfn.TOM.project
Please select object-types/relationship-types and functions (e.g. e2,a3,5)			
Object-types:			
a . SQL-Catalog (RCT)		g . SQL-Referential-Constraint (RFC)	
b . SQL-Schema (RSM)		h . SQL-Storage-Group (RSG)	
c . SQL-Base-Table (RBT)		i . SQL-User (RUS)	
d . SQL-Index (RIX)		j . Field (FLD)	
e . SQL-Uniqueness-Constraint (RUC)		k . Physical-File (PFL)	
f . SQL-Check-Constraint (RCC)		l . Volume (VOL)	
Relationship-types:			
o . Base-Table-Info (XBT)		q . Index-Info (XIX)	
p . Base-Table-Info ext.		r . Index-Info ext.	
Functions for object-types:			
1 . Display	2 . Create	3 . Change	4 . Delete
Functions without object-types:			
5 . Services			
Command:			

Bild 107 Auswahlmaske TR2000 für SESAM/SQL-Typen

Die einzelnen Masken für die SESAM/SQL-Objekttypen (Kennbuchstaben **a** bis **i** in der Auswahlmaske TR2000) sind in den folgenden Unterabschnitten beschrieben.

Mit der Kennziffer **5** können Sie aus der Maske TR2000 in die Auswahlmaske TR9000 für die Servicefunktionen wechseln.

7.4.1 RCT - SQL-Datenbank (Catalog)

RCT ist der Name einer logischen SESAM-Datenbank, der den SQL-Namenskonventionen entsprechen muß.

```

TR2110 function SESAM-CATALOG                                $userid .nkz.TOMrfn.TOM.project
-----
Name      : .....
Creation date: .....
Description : .....
Responsible : ...
Text      : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

SESAM-Attributes:
System    : .....[1]
Version   : ..... [2]
Code Table : .....[3].....

Space-Attributes:
Primary Allocation : .....[4]
Secondary Allocation: .....[5]
Sharable (Y/N)    : . [6]
Destroy (Y/N)     : . [7]
Logging (Y/N)     : . [8]
Percent Free (0-70) : .. [9]

Relationships to succeeding objects
Type      Name
.....

Attributes for relationships
Type Occurs 1 2 3 4 5 6 7 8
.....

Command: .....
    
```

Bild 108 Maske TR2110 für Objekttyp RCT

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	System	<u>SESAM/SQL</u>	Auswahl des Datenbanksystems
2	Version	<u>V2.0</u>	Auswahl der Versionsnummer
3	Code Table	zeichensatz- tabelle <u>EBCDIC_DF_03</u>	Name der BS2000-Zeichensatz- tabelle;
4	Primary Allocation	16..8388601 <u>72</u>	Primärzuweisung in PAM-Seiten
5	Secondary Allocation	24..32767 <u>24</u>	Sekundärzuweisung in PAM- Seiten
6	Sharable	Y: Ja N: <u>Nein</u>	Mehrbenutzbarkeit

Nr.	Bezeichnung	Wertebereich	Bedeutung
7	Destroy	<u>Y</u> : Ja N: Nein	Nicht mehr benötigte Daten werden mit X'00' binär überschrieben.
8	Logging	<u>Y</u> : Ja N: Nein	Protokollieren
9	Percent Free	0 bis 70 <u>20</u>	SQLV2-spezifisch: Mindestangabe des freien Plattenpeichers

Beziehungen

Folgende Beziehungen können von einem RCT-Objekt ausgehen:

- zu einem oder mehreren Datenbank-Schemata (RSM)
- zu einer oder mehreren SQL-Speicherguppen (RSG)

7.4.2 RSM - Datenbank-Schema

```

TR2050 function SQL-SCHEMA                                $userid .nkz.TOMrfn.TOM.project
-----
Name           : .....                               Creation date: .....
Description    : .....                               Last change  : .....
                                                       Responsible  : ...
                                                       Text       : .

Attributes     : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Proof-State   : ....[1]....                            Proof-Time   : ....[2].....

Relationships to succeeding objects                      Attributes for relationships
Type          Name                                     Type Occurs 1 2 3 4 5 6 7 8
.             .                                       .             . . . . .
.             .                                       .             . . . . .
.             .                                       .             . . . . .
.             .                                       .             . . . . .
.             .                                       .             . . . . .

Command: .....
    
```

Bild 109 Maske TR2050 für Objekttyp RSM

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Proof-State	UNPROOFED NOT CREATED DIFFERENCES SYNCHRONOUS	Dieses Feld wird von Sesam-DDA ausgefüllt, wenn Sie bei den Generierungsfunktionen einen Vergleich zwischen Datenbank und Data Dictionary auswählen.
2	Proof-Time	datum zeit Voreinstellung: Leerzeichen	Dieses Feld wird von Sesam-DDA ausgefüllt, wenn Sie bei den Generierungsfunktionen einen Vergleich zwischen Datenbank und Data Dictionary auswählen. Zeitstempel im Format: tt.mm.jj hh.mm.ss

Beziehungen

Folgende Beziehung kann von einem RSM-Objekt ausgehen:

- zu einer oder zu mehreren Basistabellen (RBT)

7.4.3 RBT - Basistabelle

```

TR2010 function SQL-BASE-TABLE                $userid .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
Description : .....                               Last change  : .....
          : .....                               Responsible  : ...
          : .....                               Text        : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Table specific attributes:
Type      : ..... (TABLE      : Base-Table   VIEW: View-Table) [1]
Table Style : ..... (OLDEST, OLD: Till SQL/V1 NEW : SQL/V2) [2]
View Storage : ..... (SHORT   : Short text  LONG: Long-text) [3]
Check Option (Y/N) : . (Y      : with       N : without) [4]

Relationships to succeeding objects          Attributes for relationships
Type      Name                               Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
-----
Command: .....
-----

```

Bild 110 Maske TR2010 für Objekttyp RBT

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Type	<u>TABLE</u> VIEW	Art der Tabelle: Basistabelle oder View
2	Table Style	OLD OLDEST NEW Leerzeichen	SESAM-Version, mit der gearbeitet wird: OLDEST: SESAM V14 OLD : SESAM V1 NEW : SESAM V2 Voreinstellungen: bei Typ TABLE: NEW bei Typ VIEW : Leerzeichen
3	View Storage	SHORT LONG <u>Leerzeichen</u>	Ablagebereich für den Abfrageausdruck der Viewdefinition SHORT: Kurztextbereich LONG: Langtextbereich Voreinstellungen: bei Typ TABLE: Leerzeichen bei Typ VIEW : SHORT

Nr.	Bezeichnung	Wertebereich	Bedeutung
4	Check Option	Y(es) N(o) Leerzeichen	Sätze, die über den View eingegeben oder geändert werden, müssen der in der WHERE-Klausel definierten Bedingung entsprechen. Voreinstellungen: bei Typ TABLE: Leerzeichen bei Typ VIEW : N

Beziehungen

Folgende Beziehungen können von einem RBT-Objekt (Typ Basistabelle) ausgehen:

- zu einem oder zu mehreren Feldern (FLD)
- zu einem oder zu mehreren Eindeutigkeitsbedingungen (RUC)
- zu einer oder mehreren Check-Bedingungen (RCC)
- zu einem oder zu mehreren Referenzbedingungen (RFC)
- zu einer oder zu mehreren Index-Beschreibungen (RIX)
- zu einer physikalischen Datei (PFL)

Folgende Beziehungen können von einem RBT-Objekt (Typ View) ausgehen:

- zu einem oder zu mehreren Feldern (FLD) mit Beziehungstyp XBT
- zu einer oder mehreren Basistabellen (RBT)

7.4.4 RIX - Index-Beschreibung

```

TR2030 function SQL-INDEX                                $userid .n kz.TOMrfn.TOM.project
-----
Name           : .....                               Creation date: .....
Description    : .....                               Last change  : .....
              : .....                               Responsible  : ...
              : .....                               Text        : .
Attributes     : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.
Length  [1]   : ... (total index length in Byte)

Relationships to succeeding objects                    Attributes for relationships
Type          Name                                     Type Occurs 1 2 3 4 5 6 7 8
-----
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
Command: .....
-----
    
```

Bild 111 Maske TR2030 für Objekttyp RIX

Attribut

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Length	1-256 Spalten <u>0</u>	Spaltenanzahl

Beziehungen

Folgende Beziehungen können von einem RIX-Objekt ausgehen:

- zu einem oder zu mehreren Feldern (FLD), siehe auch Beziehungstyp XIX
- zu einer physikalischen Datei (PFL)
- nur für Oldstyle-Tabellen: zu einer Eindeutigkeitsbedingung (RUC) (für den Compound-key)

7.4.5 RUC - Eindeutigkeitsbedingung (Uniqueness Constraint)

```

TR2070 function SQL-UNIQ-CONSTR                $userid .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
Description : .....                               Last change  : .....
          : .....                               Responsible  : ...
          : .....                               Text        : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Type      : ..... [1] (UNIQUE, PRIMARY-KEY)

Relationships to succeeding objects           Attributes for relationships
Type      Name                               Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
    
```

Bild 112 Maske TR2070 für Objekttyp RUC

Attribut

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Type	<u>UNIQUE</u> PRIMARY-KEY	Art der Bedingung: UNIQUE: Eindeutigkeit PRIMARY KEY: Primärschlüssel

Beziehung

Folgende Beziehung kann von einem RUC-Objekt ausgehen:

- zu einem oder mehreren Feldern (FLD)

7.4.6 RCC - Check-Bedingung (Check Constraint)

RCC ist eine Bedingungsanweisung auf Wertebereiche bestimmter Spalten einer Tabelle.

```

TR2100 function SQL-Check-Constraint          $userid .nkh.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
Description : .....                           Last change  : .....
                                           Responsible  : ...
                                           Text        : .

Attributes  : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Check Storage      : ..... [1]  (SHORT: Short-Text    LONG: TOMLIB-Text)

Relationships to succeeding objects             Attributes for relationships
Type      Name                                     Type Occurs 1 2 3 4 5 6 7 8
-----
.
.
.
.
.
.
.
.
.
.
.
-----
Command: .....
    
```

Bild 113 Maske TR2100 für Objekttyp RCC

Attribut

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Check Storage	<u>SHORT</u> LONG	Ablagebereich für die Prüfregeln (Check) SHORT: Kurztextbereich LONG: Langtextbereich

Beziehung

Folgende Beziehung kann von einem RCC-Objekt ausgehen:

- zu einem oder mehreren Feldern (FLD)

7.4.7 RFC - Referenzbedingung (Referential Constraint)

TR2020 function SQL-REF-CONSTR		\$userid	.nkz.TOMrfn.TOM.project									
Name	:	Creation date:								
Description	:	Last change		:							
	:		Responsible		: ...							
			Text		: .							
Attributes	:	1:	2:	3:	4:	5:	6:	7:	8:			
Relationships to succeeding objects			Attributes for relationships									
Type		Name	Type	Occurs	1	2	3	4	5	6	7	8
.	
.	
.	
.	
.	
.	
.	
.	
Command:												

Bild 114 Maske TR2020 für Objekttyp RFC

Beziehungen

Folgende Beziehungen können von einem RFC-Objekt ausgehen:

- zu einem oder mehreren Feldern (FLD)
- zu einer SQL-Eindeutigkeitsbedingung (RUC)

7.4.8 RSG - SQL-Speicherguppe

```

TR2120 function SESAM-STORAGE-GROUP          $userid .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
                                           Last change  : .....
Description : .....                               Responsible : ...
                                           Text         : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Proof-State      : ....[1]....   Proof-Time : ....[2].....
BS2000-Catalog-Id : ....[3]

Relationships to succeeding objects          Attributes for relationships
Type      Name                               Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
-----

```

Bild 115 Maske TR2120 für Objekttyp RSG

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Proof-State	<u>UNPROOFED</u> NOT CREATED DIFFERENCES SYNCHRONOUS	Dieses Feld wird von Sesam-DDA ausgefüllt, wenn Sie bei den Generierungsfunktionen einen Vergleich zwischen Datenbank und Data Dictionary auswählen.
2	Proof-Time	datum zeit Voreinstellung: Leerzeichen	Dieses Feld wird von Sesam-DDA ausgefüllt, wenn Sie bei den Generierungsfunktionen einen Vergleich zwischen Datenbank und Data Dictionary auswählen. Zeitstempel im Format: tt.mm.jj hh.mm.ss
3	BS2000-Catalog-Id	zeichenkette <u>Leerzeichen</u>	Katalogkennung, auf der die Speicherguppe liegt

Beziehungen

Folgende Beziehung kann von einem RSG-Objekt ausgehen:

- zu einem oder mehreren Datenträgern (VOL)

7.4.9 RUS - Eigentümer (User)

TR2130 function SQL-USER	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change :
	Responsible : ...
		Text : .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.	
Relationships to succeeding objects		Attributes for relationships
Type	Name	Type Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
.
Command:		

Bild 116 Maske TR2130 für Objekttyp RUS

Beziehungen

Folgende Beziehungen können von einem RUS-Objekt ausgehen:

- zu keiner oder einer Datenbank (RCT)
- zu keiner, einer oder mehreren SQL-Speicherguppen (RSG)
- zu keiner, einer oder zu mehreren physikalischen Dateien (PFL)
- zu keinem, einem oder mehreren Datenbank-Schemata (RSM)

7.4.10 FLD - Feld

Die `FLD`-Informationen sind auf zwei Masken verteilt: Die Maske TR0010 enthält COBOL-spezifische Attribute, die Folgemaske TR0012 SESAM- und SQL-spezifische Attribute.

Zuerst gibt TOM-REF die Maske TR0010 aus. In diese Maske muß, falls noch nicht in der vorangegangenen Maske geschehen, der Objektname festgelegt werden. Objekttyp-neutrale Attribute (z. B. Description) müssen, sofern geplant, ebenfalls bereits in die Maske TR0010 eingetragen werden. Nach dem Abschicken der Maske TR0010 mit der Taste `[DÜ]` wird die Folgemaske TR0012 ausgegeben.

Der Objektname und die objekttyp-neutralen Attribute werden in die Folgemaske TR0012 (nicht überschreibbar) übernommen. Die Masken sind ab Seite 178 beschrieben.

7.4.11 PFL - Physische Datei

Dieser Objekttyp ist dem DVS/Leasy-Teilstrukturkatalog zugeordnet und ab Seite 200 beschrieben.

7.4.12 VOL - Datenträger

Dieser Objekttyp ist dem DVS/Leasy-Teilstrukturkatalog zugeordnet und ab Seite 198 beschrieben.

7.4.13 XBT - vordefinierter Beziehungstyp View-Feld

Eine XBT-Beziehung kann zwischen Objekten folgender Typen bestehen:

- Ausgabobjekt: RBT (Typ View)
- Zielobjekt: ein oder mehrere Datenfelder (FLD)

TR2140	function relationship	\$userid	.nkz.TOMrfn.TOM.project
		Creation date:	
		Last change :	
Description :		Responsible : ...	
		Text : .	
Relationship from object			
Type	Name		
.....			
to succeeding objects		Attributes for relationships	
Type	Name	Type Occurs	1 2 3 4 5 6 7 8
.....		* XBT	
Column-usage : [1]	for	(INPUT/OUTPUT/BOTH)
First-occurence :	... [2]	from	(1..255)
Last-occurence :	... [3]	to	(1..255)
Command:			

Bild 117 Maske TR2140 für Beziehungstyp XBT

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Column-usage	INPUT OUTPUT <u>BOTH</u>	SQL-Spaltenverwendung
2	First-occurence	<u>1</u> ..255	erstes Auftreten im View
3	Last-occurence	<u>1</u> ..255	letztes Auftreten im View

7.4.14 XIX - vordefinierter Beziehungstyp Index-Feld

Eine XIX-Beziehung kann zwischen Objekten folgender Typen bestehen:

- Ausgabgsobjekt: Index (RIX)
- Zielobjekt: ein oder mehrere Datenfelder (FLD)

```

TR2040 function relationship                               $userid .nkz.TOMrfn.TOM.project
-----
Description : .....                               Creation date: .....
                                                    Last change : .....
                                                    Responsible : ...
                                                    Text       : .

Relationship from object
Type      Name
.....

to succeeding objects                               Attributes for relationships
Type      Name                                     Type Occurs 1 2 3 4 5 6 7 8
.....
* XIX .....

Index-column-Length : ... [1]

-----
Command: .....
    
```

Bild 118 Maske TR2040 für Beziehungstyp XIX

Attribut

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Index-Column-Length	1..255 NIL-Zeichen	Index-Spaltenlänge

8 Process/JCL

Eine BS2000-Prozedur beschreiben Sie im Data Dictionary mit den Objekten und Beziehungen des Process/JCL-Teistrukturkatalogs. Eine BS2000-Prozedur beschreiben heißt, daß Sie die nötigen Objekte und Beziehungen erstellen. In Kapitel 4 ist beschrieben, wie Sie Objekte und Beziehungen erstellen können.

8.1 Process/JCL-Objekte

Die Objekttypen von Process/JCL sind verschiedenen Teistrukturkatalogen zugeordnet. Die folgenden Tabellen zeigen die Objekttypen der Process/JCL-Teilstruktur den jeweiligen Teistrukturkatalogen zugeordnet mit ihren Kurznamen, Namen und einer kurzen Erläuterung.

Sie finden in diesem Kapitel auch die Beschreibung der Teistrukturkataloge für den Daten- und Funktionsentwurf, da diese den ersten Schritt bei der Programmentwicklung darstellen.

Kurzname	Langname des Objekttyps	Erläuterung
ENT	ENTER	BS2000-Batch-Prozedur
JV	JOB-VARIABLE	Logische Jobvariable in BS2000-Prozeduren
PJV	PHYSICAL-JV	Physikalische Jobvariable in BS2000-Prozeduren
PRC	PROCEDURE	BS2000-Dialog-Prozedur

Tabelle 13 Objekttypen im JCL-Teistrukturkatalog

Kurzname	Langname des Objekttyps	Erläuterung
CAL	ENTRY-CALLED	Call-Aufruf
ENP	ENTRY-INCLUDED	Entry-Punkt
MGR	MODULE-GROUP	Modulgruppe
MOD	MODULE	Modul
PGM	PROGRAM	Programm
SYS	SYSTEM	System

Tabelle 14 Objekttypen im Programmdesign-Teilstrukturkatalog

Kurzname	Langname des Objekttyps	Erläuterung
CBL	COBOL-SOURCE	COBOL-Quellprogramm
LIB	LIBRARY	Bibliothek
LMO	LOAD-MODULE	Lademodul
NOM	NON-S.OBJ.-MOD.	Nicht-Shareable Objektmodul
SOM	SHARE-OBJ.-MOD.	Shareable Objektmodul

Tabelle 15 Objekttypen im Programmrealisierungs-Teilstrukturkatalog

Die folgende Tabelle zeigt die Objekttypen der Funktionsentwurf-Teilstruktur mit ihren Kurznamen, Namen und einer kurzen Erläuterung.

Kurzname	Langname des Objekttyps	Erläuterung
CNn	CONCEPTn	Konzept
EFn	ELEMENTARY-FUNCTIONn	Elementarfunktion
EXn	EXPORTGATEN	Ausgang
FNn	FUNCTIONn	Funktion
ICn	INFORMATION-CONTAINERn	Informationsbehälter
IMn	IMPORTGATEN	Eingang
INn	INFORMATIONn	Information

Tabelle 16 Objekttypen im Funktionsentwurf-Teilstrukturkatalog



Die Objekte des Datenentwurf-Teilstrukturkatalogs sind nicht im Standard-Katalog enthalten und müssen explizit in Ihr Data Dictionary eingebracht werden.

8.2 Process/JCL-Beziehungen

Zwischen welchen Objekten des Process/JCL-Teilstrukturkatalogs Sie Beziehungen erstellen können, zeigen die folgenden Tabellen. Dabei bedeutet ein X in einem Feld, daß zwischen zwei Objekttypen eine Beziehung bestehen kann. Wie Sie diese Beziehungen erstellen, ist in Kapitel 4 beschrieben.

Objekttyp als Ausgang der Beziehung	Objekttyp als Ziel der Beziehung													
	ENT	FIL	FLD	FLG	JV	LMO	MGR	MOD	NOM	PFL	PGM	PJV	PRC	REC
ENT Enter	X	X			X	X	X	X	X	X	X	X	X	
JV Job Variable			X	X										X
PJV Physiscal JV					X									
PRC Procedure	X	X			X	X	X	X	X	X	X	X	X	

Tabelle 17 Beziehungstypen des JCL-Teilstrukturkatalogs

Objekttyp als Ausgang der Beziehung	Objekttyp als Ziel der Beziehung																							
	ARE	CBL	CAL	DAR	ENP	FIL	FLD	FLG	INL	JV	MGR	MOD	MSK	PGM	REC	RBT	RSM	SBR	SDY	SET	SSY	SUB	SYS	UTP
CAL Entry Called*)					X		X	X			X	X			X									
ENP Entry Included							X	X							X									
MGR Module Group	X		X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X
MOD Module	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X
PGM Program											X	X		X										
SYS System														X									X	

Tabelle 18 Beziehungstypen des Programm-Design-Teilstrukturkatalogs

*) Eine Beziehung von einem Objekt des Typs CAL zu einem Objekt des Typs ENP, MGR oder MOD wird von TOM-REF über Namenskonventionen abgeleitet. Sie muß nicht explizit erstellt werden.

Mehrstellige Beziehungstypen wie "Entry-called (CAL) ruft Entry-included (ENP) und übergibt Felder (FLD)" sind in TOM-REF abbildbar. Hierzu werden die Objekttypen Entry-called (CAL) und Entry-included (ENP) durch Namenskonventionen verknüpft und die Felder (FLD) werden mit CAL und/oder ENP verbunden.

Objekttyp als Ausgang der Beziehung	Objekttyp als Ziel der Beziehung									
	CBL	COM	DRP	DSP	FIL	LIB	NOM	SDB	SOM	
CBL COBOL Source	X					X	X		X	
COM Compiler Opt.						X				
INL Copy						X				
LMO Load Module						X	X		X	
MOD Module	X		X							
MSK Format						X				
NOM Non S.Obj.Mod.				X		X				
SOM Shareable O.M.						X				
PFL Physical File						X				

Tabelle 19 Beziehungstypen des Programm-Realisierung-Teilstrukturkatalogs

Zwischen welchen Objekten des Funktionsentwurf-Teilstrukturkatalogs Sie Beziehungen erstellen können, zeigt die folgende Tabelle. Dabei bedeutet ein X in einem Feld, daß zwischen zwei Objekttypen eine Beziehung bestehen kann. Wie Sie diese Beziehungen erstellen, ist in Kapitel 4 beschrieben.

Objekttyp als Ausgang der Beziehung	Objekttyp als Ziel der Beziehung								
	EFn	ETy	EXn	FLD	FLG	FNn	ICn	IMn	INn
CNn Concept	X					X	X		
EFn Element. Fct.			X					X	
EXn Export Gate			X					X	X
FNn Function	X		X			X	X	X	
ICn Inform. Cont.			X				X	X	X
IMn Import Gate								X	X
INn Information		X		X	X				X

Tabelle 20 Beziehungstypen des Funktionsentwurf-Teilstrukturkatalogs

8.3 Teilstrukturkataloge für Process/JCL

Die Teilstrukturkataloge von JCL, Programm-Design und Programm-Realisierung sind Basis für die Beschreibung einer BS2000-Prozedur im Data Dictionary. Sie legen fest, welches Objekt das Ausgangsobjekt und welches das Zielobjekt einer Beziehung sein muß. Die Pfeile zeigen vom Typ des Ausgangsobjekts zum Typ des Zielobjekts. Eine Beziehung müssen Sie immer in die Beziehungsliste des Ausgangsobjekts eintragen. Die Elementaraussagen klassifizieren die Beziehungen näher.

Eine genaue Beschreibung der in diesen Modellen verwendeten Darstellungsmittel finden Sie ab Seite 7.

8.3.1 JCL-Teilstrukturkatalog

- b) siehe Programmdesign-Teilstruktur
- c) siehe Programmrealisierung-Teilstruktur
- d) siehe UTM-Teilstruktur
- e) siehe DVS/Leasy-Teilstruktur

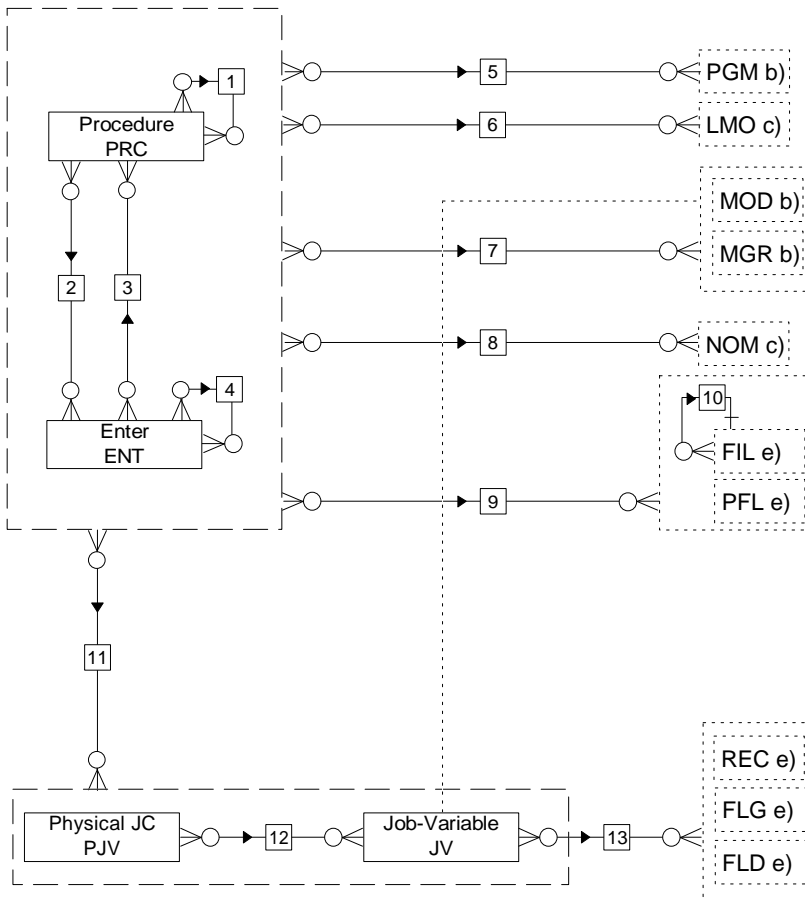


Bild 119 JCL-Teilstrukturkatalog

Die Wertigkeit des Beziehungstyps (1:1, 1:n oder n:m) ist in diesem Entity-Relationship-Diagramm zwar enthalten, aber sie wird in der aktuellen Version von TOM-REF nicht unterstützt. Sie können allerdings anhand dieses Teilstrukturkatalogs für korrekte Wertigkeit sorgen.

Elementaraussagen

1	Eine Do-Prozedur (PRC)	enthält	keine, eine oder mehrere	PRC
	Eine Do-Prozedur (PRC)	ist enthalten	in keiner, einer oder mehreren	PRC
2	Eine Do-Prozedur (PRC)	enthält	keine, eine oder mehrere	ENT
	Eine Enter-Prozedur (ENT)	ist enthalten	in keiner, einer oder mehreren	PRC
3	Eine Enter-Prozedur (ENT)	enthält	keine, eine oder mehrere	PRC
	Eine Do-Prozedur (PRC)	ist enthalten	in keiner, einer oder mehreren	ENT
4	Eine Enter-Prozedur (ENT)	enthält	keine, eine oder mehrere	ENT
	Eine Enter-Prozedur (ENT)	ist enthalten	in keiner, einer oder mehreren	ENT
5	Eine Do-Prozedur (PRC) und/oder eine Enter-Prozedur (ENT)	enthält	keine, eine oder mehrere	PGM
	Ein Programm (PGM)	ist enthalten	in keiner, einer oder mehreren	PRC und/oder ENT
6	Eine Do-Prozedur (PRC) und/oder eine Enter-Prozedur (ENT)	enthält	keine, eine oder mehrere	LMO
	Ein Lademodul (LMO)	ist enthalten	in keiner, einer oder mehreren	PRC und/oder ENT

7	Eine Do-Prozedur (PRC) und/oder eine Enter-Prozedur (ENT)	enthält	keine, eine oder mehrere	MOD und/oder MGR
	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	ist enthalten	in keiner, einer oder mehreren	PRC und/oder ENT
8	Eine Do-Prozedur (PRC) und/oder eine Enter-Prozedur (ENT)	enthält	keine, eine oder mehrere	NOM
	Ein Nicht-Shareable Objektmodul (NOM)	ist enthalten	in keiner, einer oder mehreren	PRC und/oder ENT
9	Eine Do-Prozedur (PRC) und/oder eine Enter-Prozedur (ENT)	enthält	keine, eine oder mehrere	FIL und/oder PFL
	Eine logische Datei (FIL) und/oder eine physische Datei (PFL)	ist enthalten	in keiner, einer oder mehreren	PRC und/oder ENT
10	Eine logische Datei (FIL)	ist Vorläufer	keiner, einer oder mehrerer	FIL
11	Eine Do-Prozedur (PRC) und/oder eine Enter-Prozedur (ENT)	enthält	keine, eine oder mehrere	PJV und/oder JV
	Eine physische Jobvariable (PJV) und/oder eine logische Jobvariable (JV)	ist enthalten	in keiner, einer oder mehreren	PRC und/oder ENT
12	Eine physische Jobvariable (PJV)	gehört	zu keiner, einer oder mehreren	JV
	Eine logische Jobvariable (JV)	gehört	zu einer oder mehreren	PJV
13	Eine logische Jobvariable (JV)	enthält	keine, eine oder mehrere	REC und/oder FLG und/oder FLD
	Ein Datensatz (REC) und/oder eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist enthalten	in keiner, einer oder mehreren	JV

8.3.2 Programmdesign-Teilstrukturkatalog

- a) siehe JCL-Teilstruktur
- c) siehe Programmrealisierungs-Teilstruktur
- d) siehe UTM-Teilstruktur
- e) siehe DVS/Leasy-Teilstruktur
- f) siehe UDS-Teilstruktur
- g) siehe SESAM/SQL-V2-Teilstrukturkatalog

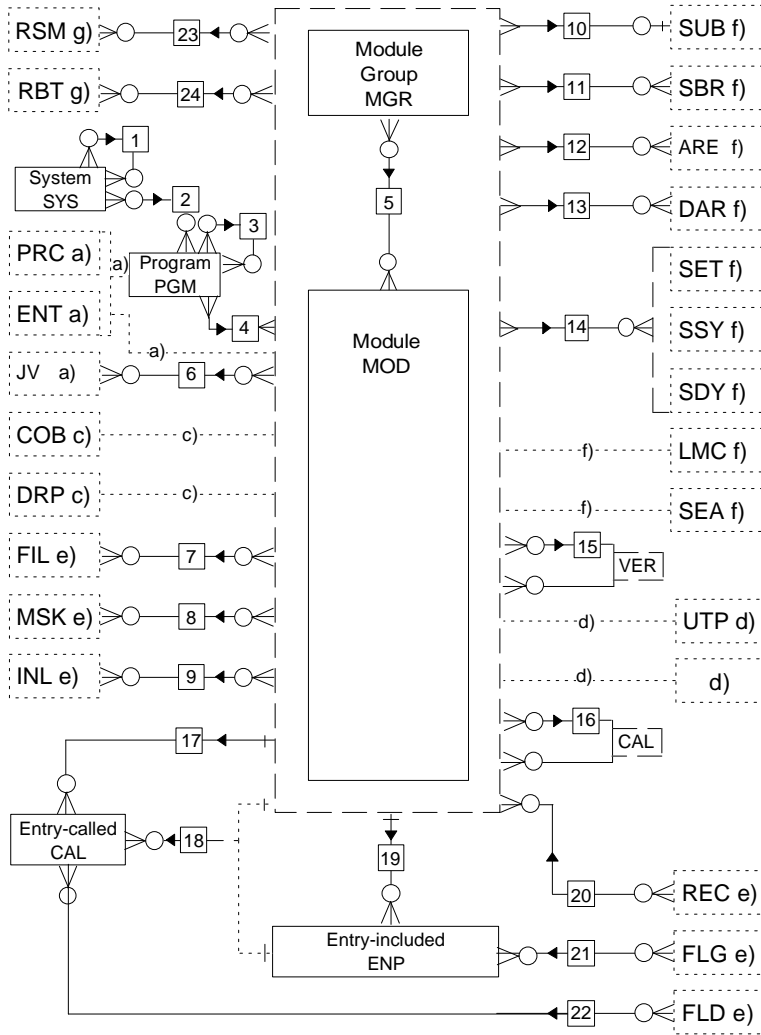


Bild 120 Programmdesign-Teilstrukturkatalog

Die Wertigkeit des Beziehungstyps (1:1, 1:n oder n:m) ist in diesem Entity-Relationship-Diagramm zwar enthalten, aber sie wird in der aktuellen Version von TOM-REF nicht unterstützt. Sie können allerdings anhand dieses Teilstrukturkatalogs für korrekte Wertigkeit sorgen.

Elementaraussagen

1	Ein System (SYS)	enthält	kein, ein oder mehrere	SYS
	Ein System (SYS)	ist enthalten	in keinem, einem oder mehreren	SYS
2	Ein System (SYS)	enthält	kein, ein oder mehrere	PGM
	Ein Programm (PGM)	ist enthalten	in keinem, einem oder mehrere	SYS
3	Ein Programm (PGM)	enthält	kein, ein oder mehrere	PGM
	Ein Programm (PGM)	ist enthalten	in keinem, einem oder mehreren	PGM
4	Ein Programm (PGM)	enthält	einen oder mehrere	MOD und/oder MGR
	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	ist enthalten	in einem oder mehreren	PGM
5	Eine Modulgruppe (MGR)	enthält	einen oder mehrere	MOD
	Ein Modul (MOD)	ist enthalten	in keinem, einem oder mehreren	MGR
6	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	verwendet	keine, eine oder mehrere	JV
	Eine Jobvariable (JV)	wird verwendet	von keinem, einem oder mehreren	MOD und/oder MGR
7	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	verwendet	keinen, einen oder mehrere	FIL
	Eine logische Datei (FIL)	wird verwendet	von einem oder mehreren	MOD und/oder MGR

8	Ein Modul (MOD) und/ oder eine Modulgruppe (MGR)	verwendet	keine, eine oder mehrere	MSK
	Eine Maske (MSK)	wird ver- wendet	von einem oder mehreren	MOD und/ oder MGR
9	Ein Modul (MOD) und/ oder eine Modulgruppe (MGR)	verwendet	keine, eine oder mehrere	INL
	Eine Copy-Strecke (INL)	wird ver- wendet	von keinem, einem oder mehreren	MOD und/ oder MGR
10	Ein Modul (MOD) und/ oder eine Modulgruppe (MGR)	verwendet	kein oder ein	SUB
	Ein UDS-Subschema (SUB)	wird ver- wendet	von einem oder mehreren	MOD und/ oder MGR
11	Ein Modul (MOD) und/ oder eine Modulgruppe (MGR)	verwendet	kein, ein oder mehrere	SBR
	Ein UDS-Subschema- Datensatz (SBR)	wird ver- wendet	von einem oder mehreren	MOD und/ oder MGR
12	Ein Modul (MOD) und/ oder eine Modulgruppe (MGR)	verwendet	kein, ein oder mehrere	ARE
	Ein UDS-Realm (ARE)	wird ver- wendet	von einem oder mehreren	MOD und/ oder MGR
13	Ein Modul (MOD) und/ oder eine Modulgruppe (MGR)	verwendet	kein, ein oder mehrere	DAR
	Ein dynamischer Realm (DAR) in UDS	wird ver- wendet	von einem oder mehreren	MOD und/ oder MGR
14	Ein Modul (MOD) und/ oder eine Modulgruppe (MGR)	verwendet	kein, ein oder mehrere	SET und/ oder SSY und/oder SDY

	Eine UDS-Set (SET) und/oder UDS-System-Set (SSY) und/oder Dynamisches Set (SDY)	wird verwendet	von einem oder mehreren	MOD und/oder MGR
15	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	verarbeitet	keinen, einen oder mehrere	MOD und/oder MGR
	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	wird verarbeitet	von keinem, einem oder mehrere	MOD und/oder MGR
16	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	steuert an	keinen, einen oder mehrere	MOD und/oder MGR
	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	wird angesteuert	von keinem, einem oder mehreren	MOD und/oder MGR
17	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	enthält	keinen, einen oder mehrere	CAL
	Ein Call (CAL)	ist enthalten	in einem	MOD und/oder MGR
18	Ein Modul (MOD) und/oder eine Modulgruppe (MGR) oder vorrangig Entry-Punkt (ENP)	steuert an mit dem Namen	keinen, einen oder mehrere	CAL
	Ein Call (CAL)	wird angesteuert mit dem Namen	über einen	MOD und/oder MGR oder vorrangig ENP
19	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	enthält	keinen, einen oder mehrere	ENP
	Ein Entry-Punkt (ENP)	ist enthalten	in einem	MOD und/oder MGR
20	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	enthält	keinen, einen oder mehrere	REC und/oder FLG und/oder FLD
	Ein Datensatz (REC) und/oder eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist enthalten	in keinem, einem oder mehreren	MOD und/oder MGR

21	Ein Entry-Punkt (ENP)	enthält	keinen, einen oder mehrere	REC und/oder FLG und/oder FLD
	Ein Datensatz (REC) und/oder eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist enthalten	in keinem, einem oder mehreren	ENP
22	Ein Call (CAL)	ist enthalten	in keinen, einen oder mehrere	REC und/oder FLG und/oder FLD
	Ein Datensatz (REC) und/oder eine Feldgruppe (FLG) und/oder ein Feld (FLD)	enthält	keinen, einen oder mehrere	CAL
23	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	verwendet	kein, ein oder mehrere	RSM
	Ein SQL-Schema (RSM)	wird verwendet	von keinem, einem oder mehreren	MOD und/oder MGR
24	Ein Modul (MOD) und/oder eine Modulgruppe (MGR)	verwendet	kein, ein oder mehrere	RBT
	Eine SQL-Basistabelle (RBT)	wird verwendet	von keinem, einem oder mehreren	MOD und/oder MGR

8.3.3 Programmrealisierung-Teilstrukturkatalog

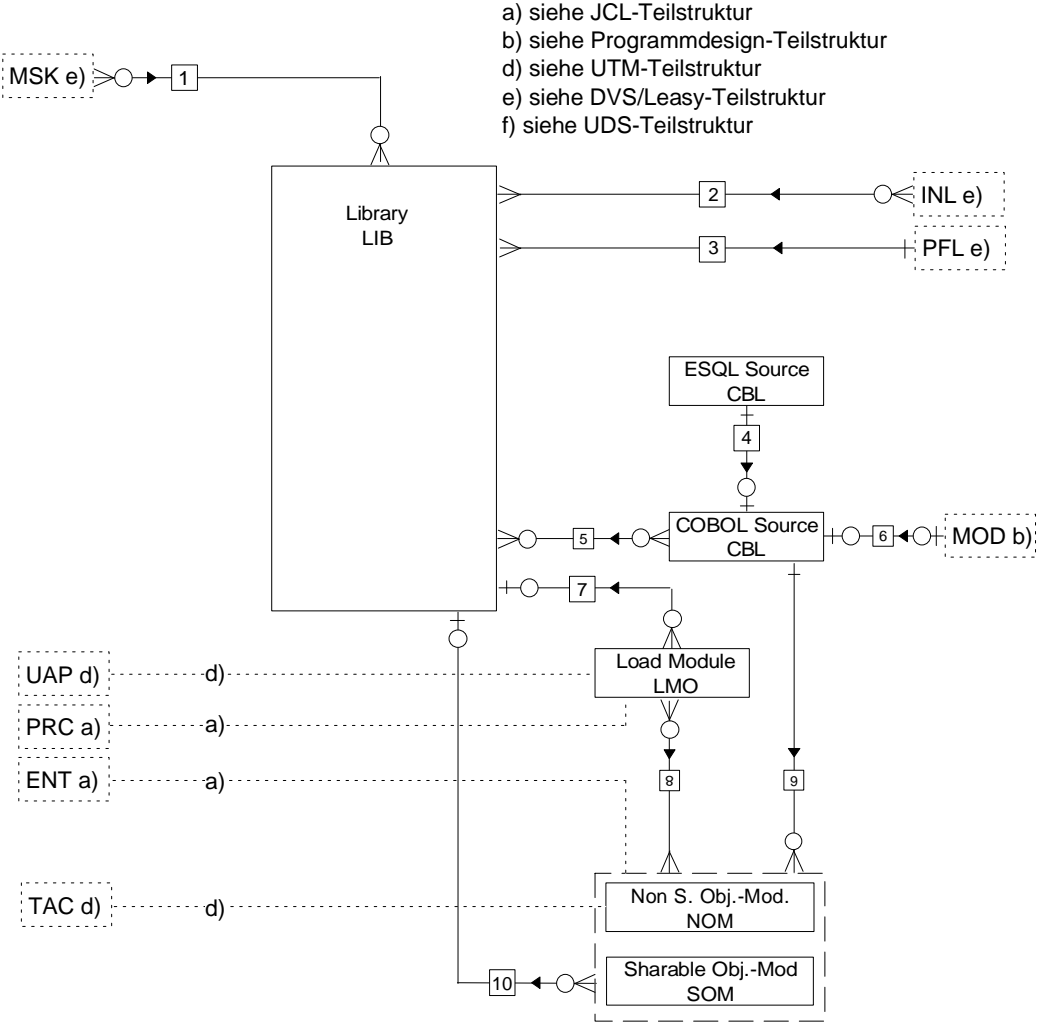


Bild 121 Programmrealisierung-Teilstrukturkatalog

Die Wertigkeit des Beziehungstyps (1:1, 1:n oder n:m) ist in diesem Entity-Relationship-Diagramm zwar enthalten, aber sie wird in der aktuellen Version von TOM-REF nicht unterstützt. Sie können allerdings anhand dieses Teilstrukturkatalogs für korrekte Wertigkeit sorgen.

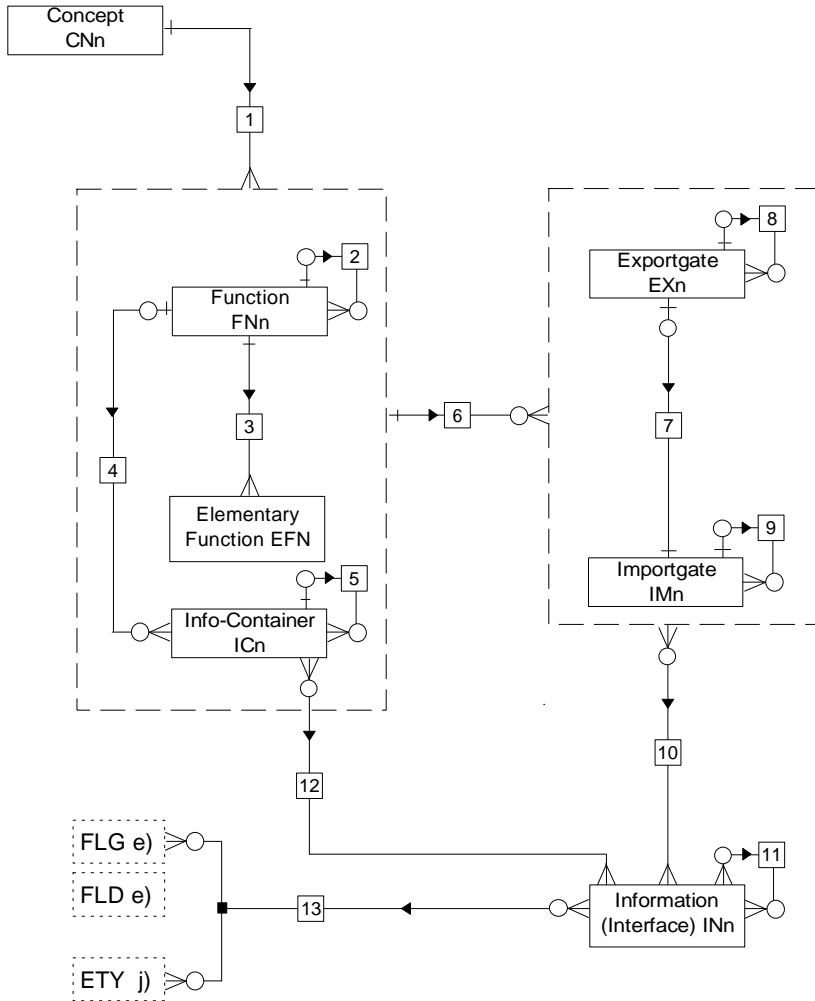
Elementaraussagen

1	Ein Format (MSK)	belegt	eine oder mehrere	LIB
	Eine Bibliothek (LIB)	wird belegt	von keiner, einer oder mehreren	MSK
2	Eine Copy-Strecke (INL)	belegt	eine oder mehrere	LIB
	Eine Bibliothek (LIB)	wird belegt	von keiner, einer oder mehreren	INL
3	Eine physische Datei (PFL)	belegt	eine oder mehrere	LIB
	Eine Bibliothek (LIB)	wird belegt	von einer	PFL
4	Eine COBOL-Source mit ESQL-Anweisungen (CBL)	ergibt	keine oder eine	CBL
	Eine COBOL-Source (CBL)	ist entstanden aus	einer	CBL
5	Eine COBOL-Source (CBL)	belegt	keine oder eine	LIB
	Eine Bibliothek (LIB)	wird belegt	von keiner, einer oder mehreren	CBL
6	Ein Modul (MOD)	ergibt	keine oder eine	CBL
	Eine COBOL-Source (CBL)	wird erzeugt	aus keinem oder einem	MOD
7	Ein Lademodul (LMO)	belegt	keine oder eine	LIB
	Eine Bibliothek (LIB)	wird belegt	von keiner, einer oder mehreren	LMO

8	Ein Lademodul (LMO)	enthält	eine oder mehrere	NOM und/oder SOM
	Ein Shareable-Objektmodul (SOM) und/oder ein Nicht-Shareable-Objektmodul (NOM)	ist enthalten	in keinem, einem oder mehreren	LMO
9	Eine COBOL-Source (CBL)	ergibt	kein, ein oder mehrere	NOM und/oder SOM
	Ein Shareable-Objektmodul (SOM) und/oder ein Nicht-Shareable-Objektmodul (NOM)	wird erzeugt	aus einer	CBL
10	Ein Shareable-Objektmodul (SOM) und/oder ein Nicht-Shareable-Objektmodul (NOM)	belegt	keine oder eine	LIB
	Eine Bibliothek (LIB)	wird belegt	von keiner, einer oder mehreren	NOM und/oder SOM
11	Eine Compiler-Option (COM)	bestimmt	keine oder eine	LIB
	Eine Bibliothek (LIB)	wird bestimmt	von keiner, einer oder mehreren	COM

8.3.4 Funktionsentwurf-Teilstrukturkatalog

e) siehe DVS/Leasy-Teilstruktur
 j) siehe Vorschlag einer Datenentwurfs-Teilstruktur



Hinweis: n in den Typnamen steht für ein numerisches Zeichen von 1-9 zur Kennzeichnung des entsprechenden Konzepts, z.B.: 1= Fachkonzept; 2 = DV-Konzept. Zunächst sind 1 und 2 vorgesehen.

Bild 122 Funktionsentwurf-Teilstrukturkatalog

Elementaraussagen

1	Ein Konzept (CNn)	besteht auf der obersten Ebene aus	ein oder mehreren	FNn und/oder EFn und/oder ICn
	Eine Funktion (FNn) und/oder Elementarfunktion (EFn) und/oder ein Informationsbehälter (ICn)	ist auf der obersten Ebene Bestandteil von	einem	CNn
2	Eine Funktion (FNn)	besteht aus	keiner, einer oder mehreren	FNn
	Eine Funktion (FNn)	ist Bestandteil von	keiner oder einer	FNn
3	Eine Funktion (FNn)	besteht aus	einer oder mehreren	EFn
	Eine Elementarfunktion (EFn)	ist Bestandteil von	einer	FNn
4	Eine Funktion (FNn)	besteht aus	keinem, einem oder mehreren	ICn
	Ein Informationsbehälter (ICn)	ist Bestandteil von	keiner oder einer	FNn
5	Ein Informationsbehälter (ICn)	besteht aus	keinem oder einem	ICn
	Ein Informationsbehälter (ICn)	ist Bestandteil von	keinem oder einem	ICn
6	Eine Funktion (FNn) und/oder Elementarfunktion (EFn) und/oder ein Informationsbehälter (ICn)	besitzt	keine, eine oder mehrere	EXn und/oder IMn
	Ein Export- (EXn) und/oder Importgate (IMn)	gehört zu	einer/einem	FNn und/oder EFn und/oder ICn

7	Ein Exportgate (EXn)	steuert an	einen	IMn
	Ein Importgate (IMn)	wird ange- steuert	von einem	EXn
8	Ein Exportgate (EXn)	besteht aus	kein, eine oder mehrere	EXn
	Ein Exportgate (EXn)	ist Bestand- teil von	in keinem oder einem	EXn
9	Ein Importgate (IMn)	besteht aus	kein, ein oder mehrere	IMn
	Ein Importgate (IMn)	ist Bestand- teil von	in keinem oder einem	IMn
10	Ein Export- (EXn) und/ oder Inportgate (IMn)	besitzt	eine oder mehrere	INn
	Eine Information (INn)	gehört zu	keinem, einem oder mehreren	EXn und/ oder IMn
11	Eine Information (INn)	enthält	keine, eine oder mehrere	INn
	Eine Information (INn)	ist enthalten	in keiner, einer oder mehreren	INn
12	Ein Informationsbe- hälter (ICn)	enthält	keine, eine oder mehrere	INn
	Eine Information (INn)	ist enthalten	in keinem, einen oder mehreren	ICn
13	Eine Information (INn)	verweist	auf kein, eine oder mehrere	FLG und/ oder FLD und/oder ETY
	Eine Feldgruppe (FLG) und/oder ein Feld (FLD) und/oder ein Entity (ETY)	ist zuge- ordnet	keinem, einem oder mehreren	INn

8.3.5 Vorschlag für einen Datenentwurf-Teilstrukturkatalog

e) siehe DVS/Leasy-Teilstruktur
 j) siehe Funktionsentwurfs-Teilstruktur

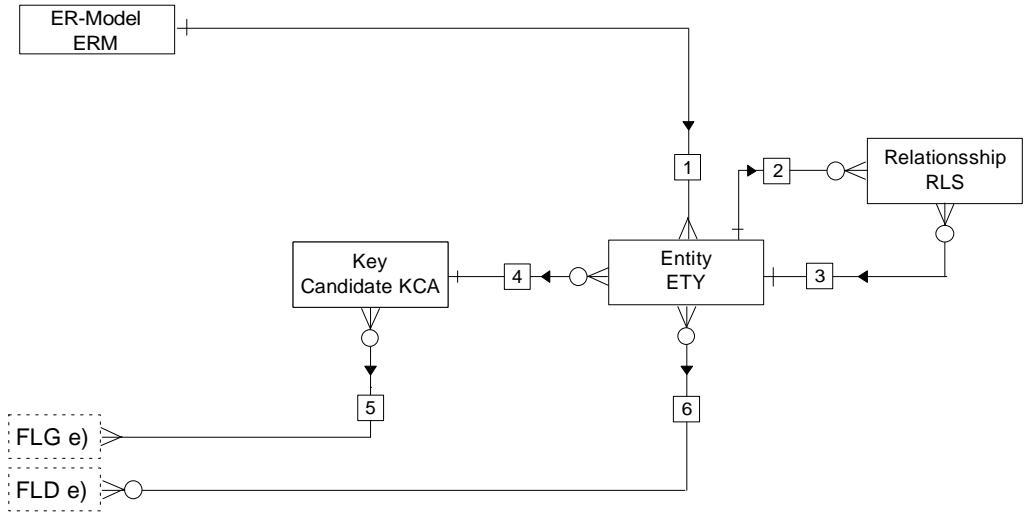


Bild 123 Datenentwurf-Teilstrukturkatalog

Elementaraussagen

1	Ein Entity-Relationship-Modell (ERM)	enthält	ein oder mehrere	ETY
	Ein Entity (ETY)	ist enthalten	in einem	ERM
2	Ein Entity (ETY)	besitzt	keine, eine oder mehrere	RLS
	Eine Beziehung (RLS)	ist zugeordnet	einem	ETY
3	Eine Beziehung (RLS)	steuert an	ein	ETY
	Ein Entity (ETY)	wird angesteuert	von keinem, einem oder mehreren	RSL
4	Ein Entity (ETY)	verweist	auf keinen, einen oder mehrere	KCA
	Ein Schlüsselkandidat (KCA)	ist zugeordnet	keiner, einer oder mehreren	ETY
5	Ein Schlüsselkandidat (KCA)	verweist	auf einen oder mehrere	FLG und/oder FLD
	Eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist zugeordnet	keinem, einem oder mehreren	KCA
6	Ein Entity (ETY)	verweist	auf keinen, einen oder mehrere	FLG und/oder FLD
	Eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist zugeordnet	keiner, einer oder mehreren	ETY

8.4 Sub-Menü Process/JCL

Die Objekttypen des Process/JCL-Teilstrukturkatalogs erstellen und bearbeiten Sie in den objekttyp- und beziehungstyp-spezifischen Masken. Diese Masken erreichen Sie über das Sub-Menü Process/JCL in der Auswahlmaske TR0003.

Das Sub-Menü Process/JCL erhalten Sie, wenn Sie in der Maske TR0001 (siehe Seite 177) den Kennbuchstaben *r* auswählen.

TR0003	Selection Process/JCL-Types	\$userid	.nkz.TOMrfn.TOM.project
Please select object-types and functions (e.g. a3,c2)			
Object-types:			
a . System (SYS)	i . Library (LIB)	r . Procedure (PRC)	
b . Program (PGM)	j . COBOL-Source (CBL)	s . Enter (ENT)	
c . Module (MOD)	k . Load-Module (LMO)	t . Job-Variable (JV)	
d . Module-Group (MGR)	l . Non-S.Obj.-Mod. (NOM)	u . Physical-JV (PJV)	
e . Entry-called (CAL)	m . Share-Obj.-Mod. (SOM)		
f . Entry-included (ENP)		x . Other	
Functions for object-types:			
1 . Display	2 . Create	3 . Change	4 . Delete
Functions without object-types:			
5 . Services			
Command:			

Bild 124 Auswahlmaske TR0003 für Process/JCL-Typen

Die einzelnen typspezifischen Masken sind in der Reihenfolge der Auswahlmaske TR0003 in den folgenden Unterabschnitten beschrieben. Die Maske TR5000 für die *other*-Objekttypen (Kennbuchstabe *x*) wurde ab Seite 203 beschrieben.

Ferner ist es möglich, aus der Maske TR0003 in die Auswahlmaske TR9000 für die Servicefunktionen zu wechseln (Kennziffer 5).

8.4.1 SYS - System

```

TR0300  function SYSTEM                                     $userid .nkz.TOMrfn.TOM.project
-----
Name      : .....                                         Creation date: .....
Description : .....                                         Last change  : .....
          : .....                                         Responsible   : ...
          : .....                                         Text         : .

Attributes : 1:.  2:.  3:.  4:.  5:.  6:.  7:.  8:.

Relationships to succeeding objects                      Attributes for relationships
Type          Name                                      Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
-----

```

Bild 125 Maske TR0300 für Objekttyp SYS

Beziehungen

Folgende Beziehungen können von einem SYS-Objekt ausgehen:

- zu einem oder zu mehreren Systemen (SYS)
- zu einem oder zu mehreren Programmen (PGM)

8.4.2 PGM - Programm

```
TR0310  function PROGRAM                               $userid .nkz.TOMrfn.TOM.project
-----
Name       : .....                               Creation date: .....
Description: .....                               Last change  : .....
           : .....                               Responsible  : ...
           : .....                               Text         : .

Attributes  : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

  Relationships to succeeding objects      Attributes for relationships
  Type          Name                       Type Occurs 1 2 3 4 5 6 7 8
  .....
  .....
  .....
  .....
  .....
  .....
  .....
  .....

-----
Command: .....
```

Bild 126 Maske TR0310 für Objekttyp PGM

Beziehungen

Folgende Beziehungen können von einem PGM-Objekt ausgehen:

- zu einem oder zu mehreren Programmen (PGM)
- zu einem oder zu mehreren Modulen (MOD)
- zu einer oder zu mehreren Modulgruppen (MGR)

8.4.3 MOD - Modul

TR0320	function MODULE	\$userid	.nkz.TOMrfn.TOM.project				
Name :				Creation date:		
Description :				Last change :		
				Responsible : ...		
				Text :		
Attributes : 1: . 2: . 3: . 4: . 5: . 6: . 7: . 8: .							
Relationships to succeeding objects				Attributes for relationships			
Type	Name	Type	Occurs	1	2	3	4
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
Command:							

Bild 127 Maske TR0320 für Objekttyp MOD

Beziehungen (Gliederung nach Teilstruktur-Katalogen)

Folgende Beziehungen können von einem MOD-Objekt ausgehen:

DVS/LEASY

- zu einer oder zu mehreren logischen Dateien (FIL)
- zu einem oder zu mehreren Datensätzen (REC)
- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einem oder zu mehreren Formaten (MSK)
- zu einem oder zu mehreren Copy-Elementen (INL)

SQL

- zu einem oder mehreren SQL-Views (RBT)
- zu einem oder mehreren SQL-Datenbank-Schemata (RSM)
- zu einem oder mehreren Tabellenspalten (FLD)

Process/JCL

- zu einem oder zu mehreren Modulen (MOD) und/oder zu einer oder zu mehreren Modulgruppen (MGR) mit dem Beziehungstyp VER (verarbeitet)
- zu einem oder zu mehreren Modulen (MOD) und/oder zu einer oder zu mehreren Modulgruppen (MGR) mit dem Beziehungstyp CAL (steuert an)
- zu einem oder zu mehreren Entry-Punkten (ENP)
- zu einem oder zu mehreren Call's (CAL)
- zu einer oder zu mehreren Jobvariablen (JV)
- zu einer Cobol-Source (CBL)

UDS

- zu einem oder zu mehreren Subschema-Records (SBR)
- zu einem UDS-Subschema (SUB)
- zu einer oder zu mehreren dynamischen Areas (DAR)
- zu einem oder zu mehreren Sets (SET)
- zu einem oder zu mehreren System-Sets (SSY)
- zu einem oder zu mehreren dynamischen Sets (SDY)
- zu einer oder zu mehreren Areas (ARE)

UTM

- zu einem UTM-Teilprogramm (UTP)
- zu einer oder zu mehreren Relationships (RLS). Vom RLS-Objekt sind folgende Beziehungen aufzubauen:
 1. Zu einem UTM-TAC (TAC) mit dem Beziehungstyp FPU (FPUT).
 2. Zu einem oder zu mehreren Modulen (MOD) und/oder zu einer oder zu mehreren Modulgruppen (MGR).
 3. Zu einem oder zu mehreren Datensätzen (REC) und/oder Feldgruppen (FLG) und/oder Feldern (FLD).

- zu einer oder zu mehreren Relationships (RLS). Vom RLS-Objekt sind folgende Beziehungen aufzubauen:
 1. Zu einem UTM-TAC (TAC) mit dem Beziehungstyp PRE (PEND RE) oder PPR (PEND PR) oder PKP (PEND KP).
 2. Zu einem oder zu mehreren Modulen (MOD) und/oder zu einer oder zu mehreren Modulgruppen (MGR).
- zu einer oder zu mehreren Relationships (RLS). Vom RLS-Objekt sind folgende Beziehungen aufzubauen:
 1. Zu einem UTM-TAC (TAC) mit dem Beziehungstyp PPA (PEND PA).
 2. Zu einem oder zu mehreren Datensätzen (REC) und/oder Feldgruppen (FLG) und/oder Feldern (FLD) mit dem Beziehungstyp MPU (MPUT-Aufruf).
 3. Zu einem oder zu mehreren Modulen (MOD) und/oder zu einer oder zu mehreren Modulgruppen (MGR).
- zu einer oder zu mehreren Relationships (RLS) mit dem Standard-Beziehungstyp FOR (Format-Information). Vom RLS-Objekt sind folgende Beziehungen aufzubauen:
 1. Zu einem oder zu mehreren Formaten (MSK).
 2. Zu einem oder zu mehreren UTM-LTERM's (LTR).
- zu einem oder zu mehreren UTM-LTERM's (LTR)
- zu einem oder zu mehreren Formaten (MSK) mit dem Standard-Beziehungstyp FOR (Format-Information)
- zu einem oder zu mehreren UTM-LSSB's (LSS) mit dem Standard-Beziehungstyp STO (Speicher-Information)
- zu einem oder zu mehreren UTM-GSSB's (GSS) mit dem Standard-Beziehungstyp STO (Speicher-Information)
- zu einem oder zu mehreren UTM-TLS's (TLS) mit dem Standard-Beziehungstyp STO (Speicher-Information)
- zu einem oder zu mehreren Datensätzen (REC) mit dem Beziehungstyp KBB. (Datenübergabe über UTM-KB-Bereich)
- zu einer oder zu mehreren Feldgruppen (FLG) mit dem Beziehungstyp KBB. (Datenübergabe über UTM-KB-Bereich)
- zu einem oder zu mehreren Feldern (FLD) mit dem Beziehungstyp KBB. (Datenübergabe über UTM-KB-Bereich)

8.4.4 MGR - Modulgruppe

TR0350	function MODULE-GROUP	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change :
		Responsible :
		Text :
Attributes	:	1: . 2: . 3: . 4: . 5: . 6: . 7: . 8: .	
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
.
.
Command:			

Bild 128 Maske TR0350 für Objekttyp MGR

Beziehungen (Gliederung nach Teilstruktur-Katalogen)

Folgende Beziehungen können von einem MGR-Objekt ausgehen:

DVS/LEASY

- zu einer oder zu mehreren logischen Dateien (FIL)
- zu einem oder zu mehreren Datensätzen (REC)
- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einem oder zu mehreren Formaten (MSK)
- zu einem oder zu mehreren Copy-Elementen (INL)

SQL

- zu einer oder mehreren SQL-Basistabellen (RBT)
- zu einem oder mehreren SQL-Datenbank-Schemata (RSM)

Process/JCL

- zu einem oder zu mehreren Modulen (MOD)
- zu einem oder zu mehreren Modulen (MOD) und/oder zu einer oder zu mehreren Modulgruppen (MGR) mit dem Beziehungstyp VER (verarbeitet)

- zu einem oder zu mehreren Modulen (*MOD*) und/oder zu einer oder zu mehreren Modulgruppen (*MGR*) mit dem Beziehungstyp *CAL* (steuert an)
- zu einem oder zu mehreren Entry-Punkten (*ENP*)
- zu einem oder zu mehreren Call's (*CAL*)
- zu einer oder zu mehreren Jobvariablen (*JV*)

UDS

- zu einem oder zu mehreren Subschema-Records (*SBR*)
- zu einem UDS-Subschema (*SUB*)
- zu einer oder zu mehreren dynamischen Areas (*DAR*)
- zu einem oder zu mehreren Sets (*SET*)
- zu einem oder zu mehreren System-Sets (*SSY*)
- zu einem oder zu mehreren dynamischen Sets (*SDY*)
- zu einer oder zu mehreren Areas (*ARE*)

UTM

- zu einem UTM-Teilprogramm (*UTP*)
- zu einer oder zu mehreren Relationships (*RLS*). Vom *RLS*-Objekt sind folgende Beziehungen aufzubauen:
 1. Zu einem UTM-TAC (*TAC*) mit dem Beziehungstyp *FPU* (*FPUT*).
 2. Zu einem oder zu mehreren Modulen (*MOD*) und/oder zu einer oder zu mehreren Modulgruppen (*MGR*).
 3. Zu einem oder zu mehreren Datensätzen (*REC*) und/oder Feldgruppen (*FLG*) und/oder Feldern (*FLD*).
- zu einer oder zu mehreren Relationships (*RLS*). Vom *RLS*-Objekt sind folgende Beziehungen aufzubauen:
 1. Zu einem UTM-TAC (*TAC*) mit dem Beziehungstyp *PRE* (*PEND RE*) oder *PPR* (*PEND PR*) oder *PKP* (*PEND KP*).
 2. Zu einem oder zu mehreren Modulen (*MOD*) und/oder zu einer oder zu mehreren Modulgruppen (*MGR*).

- zu einer oder zu mehreren Relationships (*RLS*). Vom *RLS*-Objekt sind folgende Beziehungen aufzubauen:
 1. Zu einem UTM-TAC (*TAC*) mit dem Beziehungstyp *PPA* (*PEND PA*).
 2. Zu einem oder zu mehreren Datensätzen (*REC*) und/oder Feldgruppen (*FLG*) und/oder Feldern (*FLD*) mit dem Beziehungstyp *MPU* (*MPUT*-Aufruf).
 3. Zu einem oder zu mehreren Modulen (*MOD*) und/oder zu einer oder zu mehreren Modulgruppen (*MGR*).
- zu einer oder zu mehreren Relationships (*RLS*) mit dem Standard-Beziehungstyp *FOR* (Format-Information). Vom *RLS*-Objekt sind folgende Beziehungen aufzubauen:
 1. Zu einem oder zu mehreren Formaten (*MSK*).
 2. Zu einem oder zu mehreren UTM-LTERM's (*LTR*).
- zu einem oder zu mehreren UTM-LTERM's (*LTR*)
- zu einem oder zu mehreren Formaten (*MSK*) mit dem Standard-Beziehungstyp *FOR* (Format-Information)
- zu einem oder zu mehreren UTM-LSSB's (*LSS*) mit dem Standard-Beziehungstyp *STO* (Speicher-Information)
- zu einem oder zu mehreren UTM-GSSB's (*GSS*) mit dem Standard-Beziehungstyp *STO* (Speicher-Information)
- zu einem oder zu mehreren UTM-TLS's (*TLS*) mit dem Standard-Beziehungstyp *STO* (Speicher-Information)
- zu einem oder zu mehreren Datensätzen (*REC*) mit dem Beziehungstyp *KBB*. (Datenübergabe über UTM-KB-Bereich)
- zu einer oder zu mehreren Feldgruppen (*FLG*) mit dem Beziehungstyp *KBB*. (Datenübergabe über UTM-KB-Bereich)
- zu einem oder zu mehreren Feldern (*FLD*) mit dem Beziehungstyp *KBB*. (Datenübergabe über UTM-KB-Bereich)

8.4.5 CAL - Call

TR0340	function ENTRY-CALLED	\$userid	.nkz.TOMrfn.TOM.project				
Entry-called :				Creation date:		
Description :				Last change :		
				Responsible : ...		
				Text : .		
Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.							
Relationships to succeeding objects				Attributes for relationships			
Type	Name	Type	Occurs	1	2	3	4
.....
.....
.....
.....
.....
.....
.....
Command:							

Bild 129 Maske TR0340 für Objekttyp CAL

Den Objektnamen vom Typ CAL können Sie folgendermaßen angeben:

name1(name2)nnnn

name1

Name des Moduls oder Entry-Punkts, der gerufen wird;

name2

Name des rufenden Moduls, aus dem der Aufruf abgesetzt wird.

nnnn

vierstellige Zahl zur Qualifizierung; Die Zahl wird bei der Analyse mit TOM-CA automatisch hochgezählt.

Die Namen können maximal 13 Zeichen lang sein. Für die Namen von Modulgruppen, Modulen und Entry-Punkten können Sie auch eine Variable angeben:

varname(name)nnnn

Beziehungen

Folgende Beziehungen können von einem CAL-Objekt ausgehen:

- zu einem oder zu mehreren Datensätzen (REC)
- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einem oder zu mehreren Modulen (MOD)*
- zu einer oder zu mehreren Modulgruppen (MGR)*
- zu einem oder zu mehreren Entry-Punkten (ENP)*

* Diese Beziehungen werden durch Namensgleichheit von TOM-REF hergestellt. Dabei wird die Beziehung zum Entry-Punkt (ENP) vorrangig behandelt. Ist ein Entry-Punkt mit dem angegebenen Namen vorhanden und auch ein Modul, wird die Beziehung zum Entry-Punkt aufgebaut.

Beziehungstypen

Folgende Beziehungstypen können Sie den Beziehungen REC, FLD und FLG zuordnen:

- Zugriff auf die Quelldateien (REF), das heißt, im aufrufenden Teilprogramm wird der Parameter per Referenz übergeben.
- Zugriff auf Kopie der Quelldateien (CON), das heißt, im aufrufenden Teilprogramm wird der Parameter als Wert übergeben.

Diese Beziehungstypen sind ohne typspezifische Attribute.

8.4.6 ENP - Entry-Punkt

TR0330	function	ENTRY-INCLUDED	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
			Responsible	: ...
			Text	: .
Attributes	:	1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Command:	:		

Bild 130 Maske TR0330 für Objekttyp ENP

Beziehungen

Folgende Beziehungen können von einem ENP-Objekt ausgehen:

- zu einem oder zu mehreren Feldern (FLD)
- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Datensätzen (REC)

8.4.7 LIB - Bibliothek

```

TR7010  function LIBRARY                                $userid .nzk.TOMrfn.TOM.project
-----
Name      : .....                                Creation date: .....
Description : .....                                Last change  : .....
          : .....                                Responsible  : ...
          : .....                                Text        : .

Attributes : 1:.  2:.  3:.  4:.  5:.  6:.  7:.  8:.

Relationships to succeeding objects
Type          Name                                     Attributes for relationships
                                                    Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
-----

```

Bild 131 Maske TR7010 für Objekttyp LIB

Standardmäßig dürfen von einem LIB-Objekt keine Beziehungen ausgehen.

8.4.8 CBL - COBOL-Source

TR0400	function COBOL-SOURCE	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	Responsible	: ...
	Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Type	: . (E:ESQL, C:COBOL) [1]		
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type Occurs	1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
Command:		

Bild 132 Maske TR0400 für Objekttyp CBL

Attribut

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Type	C, E	Inhalt des Quellprogramms: C: COBOL-Anweisungen E: ESQL-Anweisungen

Beziehungen

Folgende Beziehungen können von einem CBL-Objekt ausgehen:

- zu keiner oder einer eingebetteten COBOL-Source (CBL)
- zu einem oder zu mehreren Nicht-Shareable-Objektmodulen (NOM)
- zu einem oder zu mehreren Shareable-Objektmodulen (SOM)
- zu einer oder zu mehreren Bibliotheken (LIB)

8.4.9 LMO - Lademodul

```

TR0440  function LOAD-MODULE                               $userid .nzk.TOMrfn.TOM.project
-----
Name      : .....                                          Creation date: .....
Description : .....                                          Last change  : .....
          : .....                                          Responsible   : ...
          : .....                                          Text         : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Relationships to succeeding objects
Type      Name                                           Attributes for relationships
          Name                                           Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
-----

```

Bild 133 Maske TR0440 für Objekttyp LMO

Beziehungen

Folgende Beziehungen können von einem LMO-Objekt ausgehen:

- zu einem oder zu mehreren Nicht-Shareable-Objektmodul (NOM)
- zu einem oder zu mehreren Shareable-Objektmodulen (SOM)
- zu einer oder zu mehreren Bibliotheken (LIB)

8.4.10 NOM - Nicht-Shareable Objektmodul

TR0450	function	NON-S.OBJ.-MOD.	\$userid	.nzk.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	:	Responsible	: ...
	:	Text	: .
Attributes	:	1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects			Attributes for relationships	
Type	Name		Type Occurs	1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
Command:				

Bild 134 Maske TR0450 für Objekttyp NOM

Beziehungen

Folgende Beziehungen können von einem NOM-Objekt ausgehen:

- zu einem oder zu mehreren UTM-TAC's (TAC)
- zu einer oder zu mehreren Bibliotheken (LIB)

8.4.11 SOM - Shareable-Objektmodul

```

TR0430  function SHARE-OBJ.-MOD.                $userid  .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
                                           Last change  : .....
Description : .....                               Responsible  : ...
                                           Text         : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Type: . (C:COBOL, E:ESQL, V:Verbindungsmodul-ESQL) [1]

Relationships to succeeding objects           Attributes for relationships
Type      Name                               Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
-----
Command: .....
-----

```

Bild 135 Maske TR0430 für Objekttyp SOM

Attribut

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Type	<u>C</u> , E, V	Art des mehrfach benutzbaren Objektmoduls: C: COBOL-Objektmodul E: ESQL-Objektmodul V: ESQL-Verbindungsmodul

Beziehungen

Folgende Beziehungen können von einem SOM-Objekt ausgehen:

- zu einer oder zu mehreren Bibliotheken (LIB)

8.4.12 PRC - DO-Prozedur

TR6000	function PROCEDURE	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
			Last change :
Description	:	Responsible : ...
		Text : .
Attributes	:	1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.	
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
.
Command:			

Bild 136 Maske TR6000 für Objekttyp PRC

Beziehungen

Folgende Beziehungen können von einem PRC-Objekt ausgehen:

- zu einer oder zu mehreren physikalischen Dateien (PFL)
- zu einer oder zu mehreren logischen Dateien (FIL)
- zu einer oder zu mehreren DO-Prozeduren (PRC)
- zu einer oder zu mehreren ENTER-Prozeduren (ENT)
- zu einem oder zu mehreren Programmen (PGM)
- zu einem oder zu mehreren Modulen (MOD)
- zu einer oder zu mehreren Modulgruppen (MGR)
- zu einer oder zu mehreren logischen Jobvariablen (JV)
- zu einer oder zu mehreren physikalischen Jobvariablen (PJV)
- zu einem oder zu mehreren Lademodulen (LMO)
- zu einem oder zu mehreren Nicht-Shareable-Objektmodulen (NOM)

8.4.13 ENT - ENTER-Prozedur

```
TR6040  function ENTER                                $userid .nkz.TOMrfn.TOM.project
-----
Name       : .....                                Creation date: .....
Description : .....                                Last change  : .....
          : .....                                Responsible  : ...
          : .....                                Text        : .
Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

    Relationships to succeeding objects              Attributes for relationships
    Type                      Name                  Type Occurs 1 2 3 4 5 6 7 8
    .....                    .....                .....
    .....                    .....                .....
    .....                    .....                .....
    .....                    .....                .....
    .....                    .....                .....
    .....                    .....                .....
    .....                    .....                .....
    .....                    .....                .....
    .....                    .....                .....
    .....                    .....                .....

Command: .....
-----
```

Bild 137 Maske TR6040 für Objekttyp ENT

Beziehungen

Folgende Beziehungen können von einem ENT-Objekt ausgehen:

- zu einer oder zu mehreren physikalischen Dateien (PFL)
- zu einer oder zu mehreren logischen Dateien (FIL)
- zu einer oder zu mehreren DO-Prozeduren (PRC)
- zu einer oder zu mehreren ENTER-Prozeduren (ENT)
- zu einem oder zu mehreren Programmen (PGM)
- zu einem oder zu mehreren Modulen (MOD)
- zu einer oder zu mehreren Modulgruppen (MGR)
- zu einer oder zu mehreren logischen Jobvariablen (JV)
- zu einer oder zu mehreren physikalischen Jobvariablen (PJV)
- zu einem oder zu mehreren Lademodulen (LMO)
- zu einem oder zu mehreren Nicht-Shareable-Objektmodulen (NOM)

8.4.15 PJV - Physikalische Jobvariable

```

TR6030  function PHYSICAL-JV                                $userid .nzk.TOMrfn.TOM.project
-----
Name      : .....                                      Creation date: .....
Description : .....                                      Last change : .....
           .....                                      Responsible : ...
           .....                                      Text       : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Relationships to succeeding objects              Attributes for relationships
Type              Name                          Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
-----
  
```

Bild 139 Maske TR6030 für Objekttyp PJV

Beziehungen

Folgende Beziehungen können von einem PJV-Objekt ausgehen:

- zu einer oder zu mehreren logischen Jobvariablen (JV)

8.4.16 Objekte des Funktionsentwurf-Teilstrukturkatalogs

Die Objekte des Funktionsentwurf-Teilstrukturkatalogs werden in einer Standard-Maske zur Verfügung gestellt, und zwar der Maske TR-5000, die auch für die benutzereigenen Objekte verwendet wird. Im Gegensatz zu den anderen Objektmasken muß in der Maske TR-5000 der Objekttyp explizit vor den Namen des Objekts eingetragen werden. Die Objekte sind zwar vordefiniert, besitzen aber keine typspezifischen Attribute und auch die (erweiterten) Beziehungen müssen Sie selbst anlegen.

Folgende Objekte sind zwar vorbelegt, aber nicht weiter spezifiziert, wobei n für Zahlen von 1-9 steht:

- CNn Konzept
- EFn Elementarfunktion
- EXn Ausgang
- FNn Funktion
- ICn Informationsbehälter
- IMn Eingang
- INn Information

Diese Objekttypen können Sie in keiner eigenen Maske anwählen, sondern erzeugen sie direkt über Kommando. Neben den einstelligen Attributen (1 bis 8) stehen zur Beschreibung eines Objekts noch mehrstellige Attribute zur Verfügung (9 bis 12), die Sie mit einer eigenen Bedeutung belegen müssen.

Beispiel

Das erste Konzept für ein Programm soll angelegt werden:

```
create comp=test,type=cn1
```

TR5000	Creation	\$userid	.nkz.TOMrfn.project								
Object :		Creation date:								
Description :		Last change :								
		Responsible :	...								
		Text :	.								
Attributes :			1:	2:	3:	4:	5:	6:	7:	8:	
9:										
10:										
11:										
12:										
Relationships to succeeding objects			Attributes for relationships								
Type	Name	Type	Occurs	1	2	3	4	5	6	7	8
.
.
.
.
.
.
.
Command:										

Bild 140 Maske TR5000 mit Ausgabebeispiel

Für die Modellierung bestimmter Strukturen benötigen Sie eventuell die Objekttypen `RET` und `RLS`. Ihre Funktion ist auf Seite 55 näher beschrieben. Bearbeiten Sie die Objekttypen mit dieser Maske, analog zu den benutzereigenen Objekttypen.



Die Objekte des Datenentwurf-Teilstrukturkatalogs sind nicht im Standard-Katalog enthalten und müssen explizit in das Data Dictionary eingebracht werden.

9 UTM

Eine UTM-Anwendung beschreiben Sie im Data Dictionary mit den Objekten und Beziehungen des UTM-Teilstrukturkatalogs. Eine UTM-Anwendung beschreiben heißt, daß Sie die nötigen Objekte und Beziehungen erstellen. In Kapitel 4 ist beschrieben, wie Sie Objekte und Beziehungen erstellen können.

9.1 UTM-Objekte

Folgende Objekttypen können Sie zur Beschreibung einer UTM-Anwendung benutzen. Die Tabelle zeigt die Objekttypen der UTM-Teilstruktur mit ihren Kurznamen, Namen und einer kurzen Erläuterung.

Kurzname	Langname des Objekttyps	Erläuterung
GSS	UTM-GSSB	Globaler Sekundärer Speicherbereich
LSS	UTM-LSSB	Lokaler Sekundärer Speicherbereich
LTR	UTM-LTERM	UTM-Datenstation
TAC	UTM-TAC	UTM-Transaktionscode
TLS	UTM-TLS	Terminalspezifischer Langzeitspeicher
UAP	UTM-APPLICATION	UTM-Anwendung
UTP	UTM-PROGRAM	UTM-Teilprogramm-Information

Tabelle 21 Objekttypen im UTM-Teilstrukturkatalog

9.2 UTM-Beziehungen

Zwischen welchen Objekten des UTM-Teilstrukturkatalogs Sie Beziehungen erstellen können, zeigt die folgende Tabelle. Dabei bedeutet ein X in einem Feld, daß zwischen zwei Objekttypen eine Beziehung bestehen kann; bei einem S handelt es sich um Standard-Beziehungstypen, deren Namen im Data Dictionary hinterlegt sind.

Objekttyp als Ausgang der Beziehung	Objekttyp als Ziel der Beziehung														
	FLD	FLG	GSS	LMO	LSS	LTR	MGR	MOD	MSK	PFL	REC	RLS	TAC	TLS	UTP
GSS UTM GSSB	X	X									X				
LSS UTM LSSB	X	X									X				
MGR Module Group			S		S	X			S			S		S	X
MOD Module			S		S	X			S			S		S	X
NOM Non S.Obj.Mod.													X		
RLS Relationship	X	X				X	X	X	X		X		X		
TAC UTM TAC	X														
TLS UTM TLS	X	X									X				
UAP UTM Applicat.				X						X					

Tabelle 22 Beziehungstypen des UTM-Teilstrukturkatalogs

Die Standard-Beziehungstypen für UTM sind `STO` und `FOR`. Den Beziehungstyp `STO` erstellen Sie zwischen Objekten vom Typ `MOD` oder `MGR` und Objekten vom Typ `GSS` oder `LSS`, den Beziehungstyp `FOR` zwischen Objekttypen `MOD` oder `MGR` und `MSK` oder `RLS`. Wie Sie diese Beziehungen erstellen, ist in Kapitel 4 beschrieben.

9.3 UTM-Teilstrukturkatalog

Der UTM-Teilstrukturkatalog ist die Basis für die Beschreibung einer UTM-Anwendung im Data Dictionary. Er legt fest, welches Objekt das Ausgangsobjekt und welches das Zielobjekt einer Beziehung sein muß. Die Pfeile zeigen vom Typ des Ausgangsobjekts zum Typ des Zielobjekts. Eine Beziehung müssen Sie immer in die Beziehungsliste des Ausgangsobjekts eintragen. Die Elementaraussagen klassifizieren die Beziehungen näher.

Eine genaue Beschreibung der in diesem Modell verwendeten Darstellungsmittel finden Sie ab Seite 7.

- a) siehe JCL-Teilstruktur
- b) siehe Programmdesign-Teilstruktur
- c) siehe Programmrealisierung-Teilstruktur
- e) siehe DVS/Leasy-Teilstruktur
- f) siehe UDS-Teilstruktur

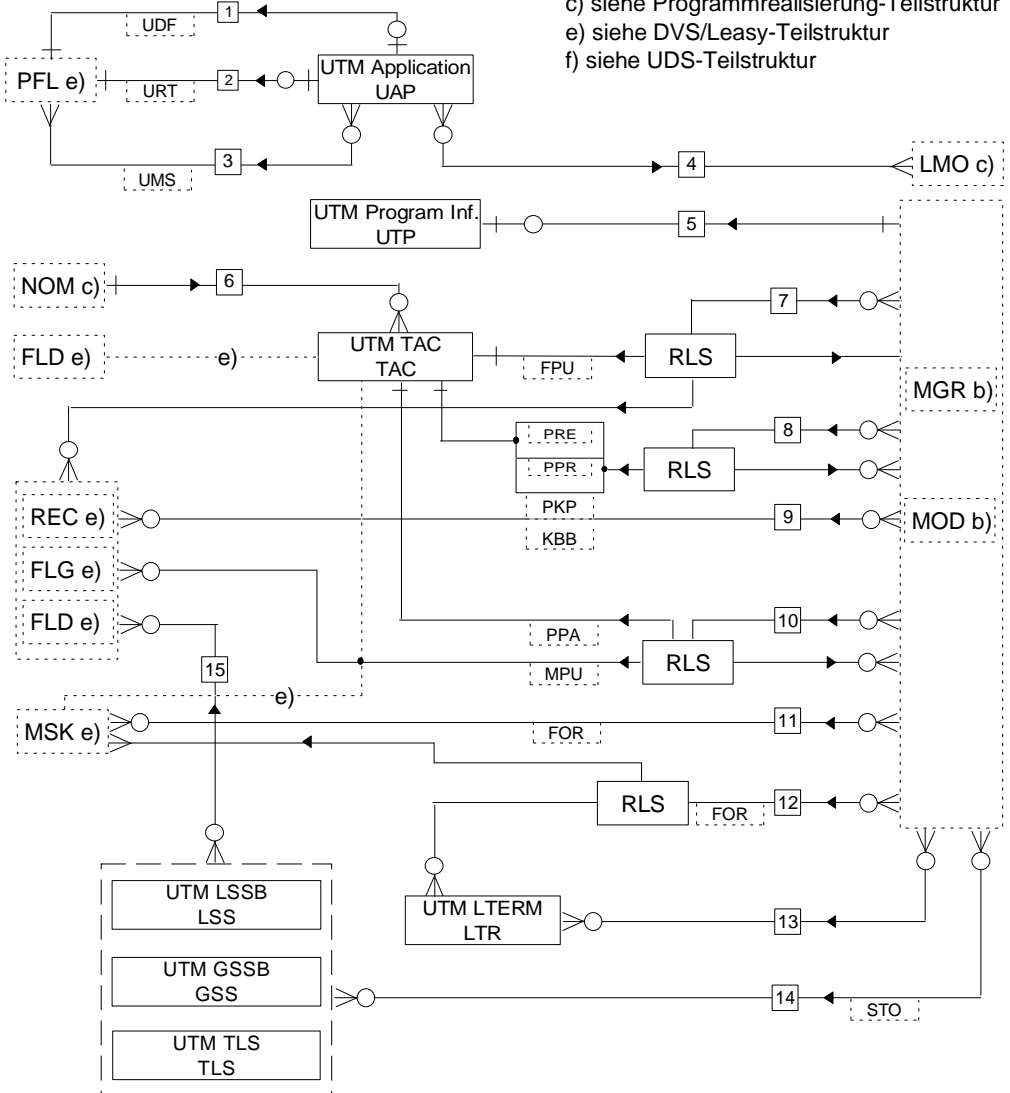


Bild 141 UTM-Teilstrukturkatalog

Die Wertigkeit des Beziehungstyps (1:1, 1:n oder n:m) ist in diesem Entity-Relationship-Diagramm zwar enthalten, aber sie wird in der aktuellen Version von TOM-REF nicht unterstützt. Sie können allerdings anhand dieses Teilstrukturkatalogs für korrekte Wertigkeit sorgen.

Elementaraussagen

1	Eine UTM-Anwendung (UAP)	benutzt	als Source-Datei für KDCDEF (Beziehungstyp: UDF)	eine	PFL
	Eine physische Datei (PFL)	wird benutzt	dto.	von keiner oder einer	UAP
2	Eine UTM-Anwendung (UAP)	benutzt	als Source-Datei für KDCROOT (Beziehungstyp: URT)	eine	PFL
	Eine physische Datei (PFL)	wird benutzt	dto.	von keiner oder einer	UAP
3	Eine UTM-Anwendung (UAP)	benutzt	als Source-Datei für die Meldungen (Beziehungstyp: UMS)	eine oder mehrere	PFL
	Eine physische Datei (PFL)	wird benutzt	dto.	von keiner, einer oder mehreren	UAP
4	Eine UTM-Anwendung (UAP)	enthält		eine oder mehrere	LMO
	Ein Lademodul (LMO)	ist enthalten		in keiner, einer oder mehreren	UAP
5	Einem Modul (MOD) und/oder einer Modulgruppe (MGR)	sind zugeordnet		keine oder eine	UTP
	Eine UTM-Teilprogramm-Information (UTP)	gehört		zu einem	MOD und/oder MGR
6	Einem Nicht-Shareable Objektmodul (NOM)	sind zugeordnet		keine, eine oder mehrere	TAC
	Ein UTM-Transaktionscode (TAC)	gehört		zu einem	NOM

7	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	steuert an	über FPUT (Beziehungstyp: FPU) mit einem UTM TAC (TAC) und einem Datensatz (REC) und/oder einer Feldgruppe (FLG) und/oder einem Feld (FLD)	keinen, einen oder mehrere	MOD und/oder MGR
	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	wird angesteuert	dto.	von keinem, einem oder mehreren	MOD und/oder MGR
8	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	steuert an	über PEND RE (Beziehungstyp: PRE) oder PEND PR (Beziehungstyp: PPR) oder PEND KP (Beziehungstyp: PKP) mit einem UTM TAC TAC	keinen, einen oder mehrere	MOD und/oder MGR
	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	wird angesteuert	dto.	von keinem, einem oder mehreren	MOD und/oder MGR
9	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	enthält	zur Datenübergabe über den KB-Bereich (Beziehungstyp: KBB)	keinen, einen oder mehrere	REC und/oder FLG und/oder FLD
	Ein Datensatz (REC) und/oder eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist enthalten	dto.	in keinem, einem oder mehreren	MOD und/oder MGR
10	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	steuert an	über PEND PA (Beziehungstyp: PPA) mit einem UTM TAC (TAC) und Datenübergabe über einen MPUT-Aufruf (Beziehungstyp: MPU) mit 0/1/n Datensatz (REC) und/oder mit 0/1/n Feldgruppe (FLG) und/oder mit 0/1/n Feld (FLD)	keinen, einen oder mehrere	MOD und/oder MGR

10	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	wird angesteuert	dto.	von keinem, einem oder mehreren	MOD und/oder MGR
11	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	benutzt	mit spezifischen Attributen (Beziehungstyp: FOR)	keine, eine oder mehrere	MSK
	Ein Format (MSK)	wird benutzt	dto.	von keiner, einer oder mehreren	MOD und/oder MGR
12	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	benutzt	mit spezifischen Attributen (Beziehungstyp: FOR) und mit einer oder mehreren Formaten (MSK)	kein, ein oder mehrere	LTR
	Ein UTM LTERM (LTR)	wird benutzt	dto.	von einem, keinem oder mehreren	MOD und/oder MGR
13	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	benutzt	ohne Format (MSK)	kein, ein oder mehrere	LTR
	Ein UTM LTERM (LTR)	wird benutzt	dto.	von einem, keinem oder mehreren	MOD und/oder MGR
14	Ein Modul (MOD) und/oder einer Modulgruppe (MGR)	benutzt	mit spezifischen Attributen (Beziehungstyp: STO)	kein, ein oder mehrere	LSS und/oder GSS und/oder TLS
	Ein UTM LSSB (LSS) und/oder UTM GSSB (GSS) und/oder UTM TLS (TLS)	wird benutzt	dto.	von keinem, einem oder mehreren	MOD und/oder MGR
15	Ein UTM LSSB (LSS) und/oder UTM GSSB (GSS) und/oder UTM TLS (TLS)	enthält		kein, ein oder mehreren	REC und/oder FLG und/oder FLD
	Ein Datensatz (REC) und/oder eine Feldgruppe (FLG) und/oder ein Feld (FLD)	ist enthalten		in keinem, einem oder mehreren	LSS und/oder GSS und/oder TLS

9.4 Sub-Menü UTM

Die Objekttypen des UTM-Teilstrukturkatalogs erstellen und bearbeiten Sie in den objekttyp- und beziehungstyp-spezifischen Masken. Diese Masken erreichen Sie über das Sub-Menü UTM in der Auswahlmaske TR0001.

Das Sub-Menü UTM erhalten Sie, wenn Sie in der Maske TR0001 (siehe Seite 177) den Kennbuchstaben *s* auswählen.

```

TR8000 Selection UTM-Types                               $userid .nkz.TOMrfn.TOM.project
-----
Please select object-types/relationship-types and functions (e.g. c2 )

Object-types:
a . UTM-Program (UTP)           i . UTM-TLS (TLS)
b . UTM-TAC (TAC)              j . UTM-LTERM (LTR)
c . UTM-Application (UAP)
d . UTM-GSSB (GSS)
e . UTM-LSSB (LSS)

Relationship-types:
g . Storage-info (STO)         o . Format-info (FOR)
h . Storage-info ext.         p . Format-info ext.

Functions for object-types/relationship-types:
1 . Display                    2 . Create                3 . Change                4 . Delete
Functions without types:
5 . Services

-----
Command: .....
-----

```

Bild 142 Auswahlmaske TR8000 für UTM-Typen

In den folgenden Unterabschnitten werden erst die Objektmasken und dann die Beziehungsmasken (*STO* und *FOR*) in der Reihenfolge der Auswahlmaske TR8000 beschrieben.

Sie können auch aus der Maske TR8000 in die Auswahlmaske TR9000 für die Servicefunktionen wechseln (Kennziffer 5).

9.4.1 UTP - UTM-Teilprogramm

```

TR8010  function UTM-PROGRAM                                $userid .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
Description : .....                               Last change  : .....
          : .....                               Responsible  : ...
          : .....                               Text        : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

KB-length      : .[1]. (Byte)           [6] Number of LSSBs      : ....
SPAB-length    : .[2]. (Byte)           [7] Number of GSSBs     : ....
Terminal-message-length : .[3]. (Byte)   [8] Number of TLS-calls : ....
Message-area-length : .[4]. (Byte)
Exit           : .[5] (S:Start,H:Shut,F:Form)

Relationships to succeeding objects           Attributes for relationships
Type      Name                               Type Occurs 1 2 3 4 5 6 7 8
.....
: .....
: .....
: .....
-----
Command: .....
-----
    
```

Bild 143 Maske TR8010 für Objekttyp UTP

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	KB-length	max. 32767	Länge des benötigten KB-Programmbereichs in Byte.
2	SPAB-length	max. 32767	Länge des benötigten Standard Primären Arbeitsbereichs in Byte.
3	Terminal-message-length	max. 32767	Benötigte FORMIO-Länge in Byte.
4	Message-area-length	max. 32767	Benötigte FORMUSR-Länge in Byte.

Nr.	Bezeichnung	Wertebereich	Bedeutung
5	Exit	S H F	Benutzerausgang: S: Laden einer Task H: Beenden einer Task F: Formatierungsroutine
6	Number of LSSBs	Ganzzahl	Anzahl der LSSBs; Vorbelegung bei CREATE: 0000
7	Number of GSSBs	Ganzzahl	Anzahl der GSSBs; Vorbelegung bei CREATE: 0000
8	Number of TLS-calls	Ganzzahl	Anzahl der TLS-Aufrufe; Vorbelegung bei CREATE: 0000

Beziehungen

Standardmäßig können von einem UTP-Objekt keine Beziehungen ausgehen.

9.4.2 TAC - UTM-TAC

TR8020	function UTM-TAC	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	:	Responsible	: ...
	:	Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
TAC-type	: . (D:Dialog,A:Asynchron) [1]		
Call-usage	: . (B:Both,F:First,N:Next) [2]		
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
:	:
:	:
:	:
:	:
:	:
Command:			

Bild 144 Maske TR8020 für Objekttyp TAC

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	TAC-type	D A	Typ des TAC's: D: TAC für Dialogprogramm. A: TAC für Asynchron-Programm
2	Call-usage	B F N	F: TAC für 1. Dialogschritt. N: TAC für Folgeprogramm. B: TAC für 1. Dialogschritt oder Folgeprogramm.

Beziehungen

Folgende Beziehung kann von einem TAC-Objekt ausgehen:

- zu einem Feld (FLD)

9.4.3 UAP - UTM-Anwendung

```

TR8030 function UTM-APPLICATION          $userid .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
Description : .....                               Last change  : .....
          : .....                               Responsible  : ...
          : .....                               Text        : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

System-environment : . (B:BS2000/S:SINIX) [1]

Relationships to succeeding objects      Attributes for relationships
Type      Name                          Type Occurs 1 2 3 4 5 6 7 8
-----
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
-----
    
```

Bild 145 Maske TR8030 für Objekttyp UAP

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	System-environment	B S	Systemumgebung: B: BS2000; S: SINIX

Beziehungen

Folgende Beziehungen können von einem UAP-Objekt ausgehen:

- zu einer physikalischen Datei (PFL) mit dem Beziehungstyp UDF.
(Source-Datei für KDCDEF)
- zu einer physikalischen Datei (PFL) mit dem Beziehungstyp URT.
(Source-Datei für KDCROOT)
- zu einer oder zu mehreren physikalischen Dateien (PFL) mit dem Beziehungstyp UMS. (Source-Datei für die Meldungen)
- zu einem oder zu mehreren Lademodulen (LMO)

9.4.4 GSS - UTM-GSSB

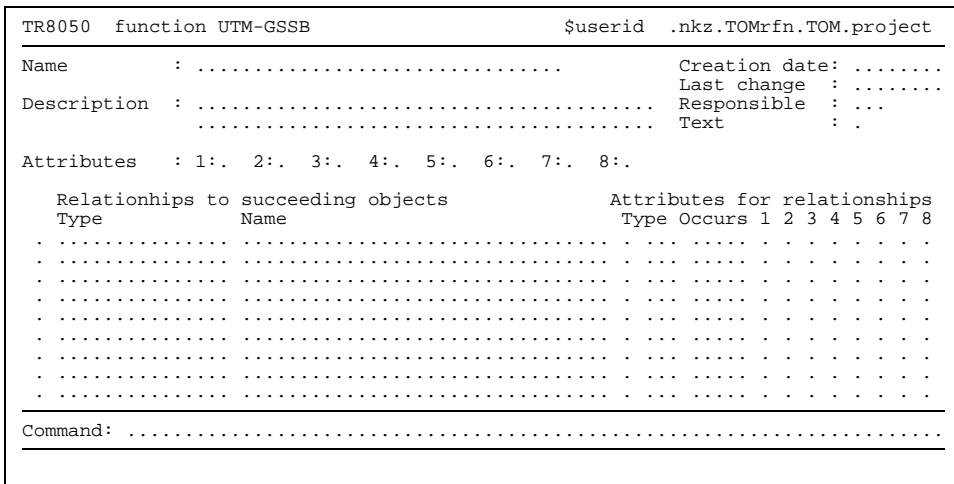


Bild 146 Maske TR8050 für Objekttyp GSS

Beziehungen

Folgende Beziehungen können von einem GSS-Objekt ausgehen:

- zu einem oder zu mehreren Feldern (FLD)
- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Datensätzen (REC)

9.4.5 LSS - UTM-LSSB

TR8060	function UTM-LSSB	\$userid	.nzk.TOMrfn.TOM.project									
Name	:	Creation date:								
Description	:	Last change	:								
		Responsible	: ...								
		Text	: .								
Attributes	:	1:.	2:.	3:.	4:.	5:.	6:.	7:.	8:.			
Relationships to succeeding objects			Attributes for relationships									
Type	Name		Type	Occurs	1	2	3	4	5	6	7	8
.
.
.
.
.
.
.
.
.
.
Command:											

Bild 147 Maske TR8060 für Objekttyp LSS

Beziehungen

Folgende Beziehungen können von einem LSS-Objekt ausgehen:

- zu einem oder zu mehreren Feldern (FLD)
- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Datensätzen (REC)

9.4.6 TLS - UTM-TLS

TR8070	function UTM-TLS	\$userid	.nzk.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	:	Responsible	: ...
		Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
Command:			

Bild 148 Maske TR8070 für Objekttyp TLS

Beziehungen

Folgende Beziehungen können von einem TLS-Objekt ausgehen:

- zu einem oder zu mehreren Feldern (FLD)
- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Datensätzen (REC)

9.4.7 LTR - UTM-LTERM

```

TR8040  function UTM-LTERM                               $userid  .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
Description : .....                               Last change  : .....
          : .....                               Responsible  : ...
          : .....                               Text        : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Terminal-type: . (D:Dialog/O:Output) [1]

Relationships to succeeding objects
Type      Name                                         Attributes for relationships
Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
    
```

Bild 149 Maske TR8040 für Objekttyp LTR

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Terminal-type	<u>D</u> O	Typ der UTM-Datenstation: D: Dialogstation O: Ausgabestation

Beziehungen

Standardmäßig dürfen von einem LTR-Objekt keine Beziehungen ausgehen.

9.4.8 STO - vordefinierter Beziehungstyp Speicher-Information

Eine STO-Beziehung kann zwischen Objekten folgender Typen existieren:

- Ausgangsobjekt: Modul (MOD) oder Modulgruppe (MGR).
- Zielobjekt: UTM-GSSB (GSS) oder UTM-LSSB (LSS) oder UTM-TLS (TLS).

```

TR8080  function relationship                $userid  .nkz.TOMrfn.TOM.project
-----
                                         Creation date: .....
                                         Last change  : .....
Description : .....                      Responsible  : ...
                                         .....

Relationship from object
Type        Name
.....

to succeeding object                      Attributes for relationships
Type        Name                          Type Occurs 1 2 3 4 5 6 7 8
.....      * STO .....

Input length : ..... (Byte) [1]
Output length : ..... (Byte) [2]

-----
Command: .....
-----
    
```

Bild 150 Maske TR8080 für Beziehungstyp STO

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Input length	max. 32767	Eingabelänge
2	Output length	max. 32767	Ausgabelänge

9.4.9 FOR - vordefinierter Beziehungstyp Format-Information

Eine FOR-Beziehung kann zwischen Objekten folgender Typen existieren:

- Ausgangsobjekt: Modul (MOD) oder Modulgruppe (MGR).
- Zielobjekt: Format (MSK) oder Relationship (RLS).

```

TR8090  function relationship                                $userid  .nkz.TOMrfn.TOM.project
-----
                                         Creation date: .....
                                         Last change  : .....
Description : ..... Responsible : ...
                                         .....

Relationship from object
Type        Name
.....

to succeeding object                                Attributes for relationships
Type        Name                                Type Occurs 1 2 3 4 5 6 7 8
.....                                * FOR .....

Format-Mode      : . (*,+,#,-) [1]
Direction       : . (I:Input,O:Output) [2]
Message part or end : .. (NT:Part of message,NE:End of message) [3]

-----
Command: .....
    
```

Bild 151 Maske TR8090 für Beziehungstyp FOR

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Format-Mode	*, + , #, -	Art des Formats
2	Direction	I O	I: Eingabeformat O: Ausgabeformat
3	Message part or end	NT NE	NT: Teilnachricht NE: Letzte Teilnachricht bzw. Gesamtnachricht.

10 UDS

Eine UDS-Datenbank beschreiben Sie im Data Dictionary mit den Objekten und Beziehungen des UDS-Teilstrukturkatalogs. Eine UDS-Datenbank beschreiben heißt, daß Sie die nötigen Objekte und Beziehungen erstellen. In Kapitel 4 ist beschrieben, wie Sie Objekte und Beziehungen erstellen können.

10.1 UDS-Objekte

Folgende Objekttypen können Sie zur Beschreibung einer UDS-Datenbank benutzen: Die Tabelle zeigt die Objekttypen der UDS-Teilstruktur mit ihren Kurznamen, Namen und einer kurzen Erläuterung.

Kurzname	Langname des Objekttyps	Erläuterung
ALI	ALIAS	Synonym
ARE	UDS-AREA	Area bzw. Realm der Datenbank
DAR	UDS-DYN.AREA	Dynamische Area bzw. Realm der Datenbank
DBK	UDS-SET-KEY	Set-Key eines Sets
DBR	UDS-DB-RECORD	Datensatz in einem Datenbankschema
DBS	UDS-DB-SCHEMA	Schema einer Datenbank
LMC	UDS-CALC-KEY	Calc-Key eines Datenbanksatzes
LMD	UDS-DIRECT-KEY	Direct-Key eines Datenbanksatzes
SBR	UDS-SUBSCH-REC	Datensatz in einem Subschema
SDY	UDS-DYNAMIC-SET	Dynamischer Set eines dynamischen Realm

Tabelle 23 Objekttypen im UDS-Teilstrukturkatalog

Kurzname	Langname des Objekttyps	Erläuterung
SEA	UDS-SEARCH-KEY	Search-Key eines Datenbanksatzes oder eines Sets
SET	UDS-SET	Set in einem Datenbankschema (kein System-Set)
SSL	UDS-SSL-SCHEMA	SSL-Schema einer Datenbank
SSY	UDS-SYSTEM-SET	System-Set in einem Datenbankschema
SUB	UDS-SUBSCHEMA	Subschema einer Datenbank
UDS	UDS-DATABASE	UDS-Datenbank

Tabelle 23 Objekttypen im UDS-Teilstrukturkatalog (Fortsetzung)

10.2 UDS-Beziehungen

Zwischen welchen Objekten des UDS-Teilstrukturkatalogs Sie Beziehungen erstellen können, zeigt die folgende Tabelle. Dabei bedeutet ein X in einem Feld, daß zwischen zwei Objekttypen eine Beziehung bestehen kann.

Objekttyp als Ausgang der Beziehung	Objekttyp als Ziel der Beziehung																							
	ALI	ARE	DAR	DBK	DBR	DBS	FLD	FLG	LMC	LMD	MOD	PFL	RBT	RIX	RSM	SBR	SDY	SEA	SET	SSL	SSY	SUB		
ALI Alias	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
ARE Area					X							X												
DAR dynam. Area												X												
DBK Set Key							X	X																
DBR DB Record							X	X	X	X								X						
DBS DB Schema		X	X		X												X		X		X	X		
LMC Calc Key							X	X			X													
LMD Direct Key							X																	
SBR Subschema Rec.							X	X					X											
SEA Search Key							X	X			X			X										
SET Set	X			X	X		X											X						
SSL SSL-Schema						X																		
SSY System Set				X	X		X											X						
SUB Subschema		X	X												X	X	X		X		X			
UDS Database						X						X								X		X		

Tabelle 24 Beziehungstypen des UDS-Teilstrukturkatalogs

10.3 UDS-Teilstrukturkatalog

Der UDS-Teilstrukturkatalog ist die Basis für die Beschreibung einer UDS-Datenbank im Data Dictionary. Er legt fest, welches Objekt das Ausgangsobjekt und welches das Zielobjekt einer Beziehung sein muß. Die Pfeile zeigen vom Typ des Ausgangsobjekts zum Typ des Zielobjekts. Eine Beziehung müssen Sie immer in die Beziehungsliste des Ausgangsobjekts eintragen. Die Elementaraussagen klassifizieren die Beziehungen näher.

Eine genaue Beschreibung der in diesem Modell verwendeten Darstellungsmittel finden Sie ab Seite 7.

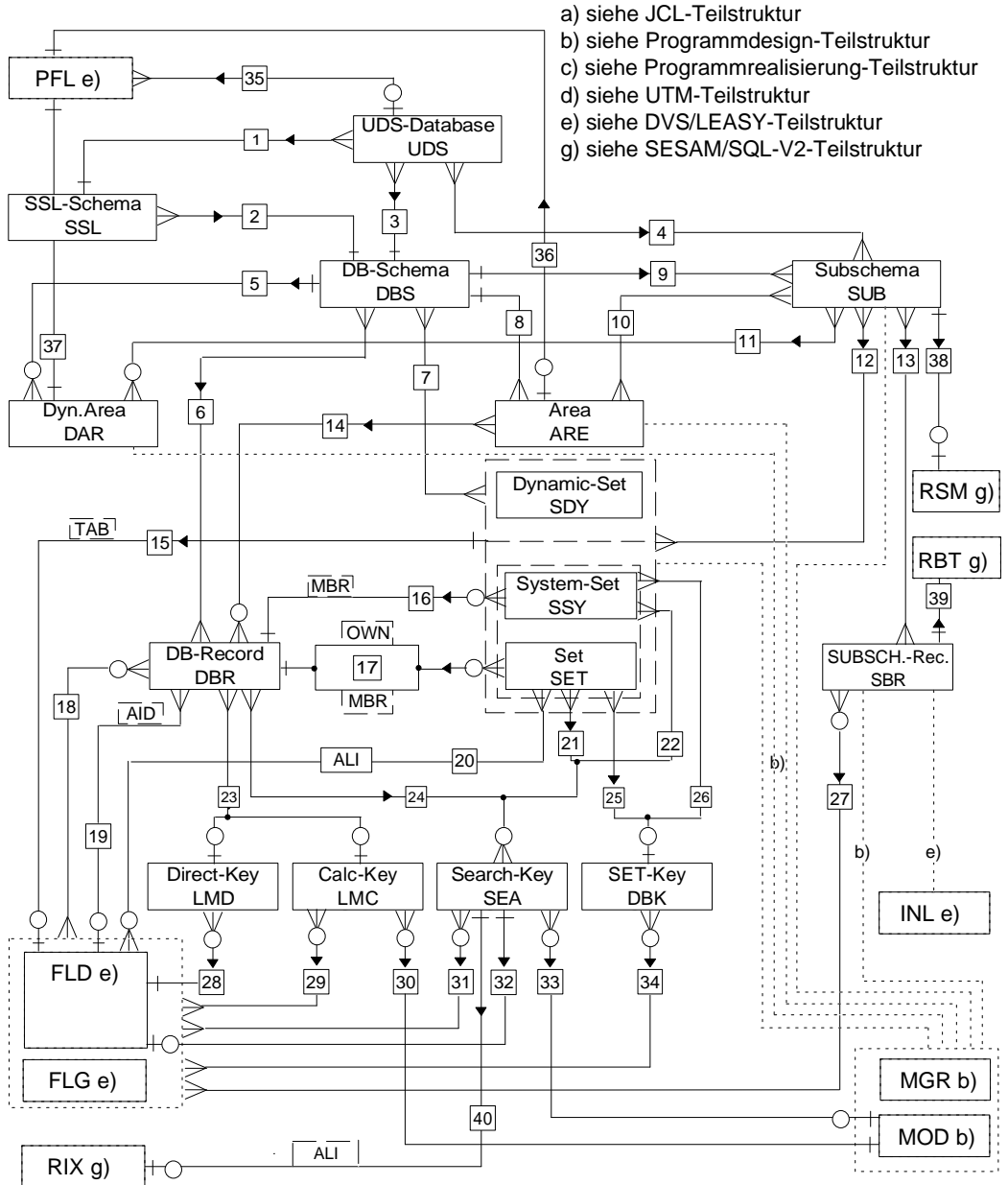


Bild 152 UDS-Teilstrukturkatalog (Stand: V2.0)

Die Wertigkeit des Beziehungstyps (1:1, 1:n oder n:m) ist in diesem Entity-Relationship-Diagramm zwar enthalten, aber sie wird in der aktuellen Version von TOM-REF nicht unterstützt. Sie können allerdings anhand dieses Teilstrukturkatalogs für korrekte Wertigkeit sorgen.

Im Teilstrukturkatalog sind auch Beziehungen enthalten, die sich gegenseitig ausschließen, wie z.B. UDS DB Record enthält einen UDS Directory key oder einen UDS Calc-key. Diese Bedingung wird ebenfalls von TOM-REF derzeit nicht unterstützt.

Elementaraussagen

1	Eine UDS-DB (UDS)	besitzt	eine	SSL
	Eine SSL (SSL)	gilt für	eine oder mehrere	UDS
2	Ein DB-Schema (DBS)	besitzt	eine oder mehrere	SSL
	Eine SSL (SSL)	gilt für	ein	DBS
3	Eine UDS-DB (UDS)	besitzt	ein	DBS
	Ein DB-Schema (DBS)	gilt für	eine oder mehrere	UDS
4	Eine UDS-DB (UDS)	besitzt	ein oder mehrere	SUB
	Ein Subschema (SUB)	gilt für	eine oder mehrere	UDS
5	Ein DB-Schema (DBS)	enthält	keine, eine oder mehrere	DAR
	Eine dyn. Area (DAR)	ist enthalten	in einem	DBS
6	Ein DB-Schema (DBS)	besitzt	eine oder mehrere	DBR
	Ein DB-Record (DBR)	ist enthalten	in einem oder mehreren	DBS
7	Ein DB-Schema (DBS)	enthält	einen oder mehrere	SET und/oder SSY und/oder SDY
	Ein Set (SET) und/oder System-Set (SSY) und/oder dynamischer Set (SDY)	ist enthalten	in einem oder mehreren	DBS

8	Ein DB-Schema (DBS)	enthält	eine oder mehrere	ARE
	Eine Area (ARE)	ist enthalten	in einem	DBS
9	Ein DB-Schema (DBS)	enthält	ein oder mehrere	SUB
	Eine Subschema (SUB)	ist enthalten	einem	DBS
10	Ein Subschema (SUB)	enthält	eine oder mehrere	ARE
	Eine Area (ARE)	ist enthalten	in einem oder mehreren	SUB
11	Ein Subschema (SUB)	enthält	keine, eine oder mehrere	DAR
	Eine dyn. Area (DAR)	ist enthalten	in einem oder mehreren	SUB
12	Ein Subschema (SUB)	enthält	einen oder mehrere	SET und/ oder SSY und/oder SDY
	Ein Set (SET) und/oder System-Set (SSY) und/oder dynamischer SET (SDY)	ist enthalten	in einem oder mehreren	SUB
13	Ein Subschema (SUB)	enthält	einen oder mehrere	SBR
	Ein Subschema-Record (SBR)	ist enthalten	in einem oder mehreren	SUB
14	Eine Area (ARE)	enthält	kein, ein oder mehrere	DBR
	Ein DB-Record (DBR)	ist enthalten	in einer oder mehreren	ARE
15	Ein System-Set (SSY) und/oder ein Set (SET)	enthält als Tabellennamen (TAB)	kein oder ein	FLD
	Ein Feld (FLD)	enthält als Tabellennamen (TAB)	in einem	SSY und/ oder SET
16	Ein System-Set (SSY)	enthält als Member (MBR)	einen	DBR
	Ein DB-Record (DBR)	ist enthalten als Member (MBR)	in keinem, einem oder mehreren	SSY

17	Ein Set (SET)	enthält als Owner (OWN) oder Member (MBR)	einen	DBR
	Ein DB-Record (DBR)	ist enthalten als Owner (OWN) oder Member (MBR)	in keinem, einem oder mehreren	SET
18	Ein DB-Record (DBR)	enthält	einen oder mehrere	FLD und/oder FLG
	Ein Feld (FLD) und/oder eine Feldgruppe (FLG)	ist enthalten	in keinem, einem oder mehreren	DBR
19	Ein DB-Record (DBR)	enthält als Identifier (AID)	kein oder ein	FLD
	Ein Feld (FLD)	ist enthalten als Area-Identifier (AID)	in einem oder mehreren	DBR
20	Ein Set (SET)	enthält als Alias-Namen (ALI)	kein, ein oder mehrere	FLD
	Ein Feld (FLD)	ist enthalten als Alias-Namen (ALI)	in einem oder mehreren	SET
21	Ein Set (SET)	enthält	keinen, einen oder mehrere	SEA
	Ein Search-Key (SEA)	ist enthalten	in einem oder mehreren	SET oder 22 oder 24
22	Ein System-Set (SSY)	enthält	keinen, einen oder mehrere	SEA
	Ein Search-Key (SEA)	ist enthalten	in einem oder mehreren	SSY oder 21 oder 24

23	Ein DB-Record (DBR)	enthält	keinen oder einen	LMD oder LMC
	Ein Direct-Key (LMD) oder ein Calc-Key (LMC)	ist enthalten	in einem oder mehreren	DBR
24	Ein DB-Record (DBR)	enthält	keinen, einen oder mehrere	SEA
	Ein Search-Key (SEA)	ist enthalten	in einem oder mehreren	DBR oder 21 oder 22
25	Ein Set (SET)	enthält	keinen oder einen	DBK
	Ein Set-Key (DBK)	ist enthalten	in einem oder mehreren	SET oder 26
26	Ein System-Set (SSY)	enthält	keinen oder einen	DBK
	Ein Set-Key (DBK)	ist enthalten	in einem oder mehreren	SSY oder 25
27	Ein Subschema-Record (SBR)	enthält	eine oder mehrere	FLD und/oder FLG
	Ein Feld (FLD) und/oder eine Feldgruppe (FLG)	ist enthalten	in keinem, einem oder mehreren	SBR
28	Ein Direct-Key (LMD)	enthält	ein	FLD
	Ein Feld (FLD)	ist enthalten	in keinem, einem oder mehreren	LMD
29	Ein Calc-Key (LMC)	enthält	ein oder mehrere	FLD und/oder FLG
	Ein Feld (FLD) und/oder eine Feldgruppe (FLG)	ist enthalten	in keinem, einem oder mehreren	LMC
30	Ein Calc-Key (LMC)	verwendet	einen	MOD
	Ein Hash-Modul (MOD)	wird verwendet	von keinem, einem oder mehreren	LMC

31	Ein Search-Key (SEA)	enthält	ein oder mehrere	FLD und/ oder FLG
	Ein Feld (FLD) und/ oder eine Feldgruppe (FLG)	ist enthalten	in keinem, einem oder mehreren	SEA
32	Ein Search-Key (SEA)	enthält als Tabellennamen (TAB)	ein oder mehrere	FLD
	Ein Feld (FLD)	ist enthalten als Tabellen- namen (TAB)	einem	SEA
33	Ein Search-Key (SEA)	verwendet	keinen oder einen	MOD
	Ein Hash-Modul (MOD)	wird ver- wendet	von keinem, einem oder mehreren	SEA
34	Ein Set-Key (DBK)	enthält	ein oder mehrere	FLD und/ oder FLG
	Ein Feld (FLD) und/ oder eine Feldgruppe (FLG)	ist enthalten	in keinem, einem oder mehreren	DBK
35	Eine UDS-Datenbank (UDS)	belegt	mehrere (genau 3: COSSD, CATALOG, CATALOG.SAVE)	PFL
	Eine physische Datei (PFL)	wird belegt	von keiner oder einer	UDS
36	Eine Area (ARE)	belegt	eine	PFL
	Eine physische Datei (PFL)	wird belegt	von einer oder keiner	ARE
37	Eine dyn. Area (DAR)	belegt	eine	PFL
	Eine physische Datei (PFL)	wird belegt	von einer oder keiner	DAR

38	Ein Subschema (SUB)	Tabellenname hat als Alias (ALI)	ein oder kein	RSM
	Ein SQL-Schema (RSM)	wird als Alias (ALI) referenziert	von einem	SUB
39	Ein System-Set (SSY) und/oder ein Set (SET)	hat als Alias (ALI)	ein oder kein *)	FLD
	Ein Datenfeld bzw. eine Spalte (FLD)	wird als Alias (ALI) referenziert	von einem	SSY und/oder SET
40	Ein Search-Key (SEA)	hat als Alias (ALI)	ein oder kein	RIX
	Ein SQL-Index (RIX)	wird als Alias (ALI) referenziert	von einem	SEA
41	Eine Subschema-Satzart (SBR)	hat als Alias (ALI)	ein oder kein	RBT
	Eine SQL-Basistabelle (RBT)	wird als Alias (ALI) referenziert	von einem	SBR

*) nicht sinnvoll bei System-Set mit Mandatory-Automatic

10.4 Sub-Menü UDS

Die Objekttypen des UDS-Teilstrukturkatalogs erstellen und bearbeiten Sie in den objekttyp- und beziehungstyp-spezifischen Masken. Diese Masken erreichen Sie über das Sub-Menü UTM in die Auswahlmaske TR1000.

Das Sub-Menü UTM erhalten Sie, wenn Sie in der Maske TR0001 (siehe Seite 177) den Kennbuchstaben *t* auswählen.

```

TR1000 Selection UDS-Types                               $userid .nkz.TOMrfn.TOM.project
-----
Please select object-types and functions (e.g. e2,a3,5)

Object-types:
a . UDS-Database (UDS)   h . UDS-Area (ARE)       l . UDS-Set (SET)
b . UDS-SSL-schema (SSL) i . UDS-Dyn.area (DAR)   m . UDS-System-set (SSY)
c . UDS-DB-schema (DBS) d . UDS-DB-record (DBR)  j . UDS-Subschema (SUB)  o . UDS-Dynamic-set (SDY)
e . UDS-DB-record (DBR) k . UDS-Subsch-rec (SBR) p . UDS-Set-key (DBK)
f . UDS-Search-key (SEA)
g . UDS-Search-key (SEA)
r . Alias (ALI)
s . Other

Functions for object-types:
1 . Display              2 . Create                3 . Change                4 . Delete

Functions without object-types:
5 . Services

Command: .....

```

Bild 153 Auswahlmaske TR1000 für UDS-Typen

Die einzelnen typspezifischen Masken sind in der Reihenfolge der Auswahlmaske TR1000 in den folgenden Unterabschnitten beschrieben. Die Maske TR5010 für den Objekttypen *Alias* (Kennbuchstabe *r*) ist im auf Seite 202 beschrieben, die Maske TR5000 für *Other*-Typen (Kennbuchstabe *s*) auf Seite 203.

Ferner ist es möglich, aus der Maske TR1000 in die Auswahlmaske TR9000 für die Servicefunktionen zu wechseln (Kennziffer *5*).

10.4.1 UDS - UDS-Datenbank

TR1010	function UDS-DATABASE	\$userid	.nkz.TOMrfn.TOM.project
DB-Name	:	Creation date:
Description	:	Last change	:
	:	Responsible	: ...
	:	Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
Command:			

Bild 154 Maske TR1010 für Objekttyp UDS

Beziehungen

Folgende Beziehungen können von einem UDS-Objekt ausgehen:

- zu einer Storage Structure Language (SSL)
- zu einem oder zu mehreren Subschemata (SUB)
- zu einem Datenbank-Schema (DBS)
- zu drei physischen Dateien (PFL): COSSD, CATALOG, CATALOG.SAVE

10.4.2 SSL - Storage Structure Language

TR1030	function	UDS-SSL-SCHEMA	\$userid	.n kz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	:	Responsible	:
	:	Text	:
Attributes	:	1:.	2:.	3:.
	:	4:.	5:.	6:.
	:	7:.	8:.	
Relationships to succeeding objects			Attributes for relationships	
Type	Name		Type Occurs	1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
<hr/>				
Command:			

Bild 155 Maske TR1030 für Objekttyp SSL

Beziehungen

Folgende Beziehung kann von einem SSL-Objekt ausgehen:

- zu einem Datenbank-Schema (DBS)

10.4.3 DBS - Datenbank-Schema

TR1020	function	UDS-DB-SCHEMA	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
		Responsible	: ...
		Text	: .
Attributes	:	1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects			Attributes for relationships	
Type		Name	Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
.
.
.
.
.
Command:				

Bild 156 Maske TR1020 für Objekttyp DBS

Beziehungen

Folgende Beziehungen können von einem DBS-Objekt ausgehen:

- zu einem oder zu mehreren Datenbank-Records (DBR)
- zu einem oder zu mehreren Sets (SET)
- zu einem oder zu mehreren System-Sets (SSY)
- zu einem oder zu mehreren Dynamischen Sets (SDY)
- zu einem oder zu mehreren Dynamischen Areas (DAR)
- zu einem oder zu mehreren Areas (ARE)
- zu einem oder zu mehreren Subschemas (SUB)

10.4.4 DBR - Datenbank-Record

```

TR1060 function UDS-DB-RECORD                                $userid .nkz.TOMrfn.TOM.project
-----
Name      : .....                                          Creation date: .....
Description : .....                                          Last change  : .....
           : .....                                          Responsible  : ...
           : .....                                          Text        : .

Attributes : 1:.  2:.  3:.  4:.  5:.  6:.  7:.  8:.

Population actual  : ..[1]... (Number of rows)
Population changed : ..[2]... (Number of rows)

Relationships to succeeding objects          Attributes for relationships
Type Name                                     Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
-----
    
```

Bild 157 Maske TR1060 für Objekttyp DBR

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Population actual	Ganzzahl	Anzahl der Tupel
2	Population changed	Ganzzahl	Anzahl der Tupel (Eintrag nur bei Änderung)

Beziehungen

Folgende Beziehungen können von einem DBR-Objekt ausgehen:

- zu einem oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einem Feld (FLD) mit dem Beziehungstyp AID (Area Identifier). Das Feld muß mit der Länge von 30 Zeichen definiert sein (PIC X(30)).
- zu einem Direct-Key (LMD)
- zu einem Calc-Key (LMC)
- zu einem oder zu mehreren Search-Keys (SEA)

10.4.5 LMD - Direct-Key

TR1070	function UDS-DIRECT-KEY	\$userid	.nzk.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	Responsible	: ...
	Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
.
Command:			

Bild 158 Maske TR1070 für Objekttyp LMD

Beziehungen

Folgende Beziehung kann von einem LMD-Objekt ausgehen:

- zu einem Feld (FLD)

10.4.6 LMC - Calc-Key

TR1080	function UDS-CALC-KEY	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	:	Responsible	: ...
		Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Duplicates	: . (Y,N) [1]		
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
Command:			

Bild 159 Maske TR1080 für Objekttyp LMC

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Duplicates	Y N	Gleiche Schlüsselwerte sind erlaubt: Y: Ja; N: Nein

Beziehungen

Folgende Beziehungen können von einem LMC-Objekt ausgehen:

- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einem Hash-Modul (MOD)

10.4.7 SEA - Search-Key

```

TR1090  function UDS-SEARCH-KEY                $userid  .nkz.TOMrfn.TOM.project
-----
Name      : .....                               Creation date: .....
Description : .....                               Last change  : .....
          : .....                               Responsible  : ...
          : .....                               Text        : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Key type   : . (Index, Calc) [1]
Duplicates : . (Y,N) [2]          Density : ... [3]

Relationships to succeeding objects          Attributes for relationships
Type      Name                               Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
-----
Command: .....
-----
    
```

Bild 160 Maske TR1090 für Objekttyp SEA

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Key type	I C	Typ des Search-Keys: I: Index C: Calc
2	Duplicates	Y N	Gleiche Schlüsselwerte sind erlaubt: Y: Ja; N: Nein
3	Density	Ganzzahl	Füllungsgrad in Prozent

Beziehungen

Folgende Beziehungen können von einem SEA-Objekt ausgehen:

- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einem Hash-Modul (MOD)
- zu einem oder zu mehreren Feldern (FLD) mit dem Beziehungstyp TAB. Die bezogenen Felder sind die Grundlage einer Search-Key-Tabelle.
- zu einer SQL-Index-Beschreibung (RIX) als Alias (Beziehungstyp ALI)

10.4.8 ARE - Area

TR1050	function UDS-AREA	\$userid	.nzk.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	Responsible	: ...
		Text	: .
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.		
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type Occurs	1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
Command:			

Bild 161 Maske TR1050 für Objekttyp ARE

Beziehungen

Folgende Beziehungen können von einem ARE-Objekt ausgehen:

- zu einem oder zu mehreren Datenbank-Records (DBR)
- zu einer Physischen Datei (PFL)

10.4.9 DAR - Dynamische Area

```

TR1150  function UDS-DYN.-AREA                                $userid  .nkz.TOMrfn.TOM.project
-----
Name      : .....                                         Creation date: .....
                                         Last change  : .....
Description : .....                                         Responsible  : ...
                                         Text         : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

  Relationships to succeeding objects                        Attributes for relationships
  Type                    Name                              Type Occurs 1 2 3 4 5 6 7 8
  .....                   .....                            .....
  .....                   .....                            .....
  .....                   .....                            .....
  .....                   .....                            .....
  .....                   .....                            .....
  .....                   .....                            .....
  .....                   .....                            .....
  .....                   .....                            .....
  .....                   .....                            .....
  .....                   .....                            .....

-----
Command: .....
-----
    
```

Bild 162 Maske TR1150 für Objekttyp DAR

Beziehungen

Folgende Beziehung kann von einem DAR-Objekt ausgehen:

- Zu einer Physischen Datei (PFL)

10.4.10 SUB - Subschema

TR1040	function UDS-SUBSCHEMA	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change	:
	:	Responsible	: ...
	:	Text	: .
Attributes	:	1:.	2:.
	:	3:.	4:.
	:	5:.	6:.
	:	7:.	8:.
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.
.
.
.
.
.
.
.
.
Command:			

Bild 163 Maske TR1040 für Objekttyp SUB

Beziehungen

Folgende Beziehungen können von einem SUB-Objekt ausgehen:

Zu folgenden Objekttypen sind Beziehungen möglich:

- zu einem oder zu mehreren Areas (ARE)
- zu einem oder zu mehreren System-Sets (SSY)
- zu einem oder zu mehreren Dynamischen Sets (SDY)
- zu einem oder zu mehreren Dynamischen Areas (DAR)
- zu einem oder zu mehreren Sets (SET)
- zu einem oder zu mehreren Subschema-Records (SBR)
- zu einem SQL-Schema (RSM)



Wenn Sie bei der DDL-Generierung mit UDS-DDA die Subschema-Parameter PRIVACY LOCK FOR COMPILE und PRIVACY LOCK FOR COPY für die DDL-Source benötigen, dann müssen Sie sie in die generierte Source eintragen.

10.4.11 SBR - Subschema-Record

TR1120	function UDS-SUBSCH-REC	\$userid	.nkz.TOMrfn.TOM.project								
Name	:	Creation date	:								
Description	:	Last change	:								
	:	Responsible	: ...								
:		Text	: .								
Attributes	: 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.										
Relationships to succeeding objects		Attributes for relationships									
Type	Name	Type	Occurs	1	2	3	4	5	6	7	8
.
.
.
.
.
.
.
.
.
.
Command:											

Bild 164 Maske TR1120 für Objekttyp SBR

Beziehungen

Folgende Beziehungen können von einem SBR-Objekt ausgehen:

- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)
- zu einer SQL-Basistabelle (RBT) als Alias (Beziehungstyp ALI)

10.4.12 SET - SET

TR1100	function UDS-SET	\$userid	.nkz.TOMrfn.TOM.project
Name	:	Creation date:
Description	:	Last change :
	:	Responsible : ...
	:	Text : .
Attributes	:	1: . 2: . 3: . 4: . 5: . 6: . 7: . 8: .	
Order is [1]	:	. (First, Last, Next, Prior, Immaterial, Sorted, sorted indexEd)	
Duplicates	:	. (Y,N) [2]	
Insert-mode	:	. (Automatic, Manual) [3]	
Connection	:	. (Mandatory, Optional) [4]	
Set-selection:	:	. (Current of set, Location mode of owner) [5]	
Relationships to succeeding objects		Attributes for relationships	
Type	Name	Type	Occurs 1 2 3 4 5 6 7 8
.....
.....
.....
Command:			

Bild 165 Maske TR1100 für Objekttyp SET

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Order is	F(IRST) L(AST) N(EXT) P(RIOR) I(MMATERIAL) S(ORTED) X =(SORTED INDEXED)	Die ORDER-Klausel dient dazu, die logische Reihenfolge zu bestimmen, die die Membersätze in der Set-Occurrence eines Sets einnehmen sollen.
2	Duplicates	Y(ES) N(O)	Der DUPLICATES-Zusatz legt fest, ob innerhalb des Sets Sätze mit gleichem Schlüssel zugelassen sind.
3	Insert-mode	A(UTOMATIC) M(ANUAL)	In dem INSERT-MODE-Teil der MEMBER-Klausel wird festgelegt, ob der Membersatz sofort beim Abspeichern in die Set-Occurrence abgelegt wird oder durch eine DML-Anweisung.

Nr.	Bezeichnung	Wertebereich	Bedeutung
4	Connection	M(ANDATORY) O(PTIONAL)	In dem CONNECTION-Teil der MEMBER-Klausel wird festgelegt, ob die Bindung eines bereits eingefügten Member-satzes an einen Ownersatz starr ist oder vom Programmierer je nach Wunsch herstellbar.
5	Set-selection	C(URRENT OF SET) L(OCATION MODE OF OWNER)	In der SET-OCCURENCE-SELECTION-Klausel wird die Set-Occurence eines Sets für den Zugriff auf die einzelnen Sätze festgelegt.

Beziehungen

Folgende Beziehungen können von eine SET-Objekt ausgehen:

- zu einem Datenbank-Record (DBR) entweder als owner (Beziehungstyp = OWN) oder als member (Beziehungstyp = MBR)
- zu einem oder zu mehreren Search-Keys (SEA)
- zu keinem oder zu einem Set-Key (DBK)
- zu einem oder mehreren Alias-Objekten (ALI), von denen aus dann eine Beziehung zu einem Feld-Objekt (FLD) aufgebaut werden muß.
- zu einem oder mehreren Feldern(FLD), die die Namen der Indextabellen tragen (Beziehungstyp = TAB), falls solche Namen verwendet werden
- zu einem Feld (FLD) als Alias (Beziehungstyp ALI)

10.4.13 SSY - System-Set

```

TR1130  function UDS-SYSTEM-SET                                $userid  .nkz.TOMrfn.TOM.project
-----
Name      : .....                                         Creation date: .....
                                         Last change  : .....
Description : .....                                         Responsible  : ...
                                         Text         : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Order is [1] : . (First, Last, Next, Prior, Immaterial, Sorted, sorted indexEd)
Duplicates   : . (Y,N) [2]
Insert-mode  : . (Automatic, Manual) [3]
Connection   : . (Mandatory, Optional) [4]

Relationships to succeeding objects
Type         Name                                           Attributes for relationships
Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....

Command: .....
-----

```

Bild 166 Maske TR1130 für Objekttyp SSY

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Order is	F(IRST) L(AST) N(EXT) P(RIOR) I(MMATERIAL) S(ORTED) X = (SORTED INDEXED)	Die ORDER-Klausel dient dazu, die logische Reihenfolge zu bestimmen, die die Mitgliedsätze in der Set-Occurrence eines Sets einnehmen sollen.

Nr.	Bezeichnung	Wertebereich	Bedeutung
2	Duplicates	Y(ES) N(O)	Der DUPLICATES-Zusatz legt fest, ob innerhalb des Sets Sätze mit gleichem Schlüssel zugelassen sind.
3	Insert-mode	A(UTOMATIC) M(ANUAL)	In dem INSERT-MODE-Teil der MEMBER-Klausel wird festgelegt, ob der Membersatz sofort beim Abspeichern in die Set-Occurrence abgelegt wird oder durch eine DML-Anweisung.
4	Connection	M(ANDATORY) O(PTIONAL)	In dem CONNECTION-Teil der MEMBER-Klausel wird festgelegt, ob die Bindung eines bereits eingefügten Membersatzes an einen Ownersatz starr ist oder vom Programmierer je nach Wunsch herstellbar.

Beziehungen

Folgende Beziehungen können von eine SSY-Objekt ausgehen:

- zu einem Datenbank-Record (DBR) als member (Beziehungstyp = MBR)
- zu einem oder zu mehreren Search-Keys (SEA)
- zu einem Set-Key (DBK)
- zu einem Feld (FLD) als Alias (Beziehungstyp ALI)
- zu einem oder mehreren Feldern mit dem Beziehungstyp TAB. Die Felder sind Grundlage einer Indextabelle.

10.4.14 SDY - Dynamischer Set

```

TR1140  function UDS-DYNAMIC-SET                                $userid .nkz.TOMrfn.TOM.project
-----
Name          : .....                                          Creation date: .....
Description   : .....                                          Last change  : .....
               : .....                                          Responsible   : ...
               : .....                                          Text         : .
Attributes    : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Relationships to succeeding objects                               Attributes for relationships
Type          Name                                              Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....
. ....

Command: .....
-----

```

Bild 167 Maske TR1140 für Objekttyp SDY

Beziehungen

Standardmäßig dürfen von einem SDY-Objekt keine Beziehungen ausgehen.

10.4.15 DBK - Set-Key

```

TR1110 function UDS-SET-KEY                                $userid .nzk.TOMrfn.TOM.project
Name      : ..... Creation date: .....
Description : ..... Last change : .....
           : ..... Responsible : ...
           : ..... Text : .

Attributes : 1:. 2:. 3:. 4:. 5:. 6:. 7:. 8:.

Indexed   : . (Y,N) [1]
Key-type  : ... (DB-key, DEFINED keys) [2]
Duplicates : . (Y,N) [3]
Sorted    : . (Ascending, Descending) [4]

Relationships to succeeding objects      Attributes for relationships
Type Name                               Type Occurs 1 2 3 4 5 6 7 8
. ....
. ....
. ....
. ....

Command: .....
    
```

Bild 168 Maske TR1110 für Objekttyp DBK

Attribute

Nr.	Bezeichnung	Wertebereich	Bedeutung
1	Indexed	Y(ES) N(O)	Der INDEXED-Zusatz in der ORDER-Klausel besagt, daß ein Direktzugriff über Tabellen durchgeführt wird.
2	Key-type	DB-key= DATABASE-KEY DEFINED KEYS	Der KEY-TYPE-Zusatz der ORDER-Klausel beschreibt den Typ des Datenbankschlüssels.
3	Duplicates	Y(ES) N(O)	Der DUPLICATES-Zusatz der ORDER-Klausel legt fest, ob mehrfache Schlüsselwerte zugelassen sind.
4	Sorted	A(SCENDING) D(ESCENDING)	Der SORTED-Zusatz der ORDER-Klausel legt fest, ob die Membersätze der Set-Occurrences aufsteigend oder absteigend nach dem Datenbankschlüssel sortiert sind

Beziehungen

Folgende Beziehungen können von einem DBK-Objekt ausgehen:

- zu einer oder zu mehreren Feldgruppen (FLG)
- zu einem oder zu mehreren Feldern (FLD)

11 Sonderfälle der Datenmodellierung

Im folgenden Kapitel sind einige Besonderheiten beschrieben, die Sie bei der Beschreibung von COBOL-Programmen beachten sollten.

11.1 Modulverbindungen modellieren

Die Sprachmittel von COBOL lassen offen, ob der Adressat einer CALL-Anweisung ein Modul oder der Entry eines Moduls ist. Eine solche fallweise Verbindung kann also nicht statisch, sondern sie muß dynamisch zum Abfragezeitpunkt aufgebaut werden.

Verbindungen von Objekttypen

MOD → CAL ... > ENP ← MOD → CAL ... > MOD →

Beispiel

A.MOD → E1.CAL ... > E1.ENP ← B.MOD → CAL ... > C.MOD

Erreicht die Pfadverfolgung ausgehend von einem Objekt mit dem Typ MOD ein Objekt mit dem Typ CAL, so prüft sie, ob ein gleichnamiges Objekt vom Typ ENP oder ersatzweise vom Typ MOD existiert. Wird ein solches Objekt gefunden, dann wird von diesem Objekt aus der Pfad weiterverfolgt.

Diese Regel ist speziell für den Objekttyp CAL definiert, falls von diesem keine weitere Beziehung in der bisherigen Wegrichtung angeboten wird. Sie wird nicht aktiviert, wenn eine solche Beziehung angelegt wurde.

Beispiele

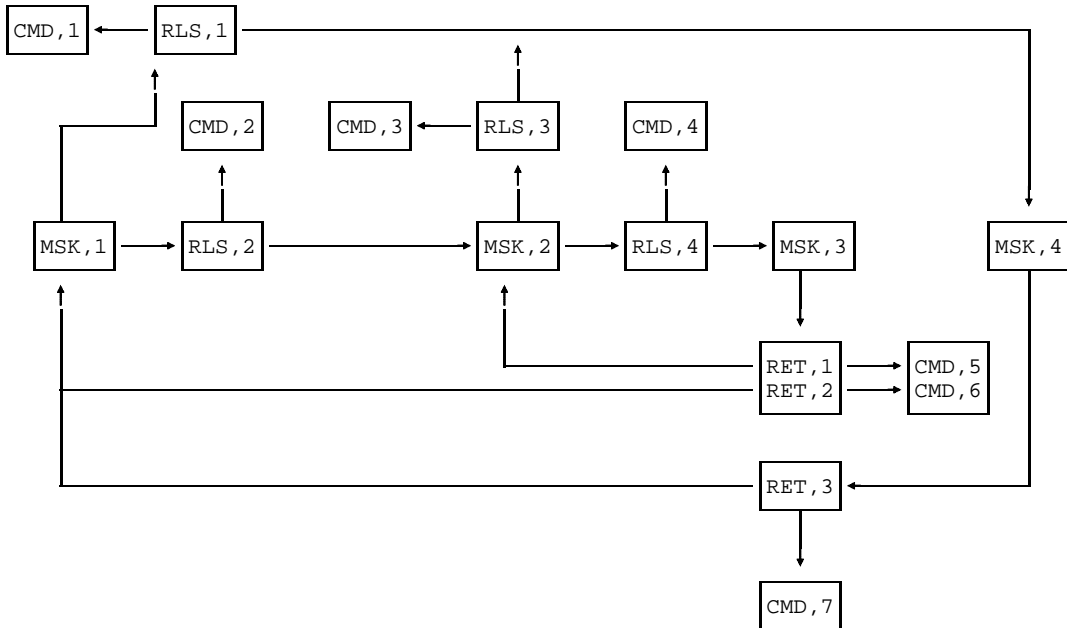
A.MOD → E1.CAL ... > E1.ENP ← B.MOD

A.MOD → B.CAL ... > B.MOD

... > bedeutet, daß diese Verbindungen nicht explizit eingegeben werden, TOM-REF jedoch intern bekannt sind.

11.2 Dialogstruktur modellieren

Die **Dialogstruktur** verstanden als Maskenfolge einer Anwendung in Abhängigkeit von Eingaben des Anwenders (in der Regel Kommandos) kann wie folgt dargestellt werden:



Legende:

MSK = Maske
 RLS/RETURN = Beziehungstyp
 CMD = Kommando/Taste

Bild 169 Modell einer Dialogstruktur

Der Aufruf einer Maske erfolgt über ein Objekt vom Typ `RLS`. An dieses Objekt sind Kommandos (Objekttyp `CMD`) gekettet, die sozusagen den Übergang vom `RLS`-Objekt zur nachfolgenden Maske kontrollieren. Mit dem Objekttyp `RET` können auch Rückverweise an weiter voranliegende Masken definiert werden.

Im Beispiel (siehe obiges Bild) gelangt der Anwender aus der Maske 1 mit Kommando 1 zur Maske 4 und mit Kommando 2 zur Maske 2. Von Maske 4 aus ist nur ein Rücksprung zur Maske 1 vorgesehen (mit Kommando 7). Maske 2 erlaubt den Übergang zu den Masken 3 und 4 (mit den Kommandos 4 und 3), jedoch keinen Rücksprung zur Maske 1. Von Maske 3 aus gelangt man wahlweise zurück zur Maske 2 (Kommando 5) oder zur Maske 1 (Kommando 6).

11.3 Integrierte Modul- und Dialogstruktur modellieren

Dialogstruktur und Modulaufrufstruktur sind als Teil der Systemstruktur miteinander integriert. Sie folgen unterschiedlichen Regeln. Die Modulaufrufstruktur wird nach den Standard-Linkage-Konventionen hierarchisch gesteuert: ein gerufenes Modul gibt beim Return die Steuerung an das aufrufende Modul zurück. Die Maskenaufrufsteuerung dagegen erlaubt einen prinzipiell beliebigen Wechsel zwischen Masken in Abhängigkeit von impliziten oder expliziten Anweisungen (Kommandos).

Die Modellierung erfolgt demgemäß in zwei Schichten:

Schicht 1: Maskenfolgen

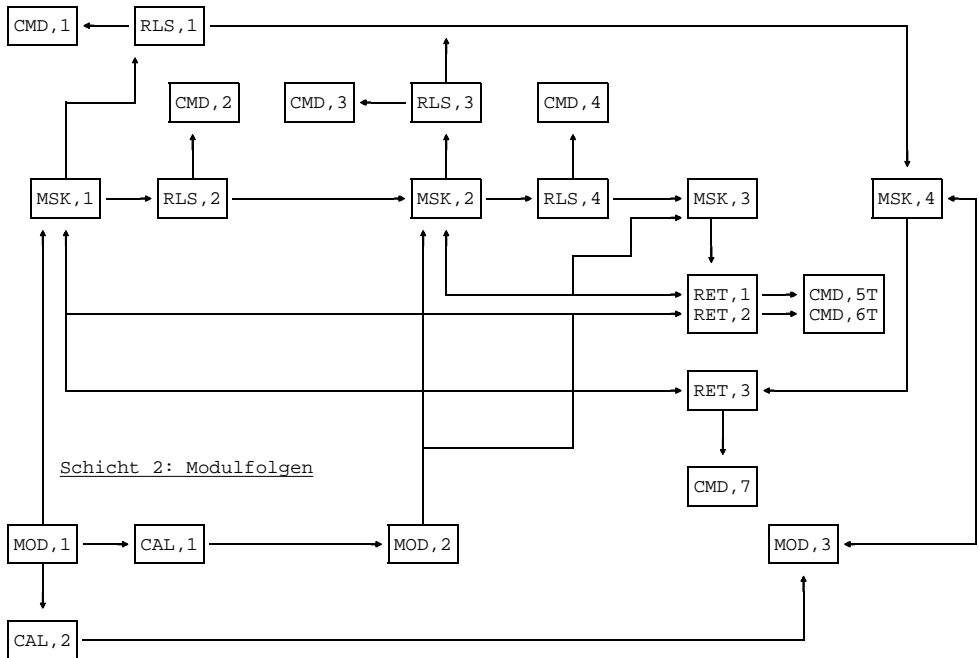


Bild 170 Modell einer integrierten Modul- und Dialogstruktur

Zu den Modulen 1 und 3 gehören die Masken 1 und 4, während das Modul 2 mit den Masken 2 und 3 verbunden ist.

12 TOM-REF Kommandos

Sie können Kommandos direkt in die Kommandozeile eingeben und damit die Menüsteuerung umgehen. TOM-REF führt zuerst diejenigen Kommandos aus, die im Kommandomodus in der Kommandozeile angegeben sind. Danach werden jene Kommandos ausgeführt, die durch Markierungen in den Masken aufgerufen werden.

Die Kommandos können nach den geltenden TOM-Konventionen auch abgekürzt eingegeben werden. Die Abkürzung muß jedoch eindeutig einem Kommando zuzuordnen sein, sonst folgt eine Fehlermeldung.

Die Kommandos können in der Auswahlmaske TR0001, in einem Sub-Menü oder auch in eine leere Objektmaske eingegeben werden, sofern diese Maske durch Ankreuzen oder durch Kurzkommando erreicht wurde.



Sie dürfen Kommandos nicht schachteln, aber miteinander verketteten.

Die Kommandos sind in diesem Kapitel in alphabetischer Reihenfolge beschrieben. Wenn die Kommandos auch in anderen als den angegebenen Masken aufgerufen werden können, ist dies gesondert angegeben. Die Darstellungsmittel finden Sie auf Seite 7.

CHANGE - Objekt oder Beziehung ändern

Mit dem Kommando `CHANGE` rufen Sie die Maske TR0010 auf, in der Sie ein bestehendes Objekt oder eine bestehende (erweiterte) Beziehung ändern können. Das Kommando `CHANGE` zählt zu den Grundfunktionen und bezieht sich auf die typneutralen und typspezifischen Attribute, die Beziehungsliste und den voreingestellten Text eines Objekts. Den Text können Sie nur bearbeiten, wenn die Textbearbeitung eingeschaltet ist.

```

CHANGE {
  COMPONENT=objektname ,TYPE=objekttyp
  COMPONENT=objektname
  TYPE=objekttyp
  RELATIONSHIP [EXTENDED] [ , TYPE=beziehungstyp]
}

```

COMPONENT=objektname
Name des Objekts

TYPE=objekttyp
Der objekttyp kann in Kurz- oder Langbezeichnung eingegeben werden.

RELATIONSHIP [EXTENDED]
Es ist zu unterscheiden, ob der gesamte Operand `RELATIONSHIP EXTENDED` oder nur `RELATIONSHIP` angegeben wird: `RELATIONSHIP EXTENDED` ist nur bei erweiterten Beziehungstypen möglich, es wird nur die Erweiterung der Beziehung bearbeitet. `RELATIONSHIP` ist bei allen Beziehungstypen möglich, es wird die gesamte Beziehung bearbeitet.

TYPE=beziehungstyp
Falls der Operand bei erweiterten Beziehungstypen angegeben wird: `beziehungstyp` muß einer der Standardnamen für erweiterte Beziehungstypen sein. (`STO`, `FOR`, `XBT`, `XIX`) Die Angabe des Operanden bewirkt, daß die Auswahlmaske für Beziehungstypen TR0008 übersprungen wird. Bei einfachen Beziehungen ist `SIR` (simple relationship) anzugeben, falls die Maske TR0008 übersprungen werden soll.

Falls bereits während des laufenden TOM-REF-Betriebs ein Objekt bearbeitet wurde, kann bei der Kommandoeingabe gegebenenfalls auf Operanden verzichtet werden. Fehlende Operanden interpretiert TOM-REF folgendermaßen:

1. `CHANGE COMPONENT=objektname`
Der zuletzt bearbeitete Objekttyp wird beibehalten.
2. `CHANGE TYPE=objekttyp`
Der Name des zuletzt bearbeiteten Objekts wird beibehalten.

3. CHANGE

Typ und Name des zuletzt bearbeiteten Objekts werden beibehalten.

Beispiele

```
CHANGE C=auftrag, T=RBT
```

Die SESAM-SQL V2-Basistabelle AUFTRAG soll geändert werden.

```
CHANGE C=kunden
```

Direkt nach der Bearbeitung der Basistabelle AUFTRAG eingegeben, wird die SESAM-SQL V2-Basistabelle KUNDEN angezeigt.

```
CH REL,TYPE=SIR
```

Es wird eine leere Beziehungsmaske ausgegeben, in die Sie Name und Typ von Ausgangs- und Zielobjekt eingeben müssen.

Siehe auch: CREATE, DELETE, DISPLAY

CHECK - Gegen den Strukturkatalog prüfen

Mit dem Kommando `CHECK` können Sie Strukturen, die mit älteren TOM-REF-Versionen erstellt wurden, gegen den Strukturkatalog der aktuellen Version prüfen oder die Übereinstimmung der Strukturkataloge verschiedener Installationen prüfen.

`CHECK`

Die fehlerhaften Objekte werden in die SYSLST-Datei eingetragen. Der Name der SYSLST-Datei ist standardmäßig: `nkz.SYSLST.LST.project`

Der Dateiname für SYSLST kann sich ändern, wenn Sie das Kommando `CHECK` bei der Servicefunktion `Print functions` absetzen. Voraussetzung dafür ist, daß Sie SYSLST in der Print-Maske in eine andere Datei umgelenkt haben.

Die SYSLST-Datei wird in der Aufruf-Prozedur zugewiesen. Sie kann nach der Beendigung von TOM-REF im Editor gelesen werden.



Während des Ablaufs der Servicefunktion `CHECK` ist das Data Dictionary global gesperrt.

Siehe auch: `PRINT`, `SERVICES`

COPY - Objekt kopieren

Mit dem Kommando `COPY` rufen Sie die Maske TR9200 auf, in dem Sie das Objekt angeben, das Sie kopieren wollen und den neuen Namen des Objekts.

`COPY`

`COPY` bietet sich an, wenn mehrere Objekte gleichen Typs erstellt werden, die sich nur gering unterscheiden. Mit dem Kommando `CHANGE` lassen sich dann die objektspezifischen Angaben leicht nachtragen. Kopiert werden sämtliche Attribute, Beziehungen und Texte. Als Erstellungsdatum wird automatisch das aktuelle Datum eingetragen. Bereits erstellte Objekte können Sie mit `COPY` nicht überschreiben.



Während des Ablaufs der Servicefunktion `COPY` ist das Data Dictionary global gesperrt.

TR9200	Rename / Copy	\$A1234567.LEO.TOMRF1.TOM.BEISPIEL
Function : . (Rename, Copy) Object : to :		
Command:		

Bild 171 Maske TR9200

Function

Hier müssen Sie ein `c` eintragen.

Object

Alte Bezeichnungen für Typ (dreistellige Kurzbezeichnung) und Name

to

Neue Bezeichnungen für Typ und Name

Siehe auch: RENAME, SERVICES

CREATE - Objekt oder Beziehung erzeugen

Mit dem Kommando `CREATE` rufen Sie die Maske TR0010 auf, in der Sie ein Objekt oder eine (erweiterte) Beziehung erzeugen können. Das Kommando `CREATE` zählt zu den Grundfunktionen und bezieht sich auf die typneutralen und typspezifischen Attribute, die Beziehungsliste und den voreingestellten Text eines Objekts.

```

CREATE {
  COMPONENT=objektname ,TYPE=objekttyp
  COMPONENT=objektname
  TYPE=objekttyp
  RELATIONSHIP [EXTENDED] [ ,TYPE=beziehungstyp]
}

```

`COMPONENT=objektname`

Name des Objekts

`TYPE=objekttyp`

Der objekttyp kann in Kurz- oder Langbezeichnung eingegeben werden.

`RELATIONSHIP [EXTENDED]`

Es ist zu unterscheiden, ob der gesamte Operand `RELATIONSHIP EXTENDED` oder nur `RELATIONSHIP` angegeben wird: `RELATIONSHIP EXTENDED` ist nur bei erweiterten Beziehungstypen möglich, es wird nur die Erweiterung der Beziehung bearbeitet.

`RELATIONSHIP` ist bei allen Beziehungstypen möglich, es wird die gesamte Beziehung bearbeitet.

`TYPE=beziehungstyp`

Bei einfachen Beziehungen ist `SIR` (simple relationship) anzugeben, falls die Maske TR0008 übersprungen werden soll.

Falls der Operand bei erweiterten Beziehungstypen angegeben wird: `beziehungstyp` muß einer der Standardnamen für erweiterte Beziehungstypen sein (`STO`, `FOR`, `XBT`, `XIX`). Die Angabe des Operanden bewirkt, daß die Auswahlmaske für Beziehungstypen TR0008 übersprungen wird.

Falls bereits während des laufenden TOM-REF-Betriebs ein Objekt bearbeitet wurde, kann bei der Kommandoeingabe gegebenenfalls auf Operanden verzichtet werden. Fehlende Operanden interpretiert TOM-REF folgendermaßen:

1. `CREATE COMPONENT=objektname`
Der zuletzt bearbeitete Objekttyp wird beibehalten.
2. `CREATE TYPE=objekttyp`
Der Name des zuletzt bearbeiteten Objekts wird beibehalten.

3. CREATE

Typ und Name des zuletzt bearbeiteten Objekts werden beibehalten.

Beispiele

```
CREATE C=auftrag, T=RBT
```

Die Beschreibung für die SESAM-SQL V2-Basistabelle AUFTRAG soll erzeugt werden.

```
CREATE C=kunden
```

Direkt nach der Bearbeitung der Basistabelle AUFTRAG eingegeben, wird die SESAM-SQL V2-Basistabelle KUNDEN erstellt.

```
CR REL,TYPE=SIR
```

Für das aktuelle Objekt soll eine einfache Beziehung erstellt werden. Es wird eine leere Beziehungsmaske ausgegeben, in die Sie Name und Typ von Ausgangs- und Zielobjekt eingeben müssen.

Siehe auch: CHANGE, DELETE, DISPLAY

DDTEXT DELETE - TOM-REF-Text löschen

Mit dem Kommando `DDTEXT DELETE` können Sie den TOM-REF-Text des zuletzt bearbeiteten Objekts zum Löschen anzeigen lassen. Das Kommando können Sie in der Auswahlmaske TR0001, in einem Sub-Menü oder in einer leeren Objektmaske eingeben.

`DDTEXT DELETE`

Der TOM-REF-Text wird in der Textmaske TR0002 mit einem `Y` in der Kommandozeile ausgegeben.

- Falls Sie den Text löschen wollen, schicken Sie die Maske unverändert mit der Taste `[DÜ]` ab.
- Falls der TOM-REF-Text nicht gelöscht werden soll, überschreiben Sie das `Y` in der Kommandozeile mit `N` und schicken Sie die Maske dann mit der Taste `[DÜ]` ab oder Sie verlassen die Maske mit `[K1]`.

Siehe auch: `TEXT DELETE`, `LIBTEXT DELETE`

DELETE - Objekt oder Beziehung löschen

Mit dem Kommando `DELETE` rufen Sie die Maske TR0010 auf, in der Sie ein bestehendes Objekt oder eine bestehende (erweiterte) Beziehung löschen können. Das Kommando `DELETE` zählt zu den Grundfunktionen und bezieht sich auf die typneutralen und typspezifischen Attribute, die Beziehungsliste und den voreingestellten Text eines Objekts.

```

DELETE {
  COMPONENT=objektname ,TYPE=objekttyp
  COMPONENT=objektname
  TYPE=objekttyp
  RELATIONSHIP [EXTENDED] [,TYPE=beziehungstyp]
}

```

`COMPONENT=objektname`

Name des Objekts

`TYPE=objekttyp`

Der objekttyp kann in Kurz- oder Langbezeichnung eingegeben werden.

`RELATIONSHIP [EXTENDED]`

Es ist zu unterscheiden, ob der gesamte Operand `RELATIONSHIP EXTENDED` oder nur `RELATIONSHIP` angegeben wird: `RELATIONSHIP EXTENDED` ist nur bei erweiterten Beziehungstypen möglich, es wird nur die Erweiterung der Beziehung bearbeitet.

`RELATIONSHIP` ist bei allen Beziehungstypen möglich, es wird die gesamte Beziehung bearbeitet.

`TYPE=beziehungstyp`

Bei einfachen Beziehungen ist `SIR` (simple relationship) anzugeben, falls die Maske TR0008 übersprungen werden soll.

Falls der Operand bei erweiterten Beziehungstypen angegeben wird: `beziehungstyp` muß einer der Standardnamen für erweiterte Beziehungstypen sein (`STO`, `FOR`, `XBT`, `XIX`). Die Angabe des Operanden bewirkt, daß die Auswahlmaske für Beziehungstypen TR0008 übersprungen wird.

Falls bereits während des laufenden TOM-REF-Betriebs ein Objekt bearbeitet wurde, kann bei der Kommandoeingabe gegebenenfalls auf Operanden verzichtet werden. Fehlende Operanden interpretiert TOM-REF folgendermaßen:

1. `DELETE COMPONENT=objektname`
Der zuletzt bearbeitete Objekttyp wird beibehalten.
2. `DELETE TYPE=objekttyp`
Der Name des zuletzt bearbeiteten Objekts wird beibehalten.

3. DELETE

Typ und Name des zuletzt bearbeiteten Objekts werden beibehalten.

Beispiel

```
DELETE C=auftrag, T=RBT
```

Die SESAM-SQL V2-Basistabelle AUFTRAG soll gelöscht werden.

```
DELETE C=kunden
```

Direkt nach der Bearbeitung der Basistabelle AUFTRAG eingegeben, wird die SESAM-SQL V2-Basistabelle KUNDEN gelöscht.

```
DE REL,TYPE=SIR
```

Eine einfache Beziehung des aktuellen Objekts soll gelöscht werden. Es wird eine leere Beziehungsmaske ausgegeben, in die Sie Name und Typ von Ausgangs- und Zielobjekt eingeben müssen.

Siehe auch: CREATE, CHANGE, DISPLAY

DELSTRUCT - Objektstruktur löschen

Mit dem Kommando `DELSTRUCT` können Sie Strukturen bestimmter Objekte und ihrer Beziehungen ganz oder teilweise löschen.

Ausgehend vom Objekt, das in der Hierarchie der Struktur an erster Stelle steht, werden der Reihe nach alle untergeordneten Objekte angezeigt. Für jedes einzelne Objekt ist zu entscheiden, ob es gelöscht werden soll oder nicht. Ein Objekt kann jedoch nur dann gelöscht werden, wenn es nicht Zielobjekt in einer Beziehung ist. Falls ein Objekt auch noch von Objekten außerhalb oder innerhalb der Struktur referenziert wird, kann es nicht gelöscht werden.

`DELSTRUCT`



Nach dem Kommando `DELSTRUCT` kann es sein, daß Objekte im Data Dictionary vorhanden sind, die nicht mehr bezogen sind. Solche Objekte können Sie sich mit der Servicefunktion `Display objects which are not part of another object` (siehe Seite 127) anzeigen lassen.

DISDIR - Inhaltsverzeichnis ausgeben

Mit dem Kommando `DISDIR` können Sie folgende Funktionen einleiten:

- Alle Objekte eines Objekttyps oder Objektgruppe anzeigen
- Eine Auswahl von Objekten eines oder mehrere Objekttypen anzeigen

Sie können das Kommando `DISDIR` in der Kommandozeile einer Auswahlmaske (TR0001 oder Sub-Menü) oder in einer leeren Objektmaske angeben.

`DISDIR [TYPE=objekttyp]`

`TYPE=objekttyp`

dreistellige Kurzbezeichnung eines Objekttyps oder Objekttyp-Gruppe. Es gibt standardmäßig folgende Gruppen von Objekttypen:

Bezeichnung	Bedeutung
*T	Alle Objekttypen
*D	Die Objekttypen FLD, FLG, REC
*M	Die Objekttypen MOD, PRG

Es ist zu unterscheiden, ob das Kommando `DISDIR` mit oder ohne Operand gegeben wird:

- `DISDIR` mit Operand:
Die Maske TR9019 mit der Anzeige der Objekte wird ausgegeben. Der weitere Ablauf ist ab Seite 112 dargestellt.
- `DISDIR` ohne Operand:
Die Auswahlmaske TR9010 wird ausgegeben. Der weitere Ablauf ist ab Seite 114 dargestellt.

Beispiel

```
DISDIR T=RBT
```

Es wird ein Verzeichnis aller SESAM/SQL V2-Basistabellen im Data Dictionary ausgegeben.

Siehe auch: `PRDIR`

DISPLAY - Objekt oder Beziehung anzeigen

Mit dem Kommando `DISPLAY` rufen Sie die Maske TR0010 auf, in der Sie sich ein bestehendes Objekt oder eine bestehende (erweiterte) Beziehung anzeigen lassen können. Das Kommando `DISPLAY` zählt zu den Grundfunktionen und bezieht sich auf die typneutrale und typspezifischen Attribute, die Beziehungsliste und den voreingestellten Text eines Objekts.

```

DISPLAY {
  COMPONENT=objektname ,TYPE=objekttyp
  COMPONENT=objektname
  TYPE=objekttyp
  RELATIONSHIP [EXTENDED] [ ,TYPE=beziehungstyp]
}

```

`COMPONENT=objektname`

Name des Objekts

`TYPE=objekttyp`

Der objekttyp kann in Kurz- oder Langbezeichnung eingegeben werden.

`RELATIONSHIP [EXTENDED]`

Es ist zu unterscheiden, ob der gesamte Operand `RELATIONSHIP EXTENDED` oder nur `RELATIONSHIP` angegeben wird: `RELATIONSHIP EXTENDED` ist nur bei erweiterten Beziehungstypen möglich, es wird nur die Erweiterung der Beziehung bearbeitet.

`RELATIONSHIP` ist bei allen Beziehungstypen möglich, es wird die gesamte Beziehung bearbeitet.

`TYPE=beziehungstyp`

Bei einfachen Beziehungen ist `SIR` (simple relationship) anzugeben, falls die Maske TR0008 übersprungen werden soll.

Falls der Operand bei erweiterten Beziehungstypen angegeben wird: `beziehungstyp` muß einer der Standardnamen für erweiterte Beziehungstypen sein (`STO`, `FOR`, `XBT`, `XIX`). Die Angabe des Operanden bewirkt, daß die Auswahlmaske für Beziehungstypen TR0008 übersprungen wird.

Falls bereits während des laufenden TOM-REF-Betriebs ein Objekt bearbeitet wurde, kann bei der Kommandoeingabe gegebenenfalls auf Operanden verzichtet werden. Fehlende Operanden interpretiert TOM-REF folgendermaßen:

1. `DISPLAY COMPONENT=objektname`
Der zuletzt bearbeitete Objekttyp wird beibehalten.
2. `DISPLAY TYPE=objekttyp`
Der Name des zuletzt bearbeiteten Objekts wird beibehalten.

3. DISPLAY

Typ und Name des zuletzt bearbeiteten Objekts werden beibehalten.

Beispiele

```
DISPLAY C=auftrag, T=RBT
```

Die Beschreibung für die SESAM-SQL V2-Basistabelle AUFTRAG soll erzeugt werden.

```
DISPLAY C=kunden
```

Direkt nach der Bearbeitung der Basistabelle AUFTRAG eingegeben, wird die SESAM-SQL V2-Basistabelle KUNDEN erstellt.

```
DIS REL,TYPE=SIR
```

Die vom aktuellen Objekt ausgehenden einfachen Beziehungen sollen angezeigt werden. Es wird eine leere Beziehungsmaske ausgegeben, in die Sie Name und Typ von Ausgangs- und Zielobjekt eingeben müssen.

Siehe auch: CHANGE, DELETE, DISPLAY

DISP-USES - Beziehungsliste anzeigen

Mit dem Kommando `DISP-USES` können Sie sich zu einem ausgewählten Objekt die Beziehungen anzeigen lassen, die von diesem Objekt ausgehen. Angezeigt werden direkte und indirekte Beziehungen.

- Direkte Beziehungen können Sie (oder ein Tool) explizit zwischen Objekten erstellen.
- Indirekte Beziehungen leitet TOM-REF automatisch aus den direkten Beziehungen ab, indem es Pfade durch das Netz der direkten Beziehungen verfolgt. TOM-REF zeigt indirekte Beziehungen bis zur Ebenentiefe 20 an. (Die Ebenentiefe einer direkten Beziehung ist 2.)

`DISP-USES`

Die weitere Vorgehensweise nach diesem Kommando finden Sie auf Seite 123.

Siehe auch: `DISP-WHERE-USED`, `SERVICES`

DISP-WHERE-USED - Inverse Beziehungen anzeigen

Mit dem Servicekommando `DISP-WHERE-USED` können Sie sich zu einem ausgewählten Objekt die inversen Beziehungen anzeigen lassen. Das sind diejenigen Beziehungen, in denen das ausgewählte Objekt Zielobjekt ist. Angezeigt werden direkte und indirekte Beziehungen.

- Direkte Beziehungen können Sie (oder ein Tool) explizit zwischen Objekten erstellen.
- Indirekte Beziehungen leitet TOM-REF automatisch aus den direkten Beziehungen ab, indem es Pfade durch das Netz der direkten Beziehungen verfolgt. TOM-REF zeigt indirekte Beziehungen bis zur Ebenentiefe 20 an. (Die Ebenentiefe einer direkten Beziehung ist 2.)

`DISP-WHERE-USED`

Die weitere Vorgehensweise nach diesem Kommando finden Sie auf Seite 118.

Siehe auch: `DISP-USES`, `SERVICES`

EDITOR - TOM-TI aufrufen

Mit dem Kommando `EDITOR` rufen Sie TOM-TI auf, um den Text eines bestimmten Objektes zu bearbeiten. Name und Texttyp des Objekts, das bearbeitet werden soll, sind als Operanden anzugeben. Eine Liste der Objekte und des zu gehörigen Texttyps finden Sie im Anhang.

```
EDITOR COMPONENT=objektname,TYPE=typ
```

COMPONENT=objektname

Name des Objekts

TYPE=typ

Texttyp des Objekts, siehe Liste im Anhang.



Um den Text eines Objekts zu bearbeiten, muß das typneutrale Attribut Text auf `Y` gesetzt sein.

Bearbeiten Sie den Text mit den TOM-TI-Kommandos. Die Taste **(K)** bewirkt die Rückkehr zu TOM-REF. Dabei legt TOM-TI den Text als Element in der TOMLIB mit dem Namen `objektname.objekttyp` ab. Informationen zum TOM-TI und zur TOMLIB finden Sie im "Referenzhandbuch TOM-M V4.2" [3]. Der Name der TOMLIB lautet `nkz.TOMLIB.TOM.project`. Falls TOM-REF als `Special function` gestartet wurde, sind die variablen Namensteile `nkz` und `project` bereits in TomDoors-M versorgt worden. Damit ist gewährleistet, daß die TOMLIB existiert. Beim Start im BS2000-Kommandomodus übernimmt TOM-REF als `nkz` und `project` Ihre Angaben.

Beispiel

```
ED C=auftrag,T=TAL
```

Der Langtext (typneutrales Attribut) des Synonyms AUFTRAG soll im TOM-TI bearbeitet werden.

Siehe auch: `N2CHARTS`, `UDS`

END, RETURN - TOM-REF beenden

Das Kommando `END` beendet TOM-REF. In der Maske angekreuzte Funktionen werden nicht mehr ausgeführt.

Mit dem Kommando `RETURN` beenden Sie TOM-REF. In der Maske angekreuzte Funktionen werden nicht mehr ausgeführt.

END

RETURN

GENERATE COPY - COBOL-Copy-Element erzeugen

Mit dem Kommando `GENERATE COPY` steuern Sie die Maske TR9060 an, in der Sie die Generierung von Copy-Elementen anstoßen können. Voraussetzung ist, daß Sie die zugehörigen Copy-Objekte erstellt haben (siehe Seite 205).

`GENERATE COPY`

Das Kommando `GENERATE COPY` zählt zu den Servicekommandos. Die weitere Vorgehensweise in der aufgerufenen Maske ist ab Seite 143 beschrieben.

Siehe auch: `SERVICES`, `UDS`

HELP - Hilfemaske ansteuern

Sämtliche Ausgaben nach dem Kommando `HELP` erfolgen in der Maske TR8888. Dessen Inhalt ist abhängig von den Operanden. Folgemasken können Sie mit der Taste `DÜ` abrufen. Die Taste `K1` bricht die Funktion ab.

```

HELP { [cmd]
      text-cmd
      serv-cmd
      TYPE
      [TYPE=objekttyp]
    }

```

[cmd]

Beliebiges TOM-REF-Kommando, zu dem Hilfe angefordert wird.

text-cmd

TOM-REF-Kommando zur Bearbeitung von Texten, dessen Syntax angezeigt werden soll.

serv-cmd

TOM-REF-Kommando zum Aufruf von Servicefunktionen, dessen Syntax angezeigt werden soll.

TYPE

Sämtliche Objekttypen, die im Strukturkatalog enthalten sind, werden angezeigt. Pro Objekttyp werden Kurz- und Langbezeichnung ausgegeben.

TYPE=objekttyp

Sämtlicher Beziehungen, die vom angegebenen Objekttyp ausgehen können, werden angezeigt. Der `objekttyp` kann in Kurz- oder Langbezeichnung eingegeben werden. Pro Beziehung werden Kurz- und Langbezeichnung des Zielobjekts angezeigt.

Wenn Sie das Kommando `HELP` ohne Operanden aufrufen, erhalten Sie den nachfolgenden Bildschirm, in dem Ihnen die Syntaxübersicht zu allen TOM-REF-Kommandos angezeigt wird.

Beispiele

HELP

```

TR8888  HELP function, display commands          $A1234567.LEO.TOMRFl.TOM.BEISPIEL
-----
CH(ANGE) C(OMPONENT)=name,T(YPE)=type          : Change an object              1)
CH(ANGE) REL(ATION),T(YPE)=rtype               : Change relationship           2)
CH(ANGE) REL(ATION) EXT(ENDED),T(YPE)=rtype    : Change extension of rel-ship  2)

CR(EATE) C(OMPONENT)=name,T(YPE)=type          : Create an object              1)
CR(EATE) REL(ATION),T(YPE)=rtype               : Create relationship           2)
CR(EATE) REL(ATION) EXT(ENDED),T(YPE)=rtype    : Create extension of rel-ship  2)

DE(LETE) C(OMPONENT)=name,T(YPE)=type          : Delete an object              1)
DE(LETE) REL(ATION),T(YPE)=rtype               : Delete relationship           2)
DE(LETE) REL(ATION) EXT(ENDED),T(YPE)=rtype    : Delete extension of rel-ship  2)

1) If name/type is not entered, last name/type is used.
2) If rtype is not entered, TR0008 follows.
   For rtype you can set FOR,STO,XBT,XIX,SIR(=simple relationship).

-----
Command: +.....
-----
Terminate HELP Function with K1 or DUE

```

Bild 172 Maske TR8888 mit Ausgabebeispiel

Alle Kommandos werden angezeigt.

HELP TYPE

```

TR8888  HELP function, display all types        $A1234567.LEO.TOMRFl.TOM.BEISPIEL
-----
CAL, ENTRY-CALLED
ENP, ENTRY-INCLUDED
MGR, MODULE-GROUP
MOD, MODULE
PGM, PROGRAM
SYS, SYSTEM
CBL, COBOL-SOURCE
DRP, DRIVE-PROCEDURE
LIB, LIBRARY
LMO, LOAD-MODULE
NOM, NON-S.OBJ.-MOD.
SOM, SHARE-OBJ.-MOD.
ARE, UDS-AREA
DAR, UDS-DYN.-AREA
DBK, UDS-SET-KEY

-----
Command: +.....
-----
Terminate HELP Function with K1 or DUE

```

Bild 173 Maske TR8888 mit Ausgabebeispiel

Alle Objekttypen werden angezeigt.

HELP TYPE=MOD

```
TR8888  HELP function, display relations          $A1234567.EBE.TOMRF1.TOM.BEISPIEL
-----
MOD,  MODULE
SUB,  UDS-SUBSCHEMA
FIL,  FILE
MSK,  FORMAT
REC,  RECORD
FLG,  FIELDGROUP
FLD,  FIELD
ENP,  ENTRY-INCLUDED
CAL,  ENTRY-CALLED
SBR,  UDS-SUBSCH-REC
DAR,  UDS-DYN.-AREA
SET,  UDS-SET
SSY,  UDS-SYSTEM-SET
SDY,  UDS-DYNAMIC-SET
MGR,  MODULE-GROUP
-----
Command: +.....
-----
Terminate HELP Function with K1 or DUE
```

Bild 174 Maske TR8888 mit Ausgabebeispiel


Alle Objekte, die das Ziel einer Beziehung von einem Modul aus sein können, werden angezeigt.

INCORPORATE - Data Dictionaries zusammenführen

Mit dem Kommando `INCORPORATE` können Sie Objekte (samt der Beziehungsliste und den Texten) von einem anderen Data Dictionary in das aktuelle Data Dictionary übertragen.

Dabei kann der Fall eintreten, daß zu einem zu übertragenden Objekt bereits ein Objekt desselben Typs und Namens im aktuellen Data Dictionary existiert. Bei solchen, sog. paarigen Objekten können Sie jeweils getrennt für Objektattribute, Beziehungsliste, TOM-REF-Text und TOM-TI-Langtext entscheiden, ob die Einträge des aktuellen Data Dictionary bestehen bleiben oder durch die entsprechenden Einträge des anderen Data Dictionary überschrieben werden.

`INCORPORATE`

 Während der Übertragung ist ein Objekt im aktuellen Data Dictionary gesperrt.

Nachdem Sie dieses Kommando mit der Taste `DU` abgeschickt haben, erhalten Sie der Maske TR9100. Das weitere Vorgehen ist ab Seite 151 beschrieben.

Siehe auch: `INITIATE`, `SERVICES`

INITIATE - Neues Data Dictionary anlegen

Mit dem Kommando `INITIATE` legen Sie ein neues Data Dictionary an.

`INITIATE`

Sie erhalten durch dieses Kommando der Maske TR9930, in dem Sie den Namen des neuen Data Dictionary eingeben.

```

TR9930  Initiate                                     $userid  .nkz.TOMrfl.TOM.project
-----
TOMREF-Data Dictionary:

Private
Catid      : :....:
Userid     : $userid
Identification : nkz
REF-name   : TOMrfl
Project    : project
Prefix     : .....
Suffix     : .....

Command: .....

```

Bild 175 Maske TR9930

Die Felder in dieser Maske sind mit den Angaben aus der Statuszeile vorbelegt. Mit Ausnahme der Felder `Catid` (Katologkennung) und `Userid` sind alle Felder überschreibbar. Das Feld `Catid` dient lediglich zur Dokumentation und ist evtl. mit einem Wert aus der Aufruf-Prozedur vorbelegt.

Falls Sie die Felder `Prefix` bzw. `Suffix` versorgen, so werden die Namen der erstellten Dateien mit der angegebenen Vorsilbe bzw. Nachsilbe versehen.

Falls zum neuen Data Dictionary noch keine TOMLIB existiert, wird diese ebenfalls automatisch angelegt.

Siehe auch: `INCORPORATE`

LIBTEXT DELETE - TOM-TI-Text löschen

Mit dem Kommando `LIBTEXT DELETE` löschen Sie den TOM-TI-Langtext, den Sie als typ-neutrales Attribut eines Objekttyps angelegt haben.

`LIBTEXT DELETE`

Der TOM-TI-Langtext wird in der TOM-TI-Maske angezeigt. Mit der Taste `(K1)` oder mit der Taste `(DÜ)` können Sie die Anzeige des TOM-TI-Langtextes beenden. Danach wird die Maske TR0005 mit der Frage ausgegeben, ob der TOM-TI-Langtext gelöscht werden soll (Meldung TR-19). Nach dem Betätigen der Taste `(DÜ)` wird der TOM-TI-Langtext gelöscht. Der TOM-TI-Langtext wird nicht gelöscht, wenn entweder das in der Kommandozeile vorgegebene `Y` zuvor mit `N` überschrieben wird oder die Maske TR0005 unmittelbar mit der Taste `(K1)` abgeschickt wird.

Das Kommando können Sie in der Auswahlmaske TR0001, in einem Sub-Menü oder in einer leeren Objektmaske eingeben.

Siehe auch: `TEXT DELETE`, `DDTEXT DELETE`

N2CHARTS - N2Charts aufrufen

Mit dem Kommando `N2Charts` rufen Sie das Programm TOMREFG auf, das als Baustein der TOM-Produktgruppe auf Ihrer BS2000-Anlage installiert sein muß. Mit diesem Programm können Sie die Systemstrukturen grafisch aufbereiten, die Sie mit TOM-REF erzeugen und verwalten.

N2CHARTS

TOM-REFG ist im Handbuch "TOM-REFG, Oberfläche für Data Dictionary-geführte Strukturen" [5] beschrieben.

Siehe auch: `EDITOR, GENERATE COPY, UDS`

PRDIR - Inhaltsverzeichnis ausdrucken

Mit dem Kommando `PRDIR` können Sie sich Inhaltsverzeichnisse des Data Dictionary ausdrucken lassen. Das Kommando `PRDIR` können Sie in der Auswahlmaske TR0001, in einem Sub-Menü oder in einer leeren Objektmaske eingeben.

`PRDIR [TYPE=objekttyp]`

objekttyp

Kurz- oder Langbezeichnung des Objekttyps

Es ist zu unterscheiden, ob das Kommando `PRDIR` mit oder ohne Operand gegeben wird:

- `PRDIR` ohne Operand:
Es wird ein Verzeichnis aller Objekte, die im Data Dictionary enthalten sind, gedruckt. Die Maske, in der das Kommando `PRDIR` gegeben wurde, wird erneut ausgegeben.
- `PRDIR` mit Operand:
Es wird ein Verzeichnis der Objekte des im Operanden anzugebenden Typs gedruckt. Die Maske, in der das Kommando `PRDIR` gegeben wurde, wird erneut ausgegeben.

Siehe auch: `DISDIR, SERVICES, PRINT`

PRINT - Auswahlmaske für den Druck ansteuern

Mit dem Kommando `PRINT` wird in der Maske TR9070 verzweigt. Dort können sämtliche Print-Funktionen angestoßen werden, die ab Seite 131 beschrieben sind. Mit der Taste **[K1]** erfolgt die Rückkehr in der Maske, in der Sie das Kommando `PRINT` eingegeben haben. Das Kommando `PRDIR` können Sie in der Auswahlmaske TR0001, in einem Sub-Menü oder in einer leeren Objektmaske eingeben.

`PRINT`

Siehe auch: `PRDIR`, `SERVICES`

PROT - Dialog protokollieren

Mit dem Kommando `PROT` wird festgelegt, ob der Ablauf des Dialogs in der logischen Systemdatei `SYSLST` protokolliert wird.

`PROT` $\left\{ \begin{array}{l} \text{ON} \\ \text{OFF} \end{array} \right\}$

`PROT ON`

Der Dialogablauf wird protokolliert.

`PROT OFF`

Der Dialogablauf wird nicht protokolliert; dies entspricht der Voreinstellung.

REESTABLISH - TOMREFI-Datei neu aufbauen

Mit dem Servicekommando `REESTABLISH` können Sie die TOMRFI-Datei (Inverse Beziehungen) wieder neu erstellen. Das Kommando `REESTABLISH` können Sie in der Auswahlmaske TR0001, in einem Sub-Menü oder in einer leeren Objektmaske eingeben.

`REESTABLISH`

Wenn Sie dieses Kommando absetzen, wird die Maske nach dem Abschicken mit der Taste `[DÜ]` erneut mit folgender Meldung ausgegeben:

```
TR-199    Do you really want to reestablish TOMREFI ? (y/n)
```

Wenn Sie die Maske mit der Taste `[DÜ]` abschicken, wird die Funktion ausgeführt.

Die Funktion wird nicht ausgeführt, wenn sie das in der Kommandozeile vorgegebene `Y` mit `N` überschreiben und dann die Maske mit der Taste `[DÜ]` abschicken. In diesem Fall wird in der Folgemaske in der Meldungszeile der Hinweis ausgegeben, daß die Funktion nicht ausgeführt wurde (Meldung TR-180).



Während des Ablaufs der Servicefunktion `REESTABLISH` ist das Data Dictionary global gesperrt.

Siehe auch: `DISP-WHERE-USED`, `RESTORE`, `SERVICES`

RENAME - Objekt umbenennen

Mit dem Kommando `RENAME` rufen Sie die Maske TR9200 auf, in dem Sie das Objekt angeben, das Sie umbenennen wollen und den neuen Namen des Objekts. Das Attribut `Letztes Änderungsdatum` wird automatisch mit dem aktuellen Datum überschrieben.

RENAME

Falls zum umbenannten Objekt Beziehungen existieren, werden diese automatisch an den neuen Namen angepaßt.



Während des Ablaufs der Servicefunktion `RENAME` ist das Data Dictionary global gesperrt.

TR9200	Rename / Copy	\$A1234567.LEO.TOMRF1.TOM.BEISPIEL
<p>Function : . (Rename, Copy)</p> <p>Object :</p> <p>to :</p>		
<hr/> <p>Command:</p> <hr/>		

Bild 176 Maske TR9200

Function

Hier müssen Sie ein `R` eintragen.

Object

Alte Bezeichnungen für Typ (dreistellige Kurzbezeichnung) und Name

to

Neue Bezeichnungen für Typ und Name

Siehe auch: `COPY`, `SERVICES`

RESTORE - Data Dictionary laden

Mit dem Servicekommando `RESTORE` können Sie das aktuelle Data Dictionary mit einem Data Dictionary überschreiben, das Sie als Element in der TOMLIB gesichert haben, Das (neue) aktuelle Data Dictionary wird reorganisiert und eine neue TOMRFI-Datei angelegt.

`RESTORE`

Mit diesem Kommando steuern Sie die Maske TR9910 an, indem Sie das Member der TOMLIB angeben können, daß Sie zum aktuellen Data Dictionary machen wollen. Das Feld im TR9910 ist mit dem Namen des aktuellen Data Dictionary vorbelegt (siehe auch Seite 165).



Während des Ablaufs von `RESTORE` ist das Data Dictionary global gesperrt.

Siehe auch: `REESTABLISH`, `SAVE`, `SERVICES`

SAVE - Data Dictionary sichern

Mit dem Servicekommando `SAVE` können Sie Ihr Data Dictionary als Member (Bibliothekselement) in der TOMLIB ablegen.

`SAVE`

Mit dem Kommando `SAVE` steuern Sie die Maske T9900 an, in dem Sie den Namen und den Typ des Members angeben, in dem das Data Dictionary in der TOMLIB gesichert werden soll. Der Name ist mit dem Teilnamen des aktuellen Data Dictionary (hier: TOMRF1) vorbelegt und überschreibbar. Der Typ `REF` ist nicht überschreibbar.

Das Feld für die Antwort auf die Frage, ob die Dateien des Data Dictionary nach dem Sichern in der TOMLIB gelöscht werden sollen, ist mit `N` (Nein) vorbelegt. Das Feld ist nur dann überschreibbar, wenn sich das Data Dictionary in der eigenen Kennung befindet. Bei einer Belegung mit `Y` (Ja) werden die Dateien nach dem Sichern gelöscht.

Siehe auch: `REESTABLISH`, `RESTORE`, `SERVICES`

SERVICES - Servicekommandos ansteuern

Mit dem Kommando `SERVICES` steuern Sie die Auswahlmaske TR9000 direkt an.

`SERVICES`

Die folgende Tabelle enthält eine Zuordnung der Kommandos zu den Servicefunktionen. Die Servicefunktionen sind in Kapitel 5 beschrieben.

Servicefunktion	Kommando
Display directory for type	DISDIR
Display where-used	DISP-WHERE-USED
Display uses	DISP-USES
Print functions - Drucken von Objektverzeichnissen (Spezielle Print-Funktion)	PRINT PRDIR
Generate functions: Copy UDS	GENERATE COPY UDS
Rename object Copy object	RENAME COPY
Delete structure	DELSTRUCT
Incorporate from another Data Dictionary	INCORPORATE
Check objects and relationships	CHECK
Administration (ISAM): - Data Dictionary in TOMLIB sichern - Data Dictionary aus TOMLIB holen - TOMRFI-Datei neu aufbauen - Neues Data Dictionary anlegen	SAVE RESTORE REESTABLISH INITIATE

TEXT - Textmaske ansteuern

Mit dem Kommando `TEXT` wird die Ausgabe einer Textmaske bei der Bearbeitung von Objekten mit den Grundfunktionen geregelt. Dieses Kommando können Sie in der Auswahlmaske TR0001, in einem Sub-Menü oder in einer leeren Objektmaske angeben.

TEXT $\left\{ \begin{array}{l} [\text{ON}] \\ [\text{ON}], \text{LIB}=\text{ON} \\ \underline{\text{OFF}} \end{array} \right.$

`TEXT [ON]`

Die TOM-REF-Textmaske TR0002 wird ausgegeben.

`TEXT, LIB=ON`

Zur Textbearbeitung wird der TOM-TI aufgerufen.

`TEXT OFF`

Es wird keine Textmaske ausgegeben.

Siehe auch: `DDTEXT DELETE`, `EDITOR`, `TEXT DELETE`

UDS - UDS-DDA aufrufen

Mit dem Kommando `UDS` rufen Sie das Programm UDS-DDA auf.

`UDS`

Das Produkt UDS-DDA (UDS-Data Dictionary Anschluß) verbindet das Datenbanksystem UDS mit TOM-REF. Die Vorkehrungen, die für den Einsatz von UDS-DDA zu treffen sind, und die Funktionen, welche UDS-DDA bietet, sind ausführlich im Handbuch "UDS-DDA Benutzerhandbuch" [13] beschrieben.

Siehe auch: `N2CHARTS`, `SERVICES`, `UDS`

13 Meldungen

- TR-00 Internal error in message handling - error-number: ;
- Bedeutung**
Bei der Meldungsbehandlung trat ein interner Fehler auf; die Fehlernummer wird ausgegeben.
- TR-01 Do you want to delete object ? (Y/N)
- Bedeutung**
Kontrollfrage, ob das angegebene Objekt wirklich gelöscht werden soll.
- TR-02 Object is deleted
- Bedeutung**
Objekt wurde gelöscht.
- TR-03 Insert relationships
- Bedeutung**
Sie können Typ und Namen von Objekten eingeben, zu denen Beziehungen aufgebaut werden sollen.
- TR-04 Please mark a function
- Bedeutung**
Sie haben vergessen, eine Funktion anzugeben.
- TR-05 Please mark an object or services
- Bedeutung**
Sie müssen mindestens ein Objekt markieren oder eine Servicefunktion auswählen.
- TR-06 Do you want to overwrite object ? (Y/N)
- Bedeutung**
Kontrollfrage, ob das angegebene Objekt überschrieben werden soll.

TR-07 Do you want to overwrite relationships ? (Y/N)

Bedeutung

Kontrollfrage, ob die angegebenen Beziehungen überschrieben werden sollen.

TR-08 Do you want to overwrite text ? (Y/N)

Bedeutung

Kontrollfrage, ob der angegebene Text überschrieben werden soll.

TR-09 Error in function codes

Bedeutung

Die Eingaben in der Markierspalte sind falsch.

TR-10 No more objects in list

Bedeutung

Diese Meldung erscheint am Ende einer Auflistung.

TR-12 Object is in central Data Dictionary

Bedeutung

Das Objekt kann nicht definiert werden, da es bereits im zentralen Data Dictionary existiert.

TR-13 Do you want to generate COPY ? (Y/N)

Bedeutung

Kontrollfrage, ob Sie ein COPY-Element generieren wollen.

TR-14 Do you want to see COPY ? (Y/N)

Bedeutung

Kontrollfrage, ob Sie das COPY-Element sehen wollen.

TR-15 COPY generated without errors

Bedeutung

Ein COPY-Element konnte fehlerfrei generiert werden.

TR-16 Error by COPY-generation - see member <copyname>.ERR in your TOMLIB

Bedeutung

Bei der Generierung eines COPY-Elementes traten Fehler auf; eine Fehlerliste wurde in der TOMLIB mit dem Namen "<copyname>.ERR" abgelegt.

TR-17 Warning - Calls to REFNUM generated

Bedeutung

Im MOVE-Statement sind für COBOL unverträgliche Typen enthalten. Überprüfen Sie das erzeugte MOVE-Statement.

- TR-18 Do you want to delete text ? (Y/N)
- Bedeutung**
Kontrollfrage, ob der angegebene Text wirklich gelöscht werden soll.
- TR-19 Do you want to delete TOMLIB-text ? (Y/N)
- Bedeutung**
Kontrollfrage, ob Sie den TOMLIB-Text löschen wollen.
- TR-20 Wanted text is not found
- Bedeutung**
Der gesuchte Text konnte nicht gefunden werden.
- TR-21 Error by write text
- Bedeutung**
Beim Schreiben eines Textes trat ein Fehler auf.
- TR-22 Error by delete text
- Bedeutung**
Der Text konnte nicht gelöscht werden.
- TR-23 Move-buffer empty!
- Bedeutung**
Eine Zeile wurde mit A oder B markiert, jedoch fehlt die Markierung mit M.
- TR-24 Add relationships
- Bedeutung**
Sie können nun Beziehungen eingeben.
- TR-25 Function not executed (line was already marked by M)
- Bedeutung**
Die Funktion kann nicht ausgeführt werden: Die Zeile ist bereits mit M markiert, diese Markierung ist jedoch durch Blättern nicht mehr sichtbar.
- TR-26 Error in mark: relationship is not extended !
- Bedeutung**
Markierfehler: Eine *- , R- oder E-Markierung darf nicht auf einfache Beziehungen angewendet werden.

TR-27 Object type is not valid

Bedeutung

Der angegebene Objekttyp ist gemäß Strukturkatalog nicht erlaubt. Sie können den Strukturkatalog ändern mit den Assemblermakros, die in Kapitel 10 beschrieben sind (Siehe A.3).

TR-28 Entered relationship is not valid

Bedeutung

Die angegebene Beziehung ist gemäß Strukturkatalog nicht erlaubt. Sie können den Strukturkatalog ändern mit den Assemblermakros, die in Kapitel 10 beschrieben sind (Siehe A.3).

TR-29 Selection-type is not valid

Bedeutung

Der angegebene Objekttyp ist falsch.

TR-30 Object found in central Data Dictionary

Bedeutung

Das Objekt wurde nicht im eigenen, sondern im zentralen Data Dictionary gefunden.

TR-31 Command TEXT DELETE only allowed when command TEXT ON[,LIB=ON] given

Bedeutung

Das Kommando TEXT DELETE ist nur erlaubt, wenn vorher das Kommando TEXT ON[,LIB=ON] abgesetzt wurde.

TR-32 Command DDTEXT DELETE only allowed when command TEXT ON given

Bedeutung

Das Kommando DDTEXT DELETE ist nur erlaubt, wenn vorher das Kommando TEXT ON abgesetzt wurde.

TR-33 Command LIBTEXT DELETE only allowed when command TEXT ON,LIB=ON given

Bedeutung

Das Kommando LIBTEXT DELETE ist nur erlaubt, wenn vorher das Kommando TEXT ON,LIB=ON abgesetzt wurde.

TR-34 It is not allowed to delete text in CENTRAL-Data Dictionary / CENTRAL-TOMLIB

Bedeutung

Es ist nicht erlaubt, Text im zentralen Data Dictionary bzw. in der zentralen TOMLIB zu löschen.

TR-35 Level greater than 20 present

Bedeutung

Eine im Data Dictionary vorhandene Verkettung mit Ebenentiefe größer als 20 wurde beim Objekt mit Ebenentiefe 20 abgebrochen.

TR-40 Wanted object is not defined

Bedeutung

Das gesuchte Objekt ist nicht definiert.

TR-41 This object is part of another object and cannot be deleted

Bedeutung

Das angegebene Objekt ist Zielobjekt in einer Beziehung; es kann deshalb nicht gelöscht werden.

TR-45 (Only) one function must be marked

Bedeutung

(Nur) eine Funktion muß markiert werden.

TR-46 At least one function must be marked

Bedeutung

Sie müssen mindestens eine Funktion markieren, bevor Sie ein Objekt bearbeiten können.

TR-47 At least one object must be marked

Bedeutung

Sie müssen mindestens ein Objekt markieren, bevor Sie eine Funktion aufrufen.

TR-48 Please mark a relationship-type

Bedeutung

Bitte markieren Sie einen Beziehungstyp.

TR-50 An object must not be referenced by itself

Bedeutung

Ein Objekt darf nicht in der eigenen Beziehungsliste enthalten sein.

TR-51 An object must be referenced only one time by the same source-object

Bedeutung

Ein Objekt darf nur einmal in der Beziehungsliste eines anderen Objekts enthalten sein.

TR-52 Both object-type and object-name must be entered

Bedeutung

Sowohl Typ als auch Name eines Objekts müssen eingegeben werden.

TR-54 Wanted TOMLIB-Text is not found

Bedeutung

Der gewünschte TOMLIB-Text wurde nicht gefunden.

TR-55 TOMLIB-text of own object not existing (please DUE)

Bedeutung

Der TOMLIB-Text des eigenen Objekts existiert nicht. Bitte drücken Sie DUE.

TR-57 Do you want to overwrite (add) TOMLIB-text ? (Y/N)

Bedeutung

Kontrollfrage, ob Sie den TOMLIB-Text überschreiben (hinzufügen) wollen.

TR-58 Incorporate-function correct finished

Bedeutung

Die Funktion `Incorporate` wurde fehlerfrei beendet.

TR-59 No input-object existing

Bedeutung

Es sind keine zu übertragenden Objekte vorhanden.

TR-60 Copy has type: DATA-AREA

Bedeutung

Generiertes Copy-Element ist vom Typ `DATA-Area`

TR-61 Copy has type: MOVE-statements

Bedeutung

Generiertes Copy-Element ist vom Typ `MOVE-Statement`

TR-62 Copy has type: INIT-statements

Bedeutung

Generiertes Copy-Element ist vom Typ `INIT-Statement`

TR-63 Copy has type: FILE-definition

Bedeutung

Generiertes Copy-Element ist vom Typ `FILE-Definition`

TR-65 Object not referenced by Copy (Copy already exists as MOVE-statement)

Bedeutung

Unter dem angegebenen Copy-Namen existiert bereits ein INL-Objekt, welches allerdings als MOVE-Statement erzeugt worden ist. (Von MOVE-Statements können keine Beziehungen ausgehen.)

TR-70 Copy/Rename completed

Bedeutung

Die Copy-/Rename-Funktion wurde beendet.

TR-71 From-type and To-type different. Do you want to copy/rename ? (Y/N)

Bedeutung

Kontrollfrage, ob die Copy-/Rename-Funktion auch den Objekttyp ändern soll.

TR-72 Please correct the marked fields

Bedeutung

Sie müssen in die markierten Feldern die korrekten Werte eintragen.

TR-74 Copy/Rename completed, but REL-TYPE OR OCCURS must have been reset

Bedeutung

Die Copy-Funktion bzw. Rename-Funktion mit Typ-Änderung wurde durchgeführt, jedoch trat mindestens einer der folgenden Fälle ein:

- Beziehungstyp wurde auf `Blank` gesetzt, weil eine erweiterte Beziehung nicht mehr möglich war.
- Die OCCURS-Angabe wurde auf `Blank` gesetzt, weil eine `Blank`-Beziehung von SRL nach FLD oder von SKE nach FLD entstand und die OCCURS-Angabe einen Wert zwischen 0 und 255 oder `Blank` haben muß.
- Die OCCURS-Angabe wurde auf `Blank` gesetzt, weil eine `Blank`-Beziehung von SRL nach SKE entstand und die OCCURS-Angabe `Blank` sein muß.

TR-75 Relationship-type is not conform with object-types

Bedeutung

Erweiterte Beziehungen sind nur zwischen bestimmten Objekttypen zulässig.

TR-76 (Ext.) Relationship found, which is impossible for new referenced type

Bedeutung

Rename-Funktion mit Typ-Änderung: Das geänderte Objekt ist bezogenes Objekt. Die Funktion wurde nicht ausgeführt, da eine unzulässige einfache Beziehung vorlag.

TR-77 (Ext.) Relationship found, which is impossible for new source-type

Bedeutung

Rename-/Copy-Funktion mit Typ-Änderung: Das geänderte Objekt ist beziehendes Objekt. Die Funktion wurde nicht ausgeführt, da eine unzulässige einfache oder erweiterte Beziehung vorlag.

TR-78 Relationship with error in OCCURS found

Bedeutung

Rename-/Copy-Funktion mit Typ-Änderung: Das geänderte Objekt ist beziehendes Objekt. Die Funktion wurde nicht ausgeführt, da in einer Blank-Beziehung unzulässige OCCURS-Angaben vorlagen (Siehe TR-74).

TR-79 Error by reading relationships

Bedeutung

Bei Einlesen der Beziehungen entsprechend der Verkettung konnte eine Beziehung nicht gelesen werden (wahrscheinlich zerstörte Verkettung). Zu reparieren mit den Servicefunktionen zur Administration `Save into TOMLIB` und anschließend `Restore from TOMLIB`.

TR-80 TOMREFI is not reestablished

Bedeutung

TOMREFI wurde nicht neu angelegt.

TR-85 TOM-TI ended with errors

Bedeutung

Bei der Bearbeitung des TOM-TI-Langtextes lieferte TOM-TI einen fehlerhaften Return-Code (z. B. wegen Pubspace-Überlauf).

TR-86 Error by read of work-file

Bedeutung

Fehler beim Lesen einer Arbeitsdatei (BS2000-Zwischendatei)

TR-87 Error by generation of work-file

Bedeutung

Fehler beim Anlegen einer Arbeitsdatei (BS2000-Zwischendatei)

TR-88 ;

Bedeutung

Dieser Text wurde von der N2Charts-Routine geliefert.

TR-89 Error loading N2Charts

Bedeutung

Beim Laden von TOM-REFG trat ein Fehler auf.

TR-90 Wanted Data Dictionary does not exist, do you want to init it ? (Y/N)

Bedeutung

Das gesuchte Data Dictionary existiert noch nicht. Sie können es neu anlegen, wenn Sie `y` eingeben.

- TR-92 No object found
- Bedeutung**
In der Export-Maske TR9300 wurde ein Objekt angegeben, das im Data Dictionary nicht existiert.
- TR-94 Error by incorporate : invalid type or invalid relationship or no space
- Bedeutung**
Fehler bei der Servicefunktion Incorporate: Bei der Übertragung von Beziehungen wurde entweder ein unzulässiger Objekttyp oder eine unzulässige Beziehung entdeckt oder der interne Beziehungs-Zähler hat den Wert 9999 erreicht. Es wird nur die Bearbeitung des betroffenen Objekts abgebrochen.
- TR-95 Error by incorporate : no space for relationships
- Bedeutung**
Fehler bei der Servicefunktion Incorporate: Der interne Beziehungs-Zähler hat den Wert 9999 erreicht. Es wird nur die Bearbeitung des betroffenen Objekts abgebrochen.
- TR-96 Error by incorporate : invalid type detected
- Bedeutung**
Fehler bei der Servicefunktion Incorporate: Ein Objekt, das übertragen werden soll, hat einen unzulässigen Typ.
- TR-97 Error by incorporate : invalid relationship detected
- Bedeutung**
Fehler bei der Servicefunktion Incorporate: Ein Objekt, das übertragen werden soll, ist Teil einer unzulässigen Beziehung.
- TR-98 Internal error in TOM-REF
- Bedeutung**
Ein interner Fehler, der noch nicht beschrieben wurde, trat auf. Verständigen Sie bitte den Kundendienst.
- TR-99 Error by access of Data Dictionary, file status: ;
- Bedeutung**
Fehler beim Zugriff auf das Data Dictionary. Die Fehlernummer wird angezeigt. Verständigen Sie Ihren Systemverwalter.
- TR-100 Invalid command
- Bedeutung**
Das angegebene Kommando gilt in TOM-REF nicht.

TR-101 Wanted object is not defined

Bedeutung

Das gesuchte Objekt ist nicht definiert.

TR-102 Wanted component-list is not defined

Bedeutung

Die gesuchte Komponenten-Liste ist nicht definiert.

TR-103 Object is already defined

Bedeutung

Das Objekt, welches Sie neu definieren wollen, existiert bereits.

TR-104 This object is part of another object and cannot be deleted

Bedeutung

Das Objekt, welches Sie löschen wollen, steht in einer Beziehung. Es kann deshalb nicht gelöscht werden.

TR-105 No more objects in list

Bedeutung

Abschlußmeldung bei der Auflistung von Objekten: keine weiteren Objekte mehr vorhanden.

TR-106 Error accessing Data Dictionary

Bedeutung

Beim Zugriff auf das Data Dictionary trat ein Fehler auf.

TR-107 No space available

Bedeutung

Es steht kein weiterer Speicherplatz mehr zur Verfügung. (Unter Umständen steht der interne Beziehungs-Zähler auf 9999. Dies ist mit den Servicefunktionen zur Administration `Save into TOMLIB` und anschließend `Restore from TOMLIB` zu reparieren.)

TR-108 Error by delete object

Bedeutung

Beim Löschen eines Objekts trat ein Fehler auf.

TR-109 Copy-name is missing

Bedeutung

Sie haben vergessen, einen Copy-Namen anzugeben.

TR-110 Object-types other than REC

Bedeutung

Objektyp ist nicht REC.

TR-111 Data Dictionary has been saved

Bedeutung

Vollzugsmeldung, wenn Sie Ihr Data Dictionary in die TOMLIB sichern konnten.

TR-112 Data Dictionary has been erased after save

Bedeutung

Vollzugsmeldung, wenn Sie Ihr Data Dictionary löschen konnten. Im nächsten Schritt wird TOM-REF beendet.

TR-113 Component-list not found

Bedeutung

Die angegebene Komponenten-Liste konnte nicht gefunden werden.

TR-114 Password is missing or invalid

Bedeutung

Sie haben vergessen, ein Paßwort anzugeben, oder Sie haben ein falsches angegeben.

TR-115 Command not yet allowed

Bedeutung

An dieser Stelle kann das Kommando nicht verarbeitet werden.

TR-116 Error by LOCK

Bedeutung

Fehler beim LOCK: Das Objekt konnte nicht gesperrt werden.

TR-117 Error by UNLOCK

Bedeutung

Fehler beim UNLOCK: Das Objekt konnte nicht freigegeben werden.

TR-118 Data Dictionary is restored

Bedeutung

Erfolgsmeldung, wenn Sie Ihr Data Dictionary wiederherstellen konnten.

TR-119 Object is not used by any object

Bedeutung

Das angegebene Objekt steht nicht in der Beziehungsliste eines anderen Objekts.

TR-120 `Command is ambiguous`

Bedeutung

Die angegebene Abkürzung für ein Kommando lässt sich nicht eindeutig interpretieren.

TR-121 `Error by creating component-list`

Bedeutung

Beim Erzeugen der Komponenten-Liste trat ein Fehler auf.

TR-122 `Component-list is created`

Bedeutung

Erfolgsmeldung, wenn Sie die Komponenten-Liste erzeugen konnten.

TR-123 `Object has no relationships`

Bedeutung

Das angegebene Objekt hat keine Beziehungen zu anderen Objekten.

TR-124 `No not-referenced objects found`

Bedeutung

Alle Objekte sind in Beziehungen enthalten.

TR-125 `No undefined objects referenced`

Bedeutung

Alle Objekte, die in Beziehungen stehen, sind auch definiert.

TR-126 `Error by restore of Data Dictionary`

Bedeutung

Bei der Funktion `Restore from TOMLIB` trat ein Fehler auf.

TR-127 `Object type is not valid`

Bedeutung

Der angegebene Objekttyp ist gemäß Strukturkatalog nicht erlaubt. Sie können den Strukturkatalog ändern mit den Assemblermakros, die in Kapitel 10 beschrieben sind (Siehe A.3).

TR-128 `Command is not allowed in this state`

Bedeutung

In der Maske, in der Sie arbeiten, können Sie dieses Kommando nicht eingeben.

TR-129 `Since no previous entries, both component and type must be entered`

Bedeutung

Sie müssen sowohl den Komponentennamen als auch den Typ angeben, da Sie bisher noch keinen anderen Eintrag gemacht haben.

- TR-130 `Data Dictionary is empty or does not exist`
- Bedeutung**
Das angegebene Data Dictionary ist leer oder existiert nicht.
- TR-131 `Data Dictionary is not shareable`
- Bedeutung**
Sie wollen auf ein Data Dictionary zugreifen, für das Sie keine Zugriffsberechtigung haben.
- TR-132 `Error by access of Data Dictionary, DMS-code(dec): ;`
- Bedeutung**
DMS-Fehler beim Zugriff auf das Data Dictionary: Die Fehlernummer wird dezimal ausgegeben.
- TR-133 `Error by OPEN of Data Dictionary`
- Bedeutung**
Beim Öffnen des Data Dictionary trat ein Fehler auf.
- TR-134 `Error by CLOSE of Data Dictionary`
- Bedeutung**
Beim Schließen des Data Dictionary trat ein Fehler auf.
- TR-135 `Userid must be filled in`
- Bedeutung**
Sie müssen Ihre Benutzerkennung angeben.
- TR-136 `Identification must be filled in`
- Bedeutung**
Sie müssen Ihr Namenskürzel angeben.
- TR-137 `Data Dictionary must be filled in`
- Bedeutung**
Sie müssen ein Data Dictionary angeben.
- TR-139 `Delete all paired objects is not allowed with dialog control`
- Bedeutung**
Im Dialog können Sie nicht alle Objekte löschen, die paarig in Beziehungen enthalten sind.
- TR-140 `Project must be filled in`
- Bedeutung**
Sie müssen den Projektnamen angeben.

TR-141 Start- or end-name only in conjunction with object-type filled in

Bedeutung

Sie können nur dann eine Auswahl über den Start- und Ende-Namen treffen, wenn Sie auch den Objekttyp angeben.

TR-142 End-name value must not be less than start-name value

Bedeutung

Der Endwert darf alphabetisch nicht kleiner als der Anfangswert sein.

TR-143 Component-list only without other incorporate specifications

Bedeutung

Wenn eine Komponentenliste angegeben ist, dürfen bei `Incorporate` keine weiteren Angaben gemacht werden.

TR-144 Services not allowed in command, already in use

Bedeutung

Sie befinden sich bereits bei den `Services` und können sie deshalb nicht noch einmal aufrufen.

TR-145 Please input Y OR N

Bedeutung

Sie können hier nur `Y` oder `N` angeben.

TR-146 Object is not defined as free type

Bedeutung

Der angegebene Objekttyp ist nicht als benutzereigener Typ definiert.

TR-147 Object has no relationships for this type

Bedeutung

Der angegebene Beziehungstyp ist nicht im Strukturkatalog enthalten.

TR-149 Object is not used by objects of this type

Bedeutung

Das Objekt wird nicht von Objekten dieses Typs benutzt.

TR-150 Object is already defined in central Data Dictionary

Bedeutung

Das angegebene Objekt ist bereits im zentralen Data Dictionary definiert.

TR-151 Invalid level-index

Bedeutung

Der Index-Level ist ungültig.

TR-152 Invalid picture-use priority

Bedeutung

Die Angabe für `Picture-Use-Priority` ist unzulässig.

TR-153 Object not filled in

Bedeutung

Sie müssen einen Objektnamen angeben.

TR-154 File control/description not C Or D

Bedeutung

Der Parameter muß entweder `C` für `Control` oder `D` für `Description` sein.

TR-155 None or too many copy-types marked

Bedeutung

Sie haben entweder keinen oder zu viele Copy-Typen markiert.

TR-156 Invalid relationshiptype

Bedeutung

Die angegebene Beziehung ist gemäß Strukturkatalog nicht erlaubt. Sie können den Strukturkatalog ändern mit dem Assemblermakros, die in Kapitel 10 beschrieben sind (Siehe A.3).

TR-157 From-object must be filled in

Bedeutung

Sie müssen ein Ausgangsobjekt angeben.

TR-158 To-object must be filled in

Bedeutung

Sie müssen ein Zielobjekt angeben.

TR-159 Invalid start level

Bedeutung

Der `Start-level` wurde falsch gewählt.

TR-160 Function must be r Or c

Bedeutung

Sie müssen als Funktion entweder `r` (`Rename`) oder `c` (`Copy`) angeben.

TR-161 Error in relationships

Bedeutung

Fehler in den Beziehungen.

TR-162 Please input both parameter-name and type

Bedeutung

Sie müssen sowohl den Parameter-Namen als auch den Typ angeben.

TR-163 Please input copy-name or component-list-name

Bedeutung

Geben Sie einen Copy-Namen oder den Namen einer Komponentenliste an.

TR-164 Please input copy-name or name of component-list

Bedeutung

Geben Sie einen Copy-Namen oder den Namen einer Komponentenliste an.

TR-165 Error found, print generated

Bedeutung

Bei der Prüfung trat ein Fehler auf. Die Liste wird nach SYSLST ausgegeben.

TR-166 No errors found

Bedeutung

Es konnten keine Fehler gefunden werden.

TR-167 No objects selected for print

Bedeutung

Sie haben keine Objekte für den Druck ausgesucht.

TR-169 ;

Bedeutung

Dieser Text wurde von der N2Charts-Routine geliefert.

TR-171 No objects selected

Bedeutung

Es wurden keine Objekte ausgewählt.

TR-172 Function-key not allowed

Bedeutung

Sie können hier keinen Function-Key angeben.

TR-174 Primary and central Data Dictionary must have different names

Bedeutung

Sie müssen für das zentrale und das private Data Dictionary verschiedene Namen angeben.

TR-175 Alternative Data Dictionary is empty or does not exist

Bedeutung

Das angegebene alternative Data Dictionary ist leer oder existiert nicht.

TR-176 Alternative Data Dictionary is not shareable

Bedeutung

Für das angegebene alternative Data Dictionary haben Sie keine Zugriffsberechtigung.

TR-177 Error by access of alternative Data Dictionary, DMS-code(dec): ;

Bedeutung

DMS-Fehler beim Zugriff auf das alternative Data Dictionary: Die Fehlernummer wird dezimal ausgegeben.

TR-178 Error by OPEN of alternative Data Dictionary

Bedeutung

Fehler beim Öffnen des alternativen Data Dictionary.

TR-179 Error by access of TOMREFI, do you want to reestablish ? (Y/N)

Bedeutung

Beim Zugriff auf die invertierte Data Dictionary-Datei TOMREFI kam es zu einem Fehler. Damit TOM-REF arbeiten kann, muß diese Datei vorhanden sein.

TR-180 TOMREFI not reestablished

Bedeutung

Es wurde keine neue TOMREFI-Datei erstellt.

TR-181 Component-list already exist, do you want to overwrite ? (Y/N)

Bedeutung

Kontrollfrage, ob eine existierende Komponenten-Liste überschrieben werden soll.

TR-183 Central-TOMLIB not existing or not accessable

Bedeutung

Auf die zentrale TOMLIB kann nicht zugegriffen werden oder sie existiert nicht.

TR-184 Alternativ-TOMLIB not existing or not accessable

Bedeutung

Auf die alternative TOMLIB kann nicht zugegriffen werden oder sie existiert nicht.

TR-185 TOMLIB existing but not accessable

Bedeutung

Auf die existierende TOMLIB kann nicht zugegriffen werden.

TR-186 Creation of TOMLIB unsuccessful - give new parameters

Bedeutung

Es konnte keine TOMLIB erzeugt werden, da entweder keine BASETOMLIB angegeben wurde oder auf die BASETOMLIB nicht zugegriffen werden kann oder kein Speicherplatz mehr vorhanden ist. Geben Sie neue Parameter ein.

TR-188 TOM-member is not present in TOMLIB

Bedeutung

In der TOMLIB konnte das Objekt nicht gefunden werden.

TR-189 Do you want to restore Data Dictionary ?

Bedeutung

Kontrollfrage, ob das Data Dictionary aus der TOMLIB geholt werden soll.

TR-190 Error in name of Data Dictionary

Bedeutung

Sie haben unzulässige Zeichen für den Data Dictionary-Namen verwendet.

TR-191 Data Dictionary is not initiated

Bedeutung

Das angegebene Data Dictionary wurde noch nicht initialisiert.

TR-192 Error by OPEN of central Data Dictionary

Bedeutung

Fehler beim Öffnen des zentralen Data Dictionary.

TR-193 Error by access of TOMLIB

Bedeutung

Fehler beim Zugriff auf die TOMLIB.

TR-194 Error by access of alternative TOMLIB

Bedeutung

Fehler beim Zugriff auf die alternative TOMLIB.

TR-195 Error by initiate Data Dictionary

Bedeutung

Beim Initialisieren eines Data Dictionary trat ein Fehler auf.

TR-196 Data Dictionary is initiated

Bedeutung

Data Dictionary wurde angelegt.

TR-197 Error by reestablish TOMREFI

Bedeutung

Beim Wiederherstellen der TOMREFI-Datei trat ein Fehler auf.

TR-198 TOMREFI reestablished

Bedeutung

Inverse Data Dictionary-Datei wurde angelegt.

TR-199 Do you really want to reestablish TOMREFI ? (Y/N)

Bedeutung

Kontrollfrage, ob die inverse Data Dictionary-Datei angelegt werden soll.

TR-200 Do you really want to initiate Data Dictionary ? (Y/N)

Bedeutung

Kontrollfrage, ob das angegebene Data Dictionary initialisiert werden soll.

TR-201 Do you really want to initiate existing Data Dictionary ? (Y/N)

Bedeutung

Kontrollfrage, ob ein existierendes Data Dictionary überschrieben werden soll.

TR-202 It is not allowed to initiate the central Data Dictionary

Bedeutung

Das zentrale Data Dictionary darf nicht initialisiert werden.

TR-203 Error in name of central Data Dictionary

Bedeutung

Fehlerhafter Name für das zentrale Data Dictionary.

TR-205 Central Data Dictionary is empty or does not exist

Bedeutung

Das zentrale Data Dictionary ist leer oder existiert nicht.

TR-206 Central Data Dictionary is not shareable

Bedeutung

Das zentrale Data Dictionary ist nicht mehrfachbenutzbar.

TR-209 Structure-level greater than 50

Bedeutung

Delete-Structure-Funktion: Die zu löschende Struktur darf nur eine Tiefe von maximal 50 haben. Die Funktion wurde nicht ausgeführt.

TR-210 From-object and To-object must be different

Bedeutung

Zwischen Ausgangsobjekt und Zielobjekt muß ein Unterschied bestehen.

TR-212 Error by exclusive OPEN

Bedeutung

Fehler bei den Servicefunktionen Initiate TOM-REF Data Dictionary oder Reestablish TOMRFI oder Restore from TOMLIB: Der für die Durchführung der Funktion nötige exklusive OPEN konnte nicht durchgeführt werden. (Unter Umständen arbeitet ein anderer Anwender mit dem Data Dictionary.)

TR-215 Error by access of Data Dictionary in foreign userid

Bedeutung

Fehler beim Zugriff auf das Data Dictionary: Das angegebene Data Dictionary ist nicht vorhanden und kann, da es sich um eine fremde Benutzerkennung handelt, nicht eingerichtet werden.

TR-216 Error by access of TOMRREFI in foreign userid

Bedeutung

Fehler beim Zugriff auf die TOMRFI-Datei. Die Datei ist nicht vorhanden und kann, da es sich um eine fremde Benutzerkennung handelt, nicht eingerichtet werden.

TR-220 You must read the object before deleting

Bedeutung

Vor dem Löschen wird das Objekt angezeigt.

TR-221 DD-Text is not deleted

Bedeutung

Der Data Dictionary-Text wurde nicht gelöscht.

TR-222 DD-Text is deleted

Bedeutung

Der Data Dictionary-Text wurde gelöscht.

TR-223 LIB-Text is not deleted

Bedeutung

Der LIB-Text wurde nicht gelöscht.

TR-224 LIB-Text is deleted

Bedeutung

Der LIB-Text wurde gelöscht.

- TR-225 `No text deleted`
- Bedeutung**
Es wurde kein Text gelöscht.
- TR-226 `Both texts are deleted`
- Bedeutung**
Beide Texte wurden gelöscht.
- TR-227 `Error in deleting Data Dictionary-Text`
- Bedeutung**
Fehler beim Löschen des Data Dictionary-Texts.
- TR-228 `Error in deleting LIB-Text`
- Bedeutung**
Fehler beim Löschen des LIB-Texts.
- TR-231 `Invalid answer - valid is: ;`
- Bedeutung**
Fehlerhafte Antwort: Die richtige Antwort wird angezeigt.
- TR-232 `Please input both type and name`
- Bedeutung**
Sie müssen sowohl den Namen als auch den Typ des Objekts angeben.
- TR-233 `Please input type and name for both from-object and to-object`
- Bedeutung**
Sie müssen sowohl für das Ausgangsobjekt als auch für das Zielobjekt den Namen und den Typ angeben.
- TR-240 `Objectname is not predefined - please input in the command`
- Bedeutung**
Der Objektname ist nicht bekannt; geben Sie den richtigen ein.
- TR-241 `Objecttype ist not predefined - please input in the command`
- Bedeutung**
Der Objekttyp ist nicht bekannt; geben Sie den richtigen ein.
- TR-242 `Objecttype and name are not predefined - please input in the command`
- Bedeutung**
Typ und Name des Objekts sind nicht bekannt; geben Sie die richtigen ein.

TR-250 Object is locked for update from another user

Bedeutung

Das Objekt ist von einem anderen Anwender gegen Änderungen gesperrt.

TR-251 File to initiate used by another user

Bedeutung

Die zu initialisierende Datei wird schon von einem anderen Anwender verwendet.

TR-252 DD is locked by another user

Bedeutung

Das Data Dictionary ist von einem anderen Anwender gesperrt.

TR-255 Data Dictionary is not restored

Bedeutung

Das Data Dictionary ist nicht wiederhergestellt.

TR-256 Restore stopped, invalid object-type detected : ;

Bedeutung

Die Restore-Funktion wurde abgebrochen: der fehlerhafte Objekttyp wird angezeigt.

TR-257 Restore stopped, invalid relationship detected : ;

Bedeutung

Die Restore-Funktion wurde abgebrochen: die fehlerhafte Beziehung wird angezeigt.

TR-258 Record found with unknown format-notation

Bedeutung

Das TOMLIB-Element, auf das die Restore-Funktion angewendet werden sollte, ist kein Data Dictionary.

TR-260 Object-name is not valid

Bedeutung

Unzulässiger Objektname.

TR-261 Object-type is not valid

Bedeutung

Unzulässiger Objekttyp.

TR-262 Name of component-list is not valid

Bedeutung

Unzulässiger Name für die Komponentenliste.

- TR-264 Element name is not valid
- Bedeutung**
Unzulässiger Elementname.
- TR-270 If object-name contains blanks, no TOMLIB-text will be handled
- Bedeutung**
Wenn der Objektname Blanks enthält, wird kein TOMLIB-Text bearbeitet.
- TR-271 Object-name begins with `_, {` or `}`. No handling of TOMLIB-text
- Bedeutung**
Der Objektname beginnt mit `_, {` oder `}`. Es wird kein TOMLIB-Text bearbeitet.
- TR-275 Error by save into TOMLIB
- Bedeutung**
Fehler beim Abspeichern in die TOMLIB.
- TR-280 REF-name must be TOM(CA,DA,DS,JA,RF,TA)(1,..,9)
- Bedeutung**
Der REF-Name muß TOM(CA,DA,DS,JA,RF,TA)(1,..9) sein.
- TR-281 You must not incorporate from your own Data Dictionary
- Bedeutung**
Die Incorporate-Funktion darf nicht auf das eigene Data Dictionary angewendet werden.
- TR-282 Please fill object-type or comp.-list in
- Bedeutung**
Bitte tragen Sie den Objekttyp oder die Komponentenliste ein.
- TR-283 If object-type = *T start-name and end-name must not be entered
- Bedeutung**
Wenn für Objekttyp *T angegeben wird, dürfen Start- und Ende-Name nicht eingegeben werden.
- TR-284 Please mark Dialog-control OR Immediate action
- Bedeutung**
Bitte markieren Sie Dialog-control oder Immediate action.
- TR-285 Object-type must be a defined type or *T
- Bedeutung**
Für den Objekttyp muß ein definierter Objekttyp oder *T angegeben werden.

TR-298 Function is not yet available in this installation

Bedeutung

Die Servicefunktion `Export` steht noch nicht zur Verfügung.

TR-299 Function is not implemented in this installation

Bedeutung

Sie haben eine Funktion aufgerufen, die in Ihrer Installation nicht implementiert ist.

TR-300 Protocol is active

Bedeutung

Die Protokollierung ist eingeschaltet.

TR-301 Protocol is deactivated

Bedeutung

Die Protokollierung ist ausgeschaltet.

TR-302 Trace is active

Bedeutung

Die Trace-Funktion ist eingeschaltet.

TR-303 Trace is deactivated

Bedeutung

Die Trace-Funktion ist ausgeschaltet.

TR-304 DD-Text is active

Bedeutung

Die Textbearbeitung (TOM-REF-Text) für das aktuelle Objekt ist aktiviert.

TR-305 LIB-Text is active

Bedeutung

Die Textbearbeitung mit dem TOM-TI für das aktuelle Objekt ist aktiviert

TR-306 Text is deactivated

Bedeutung

Die Textbearbeitung ist ausgeschaltet.

TR-403 Relationship does not exist

Bedeutung

Die Beziehung existiert nicht.

TR-404 Relationship already exists; only extension is possible

Bedeutung

Die Beziehung existiert bereits, so daß nur eine Erweiterung möglich ist.

TR-405 Extended relationship does not exist

Bedeutung

Die erweiterte Beziehung existiert nicht.

TR-406 Extended relationship already exists

Bedeutung

Die erweiterte Beziehung existiert bereits.

TR-407 Simple relationship does not exist, extension is not possible

Bedeutung

Diese einfache Beziehung existiert nicht, eine Erweiterung ist nicht möglich.

TR-408 Type of extended relationship is not allowed

Bedeutung

Der Typ der erweiterten Beziehung existiert nicht.

TR-409 Type of relationship is not conform

Bedeutung

Der Typ der Beziehung stimmt nicht.

TR-411 Do you want to delete relationship ? (Y/N)

Bedeutung

Kontrollfrage, ob Sie die Beziehung löschen wollen.

TR-412 Do you want to delete extension ? (Y/N)

Bedeutung

Kontrollfrage, ob Sie die Erweiterung entfernen wollen.

TR-415 Syntax-error in relationship-command

Bedeutung

Syntaxfehler im Relationship-Kommando.

TR-416 Given type is here invalid

Bedeutung

Der angegebene Typ ist unzulässig: Bei einem Relationship-Kommando darf als Typ nur einer der Standard-Beziehungstypen oder #SIR angegeben werden. Wenn im Kommando EXTENDED angegeben wird, ist auch #SIR unzulässig.

TR-417 Internal error in handling of relationship-command

Bedeutung

Interner Fehler bei Ausführung des Relationship-Kommandos.

TR-418 Invalid function of relationship-command

Bedeutung

Unzulässige Funktion beim Relationship-Kommando: Auf Beziehungen können nur die Funktionen `Display`, `Create`, `Change` und `Delete` angewendet werden.

TR-419 Internal error in handling of extended relationship

Bedeutung

Interner Fehler bei der Bearbeitung einer erweiterten Beziehung.

TR-420 Invalid command (blank found)

Bedeutung

Fehlerhaftes Kommando (es wurde ein Blank gefunden).

TR-421 Syntax-error in command: invalid , found

Bedeutung

Syntaxfehler im Kommando: Es wurde ein unzulässiges Komma gefunden.

TR-422 Syntax-error in command: invalid = found

Bedeutung

Syntaxfehler im Kommando: Es wurde ein unzulässiges Gleichheitszeichen gefunden.

TR-423 Syntax-error in command: object-name missing

Bedeutung

Syntaxfehler im Kommando: Angabe des Objektname fehlt.

TR-424 Syntax-error in command: object-type missing

Bedeutung

Syntaxfehler im Kommando: Angabe des Objekttyps fehlt.

TR-425 Invalid length of object-type

Bedeutung

Falsche Länge für Objekttyp.

TR-426 Length of object-name not corresponding to object-type

Bedeutung

Die Länge des Objektname stimmt nicht mit der maximalen Länge für Objektname überein, die durch den Objekttyp festgelegt ist.

TR-428 Invalid object-type

Bedeutung

Unzulässiger Objekttyp.

TR-429 Invalid uses-type

Bedeutung

Unzulässige Angabe für Uses-type.

TR-440 You are already using this service-function !

Bedeutung

Durch das abgesetzte Kommando wurde diejenige Maske angesteuert, in der das Kommando gegeben wurde.

TR-600 Length of secondary index greater length of attribute

Bedeutung

Der Wert für die Länge des Sekundärindex ist größer als die Attributlänge.

TR-800 Generate must be Y, N, or ' '

Bedeutung

Das Feld Generate muß mit einem der Werte Y, N oder einem Leerzeichen belegt sein.

TR-801 Interrupt must be Y, N, or ' '

Bedeutung

Das Feld Interrupt muß mit einem der Werte Y, N oder einem Leerzeichen belegt sein.

TR-802 Letters must be C, B, or ' '

Bedeutung

Das Feld Letters muß mit einem der Werte B, C oder einem Leerzeichen belegt sein.

TR-803 List must be U or E or N or ' '

Bedeutung

Das Feld List muß mit einem der Werte U, E, N oder einem Leerzeichen belegt sein.

TR-804 Moninfo must be L, D, B, N or ' '

Bedeutung

Das Feld Moninfo muß mit einem der Werte L, D, B, N oder einem Leerzeichen belegt sein.

TR-805 Program type must be D, A or ' '

Bedeutung

Das Feld Program type muß mit einem der Werte D, A oder einem Leerzeichen belegt sein.

TR-806 Runtime password must be D, A or ' '

Bedeutung

Das Feld `Runtime Password` muß mit einem der Werte `D`, `A` oder einem Leerzeichen belegt sein.

TR-807 TAC-type must be F, N, B or ' '

Bedeutung

Das Feld `TAC-type` muß mit einem der Werte `F`, `N`, `B` oder einem Leerzeichen belegt sein.

TR-808 XREF must be L, D, B, N or ' '

Bedeutung

Das Feld `XREF` muß mit einem der Werte `L`, `D`, `B#`, `N` oder einem Leerzeichen belegt sein.

TR-809 Password must be Y, N, or ' '

Bedeutung

Das Feld `Password` muß mit einem der Werte `Y`, `N` oder einem Leerzeichen belegt sein.

TR-810 REQM-area-length must be numerical between 1..99999

Bedeutung

Der Wert im Feld `REQM-area-length` muß numerisch sein und zwischen 1 und 99999 liegen.

TR-811 SESAM-buffer-length must be numerical between 1..32000

Bedeutung

Der Wert im Feld `SESAM-buffer-length` muß numerisch sein und zwischen 1 und 32000 liegen.

TR-812 System-environment must be B, S or ' '

Bedeutung

Das Feld `System-environment` muß mit einem der Werte `B`, `S` oder einem Leerzeichen belegt sein.

TR-813 Terminal-type must be D, O or ' '

Bedeutung

Das Feld `Terminal-type` muß mit einem der Werte `D`, `O` oder einem Leerzeichen belegt sein.

TR-814 Exit must be S, H, F or ' '

Bedeutung

Das Feld `Exit` muß mit einem der Werte S, H, F oder einem Leerzeichen belegt sein.

TR-815 I/O-area-length must be numerical between 256..32764

Bedeutung

Das Feld `I/O-area-length` muß numerisch sein und zwischen 256 und 32764 liegen.

TR-816 KB-length must be numerical between 1..32767

Bedeutung

Das Feld `KB-length` muß numerisch sein und zwischen 1 und 32767 liegen.

TR-817 SPAB-length must be numerical between 1..32767

Bedeutung

Das Feld `SPAB-length` muß numerisch sein und zwischen 1 und 32767 liegen.

TR-818 User-area-length must be nuerical between 1..32764

Bedeutung

Das Feld `User-area-length` muß numerisch sein und zwischen 1 und 32764 liegen.

TR-819 TAC-type must be D, A or ' '

Bedeutung

Das Feld `TAC-type` muß mit einem der Werte D, A oder einem Leerzeichen belegt sein.

TR-820 Call Usage must be B, F, N or ' '

Bedeutung

Das Feld `Call Usage` muß mit einem der Werte B, F, N oder einem Leerzeichen belegt sein.

TR-821 Copy-section must be C, D or ' '

Bedeutung

Das Feld `Copy-section` muß mit einem der Werte C, D oder einem Leerzeichen belegt sein.

TR-822 Picture-use-priority intern must be 1, 2, 3 or ' '

Bedeutung

Das Feld `Picture-use-priority intern` muß mit einem der Werte 1, 2, 3 oder einem NIL-Zeichen belegt sein.

TR-823 `Picture-use-priority extern` must be 1, 2, 3 or ' '

Bedeutung

Das Feld `Picture-use-priority extern` muß mit einem der Werte 1, 2, 3 oder einem NIL-Zeichen belegt sein.

TR-824 `Picture-use-priority work` must be 1, 2, 3 or ' '

Bedeutung

Das Feld `Picture-use-priority work` muß mit einem der Werte 1, 2, 3 oder einem NIL-Zeichen belegt sein.

TR-825 `Value` must be Y, N or ' '

Bedeutung

Das Feld `Value` muß mit einem der Werte Y, N oder einem Leerzeichen belegt sein.

TR-826 `Justified` must be R, L or ' '

Bedeutung

Das Feld `Justified` muß mit einem der Werte R, L oder einem Leerzeichen belegt sein.

TR-828 `Synchronized` must be X or ' '

Bedeutung

Das Feld `Synchronized` muß mit dem Wert X oder einem Leerzeichen belegt sein.

TR-829 `Access-mode` must be S, R, D, I or ' '

Bedeutung

Das Feld `Access-mode` muß mit einem der Werte S, R, D, I oder einem Leerzeichen belegt sein.

TR-830 `Assign to-more assigns` must be Y, N or ' '

Bedeutung

Das Feld `Assigns to more assigns` muß mit einem der Werte Y, N oder einem Leerzeichen belegt sein.

TR-831 `Block contains (-type)` must be C, R or ' '

Bedeutung

Das Feld `Block contains (-type)` muß mit einem der Werte C, R oder einem Leerzeichen belegt sein.

TR-832 `label-record (-type)` must be O, S, N or ' '

Bedeutung

Das Feld `label-records (-type)` muß mit einem der Werte O, S, N oder einem Leerzeichen belegt sein.

TR-833 label-record-More label-records must be Y, N or ' '

Bedeutung

Das Feld `label-records-More label-records` muß mit einem der Werte `Y, N` oder einem Leerzeichen belegt sein.

TR-834 Organization must be S, R, D, I or ' '

Bedeutung

Das Feld `Organization` muß mit einem der Werte `S, R, D, I` oder einem Leerzeichen belegt sein.

TR-835 Record-mode must be F, V or ' '

Bedeutung

Das Feld `Record-mode` muß mit einem der Werte `F, V` oder einem Leerzeichen belegt sein.

TR-836 Record-mode-type must be 1, 2, 3 or ' '

Bedeutung

Das Feld `Record-mode-type` muß mit einem der Werte `1, 2, 3` oder einem NIL-Zeichen belegt sein.

TR-837 Reserve-areas must be Y, N or ' '

Bedeutung

Das Feld `Reserve-areas` muß mit einem der Werte `Y, N` oder einem Leerzeichen belegt sein.

TR-838 Select optional must be Y, N or ' '

Bedeutung

Das Feld `Select optional` muß mit einem der Werte `Y, N` oder einem Leerzeichen belegt sein.

TR-839 Duplicates must be Y, N or ' '

Bedeutung

Das Feld `Duplicates` muß mit einem der Werte `Y, N` oder einem Leerzeichen belegt sein.

TR-840 Key-type must be I, C or ' '

Bedeutung

Das Feld `key-type` muß mit einem der Werte `I, C` oder einem Leerzeichen belegt sein.

TR-841 Connection must be M, O or ' '

Bedeutung

Das Feld `Connection` muß mit einem der Werte `M`, `O` oder einem Leerzeichen belegt sein.

TR-842 Insert mode must be A, M or ' '

Bedeutung

Das Feld `Insert mode` muß mit einem der Werte `A`, `M` oder einem Leerzeichen belegt sein.

TR-843 Order is must be F, L, N, P, I, S, X or ' '

Bedeutung

Das Feld `Order is` muß mit einem der Werte `F`, `L`, `N`, `P`, `I`, `S`, `X` oder einem Leerzeichen belegt sein.

TR-844 Set-selection must be C, L or ' '

Bedeutung

Das Feld `Set-selection` muß mit einem der Werte `C`, `L` oder einem Leerzeichen belegt sein.

TR-845 Indexed must be Y, N or ' '

Bedeutung

Das Feld `Indexed` muß mit einem der Werte `Y`, `N` oder einem Leerzeichen belegt sein.

TR-846 Sorted must be A, D or ' '

Bedeutung

Das Feld `Sorted` muß mit einem der Werte `A`, `D` oder einem Leerzeichen belegt sein.

TR-847 User mode must be A, E, U or ' '

Bedeutung

Das Feld `User mode` muß mit einem der Werte `A`, `E`, `U` oder einem Leerzeichen belegt sein.

TR-848 Base schema must be UDS, SES, DVS, LEA or ' '

Bedeutung

Das Feld `Base schema` muß mit einem der Werte `UDS`, `SES`, `DVS`, `LEA` oder einem Leerzeichen belegt sein.

TR-849 Modification must be M, C or ' '

Bedeutung

Das Feld `Modification` muß mit einem der Werte `M`, `C` oder einem Leerzeichen belegt sein.

- TR-850 View type must be BR, AV, UV or ' '
- Bedeutung**
Das Feld `view type` muß mit einem der Werte `BR`, `AV`, `UV` oder einem Leerzeichen belegt sein.
- TR-851 Key type must be DB, DEF or ' '
- Bedeutung**
Das Feld `key type` muß mit einem der Werte `DB`, `DEF` oder einem Leerzeichen belegt sein.
- TR-1000 Specified default-option needs datatype CHAR, VARCHAR with length > 17 Bytes
- Bedeutung**
Für die Anweisung im Feld `default option` haben Sie entweder einen falschen Datentyp angegeben oder die Zeichenkette zu kurz definiert.
- TR-1001 If datatype not specified, no other SQL-data allowed
- Bedeutung**
Wenn Sie keinen Datentyp angeben, können Sie auch die weiteren Felder nicht beschreiben.
- TR-1002 Datatype wrong
- Bedeutung**
Für die nachfolgenden Angaben haben Sie den falschen Datentyp gewählt.
- TR-1003 Length must be numerical between 1..32000
- Bedeutung**
Die Länge des Datentyps darf höchstens 32000 Zeichen betragen.
- TR-1004 For datatype CHAR length must be numerical between 1..256
- Bedeutung**
Der Datentyp `CHARACTER` darf höchstens 256 Zeichen lang sein.
- TR-1005 For datatype VARCHAR length must be numerical between 1..32000
- Bedeutung**
Der Datentyp `VARCHAR` darf höchstens 32000 Zeichen lang sein.
- TR-1006 For datatype NUM and DEC length/precision must be numerical between 1..31
- Bedeutung**
Die Länge und/oder die Vorkommastellen der Datentypen `NUMERIC` und `DECIMAL` dürfen 31 Stellen nicht überschreiten.

TR-1007 For datatype FLO length must be numerical beween 1..53

Bedeutung

Die Länge des Datentyps FLOAT 53 Stellen nicht überschreiten.

TR-1008 For the specified datatype no length allowed

Bedeutung

Für den angegebenen Datentyp ist keine Längenangabe erlaubt.

TR-1009 Scale must be numeric between 1..31

Bedeutung

Sie dürfen höchstens 31 Nachkommastellen angeben.

TR-1010 For Datatype NUM and DEC scale must be numerical between 1..31

Bedeutung

Für die Datentypen NUMERIC und DECIMAL dürfen Sie höchstens 31 Nachkommastellen angeben.

TR-1011 For the specified datatype no scale specification allowed

Bedeutung

Der angegebene Datentyp hat keine Nachkommastellen.

TR-1012 Invalid value for default-option 1 allowed

Bedeutung

Im Feld `default-option` dürfen Sie nur ein Literal, eine vordefinierte SQL-Funktion oder Leerzeichen angeben.

TR-1013 Default-option LITERAL not allowed for datatype TIM, STA and DAT

Bedeutung

Wenn Sie einen der Datentypen TIME, TIMESTAMP oder DATE gewählt haben, dürfen Sie im Feld `default-option` LITERAL nicht angeben.

TR-1014 For datatype VAR occurence not allowed.

Bedeutung

Eine mutiple Spalte darf nicht den Datentyp VARCHAR haben.

TR-1015 Occurence must be numerical between 1..255

Bedeutung

Die Häufigkeit des Auftretens eines Datenfeldes bzw. einer Spalte darf nur zwischen 1 und 255 liegen.

- TR-1016 Null-constraint must be NOT NULL or empty
- Bedeutung**
Bei der Prüfung auf Nullwerte können Sie nur NOT NULL oder Leerzeichen angeben.
- TR-1017 Default-option DATETIME only allowed for datatype DAT, TIM and STA
- Bedeutung**
Die SQL-Funktion DATETIME dürfen Sie im Feld `default-option` nur angeben, wenn Sie einen der Datentypen DATE, TIME oder TIMESTAMP angegeben haben.
- TR-1018 Specified default-option needs datatype CHA, VAR with length > 23 Bytes
- Bedeutung**
Für die angegebene Anweisung im Feld `default-option` müssen Sie einen der Datentypen CHARACTER oder VARCHAR mit einer Länge größer 32 Bytes angeben.
- TR-1019 Decimal scale must be smaller or equal length/precision
- Bedeutung**
Der Wert im Feld `Deciaml scale` darf höchstens gleich der Länge des gesamten Wertes oder der Vorkommastellen sein.
- TR-1020 Table-type must be "VIEW", "TABLE" or empty
- Bedeutung**
Sie haben im Feld `table-type` eine falsche Angabe gemacht.
- TR-1021 Table-stale must bei "OLD", "NEW" or empty
- Bedeutung**
Sie haben im Feld `table-style` eine falsche Angabe gemacht.
- TR-1022 For table "VIEW" table-style must be empty
- Bedeutung**
Wenn Sie einen View erzeugen wollen, dürfen Sie im Feld `table-style` keine Angabe machen.
- TR-1023 View-storage must be "LONG, "SHORT" or empty
- Bedeutung**
Sie haben im Feld `view-storage` eine falsche Angabe gemacht.
- TR-1024 For table-type "TABLE" view-storage must be empty
- Bedeutung**
Wenn Sie eine Tabelle erzeugen wollen, dürfen Sie im Feld `view-storage` keine Angabe machen.

TR-1025 `Check-option must be "Y", "N" or empty`

Bedeutung

Sie haben im Feld `check-option` eine falsche Angabe gemacht.

TR-1026 `For table-type "TABLE" check-option must be empty`

Bedeutung

Wenn Sie eine Tabelle erzeugen wollen, dürfen Sie im Feld `check-option` keine Angabe machen.

TR-1030 `Length must be numerical between 1..256`

Bedeutung

Der Text darf nicht länger als 256 Zeichen sein.

TR-1040 `Column-usage must be BOTH, INPUT or OUTPUT`

Bedeutung

Sie haben im Feld `column-usage` eine falsche Angabe gemacht.

TR-1041 `First-occurence must be numerical between 1..255`

Bedeutung

Das erste Auftreten des Feldes müssen Sie mit einer Zahl zwischen 1 und 255 angeben.

TR-1042 `Last-occurence must be numerical between 1..255`

Bedeutung

Das letzte Auftreten des Feldes müssen Sie mit einer Zahl zwischen 1 und 255 angeben.

TR-1043 `Last-occurence must be greater then first-occurrence`

Bedeutung

Der Wert des letzten Auftretens eines Feldes muß größer sein, als der Wert des ersten Auftretens.

TR-1050 `Check-storage: only SHORT or LONG allowed`

Bedeutung

Im Feld `Check-storage` sind nur die Werte `SHORT` oder `LONG` erlaubt.

TR-1060 `Primary-allocation must be numerical`

Bedeutung

Bei der Primärzuweisung dürfen Sie nur ganze Zahlen angeben.

TR-1061 Secondary-allocation must be numerical

Bedeutung

Bei der Sekundärzuweisung dürfen Sie nur ganze Zahlen angeben.

TR-1064 Sharable: only Y or N or space allowed

Bedeutung

Im Feld `Sharable` sind nur die Werte `Y`, `L` oder Leerzeichen erlaubt.

TR-1065 Logging: only Y or N or space allowed

Bedeutung

Im Feld `Logging` sind nur die Werte `Y`, `L` oder Leerzeichen erlaubt.

TR-1066 Destroy: only Y or N or space allowed

Bedeutung

Im Feld `Destroy` sind nur die Werte `Y`, `L` oder Leerzeichen erlaubt.

TR-1067 Percent-free: Input is not numeric

Bedeutung

Im Feld `Percent-free` dürfen Sie nur numerische Werte angeben.

TR-1068 Percent-free: Domain is 10 to 90 percent

Bedeutung

Im Feld `Percent-free` dürfen Sie nur Werte zwischen 10 und 90 Prozent angeben.

TR-1070 Type must be "UNIQUE" or "PRIMARY-KEY"

Bedeutung

Sie haben im Feld `Type` eine falsche Angabe gemacht.

TR-1100 Check-Storage must be "SHORT" or "LONG"

Bedeutung

Sie haben im Feld `check-storage` eine falsche Angabe gemacht. Es sind nur die Werte `SHORT` und `LONG` erlaubt.

TR-1107 Index-Column-length: Domain is 1 to 256

Bedeutung

Im Feld `Index-Column-length` sind nur Werte von 1 bis 256 erlaubt.

TR-1110 Primary-allocation must be numerical between 16..8388601

Bedeutung

Die Primärzuweisung für die Datenbank muß zwischen 16 und 8388601 PAM-Seiten liegen.

TR-1111 First-occurence: Input is not numeric

Bedeutung

Sie müssen im Feld `First occurrence` eine Ganzzahl eingeben.

TR-1112 Secondary-allocation must be numerical between 24..32767

Bedeutung

Die Sekundärzuweisung für die Datenbank muß zwischen 24 und 32767 PAM-Seiten liegen.

TR-1113 Last-occurence: Input is not numeric

Bedeutung

Sie müssen im Feld `Last occurrence` eine Ganzzahl eingeben.

TR-1114 Sharable must be 'Y', 'N' or empty

Bedeutung

Sie haben im Feld `Sharable` eine falsche Angabe gemacht. Sie dürfen nur Y, N oder ein Leerzeichen angeben.

TR-1115 Destroy must be 'Y', 'N' or empty

Bedeutung

Sie haben im Feld `Destroy` eine falsche Angabe gemacht. Sie dürfen nur Y, N oder ein Leerzeichen angeben.

TR-1116 Logging must be 'Y', 'N' or empty

Bedeutung

Sie haben im Feld `Logging` eine falsche Angabe gemacht. Sie dürfen nur Y, N oder ein Leerzeichen angeben.

TR-1117 Percent-free must be numerical between 0..70

Bedeutung

Im Feld `Percent-free` dürfen Sie nur Werte zwischen 0 und 70 angeben.

TR-1140 Column-length must be numerical between 1..256

Bedeutung

Im Feld `Column-length` sind nur numerische Werte zwischen 1 und 256 erlaubt.

TR-1220 If Default-literal specified, Default option must be 'LITERAL'

Bedeutung

Wenn Sie das Feld `Default-literal` mit einem Wert belegt haben, darf im Feld `Default-option` nur der Wert `LITERAL` stehen.

TR-1221 If Default-option is 'LITERAL', Default-literal must be specified

Bedeutung

Wenn Sie im Feld `Default-option` den Wert `LITERAL` angegeben haben, müssen Sie das Feld `Default-literal` mit einem Wert belegen.

TR-2001 Additional messages are written in SYSLST

Bedeutung

Bei der Bearbeitung einer Print-Funktion wurden zusätzliche Hinweise in die Liste aufgenommen. (Die Hinweis-Zeilen beginnen jeweils mit ***)

TR-2002 Wanted objects and directories written

Bedeutung

Die Bearbeitung der Print-Funktion ist fehlerfrei durchgeführt worden.

TR-2003 Output-file must be SYSLST or specified file

Bedeutung

In der Maske TR9070 darf für `Output-file` nur entweder `SYSLST` angekreuzt oder ein expliziter Datei-Teilname angegeben werden.

TR-2004 At least one filename must be entered

Bedeutung

In der Maske TR9070 muß für `Output-file` entweder `SYSLST` angekreuzt oder ein expliziter Datei-Teilname angegeben werden.

TR-2006 Text must be N, S, L OR B

Bedeutung

In der Maske TR9070 sind im Feld `Text` nur folgende Werte erlaubt: N, S, L, B.

TR-2007 Please fill in "Other-type"

Bedeutung

Ein Objekt-Verzeichnis (Directory) für einen Other-Typ kann nur bei explizit angegebenem Other-Typ gedruckt werden.

TR-2008 Wanted SYSLST-file is locked, please select another filename

Bedeutung

Die angegebene SYSLST-Datei wird bereits von einem anderen Anwender benutzt. Deshalb ist ein anderer Dateiname zu wählen.

TR-2009 Filename is not valid

Bedeutung

Der Dateiname genügt nicht den BS2000-Konventionen. Deshalb ist ein anderer Dateiname zu wählen.

TR-2010 Directory for wanted type is written

Bedeutung

Gutquittung für den Druck des Objektverzeichnisses des angekreuzten Typs.

TR-2011 Directory is written

Bedeutung

Gutquittung für den Druck des Objektverzeichnisses des gesamten Data Dictionary.

TR-2012 Directory for wanted type is empty

Bedeutung

Für den angekreuzten Typ ist kein Eintrag im Data Dictionary vorhanden.

TR-2013 Directory is empty

Bedeutung

In diesem Data Dictionary ist kein Objekteintrag enthalten.

TR-2014 Reset of SYSLST impossible, select another file

Bedeutung

Die Rücksetzung auf Standard-SYSLST ist nicht möglich. Deshalb ist ein anderer Dateiname zu wählen.

TR-2015 If you want Standard-SYSLST, mark x in Output-file is SYSLST

Bedeutung

Die ausgewählte SYSLST-Datei ist die durch die Startprozedur festgelegte Standard-SYSLST-Datei. Die Standard-SYSLST-Datei kann nur durch Ankreuzen des Feldes `Output-file is SYSLST` ausgewählt werden. Falls explizit eine SYSLST-Datei angegeben werden soll, so muß sich diese in mindestens einer der Komponenten `nkz`, `teilname` oder `project` von der Standard-SYSLST-Datei unterscheiden.

TR-2017 Please look in SYSLST

Bedeutung

Bitte sehen Sie in der SYSLST-Datei nach.

TR-2018 Error on accessing Indgaard

Bedeutung

Fehler beim Bearbeiten der invertierten Datei.

TR-2019 Only one function is allowed.

Bedeutung

Es ist nur genau eine Funktion erlaubt.

TR-2020 Only Data Transfer will be executed. Please delete other entries.

Bedeutung

Es wird nur der Parameter `Data Transfer` ausgewertet. Bitte löschen Sie Einträge in anderen Feldern der Maske.

TR-2021 Both mark and name of component-list must be entered.

Bedeutung

Es muß zum einen das Feld `Comp.list` markiert werden und zum anderen der Name der Komponentenliste angegeben werden.

TR-2022 Only object-type-function will be executed. Delete other entries.

Bedeutung

Es werden nur die Einträge im Block `Object-type` ausgewertet. Bitte löschen Sie Einträge in anderen Feldern der Maske.

TR-2023 Both mark and name of object-type must be entered.

Bedeutung

Es muß zum einen das Feld `Object-type` markiert werden und zum anderen die Kurzbezeichnung eines Objekttyps angegeben werden.

TR-2024 Please enter type and name

Bedeutung

Geben Sie die Kurzbezeichnung eines Objekttyps und den Objektname ein.

TR-2025 Only structure will be executed. Please delete other entries.

Bedeutung

Es werden nur Einträge im Block `Object-type` ausgewertet. Bitte löschen Sie Einträge in anderen Feldern der Maske.

TR-2026 Only total-dictionary will be executed. Please delete other entries.

Bedeutung

Es wird nur der Eintrag im Feld `Total dictionary` ausgewertet. Bitte löschen Sie Einträge in anderen Feldern der Maske.

TR-2027 Only comp.list will be executed. Please delete other entries.

Bedeutung

Es wird nur der Eintrag im Feld `Comp.list` ausgewertet. Bitte löschen Sie Einträge in anderen Feldern der Maske.

TR-2028 Please mark a function.

Bedeutung

Bitte markieren Sie eine Funktion.

TR-2030 Please input "D" or "E"

Bedeutung

Bitte geben Sie "D" oder "E" ein.

TR-2031 Please enter "MOD", "NEW" or "NOM"

Bedeutung

Bitte geben Sie "MOD", "NEW" oder "NOM" ein.

TR-2033 Internal error (memory error)

Bedeutung

Interner Fehler: Dynamische Speicherplatzanforderung kann nicht befriedigt werden.

TR-2034 File not accessable

Bedeutung

Auf die Protokoll- oder die Dictionary-Export-Datei kann nicht zugegriffen werden.

TR-2035 Internal error

Bedeutung

Interner Fehler bei Dateizuweisung.

TR-2037 Wanted object is not defined

Bedeutung

Das gesuchte Objekt ist nicht definiert.

TR-2040 REFVAL-Error

Bedeutung

Bei der REFVAL-Bearbeitung ist ein unerwarteter Fehler aufgetreten.

TR-2043 Internal error (REFTAB-Error)

Bedeutung

Beim Anlegen eines erweiterten Beziehungstyps ist ein unerwarteter Fehler aufgetreten.

TR-2044 Internal error (Heap-error)

Bedeutung

Bei der dynamischen Speicherplatz-Verwaltung ist ein interner Fehler aufgetreten.

TR-2051 It is not possible to transfer the whole CLS

Bedeutung

Es war nicht möglich, alle Objekte der Komponentenliste zu übertragen, da einige oder alle Objekte der Komponentenliste nicht mehr existieren. Die gefundenen Objekte wurden übertragen.

14 Anhang: TOM-REF-Objekte

Objekttyp	zugehöriger Teil-Strukturkatalog	Texttyp	Bedeutung
ALI	UDS	TAL	Synonym
ARE	UDS	TAR	Area der Datenbank
CAL	Prog.-Design	TCA	Call-Aufruf
CBL	Prog.-Real.	TCS	COBOL-Quellprogramm
CMD	DVS-Leasy	TCM	Kommando
CNn	Funkt-Entw.	TCn	Konzept
DAR	UDS	TDA	dynamischer Area der DB
DBR	UDS	TBR	Datensatz im DB-Schema
DBK	UDS	TDK	Set-key eines Sets
DBS	UDS	TDB	Datenbank-Schema
EFn	Funkt-Entw.	TEn	Elementarfunktion
ENP	Funkt-Entw.	TEN	Einsprungspunkt
ENT	JCL	TET	BS2000-Batch-Prozedur
EXn	Funkt-Entw.	TXn	Ausgang
FIL	DVS/Leasy	TFI	Logische Dateibeschreibung in einem COBOL-Programm
FKY	DVS/Leasy	TFK	Funktionstaste
FLD	DVS/Leasy	TFD	Datenfeld eines COBOL-Programms, Datenbank...

Objekttyp	zugehöriger Teil-Strukturkatalog	Texttyp	Bedeutung
FLG	DVS/Leasy	TFL	Datengruppe im COBOL-Programm oder UDS-DB
FNn	Funkt-Entw.	TFn	Funktion
GSS	UTM	TGS	globaler sekundärer Speicherbereich
ICn	Funkt-Entw.	TIn	Informationsbehälter
IMn	Funkt-Entw.	TM1	Eingang
INn	Funkt-Entw.	TN1	Information
INL	DVS/Leasy	TIN	Copy-Strecke für COBOL-Prog.
JV	JCL	TJV	Logische Jobvariable in BS2000-Proz
LIB	Prog.-Real.	TLI	Bibliothek
LMC	UDS	TLC	Calc-Key eines DB-Satzes
LMD	UDS	TLD	Direct-Key eines DB-Satzes
LMO	Prog.-Real.	TLM	Lademodul
LSS	UTM	TUL	Lokaler sekundärer Speicherbereich
LTR	UTM	TLT	UTM-Datenstation
MGR	Prog.-Design	TMG	Modulgruppe
MOD	Prog.-Design	TMO	Modul
MSK	DVS/Leasy	TMS	Maske
NOM	Prog.-Real.	TNO	nicht-sharables Objektmodul
PFL	DVS/Leasy	TPF	physische Datei
PGM	Prog.-Design	TPG	Programm
PJV	JCL	TPJ	physikalische Jobvariable in BS2000-Prozedur
PKY	DVS/Leasy	TPK	Primärschlüssel einer ISAM-Datei
PRC	JCL	TPC	BS2000-Dialog-Prozedur

Objekttyp	zugehöriger Teil-Strukturkatalog	Texttyp	Bedeutung
RBT	SESAM/SQL	TRB	Basistabelle
RCC	SESAM/SQL	TCC	Wertebedingung für Tabellenspalten
RCT	SESAM/SQL	TCT	Datenbankname
REC	DVS/Leasy	TRE	Datensatz
RET	DVS/Leasy	TRT	OTHER, benutzerdefinierte Objekttypen
RFC	SESAM/SQL	TRC	Referenzbedingung
RIX	SESAM/SQL	TRI	Index-Beschreibung
RLS	DVS/Leasy	TRL	OTHER, benutzerdefinierte Objekttypen
RSG	SESAM/SQL	TSG	Plattenspeichergruppe
RSM	SESAM/SQL	TRS	Datenbank-Schema
RUC	SESAM/SQL	TRU	Eindeutigkeitsbedingung
RUS	SESAM/SQL	TUS	Datenbankbenutzer
SBR	UDS	TSB	Subschema-Datensatz
SDY	UDS	TDY	dynam. Set eines dynam. Realm
SEA	UDS	TSA	Suchschlüssel eines DB-Satzes oder Sets
SEQ	DVS/Leasy	TSQ	Formalsequenz
SET	UDS	TSE	Set im Datenbankschema
SKY	DVS/Leasy	TSK	Sekundärschlüssel einer DVS/Leasy-Anwendung
SOM	Prog.-Real.	TSO	sharables Objektmodul
SSL	UDS	TSL	SSL-Schema einer DB
SSY	UDS	TSS	System-Set im DB-Schema
SUB	UDS	TSU	Subschema einer DB

Objekttyp	zugehöriger Teil-Strukturkatalog	Texttyp	Bedeutung
SYS	Prog.-Design	TSY	System
TAC	UTM	TTA	UTM-Transaktionscode
TLS	UTM	TTL	Terminalspezifischer Langzeitspeicher
UAP	UTM	TUA	UTM-Anwendung
UDS	UDS	TUD	Datenbank
UTP	UTM	TUT	UTM-Teilprogramm-Info
VOL	DVS/Leasy	TVO	Datenträger

Glossar

Analyse-Programm

Analyser

Bestehende Anwendungen, Programme und Prozeduren werden von speziellen Werkzeugen analysiert, entsprechend dem Teilstrukturkatalog aufbereitet und in einem Data Dictionary abgelegt.

Anweisung

statement

Die einzelnen Funktionen des Editors TOM-TI aktivieren Sie durch Eingabe von Anweisungen. Die Operanden einer Anweisungen geben zusätzliche Informationen zur Ausführung der jeweiligen Funktion. Neben den TOM-TI-spezifischen Anweisungen können Sie auch alle EDT-Anweisungen eingeben (siehe auch "EDT-Handbuch" [17]). Alle TOM-TI-spezifischen Anweisungen müssen mit dem Anweisungssymbol Stern (*) versehen sein.

Art des Beziehungstyps

mode of relationship type

Attribut, das angibt, wieviele Beziehungen eines Beziehungstyps zwischen zwei Objekttypen bestehen bzw. bestehen dürfen, z.B.

1:1: ein Objekttyp hat genau eine Beziehung eines bestimmten Typs zu einem bestimmten anderen Objekttyp, auch binärer Beziehungstyp.

1:n: ein Objekttyp hat Beziehungen eines bestimmten Typs zu einem oder mehreren anderen Objekten eines bestimmten Typs, auch mehrstelliger Beziehungstyp.

m:n: ein Objekttyp hat eine Beziehung eines bestimmten Typs zu einem oder mehreren Objekten eines bestimmten Typs und umgekehrt.

Attribut

attribute

Ausprägung eines Attributtyps, z.B. Kundenname Fa. Binding, Programm X ruft Programm Y auf, Programm wurde am 15.7.88 fertiggestellt.

Attributtyp

attribute type

Eigenschaftsart, die einem Objekttyp oder einem Beziehungstyp (Entity oder Relationship) zur Identifikation und weiteren Beschreibung zugeordnet wird, z.B. "Kundenname" für den Objekttyp Kunde oder "aufrufen" für die Beschreibung einer Beziehung zwischen Programmen.

Aufrufprozedur

start-procedure

Prozedur, die TOM-REF in einer bestimmten Konfiguration startet. Die Aufrufprozedur legt der TOM-Administrator bei der Initialisierung der Phase TOM fest. Der TOM-Administrator bestimmt, welche TOM-Produkte, z.B. TOM-REF, als `Special function` aufgerufen werden können und welcher Parametersatz verwendet werden soll. Die Aufrufprozedur für TOM-REF heißt `TOMREF.SPF` und kann in TomDoors-M oder auf Kommandoebene aufgerufen werden.

Ausgangsobjekt

start-object

Objekt, von dem aus eine Beziehung aufgebaut wird.

Benutzer-eigener [Dictionary-]Attributtyp

user defined [dictionary] attribute type

Attributtyp im Data Dictionary, der nicht lt. Strukturkatalog herstellerseitig definiert ist. Mit ihnen können Sie selbst Teilstrukturkataloge bilden, mit dem Strukturkatalog verbinden und im Data Dictionary ablegen.

Benutzer-eigener [Dictionary-] Beziehungstyp

user defined [dictionary] relationship type

Beziehungstyp im Data Dictionary, der nicht lt. Strukturkatalog herstellerseitig definiert ist. Mit ihnen können Sie selbst Teilstrukturkataloge bilden, mit dem Strukturkatalog verbinden und im Data Dictionary ablegen.

Benutzer-eigener [Dictionary-] Objekttyp

user defined [dictionary] entity type

Objekttyp im Data Dictionary, der nicht lt. Strukturkatalog herstellerseitig definiert ist. Mit ihnen können Sie selbst Teilstrukturkataloge bilden, mit dem Strukturkatalog verbinden und im Data Dictionary ablegen.

Beziehung

relationship

Ausprägung eines Beziehungstyps (bestimmter Beziehungstyp), z.B. Auftrag Nr. A50001 enthält Artikel Sportschuh Leopard, Programm X ruft Programm Y auf.

Beziehungsgefüge

entity-relationship-modell

Beschreibt ein Problem mit Objekttypen, Attributstypen und Beziehungstypen. Z.B. einen Datenentwurf oder allgemein ein Data Dictionary bzw. eine Systembibliothek.

Beziehungsliste

relationship list

Liste von Beziehungen, die von einem Objekt ausgehen.

Beziehungstyp

relationship type

Art der Beziehung zwischen Objekttypen, z.B. Auftrag enthält Artikel oder Programm ruft Programm auf.

Binärer Beziehungstyp

binary relationship type

Beziehungstyp, zu dem ein Objekttyp als Beginn der Beziehung und einer als Ziel gehört.

Blättern

paging

Art der Navigation in Ausgabelisten in Masken:

- + eine Seite vorwärts
- eine Seite rückwärts
- ++ letzte Masken
- erste Masken
- ⌵ Blättern beenden

Data Dictionary

data dictionary

Verzeichnis der Typen von Informationen, die in einem Unternehmen existieren. Es dient als Nachschlagewerk und kann somit zur Dokumentation (passiv) oder während des Verarbeitungsprozesses selbst (aktiv) genutzt werden. Die Datenhaltung des Data Dictionary kann verschiedenartig sein. Das Data Dictionary-System TOM-REF benutzt als Datenhaltung ISAM-Dateien (je Data Dictionary zwei ISAM-Dateien).

[Data] Dictionary-Konfiguration

[data]dictionary configuration

Konfiguration, die aus einem oder keinem zentralen und/oder einem privaten Data Dictionary besteht. Die Datenhaltungen der verschiedenen Data Dictionaries kann unterschiedlich sein und ggf. auf verschiedenen Rechnern liegen. Bedingung ist, daß das Data Dictionary-System solche Konfigurationen bearbeiten kann.

Dialogbetrieb

interactive mode

Betriebsart, in der ein Benutzerauftrag als Folge einzelner Auftragschritte gestellt wird, meist interaktiv von einem Datensichtgerät aus. Der Dialogbetrieb ist zu unterscheiden vom Stapelbetrieb.

ESQL

ESQL

(embedded SQL) SQL-Anweisungen, die innerhalb eines Programms einer Wirtssprache (z.B. COBOL, C) angegeben werden und deren Anfang (mit EXEC SQL) und Ende (mit END-EXEC oder ';') markiert ist. Diese Markierungen heben die SQL-Anweisungen von den Anweisungen der Wirtssprache ab, so daß sie precompiliert werden können.

Identifizierendes Attribut

identifying attribute

Attribut, das es erlaubt, ein Objekt oder eine Beziehung zu identifizieren.

Informationsmodell

information modell

Siehe Strukturkatalog.

Inverse Beziehung

inverse relationship

Die Beziehung eines Objektes zu einem anderen Objekt, bei der das erstgenannte Objekt nicht der Beginn der Beziehung ist, sondern das Ziel.

Katalogkennung (catid)

catalog ID

bezeichnet das Public Volume Set (PVS), auf dem die BS2000- oder TOM-Dateien gespeichert sind. Die Katalogkennung wird dem Dateinamen in der Form :catid:userid vorangestellt.

Kennzeichen

flag

TomDoors-M vergibt bei einigen Funktionen bestimmte Kennzeichen an das bearbeitete Objekt, die neben Objektname und -typ den Arbeitsstand des Objekts kennzeichnen. TomDoors-M vergibt folgende Kennzeichen:

- .@ Temporäre Sicherungsdatei, die im Editor mit der Anweisung `*STORE` erzeugt wird.
- .# Objekt, das beim Übersetzungsnachlauf (bei Übersetzungsfehlern) bzw. über eine COLUMBUS-Funktion erzeugt wird. Es enthält die Quelldatei mit den eingemischten Fehlermeldungen des Precompilers, Compilers oder Assemblers bzw. die Kommentare der Strukturfunktionen. Die #-Datei wird auch als TOMSAVLST bezeichnet.
- .\$ Permanente Sicherungsdatei, die beim Zurückschreiben des aktuellen Objekts im Editor aus dem Arbeitsbereich (0) durch Kopie des vorherigen Status erzeugt wird.
- .nnn dreistellige Ziffer (bei COLUMBUS-Durchlaufzählerdateien), die pro Testdurchlauf um 1 hochgezählt wird.

Kommando

command

Ein TOM-REF-Kommando besteht aus einem oder zwei Kommandoschlüsselwörter ohne oder mit einem oder mehreren Operanden. Das Schlüsselwort beschreibt die Funktion, die ausgeführt werden soll. Die Operanden können folgende Ausführungsbedingungen festlegen:

- Objekt benennen, das bearbeitet werden soll
- Parametersatz angeben, der bei der Ausführung verwendet werden soll
- Festlegen, ob die betreffende Funktion im Dialog oder als Batchprozeß ablaufen soll

Kommandos können Sie miteinander verketteten. Der Strichpunkt (;) dient dann als Trenner zwischen den Kommandos. Alternativ zu den Kommandos können Sie auch Kurzkommandos eingeben oder die einzelnen Funktionen in den Masken ankreuzen.

Komponente

component

Siehe Objekt.

Kurzkommando

Shortcut

In den einleitenden Auswahlmasken (TomDoors-M: 02 Selection, TOM-Ref: TR0001 Selection) können Sie Kurzkommandos verwenden. Allen Funktionen ist in diesen Masken eine Nummer oder ein Buchstabe fest zugeordnet, die bzw. den Sie einzeln oder in Kombination als Kurzkommando angeben können. Kurzkommandos können Sie wie Kommandos verketteten.

Maske

form

Am Bildschirm dargestelltes Schema zur Anzeige und Eingabe von Daten. Die Oberflächen der TOM-Tools sind maskengesteuert.

Mehrstelliger Beziehungstyp

n-ary relationship type

Beziehungstyp, zu dem ein Objekttyp als Ausgang der Beziehung und mehrere als Ziel gehören.

Member

Bibliothekselement in der TOMLIB.

Modul

module

Übersetzte Quellprogramme werden als Module oder Bindemodule bezeichnet. Sie stehen in der benutzereigenen TOMLIB. Übersetzte und gebundene Module sind sogenannte Lademodule (Objektmodule, Phasen). Sie werden in TomDoors-M als BS2000 Dateien geführt.

N²Chart

Grafische Dokumentationstechnik und Darstellungsform zur Wiedergabe von Kommunikationsstrukturen.

Nachdokumentation

Analysation

Siehe Analyse.

Namensbezogene Objektmenge

name dependent quantity of entities

Menge von Objekten mit gleichem Namen aber unterschiedlichem Typ; z.B. können Sie sich mit dem Kommando `DISPLAY C=objektname` alle Objekte anzeigen lassen, die denselben Namen haben, sich aber im Typ unterscheiden.

Objekt

entity

Die einzelnen, in einer Bibliothek gespeicherten Einheiten werden als Objekte bezeichnet. Sofern sie in einer TOMLIB stehen, werden sie mit `objektname.objekttyp` näher beschrieben.

Diese Objektidentifikation ist in Übereinstimmung mit den in TOM gültigen Namenskonventionen frei wählbar.

Mit `objekttyp` legen Sie fest, um welche Art Objekt es sich handelt, beispielsweise um

- freien Text
- Quellcode
- Bindemodul
- Parametersatz zu einer bestimmten Funktion

TOM-REF stellt vordefinierte und frei zu definierende Objekttypen zur Verfügung. Im Rahmen des Entity-Relationship-Modells wird Objekt auch als Entity bezeichnet und definiert die konkrete Ausprägung eines Objekttyps. Ein Objekt ist zugleich die kleinste Verwaltungseinheit der Systembibliothek.

Objektidentifikation

entity identification

Summe der Attribute, die ein Objekt eindeutig beschreiben. In der TOM-SPU sind dies: `objektname.objekttyp`.

Objektkatalog

directory

Ausprägung eines Objekttypkatalogs, z.B. die in einem Data Dictionary erfaßten Objekte.

Objektliste

entity list

Menge von Objekten mit beliebigem Namen und von beliebigem Typ.

Objektname

entity name

Sie bestimmen als Anwender den Namen des Objekts, das Sie bearbeiten wollen. Bei der Festlegung des Namens müssen Sie die TOM-Namenskonvention beachten. Ein Objektname darf maximal 8 Zeichen lang sein.

Objekttyp

entity type

Objekte des gleichen Aufbaus, z.B. Kunde, Artikel, Programm, Maske, Datenbeschreibung oder COBOL-Quellprogramm. TOMREF bietet für eine Reihe von Objekten fest definierte Objekttypen an, z.B. COB für ein COBOL-Quellprogramm. Darüber hinaus können Sie selbst für Objekte freie Objekttypen analog der TOM-Namenskonventionen definieren. Ein Objekttyp darf maximal 3 Zeichen lang sein.

Objekttypkatalog

entity type catalogue

Menge von Objekttypen eines Informationsmodells, z.B. die Objekttypen eines Data Dictionary.

Objekttyp-Teilkatalog

entity type relationship model

Teilmenge von Objekttypen eines Beziehungsgefüges, z.B. die Objekttypen für TOMLIB (LMS), Data Dictionary, KMS und SINIX-Filesystem als Bestandteile der TOM-Systembibliothek, die Objekttypen zur Definition einer SESAM-Datenbank oder *D für die Objekttypen REC, FLG und FLD im Data Dictionary.

Parameter

parameter

beeinflussen die Umgebung und die Funktion von TomDoors-M. Parameter für den Ablauf von TomDoors-M sind zu Parametersätzen zusammengefaßt. Ein solcher Parametersatz kann individuell festgelegt werden oder als Standard-Parametersatz entsprechend der Initialisierung von TomDoors-M übernommen werden.

Phase

program

Ablauffähiges Programm (übersetztes und gebundenes Modul, Objektmodul), das von TOM als eigene BS2000-Datei erstellt wird.

Privates Data Dictionary

private data dictionary

Data Dictionary für einen Teilbereich eines Unternehmens bzw. der Informationsverarbeitung. Es wird von einer bestimmten Person oder Personengruppe, z.B. den an einem bestimmten Entwicklungsprojekt Beteiligten genutzt.

Projektbibliothek

project library

Zusätzliche TOMLIB, in der alle an einem Projekt beteiligten Entwickler ihre fertigen Module ablegen. Bei größeren Projekten ist es ratsam, eine eigene Projektbibliothek anzulegen.

Referenz-Datei

reference file

Siehe Data Dictionary.

Sicherungsdatei (permanent)

back up

Beim Überschreiben des aktuellen Objekts im Editor aus dem Arbeitsbereich (0) legt TomDoors-M eine sogenannte permanente Sicherungsdatei an. Diese Datei enthält den vorangegangenen Stand des jeweiligen Objekts.

Dabei fügt TomDoors-M an die bisherige Objektidentifikation noch zusätzlich das Kennzeichen.\$ an: `objektname.objekttyp.$`. Deshalb werden diese Dateien auch \$-Dateien genannt.

Sicherungsdatei (temporär)

back up

Wenn Sie bei der Bearbeitung eines Objekts im Arbeitsbereich (0) des Editors dieses zwischenzeitlich mit Hilfe der Anweisung `*STORE` (ohne Operanden) sichern, ohne dabei den Editor zu verlassen, vergibt TomDoors-M bei diesem expliziten Sichern das Kennzeichen `.@`.

Dieses Kennzeichen tritt an das Ende der bisherigen Objektidentifikation: `objektname.objekttyp.@..` Damit ist die temporäre Sicherungsdatei als `@`-Datei gekennzeichnet. Beim Verlassen des Editors wird diese `@`-Datei wieder gelöscht.

Standard-Beziehungstyp

predefined relationship type

Von Herstellerseite vordefinierter Beziehungstyp zwischen zwei oder mehr Objekten.

Stapelbetrieb

batch mode

Betriebsart, in der ein Benutzerauftrag vollständig gestellt ist und von der Auftragstellung zeitlich entkoppelt abgewickelt werden kann. Die Betriebsart Stapelbetrieb unterscheidet sich dadurch vom Dialogbetrieb.

Strukturiertes Objekt

structured entity

Objekt, das sich aus mehreren, möglicherweise wieder strukturierten Objekten zusammensetzt, z.B. der Objektteilkatalog einer bestimmten SESAM/SQL V2-Datenbank.

Strukturkatalog

entity relationship model

Katalog der Objekttypen, Beziehungstypen und Attributtypen, die in einem Data Dictionary enthalten sind.

Systembibliothek

system library

Integrierte Datenhaltung für die Anwendungsentwicklung.

Teilname

partial name

Im BS2000 bezeichnet ein vollqualifizierter Dateiname eine Datei. Er kann aus mehreren Teilnamen bestehen. Teilnamen sind durch einen Punkt voneinander getrennt. Hinter dem letzten Teilnamen steht kein Punkt.

Teilstrukturkatalog

sub-structure catalog

Teil des Strukturkatalogs, der herausgegriffen wurde, weil dadurch eine übersichtliche Darstellung ermöglicht wird bzw. von diesem Teil bestimmte Data-Dictionary-Anwendungen bedient werden.

TOMLIB

TOMLIB

PLAM-Bibliothek, die mit LMS bearbeitet werden kann. TOM unterscheidet zwischen zentraler und benutzereigener TOMLIB.

Die zentrale TOMLIB wird beim TOMSTART-Lauf erstellt oder aktualisiert und enthält:

- Standard-Parametersätze
- ggf. weitere Objekte mit Namen STANDARD
- zentrale `Special functions`
- die zentralen Infomember

Die benutzereigene TOMLIB entsteht beim Aufruf von TomDoors-M nach dem Ausfüllen der Maske `01 Identification` und enthält:

- alle vom Benutzer erstellten Objekte und durch Funktionen von TomDoors-M erstellten Objekte
- die aus der zentralen TOMLIB automatisch bei der Erstellung kopierten STANDARD-Parametersätze und ggf. weitere Objekte mit dem Namen STANDARD

TOM-Namenskonvention

Für die Namensbestandteile, die Sie bei der Arbeit mit TOM selbst bilden, wie z.B. "Objektname", "Namenskurzzeichen" (nkz) oder "Projekt", gelten bestimmte Regeln:

- Sie dürfen alle Buchstaben, alle Ziffern und die Sonderzeichen -, \$, @ und # verwenden.
- Ein einstelliger Namensbestandteil darf nicht aus \$, @ oder # bestehen.
- Präfix, Suffix, Namenskurzzeichen und Objektnamen müssen mindestens einen Buchstaben enthalten oder mit \$, @ oder # beginnen. D.h. Projektnamen und Objekttypen dürfen auch rein numerisch sein.
- Ein Namensbestandteil darf keinen Punkt enthalten.

TOMRFI-Datei

Datei des Data Dictionary, in der die inversen Beziehungen hinterlegt sind.

TOM-Systembibliothek

TOM system library

Besteht aus TOM-Bibliothek (LMS) und REF-Bibliothek (Data Dictionary).

Typbezogene Objektmenge

type dependend quantity of entities

Menge von Objekten gleichen Typs mit beliebigen Namen.

Typ-neutraler Attributtyp

common attribute type

Typ des Attributs eines Objekts oder einer Beziehung, der bei allen Objekt- bzw. Beziehungstypen vorkommt und daher so etwas wie eine Grundausstattung darstellt.

Typ-spezifischer Attributtyp

type specific attribute type

Typ des Attributs eines Objekts oder einer Beziehung, der nur bei bestimmten Objekt- bzw. Beziehungstypen vorkommt.

Verfahrenstechnik

process-engineering

regelt die Abfolge der Meilensteinergebnisse, deren Inhalt pro Meilenstein und die sprachliche, methodische und automatisierte/werkzeuggestützte Transformation von Meilenstein zu Meilenstein innerhalb eines Produktionsprozesses.

Vordefinierter Objekttyp

predefined entity type

Objekttyp im Data Dictionary, der den Eintrag bestimmter Objekte, ihrer Attribute und Beziehungen im Strukturkatalog ermöglicht. Er ist herstellerseitig vorgesehen, damit bestimmte Anwendungen des Data Dictionary arbeiten können.

Wertigkeit des Beziehungstyps

cardinality of relationship type

Die Wertigkeit des Beziehungstyps gibt an, welche Anzahl von Ausprägungen der beteiligten Objekttypen vorkommen. Es gibt folgende Wertigkeiten: 1:1, 1:n, n:m, der Einschluß des Nullwerts (1:0/1, 1:0/n, n:0/m), sowie jeweils die Umkehrung n:1, m:n usw.

Zentrales Data Dictionary

central data dictionary

Von allen Beteiligten der Informationsverarbeitung genutztes Data Dictionary. Es enthält entsprechend globale Daten, die meistens besonders schutzwürdig sind.

Zielobjekt

target object

Objekt, zu dem eine Beziehung aufgebaut wird.

Abkürzungen

BCAM	Basic Communication Access Method
DCAM	Data Communication Access Method
DDL	Data Definition Language
DDA	Data Dictionary Anschluß
DML	Data Manipulation Language
DSS	Datensichtstation
DVS	Datenverwaltungssystem
FE	Front End
FHS	Format Handling System
FT	File-Transfer
IFG	Interaktiver Formatgenerator
ISAM	Index Sequential Access Method
JCL	Job Class Language
LAN	Local Area Network
LEASY	Lineares Ein-/Ausgabe-System
LMS	Library Maintenance System
OMNIS	Steuersystem zur zentralen Bedienung mehrerer Anwendungen
PLAM	Program Library Access Method
SAM	Sequential Access Method
SDF	System Dialog Facility
SESAM	System zur elektronischen Speicherung alphanumerischer Methoden
SQL	Structured Query Language
SSL	Structured Storage Language
TIAM	Terminal Interactive Access Method

TOM-CA	Tool Manager Code-Analyse
TOM-DOC	Tool Manager Dokumentaufbereitung
TOM-GEN	Tool Manager Generatorbaustein
TOM-JA	Tool Manager JCL-Analyse
TOMLIB	TOM-Library
TOM-MP	Tool Manager Masken-Prototyping
TOM-MTEST	Tool Manager Modultest
TOM-RTEST	Tool Manager Repeattest
TOM-REF	Tool Manager Reference System
TOM-REFG	Tool Manager Reference Grafik
TOM-TA	Tool Manager Textanalyse
TOM-TI	Tool Manager Tool Interface
UDS	Universelles Datenbank-System
UTM	Universeller Transaktionsmonitor
VTSU	Virtual Terminal Support

Literatur

Das Literaturverzeichnis enthält alle Handbücher und weiterführende Literatur, auf die in dem vorliegenden Handbuch verwiesen wird.

Literatur zur TOM-Entwicklungsumgebung

- [1] **TomDoors-M V5.0** (BS2000/OSD)
Grafische Oberfläche für den Toolmanager-Monitor
Benutzerhandbuch

Zielgruppe

TOM-M Anwender, die die Vorteile einer grafischen Oberfläche nutzen wollen.

Inhalt

Anhand einer Beispielsitzung wird der Anwender in die Arbeit mit der grafischen Oberfläche von TomDoors-M eingeführt. Das Handbuch enthält zudem eine Beschreibung des Fensteraufbaus, ein Konfigurationsbeispiel für den Ablauf unter FHS-DOORS und eine Zuordnungstabelle von BS2000-Bildschirmen zu den Windows-Fenstern.

- [2] **TOM-M V4.2** (BS2000)
Toolmanager-Monitor
Einführungshandbuch

Zielgruppe

BS2000-Software-Entwickler

Inhalt

Eine Beispielsitzung führt den Anwender in das Arbeiten mit TOM-M und in die maskenorientierte Menüführung ein. Er erhält zudem einen Überblick, wie er mit der TOM-SPU Programme erstellen und übersetzen kann.

- [3] **TOM-M V4.2** (BS2000)
Toolmanager-Monitor
Referenzhandbuch

Zielgruppe

BS2000-Software-Entwickler

Inhalt

Das Referenzhandbuch beschreibt alle Funktionen von TOM-M. Der Anwender erfährt, wie er mit der TOM-SPU Programme erstellt, editiert und übersetzt, bindet und testet. Der TOM-Administrator erfährt, wie er TOM initialisiert, administriert und Parametersätze für den Betrieb zusammenstellt.

[4] **TOM-TI (BS2000/OSD)**

Tool-Interface

Benutzerhandbuch

Zielgruppe

TomDoors-M Anwender

Inhalt

Mit TOM-TI können Texte und Programme von einer einheitlichen Oberfläche erfaßt, generiert, formatiert und druckaufbereitet werden. Darüber hinaus bietet TOM-TI:

- Start eigener Programme von der TOM-TI-Oberfläche aus
- Nebenprozeßtechnik und zusätzliche Leistungen im Rahmen der TOM-TI-Toolbörse

[5] **TOM-REFG V2.1 (BS2000)**

Oberfläche für Data Dictionary-geführte Strukturen

Benutzerhandbuch

Zielgruppe

TOM-REF-Anwender

Inhalt

TOM-REFG ist ein Oberflächenwerkzeug für das Data Dictionary-System TOM-REF. Mit TOM-REFG lassen sich Systemstrukturen kommerzieller DB/DC-Anwendungen grafisch darstellen und entwickeln.

[6] **TOM-JA (BS2000)**

JCL-Analyseprogramm

Benutzerhandbuch

Zielgruppe

BS2000-Programmierer

Inhalt

TOM-JA dient der Analyse von BS2000-Do- und -Enter-Prozeduren. Gegenstand der Analyse sind die Aufrufbeziehungen dieser Prozeduren untereinander, welche Programme und Module in diesen Prozeduren aufgerufen werden, und welche Dateien, Linknamen und Jobvariablen in diesen Prozeduren verwendet werden.

- [7] **TOM-CA V3.0** (BS2000/OSD)
COBOL-Analyseprogramm
Benutzerhandbuch
- Zielgruppe*
COBOL-Programmierer
- Inhalt*
TOM-CA dient der Analyse von COBOL-, ESQL-COBOL- und UTM-Programmen. Gegenstand der Analyse sind die Aufrufbeziehungen der Programme untereinander, welche Unterprogramme und Module in den Programmen aufgerufen werden, und welche Dateien, Linknamen und Variablen in diesen Prozeduren verwendet werden.
- [8] **TOM-TA V1.0B** (BS2000)
Text-Analyse
Benutzerhandbuch
- Zielgruppe*
TomDoors-M-Anwender
- Inhalt*
TOM-TA analysiert beliebige Texte auf Beziehungen zu anderen Strukturelementen, z.B. zu anderen Texten und Daten.
- [9] **TOM-MTEST** (BS2000)
Modultest
Benutzerhandbuch
- Zielgruppe*
Software-Entwickler, Tester von COBOL-Programmen
- Inhalt*
TOM-MTEST unterstützt den Entwickler in der Testphase. Es können COBOL- bzw. COLUMBUS-COBOL-Module und Modulgruppen getestet werden.
- [10] **TOM-MP** (BS2000)
Maskenprototyping
Benutzerhandbuch
- Zielgruppe*
Software-Entwickler, Tester von COBOL-Programmen
- Inhalt*
Mit TOM-MP können Dialogstrukturen modelliert und in Form von Masken-Sequenzen ausgeführt werden. TOM-MP nutzt das Data Dictionary-System TOM-REF und die Benutzeroberfläche von TOM-REFG.

- [11] **TOM-RTEST (BS2000)**
Repeattest
Benutzerhandbuch
- Zielgruppe*
Software-Entwickler, Tester von COBOL-Programmen
- Inhalt*
TOM-RTEST erweitert die Testfunktion von TomDoors-M, schneidet die Bildschirm-ein- und -ausgaben dialogorientierter Programme mit und verwendet die mitgeschnittenen Dateneingaben zur automatischen Wiederholung des Programmablaufs. Die Testprotokolle können ausgewertet und mit Vergleichsmasken kann ein automatischer Soll-/Ist-Vergleich durchgeführt werden.
- [12] **SESAM/SQL-DDA V3.0 (BS2000/OSD)**
Data Dictionary Anschluß
Benutzerhandbuch
- Zielgruppe*
Datenbankverwalter, Datenbankdesigner
- Inhalt*
Der Anwender erfährt, wie er mit SESAM/SQL-DDA Datenbanken nachdokumentiert, und neue Datenbanken bequem erstellen und bearbeiten kann.
- [13] **UDS-DDA V1.0 (BS2000)**
UDS-Data Dictionary Anschluß
Benutzerhandbuch
- Zielgruppe*
Datenbankverwalter, Datenbankdesigner
- Inhalt*
Der Anwender erfährt, wie er mit UDS-DDA Datenbanken erstellt und bestehende DDL-Sourcen und ESQL-COBOL-Programme nachdokumentieren kann.
- [14] **TOM-DOC (SINIX)**
Textaufbereitung
Benutzerhandbuch
- Zielgruppe*
SINIX-Benutzer
- Inhalt*
- Programm zur Textformatierung einschließlich der Formatierkommandos
 - Programme zur Datenübertragung zwischen SINIX und BS2000.

Weiterführende Literatur

- [15] **LEASY** (BS2000)
Taschenbuch

Zielgruppe

LEASY-Anwender

Inhalt

LEASY ist ein im BS2000 ablauffähiges Datenverwaltungs- und Zugriffssystem. Der Zugriff kann aus COBOL-, Assembler- oder RPG-Programmen erfolgen.

- [16] **BS2000/OSD-BC
Kommandos Band 1, A-L**
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich sowohl an den nichtprivilegierten BS2000/OSD-Anwender als auch an die Systembetreuung.

Inhalt

Es enthält die BS2000/OSD-Kommandos ADD-... bis LOGOFF (Grundausbau und ausge wählte Produkte). Die Einleitung gibt Hinweise zur Kommandoeingabe. Die Kommandobe

- [17] **EDT** (BS2000)
Benutzerhandbuch

Zielgruppe

EDT-Einsteiger und EDT-Anwender

Inhalt

Bearbeiten von SAM- und ISAM-Dateien und Elementen aus Programm-Bibliotheken und POSIX-Dateien.

- [18] **BS2000/OSD-BC
Kommandos Band 2, M-SG**
Benutzerhandbuch

Zielgruppe

Das Handbuch wendet sich sowohl an den nichtprivilegierten BS2000/OSD-Anwender auch an die Systembetreuung.

Inhalt

Es enthält die BS2000/OSD-Kommandos MODIFY... bis SET... (Grundausbau und ausge wählte Produkte). Die Kommandobeschreibung zeigt u.a. die Privilegien zur Butzung des Kommandos.

- [19] **UTM, UTM-D**
Verteilte Transaktionsverarbeitung
Beschreibung
Zielgruppe
Programmierer
UTM-Administrator
Generierer
Inhalt
– Konzept der verteilten Transaktionsverarbeitung
– Programm-, Generierungs- und Administrationsschnittstelle von UTM-D zur
– Realisierung der verteilten Transaktionsverarbeitung
– Einsatz
– BS2000-Transaktionsbetrieb
- [20] Martin, James and Carma McClure: * **Diagramming Techniques for Analysts and Programmers.** Prentice Hall 1985
- [21] Martin, James: * **Recommended Diagramming Standards for Analysts and Programmers.** A basis for automation. Prentice-Hall 1987.
- [22] Rom, Narayan: * **Data Dictionary - Implementation, Use and Maintenance.** Prentice-Hall 1988.

Mit * markierte Titel sind nicht von der Siemens Nixdorf Informationssysteme AG oder der Siemens AG herausgegeben.

Handbücher bestellen

Die aufgeführten Handbücher finden Sie mit ihren Bestellnummern im *Druckschriftenverzeichnis* der Siemens Nixdorf Informationssysteme AG. Neu erschienene Titel finden Sie in den *Druckschriften-Neuerscheinungen*.

Beide Veröffentlichungen erhalten Sie regelmäßig, wenn Sie in den entsprechenden Verteiler aufgenommen sind. Wenden Sie sich bitte hierfür an Ihre zuständige Geschäftsstelle. Dort können Sie auch die Handbücher bestellen.

Stichwörter

A

- Abbrechen, Funktion 92
- Ablauf, von TOM-REF 63
- Ablauflogik 28
- Administration (ISAM)163
- Administrator 31, 44
- Ändern
 - Beziehung 86, 336
 - erweiterte Beziehung 336
 - Objekt 336
- ALI, DVS/Leasy-Objektyp 202
- Alias (syn.) →Synonym 39, 202
- Analysator 26
- Analyse-Programm **417**
- Analyser **417**
- Analysis **422**
- Ankreuzen, Funktionsauswahl 66
- Anlegen, Data Dictionary 358
- Ansteuern
 - Druckmaske 362
 - Servicemaske 368
 - Textmaske 369
- Anweisung **417**
- Anwendungssystem 15
- Anzeigen
 - alle Beziehungen 123, 349
 - alle inversen Beziehungen 118, 350
 - alle Objekte eines Objektyps 112
 - Auswahl von Objekten 114
 - Beziehung 347
 - Beziehungen zu Objektyp 125, 349
 - erweiterte Beziehung 347
 - Inhaltsverzeichnis 346
 - inverse Beziehung eines Objekts 120, 350

- nicht-definierte Objekte 129
- Objekt 75, 347
- Objektgruppe 346
- Objekt ohne inverse Beziehung 127
- Objekttypen 346
- Arbeitsbereich 60
- ARE, UDS-Objekttyp 320
- Art des Beziehungstyps **417**
- Arten von Beziehungstypen, Strukturkatalog 9
- Attribut 19, **417**
 - Creation Date 38
 - einstellig 38
 - Last Change 38
 - Name 38
 - OCCURS 39
 - Responsible 38
 - Short Description 38
 - Text 38
 - Type 38
- attribute type **418**
- attribute **417**
- Attributtyp **418**
 - Benutzereigen 42
 - frei benutzbar 39
 - neutral 19, 37
 - spezifisch 19
 - Text 42
 - typspezifisch 37
- Aufruf 269
- Aufrufen
 - Hilfemaske 354
 - Servicefunktion 111
 - TOM-REFG 360
 - TOM-TI 351
 - UDS-DDA 370
- Aufrufprozedur 59, **418**
 - Textausgabe einstellen 89
- Ausdruck, Hinweise 140
- Ausgabeliste
 - Formatierung 134
 - Hinweise 134
- Ausgang, Funktionsentwurf 281
- Ausgangsobjekt **418**
- Ausprägung 18, 33

Ausschalten, Protokoll 363
Auswählen
 Copy-Element 67
 Data Dictionary 61
 durch Ankreuzen 66
 Funktion 64
 Objektyp 69
 zurücksetzen 67
Auswerten, Data Dictionary 22
Auswertungsfunktion 52

B

back up **425**
Basis-Symbol, Strukturkatalog 8
Basis-TOMLIB 63
Basistabelle 226
batch mode **426**
BCAM 429
Bearbeiten
 Beziehung 100
 Text 89, 369
 Text mit TOM-TI 90
Bedingung
 Check 230
 Eindeutigkeit 229
 Referenz 231
Beenden
 TOM-REF 60, 352
Behälter 19
Benutzereigen
 Attributtyp **418**
 Beziehungstyp **418**
 Objektyp **418**
 Objekt (RET) 203
 Objekt (RLS) 203
Berechnungsschlüssel 317
Bereich 320
Beziehung 18, **418**
 ändern 336
 alle anzeigen 123, 349
 anzeigen 347
 bearbeiten 100
 direkt 117
 drucken 133

- einfach 100
- erstellen 340
- erweiterbar 100
- erweitern 78
- erweitert 100, 104
- erweiterte (SQL) 237
- erweiterte (XBT) 236
- Erweiterung löschen 86
- indirekt 117
- inverse 37
- inverse anzeigen 350
- löschen 343
- markieren 86
- Objekt ohne inverse anzeigen 127
- zu Objekttyp anzeigen 125, 349
- Beziehungsgefüge **419**
- Beziehungsliste 73, **419**
 - bearbeiten 84
 - blättern 89
 - Maske 84
 - Maske 85
- Beziehungstyp 18, 86, **419**
 - Benutzereigen 41
 - FOR (UTM) 300
 - mehrstellig 41, 242
 - STO (UTM) 299
 - vordefinierter 216
 - XBT 236
 - XIX 237
- Bibliothek 272
- Binärer Beziehungstyp **419**
- binary relationship type **419**
- Blättern 89, **419**
- C**
- CAL, Process/JCL-Objekttyp 269
- Call 269
- cardinality of relationship type **428**
- catalog ID **420**
- catid **420**
- CBL, Process/JCL-Objekttyp 273
- central data dictionary **428**
- CHANGE 49, 80
 - Kommando 336

CHECK 49
 Kommando 338
Check objects 162
Check-Bedingung 230
CMD, DVS/Leasy-Objektyp 197
COBOL-Quellprogramm 273
COBOL-Source 273
Code-Generator 28
command **421**
common attribute type **428**
component **421**
COPY 49, 146
 Kommando 339
Copy-Element
 auswählen 67
 Auswahlmaske 205
 beschreiben 205
 Data-Area (Objektyp) 207
 erzeugen 353
 FILE-Description (Objektyp) 213
 generieren 143
 INIT-Statements 211
 INL (Objektyp) 205
 MOVE-Statements 209
COPY-Strecken-Generator 79
CREATE 49, 77
 Kommando 340
Creation Date, Attribut 38

D

DAR, UDS-Objektyp 321
Darstellungsmittel
 Kommando 7
 Strukturkatalog 8
Data Dictionary **419**
 aus TOMLIB holen 165, 366
 Auswahl 61
 Daten einbringen 20
 Definition 15
 Einsatzbereiche 15
 Inhalt auswerten 22
 in TOMLIB sichern 164
 mischen 151, 357
 Nutzen 16, 22

- privates 24, 43, 61
- privates anlegen 358
- Projekt- 25
- Struktur 19
- Tools 35
- zentrales 24, 43, 61
- zentrales anlegen 358
- data dictionary **419**
- Data Dictionary-Datei löschen 165
- Data Dictionary-Konfiguration **420**
- data dictionary configuration **420**
- Data-Area, COPY-Element 207
- Datei
 - logisch 188
 - physikalisch 200
- Dateinamen 46
- Datenbank
 - generieren 29
 - Spalte 235
- Datenbank (SESAM/SQL) 223
- Datenbank-Record 315
- Datenbank-Schema (SESAM/SQL) 225
- Datenbank-Schema (UDS) 314
- Datenentwurf, Objekte 241, 282
- Datenhaltung 43
- Datenorganisation 27
- Datensatz 186
- Datenträger 198
- DBK, UDS-Objektyp 329
- DBR, UDS-Objektyp 315
- DBS, UDS-Objektyp 314
- DCAM 429
- DD-Administrator 24
- DDA 429
- DDL 429
 - erzeugen 29
- DDTEXT DELETE 50, 342
- Definieren
 - benutzereigenes Objekt (RET) 203
 - Objekt (RLS) 203
- Definition, Data Dictionary 15
- DELETE 49, 83
 - Kommando 343
- Delete structure 148

DELSTRUCT 49
 Kommando 345
Dialog, protokollieren 363
Dialogbetrieb **420**
directory **423**
Direktschlüssel 316
DISDIR 49, 346
DISP-USES 49, 122, 349
DISP-WHERE-USED 49, 117, 350
DISPLAY 49, 75
 Kommando 347
Display Directory 112
Display Objects 127
Display Referenced Objects 129
DML 429
DO-Prozedur 277
Drucken 131
 Beziehungen 133
 DVS/Leasy-Objekte 139
 Inhaltsverzeichnis 361
 inverse Beziehungen 133
 JCL-Objekte 137
 Objekteintrag 132, 135
 Objektverzeichnis 132, 135
 Process-Objekte 138
 SESAM/SQL-Objekte 140
 UDS-Objekte 139
 UTM-Objekte 138
Druckfunktionen, ansteuern 362
Druckmaske, ansteuern 362
DSS 429
DVS 429
DVS/Leasy 177
 ALI (Objektyp) 202
 CMD (Objektyp) 197
 FIL (Objektyp) 188
 FKY (Objektyp) 196
 FLD (Objektyp) 178
 FLG (Objektyp) 183
 MSK (Objektyp) 195
 Objekte drucken 139
 PFL (Objektyp) 200
 PKY (Objektyp) 193
 REC (Objektyp) 186

SEQ (Objekttyp) 199
SKY (Objekttyp) 194
VOL (Objekttyp) 198
Dynamischer Bereich 321
Dynamischer Satz 328

E

EDITOR 50, 351
Eigentümer 234
Eindeutigkeitsbedingung 229
Einfügen, Beziehung 86
Eingang, Funktionsentwurf 281
Eingeben
 Kommando 68
 Kurzkommando 68
Einschalten, Protokoll 363
Einsprungspunkt 271
Einstellig, Attribut 38
Elementarfunktion, Funktionsentwurf 281
embedded SQL **420**
END 50, 352
ENP, Process/JCL-Objekttyp 271
ENT, Process/JCL-Objekttyp 278
ENTER-Prozedur 278
entity **423**
entity identification **423**
entity list **423**
entity name **423**
entity relationship model **426**
entity type catalogue **424**
entity type relationship model **424**
entity type **424**
Erstellen
 Beziehung 340
 erweiterte Beziehung 340
 Objekt 77, 340
Erweiterte Beziehung
 ändern 336
 anzeigen 347
 erstellen 340
 Index-Feld 237
 löschen 86, 343
Erzeugen
 COPY-Element 143, 353

DDL für Datenbank 29
neues Data Dictionary 358
ESQL **420**

F

FE 429
Feld 178
Feldgruppe 183
FHS 429
FIL, DVS/Leasy-Objektyp 188
FILE-Description, COPY-Element 213
FILLER 185
FKY, DVS/Leasy-Objektyp 196
flag **421**
FLD, DVS/Leasy-Objektyp 178
FLG, DVS/Leasy-Objektyp 183
FOR, UTM-Beziehungstyp 300
form **422**
Format-Information 300
Formatbeschreibung 195
Formatierung, Ausgabeliste 134
FT 429
Funktion
 abbrechen 92
 auswählen 64
 Funktionsentwurf 281
 mit Kurzkommandos verketten 96
 verketten (Ankreuzen) 67, 92
Funktionsauswahl
 durch Ankreuzen 66
 löschen 67
Funktionsentwurf
 Ausgang 281
 Eingang 281
 Elementarfunktion 281
 Funktion 281
 Information 281
 Informationsbehälter 281
 Konzept 281
 Objekte 281
Funktionskettung 91
Funktionsschlüssel 196
Funktionstasten 91
Funktionsverbund, TOM-Werkzeuge 1

G

GENERATE COPY 49, 353
Generatorbaustein 29
Generieren, COPY-Element 143
Generierung 28, 142
Gesamtprojekt 23
Gesamtsystem 31
Grundfunktion 26, **47**, 73
GSS, UTM-Objektyp 295

H

HELP 50, 354
Hilfe, aufrufen 354
Hinweise
 Ausgabeliste 134
 im Ausdruck 140
Holen, Data Dictionary aus TOMLIB 165, 366

I

Identifizierendes Attribut **420**
identifying attribute **420**
IFG 429
INCORPORATE 49, 151, 357
Incorporate Data Dictionary (syn.) →Übertragung 152
Index-Beschreibung 228
Index-Feld, erweiterte Beziehung 237
Information, Funktionsentwurf 281
information model **420**
Informationsbehälter, Funktionsentwurf 281
Informationsmodell **420**
Inhaltsverzeichnis, drucken 361
INIT-Statement, COPY-Element 211
INITIATE 49, 358
INL
 COPY-Element 205
 Data-Area (Copy-Element) 207
Integration, Phase 23
interactive mode **420**
Inverse Beziehung **420**
 alle anzeigen 118, 350
 drucken 133
 eines Objekts Anzeigen 120, 350
inverse relationship **420**
ISAM 429
Ist-Zustand 26

J

JCL, Objekte drucken 137
JCL 429
Jobvariable
 logisch 279
 physikalisch 280
JV, Process/JCL-Objekttyp 279

K

Katalogkennung **420**
Kennzeichen **421**
Kommando 197, **421**
 absetzen 335
 CHANGE 336
 CHECK 338
 COPY 339
 CREATE 340
 Darstellungsmittel 7
 DDTEXT DELETE 342
 DELETE 343
 DELSTRUCT 345
 DISDIR 346
 DISP-USES 349
 DISP-WHERE-USED 350
 DISPLAY 347
 EDITOR 351
 eingeben 68
 END 352
 GENERATE COPY 353
 HELP 354
 Hilfe ausgeben 354
 INCORPORATE 357
 INITIATE 358
 LIBTEXT DELETE 359
 N2CHARTS 360
 PRDIR 361
 PRINT 362
 PROT 363
 REESTABLISH 364
 RENAME 365
 RESTORE 366
 RETURN 352
 SERVICES 368
 TEXT 369

UDS 370

Kommandomodus 335

Kommandozeile 60

Komponente 28, **421**

Komponentenliste 29, 127

Komponententest 28

Konfiguration 43

Konzept, Funktionsentwurf 281

Kopieren, Objekt 147, 339

Kurzkommando 65, **421**

 eingeben 68

 Funktionen verketteten 96

 verketteten 68

L

Lade-Modul 274

LAN 429

Last Change, Attribut 38

LEASY 429

LIB, Process/JCL-Objektyp 272

LIBTEXT DELETE 50, 359

Listing, Hinweis 140

LMC, UDS-Objektyp 317

LMD, UDS-Objektyp 316

LMO, Process/JCL-Objektyp 274

LMS 429

Löschen

 Beziehung 86, 343

 Data Dictionary-Datei 165

 erweiterte Beziehung 343

 Funktionsauswahl 67

 Objekt 83, 343

 Struktur 149, 345

 TOM-REF-Text 342

 TOM-TI-Text 359

Logische Datei 188

Logische Jobvariable 279

LSS, UTM-Objektyp 296

LTR, UTM-Objektyp 298

M

Markieren, in der Maske 86
Markierfunktion 86
 verketteten 88
Markierspalte 86, 135
Maske **422**
 Aufbau 60
 Beziehungsliste 85
 markieren 86
 TOM-REF-Text 90
Maskeninhalte, Wiederherstellen 91
Mehrfach-benutzbares Modul 276
Mehrstufiger Beziehungstyp **422**
Meldungsfenster 60
Member **422**
Menü, DVS/Leasy 177
Meta-Daten 15
MGR, Process/JCL-Objekttyp 266
Mischen, Incorporate 151, 357
MOD, Process/JCL-Objekttyp 263
mode of relationship type **417**
Modell
 speichern 19
 Systemstruktur 17
Modellierung 17
Modul 263, **422**
 ladbar 274
 mehrfach-benutzbar 276
 nicht mehrfach-benutzbar 275
module **422**
Modulgruppe 266
Modulrahmensource 28
Modulverständigung 28
Modus, Übertragung 153
MOVE-Statement, COPY-Element 209
MSK, DVS/Leasy-Objekttyp 195

N

N²Chart **422**
n-ary relationship type **422**
N2CHARTS 50, 51, 360
Nachdokumentation 26, **422**
Name, Attribut 38
name dependent quantity of entities **422**

Namensbezogene Objektmenge **422**
Namenskonventionen, TOM-REF 46
Neues Data Dictionary anlegen 167
neutral, Attributtyp 37
Nicht mehrfach-benutzbares Modul 275
NOM, Process/JCL-Objekttyp 275

O

Objekt 18, **423**
 ändern 80, 336, 347
 anzeigen 75
 bearbeiten 71
 Datenentwurf 241, 282
 drucken (DVS/Leasy) 139
 drucken (JCL) 137
 drucken (Process) 138
 drucken (SESAM/SQL) 140
 drucken (UDS) 139
 drucken (UTM) 138
 Eintrag drucken 132
 erstellen 77, 340
 Funktionsentwurf 281
 inverse Beziehungen anzeigen 120, 350
 kopieren 146, 339
 löschen 83, 343
 Maske 71
 nicht definiertes anzeigen 129
 ohne inverse Beziehung anzeigen 127
 paarig 151, 357
 übertragen 151, 357
 umbenennen 146, 365
 Verzeichnis drucken 132
Objekt (RET), benutzereigenes definieren 203
Objekt (RLS), benutzereigenes definieren 203
Objekte
 eines Typs anzeigen 114
 Texttypen 413
 von TOM-REF 413ff
Objekteintrag, drucken 135
Objektgruppe, anzeigen 346
Objektidentifikation **423**
Objektkatalog **423**
Objektliste **423**
Objektname **423**

Objektstruktur, löschen 345
Objekttyp 18, **424**
 ALI (DVS/Leasy-Objekttyp) 202
 anzeigen 346
 ARE (UDS) 320
 auswählen 69
 Benutzereigen 40
 Beziehungen anzeigen 125, 349
 CAL(Process/JCL) 269
 CBL (Process/JCL) 273
 CMD (DVS/Leasy-Objekttyp) 197
 DAR (UDS) 321
 DBK (UDS) 329
 DBR (UDS) 315
 DBS (UDS) 314
 ENP(Process/JCL) 271
 ENT (Process/JCL) 278
 FIL (DVS/Leasy-Objekttyp) 188
 FILE-Description (Copy-Element) 213
 FKY (DVS/Leasy-Objekttyp) 196
 FLD (DVS/Leasy-Objekttyp) 178
 FLG (DVS/Leasy-Objekttyp) 183
 Gruppe 112
 GSS (UTM) 295
 Hilfe anfordern 354
 INIT-Statement (Copy-Element) 211
 INL (Copy-Element) 205
 JV(Process/JCL) 279
 LIB (Process/JCL) 272
 LMC (UDS) 317
 LMD (UDS) 316
 LMO (Process/JCL) 274
 LSS (UTM) 296
 LTR (UTM) 298
 MGR (Process/JCL) 266
 MOD (Process/JCL) 263
 MOVE-Statement (Copy-Element) 209
 MSK (DVS/Leasy-Objekttyp) 195
 NOM (Process/JCL) 275
 Other 203, 282
 PFL (DVS/Leasy-Objekttyp) 200
 PGM (Process/JCL) 262
 PJV (Process/JCL) 280
 PKY (DVS/Leasy-Objekttyp) 193

PRC (Process/JCL) 277
RBT (SESAM/SQL) 226
RCC (SQL) 230
RCT (SESAM/SQL)223
REC (DVS/Leasy-Objekttyp) 186
RFC (SQL) 231
RIX (SESAM/SQL) 228
RSG (SESAM/SQL) 232
RSM (SESAM/SQL) 225
RUC (SESAM/SQL) 229
RUS (SESAM/SQL) 234
SBR (UDS) 323
SDY (UDS) 328
SEA (UDS) 318
SEQ (DVS/Leasy-Objekttyp) 199
SET (UDS) 324
SKY (DVS/Leasy-Objekttyp) 194
SOM (Process/JCL) 276
SSL (UDS) 313
SSY (UDS) 326
Standard 36
SUB (UDS) 322
SYS (Process/JCL) 261
TAC (UTM) 293
TLS (UTM) 297
UAP (UTM) 294
UDS (UDS) 312
UTP (UTM) 291
VOL (DVS/Leasy-Objekttyp) 198
Objekttyp-Teilkatalog **424**
Objekttypkatalog **424**
Objektverzeichnis, drucken 135
OCCURS, Attribut 39
OMNIS 429
Other, Objekttyp 203, 282

P
Paging **419**
Parameter **424**
 Übertragung 153
parameter **424**
partial name **426**
PFL, DVS/Leasy-Objekttyp 200
PGM, Process/JCL-Objekttyp 262

Phase 23, **424**
Physikalische Datei 200
Physikalische Jobvariable 280
PJV, Process/JCL-Objekttyp 280
PKY, DVS/Leasy-Objekttyp 193
PLAM 429
 Bibliothek 63
Platte 198, 235
PRC, Process/JCL-Objekttyp 277
PRDIR 50, 361
predefined entity type **428**
Predefined relationship type **425**
Primärschlüssel 193
PRINT 50, 362
Print Functions 131
Printmasken 137
private data dictionary **425**
Privates Data Dictionary **425**
Problemanalyse, Phase 23
Process, Objekte drucken 138
process-engineering **428**
Process/JCL
 Menü 260
 Objekttyp (CAL) 269
 Objekttyp (CBL) 273
 Objekttyp (ENP) 271
 Objekttyp (ENT) 278
 Objekttyp (JV) 279
 Objekttyp (LIB) 272
 Objekttyp (LMO) 274
 Objekttyp (MGR) 266
 Objekttyp (MOD) 263
 Objekttyp (NOM) 275
 Objekttyp (PGM) 262
 Objekttyp (PJV) 280
 Objekttyp (PRC) 277
 Objekttyp (SOM) 276
 Objekttyp (SYS) 261
Programm 262
project library **425**
Projektbibliothek **425**
Projektmanagement 24
Projektphase
 Integration 23

- Problemanalyse 23
- System-Entwurf 23
- Systemstruktur 23
- PROT 363
- PROT ON/OFF 50
- Protokoll, ein-/ausschalten 363
- Prozedur
 - DO 277
 - ENTER 278
- Prüfen, gegen den Strukturkatalog 338
- Pseudocode 28

R

- RBT, SESAM/SQL-Objektyp 226
- RCC, SQL-Objektyp 230
- RCT, SESAM/SQL-Objektyp 223
- REC, DVS/Leasy-Objektyp 186
- REESTABLISH 50, 364
- REF-Name 46, **425**
- reference file **425**
- Referenz-Datei **425**
- Referenzbedingung 231
- REFGRP 57
- REFOTH 57
- REFREL 57
- REFVAL 55
- relationship **418**
- relationship-type **419**
- RENAME 50, 146, 365
- Responsible, Attribut 38
- RESTORE 50, 366
- RET, benutzereigenes Objekt 203
- RETURN 50, 352
- RFC, SQL-Objektyp 231
- RIX, SESAM/SQL-Objektyp 228
- RLS, benutzereigenes Objekt 203
- RSG, SESAM/SQL-Objektyp 232
- RSM, SESAM/SQL-Objektyp 225
- RUC, SESAM/SQL-Objektyp 229
- RUS, SESAM/SQL-Objektyp 234

S

SAM 429
Satz 324
Satzschlüssel 329
SAVE 50
SBR, UDS-Objektyp 323
Schlüssel
 Berechnung 317
 direkter 316
 funktionaler 196
 primärer 193
 Satz 329
 sekundärer 194
 Suche 318
Schnittstelle 27
SDF 429
SDY, UDS-Objektyp 328
SEA, UDS-Objektyp 318
Sekundärschlüssel 194
SEQ, DVS/Leasy-Objektyp 199
Sequenz 199
Servicefunktion 26, **52**, 109
 aufrufen 111
 Hilfe anfordern 354
Servicemaske, ansteuern 368
SERVICES 49, 368
SESAM 429
SESAM-DDA 29, 35
SESAM/SQL, Objekte drucken 140
SET, UDS-Objektyp 324
Short Description, Attribut 38
Shortcut **421**
Sicherungsdatei (permanent) **425**
Sicherungsdatei (temporär) **425**
SIR 100
SKY, DVS/Leasy-Objektyp 194
Soll-Vorstellung 26
SOM, Process/JCL-Objektyp 276
Source-Vorschlag 28
Space 235
Spalte 235
Speicherguppe 232
Speicherinformation 299
spezifisch, Attributtyp 37

- SQL 429
 - erweiterte Beziehung (XBT) 236
 - Menü 222
 - Objektyp RBT 226
 - Objektyp RCC 230
 - Objektyp RCT 223
 - Objektyp RFC 231
 - Objektyp RIX 228
 - Objektyp RSG 232
 - Objektyp RSM 225
 - Objektyp RUC 229
 - Objektyp RUS 234
- SQL-Beziehungstyp
 - Standard 216
 - Wertigkeit 216
- SSGEN 29, 35, 53, 79
- SSL 429
 - UDS-Objektyp 313
- SSY, UDS-Objektyp 326
- Standard
 - Beziehungstyp **425**
 - Objektyp 27
 - SQL-Beziehungstyp 216
 - UTM-Beziehungstyp 284
- Stapelbetrieb **426**
- start-object **418**
- start-procedure 418
- Starten, TOM-REF 59
- statement **417**
- Statuszeile 60
- STO, UTM-Beziehungstyp 299
- structured entity **426**
- Struktur
 - Data Dictionary 19
 - löschen 148, 345
- Strukturiertes Objekt **426**
- Strukturkatalog 34, **426**
 - Arten von Beziehungstypen 9
 - Basis-Symbol 8
 - benutzereigene Erweiterung 40
 - Darstellungsmittel 8
 - prüfen 162, 338
 - Standard 36
 - Wertigkeit von Beziehungstypen 8

SUB, UDS-Objektyp 322
Sub-Menü 65
 Copy 205
 Process/JCL 260
 SESAM/SQL 222
 SQL 260
 UDS 311
 UTM 290
sub-structure catalog **426**
Subschema 322
Subschema-Record 323
Suchschlüssel 318
Synonym (syn.) →Alias 39, 202
SYS, Process/JCL-Objektyp 261
System 261
system library **426**
System-Entwurf 24, 27
 Phase 23
Systembibliothek **426**
Systemkomponenten 27
Systemsatz 326
Systemstruktur
 modellieren 17
 Phase 23
Systemumgebung 28
Systemverhalten 23, 31

T

Tabellenspalte (SQL) 181
TAC, UTM-Objektyp 293
Taste
 K1 (Wirkung) 92
 K2 (Wirkung) 91
 K3 (Wirkung) 91
Teilname **426**
Teilprogramm (UTM) 291
Teilstrukturkatalog 34, **426**
Testdaten 24
Testversion 28
TEXT, Kommando 369
Text 73
 Attribut 38
 Attributtyp 42
 bearbeiten 89, 369

- im TOM-TI bearbeiten 90
- in der Aufrufprozedur einstellen 89
- Maske 89
- TOM-REF-Text 89
- TOM-TI-Langtext 89
- TEXT DELETE 50
- TEXT ON 50
- TEXT ON/OFF 369
- Textbearbeitung, Hilfe anfordern 354
- Textmaske, ansteuern 369
- Texttyp, eines Objekts 413
- TIAM 429
- TLS, UTM-Objekttyp 297
- TOM system library **427**
- TOM-CA 35, 430
- TOM-DOC 430
- TOM-GEN 430
- TOM-JA 35, 430
- TOM-MP 35, 430
- TOM-MTEST 29, 35, 53, 430
- TOM-Namenskonvention **427**
- TOM-REF 35, 430
 - Ablauf 63
 - beenden 60, 352
 - Datenhaltung 43
 - Funktion 47
 - im Funktionsverbund 1
 - Kommando 49, 335ff
 - Masken-Layout 47
 - Namenskonventionen 46
 - Objekte 413ff
 - Starten 59
 - Texttyp eines Objekts 413
 - Unterbrechen 91
- TOM-REF-Text
 - löschen 342
 - übertragen 158
 - Maske 90
- TOM-REFG 35, 51, 430
 - aufrufen 360
- TOM-RTEST 430
- TOM-Systembibliothek **427**
- TOM-TA 35, 430
- TOM-TI 430

aufrufen 351
Texttyp eines Objekts 413
TOM-TI-Text
bearbeiten 90
löschen 359
übertragen 160
TOMLIB 63, 430, **426**
Data Dictionary holen 165, 366
TOMRFI-Datei 44, 46, 166, 364, **427**
neu aufbauen 166, 364
Toolsteuerung 29
TR0000 62
TR0001 64, 177
TR0002 90
TR0003 260
TR0005 160, 359
TR0008 102
TR0010 178
TR0012 181
TR0020 183
TR0030 186
TR0040 188
TR0042 190
TR0050 195
TR0060 206
TR0062 207
TR0063 209
TR0064 211
TR0065 213
TR0070 193
TR0080 194
TR0090 197
TR0100 196
TR0110 199
TR0120 198
TR0200 103
TR0300 261
TR0310 262
TR0320 263
TR0330 271
TR0340 269
TR0350 266
TR0400 273
TR0430 276

TR0440 274
TR0450 275
TR1000 311
TR1010 312
TR1020 314
TR1030 313
TR1040 322
TR1050 320
TR1060 315
TR1070 316
TR1080 317
TR1090 318
TR1100 324
TR1110 329
TR1120 323
TR1130 326
TR1140 328
TR1150 321
TR2000 222
TR2010 226
TR2020 231
TR2030 228, 230
TR2040 237
TR2050 225
TR2070 229
TR2110 223
TR2120 232
TR2130 234
TR2140 236
TR5000 203, 282
TR5010 202
TR6000 277
TR6010 200
TR6020 279
TR6030 280
TR6040 278
TR7010 272
TR8000 290
TR8010 291
TR8020 293
TR8030 294
TR8040 298
TR8050 295
TR8060 296

TR8070 297
TR8080 299
TR8090 300
TR8888 355
TR9000 109
TR9010 114
TR9019 113
TR9020 118
TR9030 123
TR9040 127
TR9050 129
TR9060 143
TR9070 131
TR9072 137
TR9073 139
TR9074 138
TR9075 139
TR9078 138
TR9079 137
TR9083 140
TR9084 115
TR9090 145
TR9100 152
TR9102 156
TR9102 158
TR9200 146, 339, 365
TR9210 149
TR9800 163
TR9900 164
TR9910 165
TR9930 167
Typ 33
 Attribut 38
Typ-neutraler Attributtyp **428**
Typ-spezifischer Attributtyp **428**
Typbezogene Objektmenge **427**
type dependend quantity of entities **427**
type specific attribute type **428**

U

UAP, UTM-Objekttyp 294

UDS 370, 430

ARE (Objekttyp) 320

DAR (Objekttyp) 321

DBK (Objekttyp) 329

DBR (Objekttyp) 315

DBS (Objekttyp) 314

LMC (Objekttyp) 317

LMD (Objekttyp) 316

Objekte drucken 139

SBR (Objekttyp) 323

SDY (Objekttyp) 328

SEA (Objekttyp) 318

SET (Objekttyp) 324

SSL (Objekttyp) 313

SSY (Objekttyp) 326

SUB (Objekttyp) 322

UDS (Objekttyp) 312

UDS-Objekttyp 312

UDS-Datenbank 312

UDS-DDA 29, 35, 145

aufrufen 370

UDS-Schema 313

Übertragen

TOM-REF-Text 158

TOM-TI-Text 160

Übertragung

Modus 153

Parameter 153

Umfang (Incorporate) 152

Umbenennen, Objekt 147, 365

Umfang (Incorporate) , Übertragung 152

Umfeldkomponente 29

Undefiniert, Objekt anzeigen 129

Unterbrechen, TOM-REF 91

user defined [dictionary] attribute type **418**

user defined [dictionary] entity type **418**

user defined [dictionary] relationship type **418**

UTM 430

Anwendung 294

FOR (Beziehungstyp) 300

GSS (Objekttyp) 295

LSS (Objekttyp) 296

LTR (Objekttyp) 298
Objekte drucken 138
STO (Beziehungstyp) 299
TAC (Objekttyp) 293
TLS (Objekttyp) 297
UAP (Objekttyp) 294
UTP (Objekttyp) 291
UTM-Beziehungstyp
 Standard 284
 Wertigkeit 284
UTM-GSSB 295
UTM-LSSB 296
UTP, UTM-Objekttyp 291

V

Verfahrenstechnik **428**
Verketten
 Funktion 92
 Funktionen mit Kurzkommandos 96
 Funktionsauswahl 67
 Kurzkommandos 68
 Markierungen 88
Verwendungsnachweis 26
View 226
VOL, DVS/Leasy-Objekttyp 198
Vorbelegung des Objektname 74
Vordefiniert
 Beziehungstyp 216
 Objekttyp **428**
Voreinstellung 65
VTSU 430

W

Wertigkeit
 des Beziehungstyps **428**
 SQL-Beziehungstyp 216
 UTM-Beziehungstyp 284
 von Beziehungstypen, Strukturkatalog 8
Wiederherstellen , Maskeninhalte 91

X

XBT, SESAM/SQL-Beziehung 236

XIX, SQL-Beziehung 237

Z

Zentrales Data Dictionary **428**

Zielobjekt **428**

Zurücksetzen, Auswahl 67

Inhalt

1	Einleitung	1
1.1	Kurzbeschreibung des Produkts	1
1.1.1	Strukturkatalog	3
1.1.2	Datenhaltung	3
1.1.3	Ein-/Ausgabe	4
1.1.4	Unterstützung der Software-Entwicklung	4
1.2	Zielsetzung und Zielgruppe des Handbuchs	5
1.3	Konzept des Handbuchs	5
1.4	Änderungen gegenüber dem Handbuch TOM-REF V2.2	6
1.5	Darstellungsmittel	7
2	Data Dictionary: Bedeutung, Nutzen und Einsatz	15
2.1	Begriffsklärung	15
2.2	Nutzen eines Data Dictionary	16
2.2.1	Problemstellung: Komplexität der Systeme	16
2.2.2	Lösungsschritt 1: Systemstruktur modellieren	17
2.2.3	Lösungsschritt 2: Modell speichern	19
2.2.4	Lösungsschritt 3: System speichern	20
2.2.5	Lösungsschritt 4: Gespeicherte Daten auswerten	22
2.2.6	Problemlösung: Der erkennbare Nutzen	22
2.3	TOM-REF in den Projektphasen einsetzen	23
2.3.1	Projektorganisation in Grundzügen	23
2.3.2	Problemanalyse	26
2.3.3	Systementwurf	27
2.3.4	Realisierung	28
2.3.5	Integration	31
2.3.6	Einsatz	31
3	Konzept von TOM-REF	33
3.1	Der Strukturkatalog	34
3.1.1	Standard-Einträge	36
3.1.2	Benutzereigene Erweiterungen	40

3.2	Datenhaltung	43
3.2.1	Konfigurationen	43
3.2.2	Datenhaltungssystem	44
3.2.3	Namenskonventionen	46
3.3	Funktionen	47
3.3.1	Grundfunktionen	47
3.3.2	Servicefunktionen	52
3.4	Strukturkatalog erweitern	55
4	Mit TOM-REF arbeiten	59
4.1	TOM-REF starten und beenden	59
4.2	Allgemeiner Maskenaufbau	60
4.3	Data Dictionary auswählen	61
4.4	Funktionen auswählen	64
4.4.1	Der Arbeitsbereich der Auswahlmaske TR0001	66
4.4.2	Eingaben in die Auswahlmaske TR0001	66
4.4.3	Typ im Sub-Menü auswählen	69
4.5	Objekte bearbeiten	71
4.5.1	Die Objektmaske	71
4.5.2	Grundfunktionen	73
4.5.2.1	Objekt anzeigen [DISPLAY]	75
4.5.2.2	Objekt erstellen [CREATE]	77
4.5.2.3	Objekt ändern [CHANGE]	80
4.5.2.4	Objekte löschen [DELETE]	83
4.5.3	Beziehungsliste bearbeiten	84
4.5.3.1	Die Beziehungslisten-Maske	84
4.5.3.2	Beziehungen markieren	86
4.5.3.3	In der Beziehungsliste blättern	89
4.5.4	Texte bearbeiten	89
4.5.4.1	Die TOM-REF-Textmaske	89
4.5.4.2	Texte im TOM-TI bearbeiten	90
4.5.5	Funktionstasten und Funktionskettung	91
4.5.5.1	Wirkung der Funktionstasten	91
4.5.5.2	Funktionen durch Ankreuzen verketteten	92
4.5.5.3	Funktionen mit Kurzkommandos verketteten	96
4.6	Beziehungen bearbeiten	100
4.6.1	Einfache Beziehungen	101
4.6.2	Erweiterte Beziehungen bearbeiten	104
5	Servicefunktionen	109
5.1	Display directory for type	112
5.1.1	Alle Objekte eines Objekttyps oder einer Objekttyp-Gruppe anzeigen	112
5.1.2	Auswahl von Objekten eines oder mehrerer Objekttypen anzeigen	114
5.2	Display where-used	117
5.2.1	Alle inversen Beziehungen anzeigen	118

5.2.2	Inverse Beziehungen anzeigen, die von Objekten eines Typs ausgehen	120
5.3	Display uses	122
5.3.1	Alle Beziehungen anzeigen	123
5.3.2	Beziehungen zu Objekten eines bestimmten Typs anzeigen	125
5.4	Display objects which are not part of another object	127
5.5	Display referenced objects which are not defined	129
5.6	Print functions	131
5.6.1	Objektverzeichnisse und Objekteinträge drucken	135
5.6.2	Hinweise in den Listen	140
5.7	Generate functions	142
5.7.1	Copy-Elemente erzeugen	143
5.7.2	UDS-DDA: Anschluß von UDS an TOM-REF	145
5.8	Rename/copy object	146
5.9	Delete structure	148
5.10	Incorporate from another Data Dictionary	151
5.10.1	Allgemeiner Ablauf	151
5.10.2	Modus Dialog Control	155
5.10.2.1	Behandlung der Objektattribute	155
5.10.2.2	Behandlung der Beziehungslisten	156
5.10.2.3	Behandlung des TOM-REF-Textes	158
5.10.2.4	Behandlung der TOM-TI-Langtexte	160
5.10.2.5	Ende der Übertragung	161
5.11	Check objects and relationships	162
5.12	Administration (ISAM)	163
5.12.1	Save into TOMLIB - Data Dictionary in TOMLIB sichern	164
5.12.2	Restore from TOMLIB - Data Dictionary aus TOMLIB holen	165
5.12.3	Reestablish TOMREFI - TOMRFI-Datei neu aufbauen	166
5.12.4	Initiate TOM-REF Data Dictionary - Neues Data Dictionary anlegen	167
6	DVS/Leasy	169
6.1	DVS/Leasy-Objekte	169
6.2	DVS/Leasy-Beziehungen	170
6.3	DVS/Leasy-Teilstrukturkatalog	171
6.4	Menü DVS/Leasy	177
6.4.1	FLD Feld	178
6.4.2	FLG Feldgruppe	183
6.4.3	REC Datensatz	186
6.4.4	FIL Logische Datei	188
6.4.5	PKY Primärschlüssel	193
6.4.6	SKY Sekundärschlüssel	194
6.4.7	MSK Format	195
6.4.8	FKY Funktionstasten	196
6.4.9	CMD Kommando	197
6.4.10	VOL Datenträger	198

6.4.11	SEQ Sequenz	199
6.4.12	PFL Physikalische Datei	200
6.4.13	ALI Alias	202
6.4.14	RET, RLS, Other - Benutzereigene Objekttypen	203
6.5	INL - Standard-Objekttyp für Copy-Elemente	205
6.5.1	Copy-Objektmasken aufrufen	205
6.5.2	Copy-Element Data-Area	207
6.5.3	Copy-Element MOVE-Statements	209
6.5.4	Copy-Element INIT-Statements	211
6.5.5	Copy-Element FILE-Description	213
7	SESAM/SQL V2	215
7.1	SESAM/SQL V2-Objekte	215
7.2	SESAM/SQL V2-Beziehungen	216
7.3	SESAM/SQL V2-Teilstrukturkatalog	217
7.4	Sub-Menü SESAM/SQL	222
7.4.1	RCT SQL-Datenbank (Catalog)	223
7.4.2	RSM Datenbank-Schema	225
7.4.3	RBT Basistabelle	226
7.4.4	RIX Index-Beschreibung	228
7.4.5	RUC Eindeutigkeitsbedingung (Uniqueness Constraint)	229
7.4.6	RCC Check-Bedingung (Check Constraint)	230
7.4.7	RFC Referenzbedingung (Referential Constraint)	231
7.4.8	RSG SQL-Speichergruppe	232
7.4.9	RUS Eigentümer (User)	234
7.4.10	FLD Feld	235
7.4.11	PFL Physische Datei	235
7.4.12	VOL Datenträger	235
7.4.13	XBT vordefinierter Beziehungstyp View-Feld	236
7.4.14	XIX vordefinierter Beziehungstyp Index-Feld	237
8	Process/JCL	239
8.1	Process/JCL-Objekte	239
8.2	Process/JCL-Beziehungen	241
8.3	Teilstrukturkataloge für Process/JCL	243
8.3.1	JCL-Teilstrukturkatalog	244
8.3.2	Programmdesign-Teilstrukturkatalog	247
8.3.3	Programmrealisierung-Teilstrukturkatalog	252
8.3.4	Funktionsentwurf-Teilstrukturkatalog	255
8.3.5	Vorschlag für einen Datenentwurf-Teilstrukturkatalog	258
8.4	Sub-Menü Process/JCL	260
8.4.1	SYS System	261
8.4.2	PGM Programm	262
8.4.3	MOD Modul	263
8.4.4	MGR Modulgruppe	266

8.4.5	CAL Call	269
8.4.6	ENP Entry-Punkt	271
8.4.7	LIB Bibliothek	272
8.4.8	CBL COBOL-Source	273
8.4.9	LMO Lademodul	274
8.4.10	NOM Nicht-Shareable Objektmodul	275
8.4.11	SOM Shareable-Objektmodul	276
8.4.12	PRC DO-Prozedur	277
8.4.13	ENT ENTER-Prozedur	278
8.4.14	JV Logische Jobvariable	279
8.4.15	PJV Physikalische Jobvariable	280
8.4.16	Objekte des Funktionsentwurf-Teilstrukturkatalogs	281
9	UTM	283
9.1	UTM-Objekte	283
9.2	UTM-Beziehungen	284
9.3	UTM-Teilstrukturkatalog	285
9.4	Sub-Menü UTM	290
9.4.1	UTP UTM-Teilprogramm	291
9.4.2	TAC UTM-TAC	293
9.4.3	UAP UTM-Anwendung	294
9.4.4	GSS UTM-GSSB	295
9.4.5	LSS UTM-LSSB	296
9.4.6	TLS UTM-TLS	297
9.4.7	LTR UTM-LTERM	298
9.4.8	STO Standard-Beziehungstyp Speicher-Information	299
9.4.9	FOR Standard-Beziehungstyp Format-Information	300
10	UDS	301
10.1	UDS-Objekte	301
10.2	UDS-Beziehungen	302
10.3	UDS-Teilstrukturkatalog	303
10.4	Sub-Menü UDS	311
10.4.1	UDS UDS-Datenbank	312
10.4.2	SSL Storage Structure Language	313
10.4.3	DBS Datenbank-Schema	314
10.4.4	DBR Datenbank-Record	315
10.4.5	LMD Direct-Key	316
10.4.6	LMC Calc-Key	317
10.4.7	SEA Search-Key	318
10.4.8	ARE Area	320
10.4.9	DAR Dynamische Area	321
10.4.10	SUB Subschema	322
10.4.11	SBR Subschema-Record	323
10.4.12	SET SET	324

10.4.13	SSY System-Set	326
10.4.14	SDY Dynamischer Set	328
10.4.15	DBK Set-Key	329
11	Sonderfälle der Datenmodellierung	331
11.1	Modulverbindungen modellieren	331
11.2	Dialogstruktur modellieren	332
11.3	Integrierte Modul- und Dialogstruktur modellieren	333
12	TOM-REF Kommandos	335
	CHANGE Objekt oder Beziehung ändern	336
	CHECK Gegen den Strukturkatalog prüfen	338
	COPY Objekt kopieren	339
	CREATE Objekt oder Beziehung erzeugen	340
	DDTEXT DELETE TOM-REF-Text löschen	342
	DELETE Objekt oder Beziehung löschen	343
	DELSTRUCT Objektstruktur löschen	345
	DISDIR Inhaltsverzeichnis ausgeben	346
	DISPLAY Objekt oder Beziehung anzeigen	347
	DISP-USES Beziehungsliste anzeigen	349
	DISP-WHERE-USED Inverse Beziehungen anzeigen	350
	EDITOR TOM-TI aufrufen	351
	END, RETURN TOM-REF beenden	352
	GENERATE COPY COBOL-Copy-Element erzeugen	353
	HELP Hilfemaske ansteuern	354
	INCORPORATE Data Dictionaries zusammenführen	357
	INITIATE Neues Data Dictionary anlegen	358
	LIBTEXT DELETE TOM-TI-Text löschen	359
	N2CHARTS N2Charts aufrufen	360
	PRDIR Inhaltsverzeichnis ausdrucken	361
	PRINT Auswahlmaske für den Druck ansteuern	362
	PROT Dialog protokollieren	363
	REESTABLISH TOMREFI-Datei neu aufbauen	364
	RENAME Objekt umbenennen	365
	RESTORE Data Dictionary laden	366
	SAVE Data Dictionary sichern	367
	SERVICES Servicekommandos ansteuern	368
	TEXT Textmaske ansteuern	369
	UDS UDS-DDA aufrufen	370

13 **Meldungen** 371

14 **TOM-REF-Objekte** 413

Glossar 417

Abkürzungen 429

Literatur 431

 Literatur zur TOM-Entwicklungsumgebung 431

 Weiterführende Literatur 435

Stichwörter 438

TOM-REF V3.0 (BS2000/OSD)

Data-Dictionary-System

Zielgruppe

BS2000-Entwickler, die Projekte und Software sinnvoll verwalten wollen

Inhalt

Im Einführungsteil ist das Arbeiten mit einem Data Dictionary im allgemeinen und TOM-REF im besonderen beschrieben, sowie der Einsatz der Grund- und Servicefunktionen. Im Nachschlageteil sind die produktspezifischen Informationsmodelle, die Sonderfälle der Datenmodellierung, Kommandos und Meldungen dargestellt.

Ausgabe: Juli 1995

Datei: TOM_REF.PDF

BS2000 ist ein eingetragenes Warenzeichen der
Siemens Nixdorf Informationssysteme AG

Copyright © Siemens Nixdorf Informationssysteme AG, 1994.

Alle Rechte vorbehalten, insbesondere (auch auszugsweise) die der Übersetzung, des Nachdrucks, Wiedergabe durch Kopieren oder ähnliche Verfahren.

Zu widerhandlungen verpflichtet zu Schadenersatz.

Alle Rechte vorbehalten, insbesondere für den Fall der Patenterteilung oder GM-Eintragung.

Liefermöglichkeiten und technische Änderungen vorbehalten.



Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@ts.fujitsu.com.

The Internet pages of Fujitsu Technology Solutions are available at

[http://ts.fujitsu.com/...](http://ts.fujitsu.com/)

and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@ts.fujitsu.com.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter

[http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009