CMX V6.0B (Solaris)

Betrieb und Administration

Ausgabe Juni 2005

Kritik… Anregungen… Korrekturen…

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Für Ihre Kommentare stehen Ihnen Fax-Formulare auf den letzten Seiten dieses Handbuchs zur Verfügung.

Dort finden Sie auch die Adressen der zuständigen Redaktion.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2000

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2000 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Dieses Handbuch wurde
auf chlorfrei gebleichtem
Papier gedruckt.

weitere Kapitel ▶

Inhalt

1 Einleitung

Der Communications Manager for UNIX Systems (CMX) ist das Basisprodukt der Solaris-Kommunikationssoftware. Zusammen mit den Communication Control Programs (CCPs) realisiert CMX ein offenes Kommunikationssystem. CMX vermittelt zwischen unterschiedlichen Transportsystemen und Programmschnittstellen und bietet Ihnen die Möglichkeit, Programm-Programm-Kommunikation unabhängig von den verwendeten Transportsystemen zu betreiben. Gleiches gilt für Ihre eigenen Anwendungen, die Sie mit Hilfe der angebotenen Programmschnittstellen (ICMX, XTI) erstellen können.

Mit CMX und den entsprechenden Communication Control Programs (CCP) können Sie alle gängigen Kommu nikationsdienste nutzen:

Network Access Services bieten Ihnen Zugang

– zu Wide Area Networks (WANs) wie PSDN, CSDN, PSTN, Frame Relay

– zu Ethernet-, Fast Ethernet-, Gigabit-Ethernet-, Token Ring- und FDDI-LANs

– zu ISDN über S_0 und S_2-Anschlüsse

End-to-End Services unterstützen

– RFC 1006 über TCP/IP-, OSI- und TRANSDATA NEA-Protokolle im Transportsystem

– umfassende Integration Ihres Solaris-Systems in SNA-Netze

– X.25-Kopplung von Systemen und Terminals über WAN/ISDN ohne Transportprotokoll

Die folgende Abbildung zeigt die Produktstruktur der Solaris-Kommunikation zusammen mit den Subnetzen, die die CCP-Produkte bedienen.

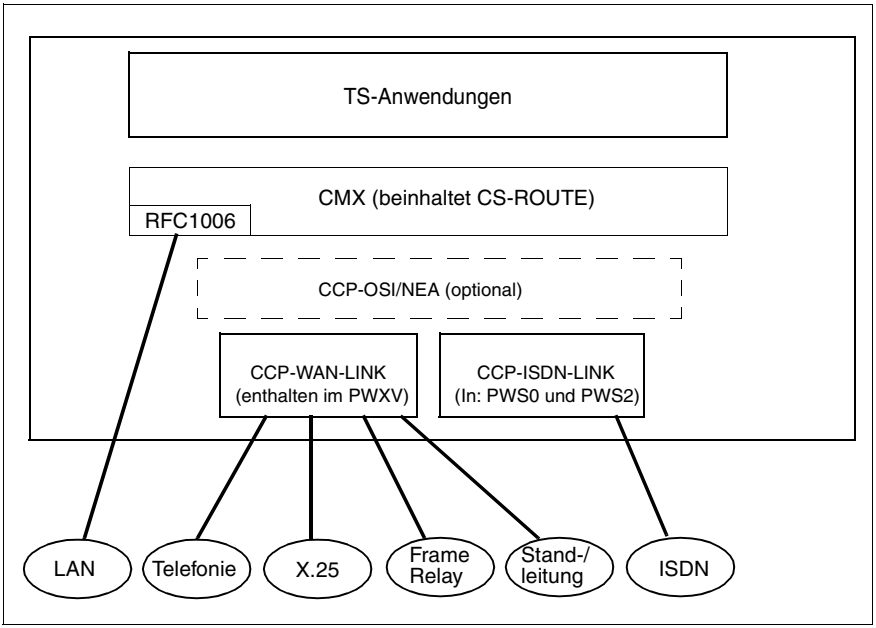


Bild 1: Übersicht der Produktstruktur der Solaris-Kommunikation

CMX bietet alle notwendigen Funktionen zur Konfiguration und Administration Ihrer Kommunikationsmittel. Sie können sowohl über die Kommandoschnittstelle als auch über Menü sämtliche Administrations-, Wartungs- und Diagnoseaufgaben erledigen. Hierzu ist keine root-Berechtigung mehr unbedingt erforderlich. Sämtliche Administrations-, Wartungs- und Diagnoseaufgaben sind unter der Benutzerrolle *cmxadm* realisierbar.

Bestandteil des Produktes CMX ist das Transportprofil TCP/IP-RFC1006.

1.1 Zielgruppe des Handbuchs

Dieses Handbuch richtet sich an Systemverwalter. Um mit CMX arbeiten zu können, benötigen Sie Kenntnisse des Betriebssystems Solaris. Ferner sind für das Verständnis Kenntnisse über Prinzipien und Methoden der Datenfernverarbeitung hilfreich, insbesondere das OSI-Referenzmodell, wie es in ISO7498 normiert ist.

1.2 Konzept des Handbuchs

Die Beschreibung des gesamten Produkts CMX umfasst zwei Handbücher:

– Das vorliegende CMX-Benutzerhandbuch für Systemverwalter

– CMX-Benutzerhandbuch „Anwendungen programmieren“ für den Programmierer von TS-Anwendungen (TS = Transport Service), die die CMX-Programmschnittstellen benutzen

Das Handbuch beschreibt in Kapitel 2 die Architektur der Solaris-Kommunikationssoftware. Sie erhalten dort einen Überblick über die erhältlichen CCP-Produkte und ihre Funktionalität sowie wichtige Basisinformation zur CMX-Systemverwaltung.

Kapitel 3 widmet sich der Benutzerrolle *cmxadm*.

Kapitel 4 enthält grundlegende Informationen zur Adressierung mit Hilfe der Komponenten TNS und FSS. Sie benötigen diese Kenntnisse, um Ihr Rechnernetz konfigurieren zu können.

Wie Sie bei der Installation vorgehen müssen, erfahren Sie in Kapitel 5.

Kapitel 6 stellt die Vorgehensweise beim Konfigurieren und Administrieren mit der zeichenorientierten Bedienoberfläche CMXCUI vor.

Kapitel 7 beschreibt Eingabe- und Dateiformat der CMX-Komponenten TNS und FSS.

Kapitel 8 widmet sich der web-basierten CMX-Administration. Es beschreibt Installation, Konfiguration und Start des Paketes CMXwca, das den Zugang zur CMX-Administration aus der web-basierten Oberfläche heraus ermöglicht, sowie die Lösung von Problemfällen und Sicherheitsaspekte.

Kapitel 9 enthält Adress- und Kommando-Informationen zum Transportdienst RFC1006 über TCP/IP, der als Bestandteil von CMX ausgeliefert wird.

Kapitel 10 beschreibt die für die CMX-Systemverwaltung relevanten Kommandos mit ihrer Syntax.

Kapitel 11 beschreibt eine zusätzliche Komponente von CMX, den CMX-Agenten zur Fernadministration von CMX über SNMP. Es enthält Informationen zu Funktion und Betrieb des CMX-Agenten.

Kapitel 12 beschreibt den Ablauf von TLI-Anwendungen über CMX.

Das CMX-Benutzerhandbuch „Anwendungen programmieren“ beschreibt die Programmschnittstellen von CMX, d. h. alle Programmaufrufe, die Sie benötigen, um selbst Anwendungen zu entwickeln.

Man Pages

Mit dem Produkt CMX werden *Online Man Pages* ausgeliefert. Sie dokumentieren über die im Handbuch beschriebenen Kommandos hinaus weitere Experten-Kommandos, die Sie bei Bedarf für Ihre Systemverwaltung nutzen können. Innerhalb dieses Handbuches finden Sie gelegentlich Referenzen auf diese *Man Pages*.

Verweise auf die Freigabemitteilung

Betriebssystemspezifische Besonderheiten werden nicht in diesem Benutzerhandbuch, sondern in der Freigabemitteilung (produktspezifischen Readme-Datei) beschrieben.

Verweise auf andere Handbücher

Im Text wird mit „siehe Handbuch ’Handbuchtitel’ [*n*]" auf andere Handbücher verwiesen, die weiterführende Informationen enthalten. Dabei ist *n* eine Ziffer. Unter [*n*] finden Sie die Titel der entsprechenden Handbücher im Literaturverzeichnis zusammen mit einer kurzen Inhaltsangabe.

1.3 Änderungen gegenüber der Vorgängerversion

Gegenüber dem Vorgängerhandbuch CMX V6.0A (Solaris), Ausgabe Juni 2003, gibt es folgende Änderungen:

CMX ist Live Upgrade fähig

CMX kann jetzt bei laufendem Berieb in einer alternativen Boot-Umgebung nachinstalliert werden. Details finden Sie im Kapitel „Installation und Inbetriebnahme“ auf Seite 53.

Erweiterte Web-basierte Administration

Die Web-basierte Administration wird jetzt mit Hilfe von ServerView realisiert. Damit können die CMX-Kommunikationsserver wahlweise über eine LAN-Konsole/SMC oder über ServerView-basierte Windows-Administrationsclients verwaltet werden.

Zusätzlich ist es jetzt möglich, die Kommunikation zwischen Administrationsclient und Kommunikationsserver per SSL/TLS zu verschlüsseln.

Nähere Informationen siehe Kapitel „Web-basierte CMX-Administration“ auf Seite 137.

Neue und erweiterte CMX-Administrationskommandos

Die Kommandoschnittstelle wurde um folgende Kommandos erweitert:

– **cmxconf**
Konfiguration einer CMX-Anwendung prüfen, siehe Seite 231.

– **cmxstat**
TSP-spezifische Statusinformationen ausgeben, siehe Seite 283.

– **cmxtrc**
Traces für eine Transportsystem ein- und ausschalten, siehe Seite 287.

Außerdem wurden folgende Kommandos geändert:

- **cmxprod**
Produktinformation kann nach Boot-Umgebung und Root-Directory differenzert ausgegeben werden, siehe Seite 281.

- **comtr**
Trace-Daten können direkt nach *stdout* ausgegeben werden, siehe Seite 292.

- **StartStop-Kommandos**
In- und Außerbetriebnhme von CMX und/oder seinen Komponenten bei laufendem System im Rahmen der Live Upgrade Installation, siehe Seite 303.

1.4 Readme-Dateien

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. den produktspezifischen Freigabemitteilungen. Diese finden Sie im Readme-Paket, welches mit dem jeweiligen Produkt ausgeliefert wird.

1.5 Darstellungsmittel

Soweit möglich hält sich die Kommandobeschreibung an ein festes Raster:

– Beschreibung des Kommandos
– Syntax
– Syntaxbeschreibung
– Ausgabeformat
– Ende-Status
– Fehlermeldungen
– Dateien
– Beispiel
– Siehe auch

Die hier aufgeführten Bestandteile werden im Folgenden erläutert.

Beschreibung des Kommandos

Im ersten Abschnitt jeder Kommandobeschreibung ist Folgendes dargestellt:

– die Arbeitsweise des Kommandos
– die unterschiedlichen Aufgaben der verschiedenen Kommando-Formate, falls mehrere Formate vorhanden sind
– in welcher Umgebung das Kommando zu verwenden ist (z. B. Einträge in Dateien, Zugriffsrechte)
– Hintergrundinformationen

Syntax

cmd [␣**-a**] [␣**-b**] [␣**-c**] [␣**-d**␣arg1] [␣**-f**␣arg2]␣datei␣...

Sie müssen *cmd* eingeben sowie für *datei* eine oder mehrere Dateien, die jeweils durch ein Leerzeichen voneinander getrennt werden. Sie können zusätzlich angeben:

– eine oder mehrere Optionen **-a, -b, -c**. Diese Optionen können Sie einzeln (**-a**␣**-b**␣**-c**) oder zusammen (**-abc**) angeben.

– die Option *-d*, wobei *arg1* durch ein Argument ersetzt werden muss

– die Option *-f*, wobei *arg2* durch ein Argument ersetzt werden muss

Halbfette Zeichen
Konstanten. Halbfett gedruckte Zeichen müssen genau wie dargestellt eingegeben werden.

Normale Zeichen
: Variablen. Diese Zeichen sind Stellvertreter für andere Zeichen, die Sie auswählen und eingeben.

[ ] Optionen. Argumente in eckigen Klammern sind optional und müssen nicht angegeben werden. Die eckigen Klammern sind nicht einzugeben, es sei denn, es wird ausdrücklich darauf hingewiesen.

␣ Leerzeichen, das Sie eingeben müssen.

.. Der vorherige Ausdruck kann wiederholt werden. Falls zwischen den Wiederholungen Leerzeichen eingegeben werden müssen, die nicht im Ausdruck enthalten sind, steht vor ... ein ␣(Leerzeichen).

{ | } Auswahlmöglichkeit. Wählen Sie genau einen der Ausdrücke aus, die durch den Strich getrennt sind.

<u>unterstrichen</u>
: Voreinstellung

Ist es bei einem Kommando möglich, für eine Option mehrere alternative Angaben zu machen, so wird die Kommandosyntax zweimal aufgeführt. Einmal wird für die entsprechende Option ein Stellungsparameter angegeben. In der zweiten Darstellung stehen statt des Stellungsparameters alle möglichen Angaben für die Option.

Die zweite Darstellung soll dem Profi ein schnelles Nachschlagen ermöglichen. Die Rasterung ist zwischen den beiden Darstellungen unterbrochen. Im ersten Raster steht jeweils die Darstellungsform mit Stellungsparametern.

Syntaxbeschreibung

Hier finden Sie die Beschreibung von Optionen und Argumenten (Eingabedateien, Parameter, Variablen etc.), die Sie beim Aufruf eines Kommandos eingeben können. Im Fließtext wird nicht zwischen Konstanten und Variablen unterschieden. Alle Syntaxelemente sowie Dateinamen, Pfadnamen und Kommandos sind dort in *kursiver* Schrift dargestellt.

Ausgabeformat

In diesem Abschnitt werden die Ausgabeformate der Kommandos erläutert.

Ende-Status

Ein Ende-Status ist der Wert, den ein Kommando nach seiner Ausführung an den aufrufenden Prozess zurückliefert. Der Wert gibt Auskunft darüber, wie das Kommando abgelaufen ist. Der Ende-Status ist ein Zahlenwert und wird in der Variablen ? abgelegt. Sie fragen den Ende-Status mit dem Befehl *echo $?* ab.

Der Ende-Status wird nur dann beschrieben, wenn er von folgendem Regelfall abweicht:

0 nach korrekter Durchführung des Kommandos

≠0 bei Fehler

Fehler

Hier werden wichtige Fehlermeldungen angegeben und erläutert sowie Hinweise zur Fehlervermeidung und -behebung gegeben.

Fehlermeldungen werden generell auf die Standard-Fehlerausgabe *stderr* ausgegeben. Normalerweise ist der Bildschirm die Standard-Fehlerausgabe.

Dateien

Hier werden Dateien angegeben, auf die das betreffende Kommando zugreift oder die von dem Kommando erzeugt werden.

Beispiel

Beispiele sollen die Hauptfunktion des Kommandos, den Einsatz wesentlicher Optionen sowie sinnvolle Kombinationen von Optionen und Argumenten veranschaulichen. In Anwendungsbeispielen sind Eingaben in das System dicktengleich halbfett dargestellt. Alle diese Eingabezeilen werden mit der ↵-Taste abgeschlossen. Die Taste wird daher am Ende der Zeilen nicht angegeben.

Ausgaben des Systems werden, außer im Fließtext, `dicktengleich` dargestellt. Im Fließtext erscheinen die Ausgaben *kursiv*.

Siehe auch

Hier finden Sie Verweise auf andere Kommandos, die eine ähnliche Funktionsweise haben oder mit dem betreffenden Kommando zusammenarbeiten. Außerdem wird auf weitere Literatur zu diesem Kommando verwiesen.

Hinweise und Warnungen

Dieses Symbol weist auf besonders wichtige Informationen hin, die Sie unbedingt beachten sollten.

Vorsicht!

Dieses Symbol weist auf Gefahren hin, die zu Datenverlust oder Geräteschaden führen können.

2 Architektur der Solaris-Kommunikation

In diesem Kapitel erhalten Sie einen Überblick über die Solaris-Kommunikationsprodukte. Sie lernen die wesentlichen Eigenschaften dieser Produkte kennen und erfahren, welche Rolle CMX in der Solaris-Kommunikation spielt, und welche Dienste CMX Ihnen anbietet. Alle zentralen Begriffe, die in den nachfolgenden Kapiteln verwendet sind, werden hier erklärt.

Anhand der Solaris-Kommunikationsprodukte wird die Vielzahl der Vernetzungsmöglichkeiten aufgezeigt, und die von CMX angebotenen Dienste werden aufgezählt. Sie erfahren so im Überblick, welche Leistungen von den Produkten erbracht werden.

Da für den Einsatz von CMX ein allgemeines Verständnis der Architektur der Solaris-Kommunikation erforderlich ist, beschreiben die folgenden Abschnitte, wie diese Leistungen erbracht werden. Anhand der Architektur wird die Terminologie eingeführt, wie sie in den Handbüchern zur Solaris-Kommunikation verwendet ist. Voraussetzung zum Verständnis sind lediglich Grundkenntnisse der Datenkommunikation sowie der Basis-Struktur des Solaris-Betriebssystems.

2.1 Leistungsumfang von CMX und CCPs

Die Familie der Solaris-Kommunikationsprodukte bietet für alle wichtigen Datennetze die entsprechenden Transportprofile. Zum Erstellen von Anwendungsprogrammen für die Kommunikation werden mehrere Programmschnittstellen angeboten.

Um zwischen den verschiedenartigen Transportprofilen und Programmschnittstellen zu vermitteln, präsentiert CMX den TS-Anwendungen ein einheitliches Bild des Transportsystems. Dadurch haben Sie den Vorteil, TS-Anwendungen unabhängig vom Transportsystem entwickeln zu können. Auf welche CCP-Profile die Anwendungen aufsetzen, wird erst zum Ablaufzeitpunkt durch die Konfigurierung entschieden (siehe Abschnitt „Ablauf eines Verbindungsaufbaus“ auf Seite 50).

CMX und CCPs benötigen entsprechende Systemgrenzwerte für uneingeschränkte Kommunikation. Beachten Sie hierzu die Hinweise in Kapitel „Installation und Inbetriebnahme“ auf Seite 53 sowie in der Freigabemitteilung.

Administration und Diagnose

CMX bietet Kommandos zur Abfrage von Informationen über die Auslastung der Kommunikationsmittel sowie zu Konfiguration und Grenzwerten der Kommunikationsprodukte. Mit weiteren Kommandos kann der Systemverwalter bei Problemen Diagnoseinformationen abfragen. Mit der Menüoberfläche CMXCUI (siehe Kapitel „Konfigurieren und Administrieren im Menü“ auf Seite 59) aktivieren Sie diese Funktionen auf einfache Weise.

Für die Nutzung dieser Funktionen ist keine root-Berechtigung mehr erforderlich. Sämtliche Funktionen der Konfiguration, Administration und Wartung von CMX sind auch unter der Benutzerrolle *cmxadm* ausführbar. Jeder Systembenutzer, der diese Rolle einnehmen darf, kann CMX administrieren. Im folgenden sind die Begriffe Systemverwalter und Administration unter der Benutzerrolle cmxadm als synonym aufzufassen.

2.2 Netzzugänge und Transportprofile

CMX unterstützt den Anschluss von Solaris-Systemen an alle marktrelevanten Netze. Das bedeutet Integration in die wichtigsten Kommunikations-Architekturen TCP/IP, OSI, TRANSDATA NEA und SNA und alle gängigen physikalischen Netze. Die folgende Abbildung zeigt diese Anschlussmöglichkeiten.

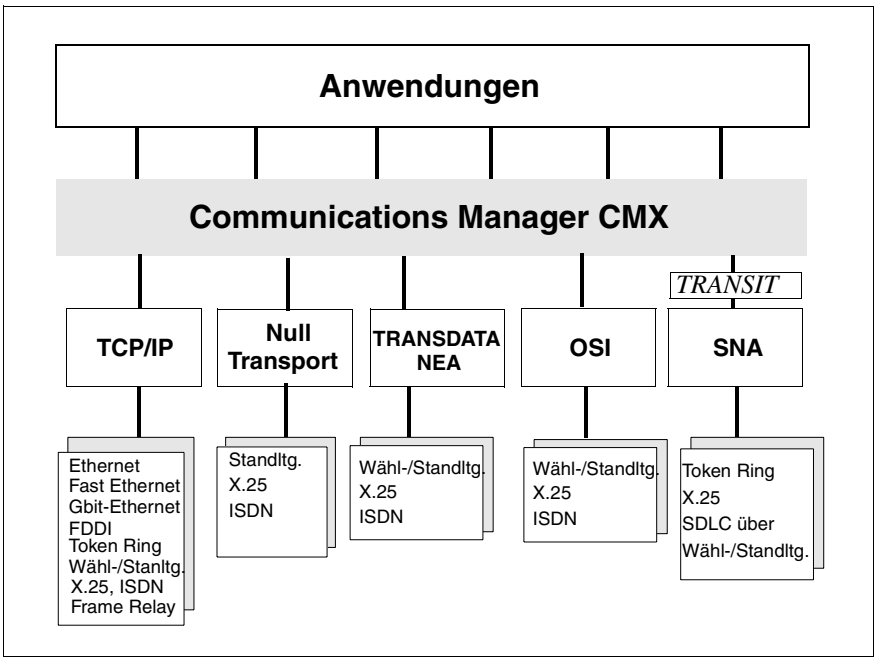


Bild 2: Netzzugänge und Transportprofile

Damit Rechner in lokalen Netzen oder über Weitverkehrsnetze hinweg kommunizieren können, müssen sie über eindeutige Adressen erreichbar sein. Die Adressierung ist jedoch von den verwendeten Protokollen abhängig und daher in den einzelnen Netzarchitekturen unterschiedlich. CMX unterstützt die Verwendung von TCP/IP-, ISO- und TRANSDATA NEA-Netzadressen und ermöglicht dadurch netzunabhängige Kommunikation. Im Folgenden werden Netzarchitekturen sowie die Komponenten und Merkmale der wichtigsten Adressen vorgestellt.

2.2.1 TCP/IP-Architektur

TCP/IP kann über lokale und Weitverkehrsnetze aller Art betrieben werden.

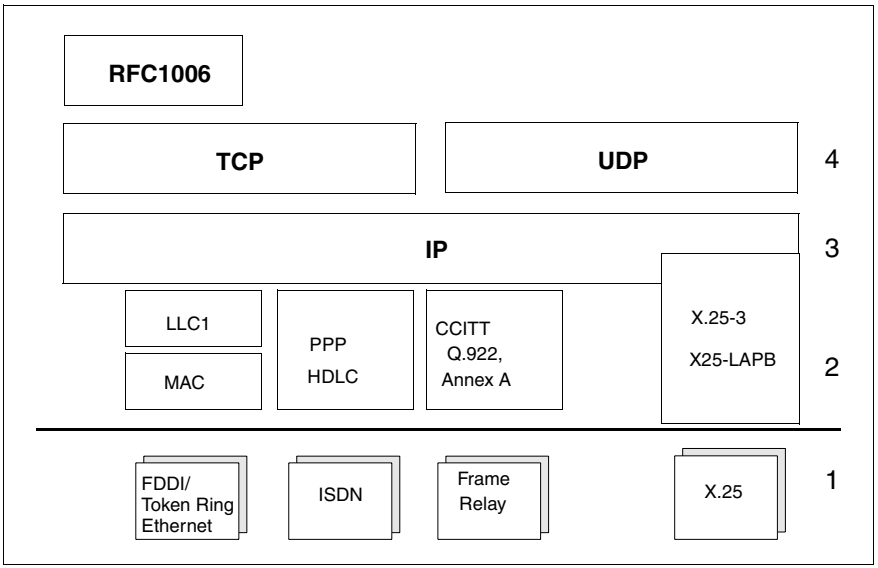


Bild 3: TCP/IP-Protokollstack

Schicht 2 des TCP/IP-Transportprofils für LANs ist in die MAC-Teilschicht und die LLC-Teilschicht gegliedert. In der MAC-Teilschicht (Medium Access Control) wird der Zugang auf das jeweilige physische LAN geregelt; in der LLC-Teilschicht (Logical Link Control) werden die Nachrichten ausgetauscht.

IP verwendet in Token Ring und FDDI-Netzen das verbindungslose Protokoll LLC1 mit einer IP-spezifischen Erweiterung namens SNAP. In Ethernet-LANs wird, abhängig von der MAC-Variante, kein LLC-Protokoll oder ebenfalls LLC1 verwendet.

Für IP über ISDN wird HDLC (mit oder ohne T.70-3) oder das Point-to-Point-Protokoll (PPP) verwendet. Oberhalb eines Frame-Relay-Subnetzes gilt das CCITT-Protokoll Q.922, Annex A, während über paketvermittelnde Netze X25-LAPB eingesetzt wird.

TCP/IP-Adressen

Eine TCP- oder UDP-Anwendung im TCP/IP-Netz (= Internet) wird über ihre Portnummer und eine der IP-Adressen des Rechners adressiert.

IP-Adresse

Es ist zwischen IPv4- und IPv6-Adressen zu unterscheiden:

– Eine IPv4-Adresse ist 32 Bit lang. Als Text, z. B. bei der Eingabe oder Ausgabe am Bildschirm, wird sie dargestellt, indem jeweils 8 Bit als Dezimalzahl wiedergegeben und durch Punkt voneinander getrennt werden (z. B. 139.22.112.88).

– Eine IPv6-Adresse ist 128 Bit lang. Hier werden in der Textdarstellung jeweils 16 Bit zusammengefasst, als Sedezimalzahl wiedergegeben und durch Doppelpunkt getrennt.

  Eine oder mehrere benachbarte Sedezimalzahlen mit Wert 0 können einmal pro Adresse als 2 unmittelbar aufeinanderfolgende Doppelpunkte abgekürzt werden (z. B. FE80::280:17FF:FE28:7B08).

  Bei speziellen IPv6-Adressen, die aus IPv4-Adressen abgeleitet sind, kann der IPv4-Adressanteil auch in der IPv4-Schreibweise dargestellt werden (z.B. ::FFFF:139.22.112.88).

Portnummer

Eine Portnummer ist 16 Bit lang. Sie wird im Text normalerweise als Dezimalzahl dargestellt.

Beachten Sie bei der Vergabe von Portnummern für Ihre eigenen Anwendungen, dass für Standard-Anwendungen bestimmte Portnummern reserviert sind. So adressiert beispielsweise die Anwendung TELNET über die Portnummer 23, die Anwendung FTP über Portnummer 21.

Sie sollten für den eigenen Bedarf grundsätzlich nur Portnummern größer als 1024 konfigurieren. Die weltweit reservierten Portnummern werden regelmäßig als RFC (Request for Comment) von der International Electrotechnical Commission (IEC) veröffentlicht.

RFC1006

Der Internet-Standard RFC1006 definiert, wie durch eine weitere Protokollschicht oberhalb von TCP ein verbindungsorientierter OSI-Transportdienst realisiert werden kann. Diesem Dienst ist die TCP-Portnummer 102 fest zugeord-

net. OSI-Anwendungen, die diesen Dienst verwenden, adressieren sich zusätzlich zur TCP/IP-Adresse über einen sogenannten T-Selektor (siehe Abschnitt „OSI-Architektur“ auf Seite 17).

In Partner-Rechnern kommt jedoch auch eine RFC1006-Implementierung vor, bei der die Anwendungen, die den RFC1006-Dienst nutzen, nicht mit dem Tripel IP-Adresse – TCP-Portnummer 102 – T-Selektor adressiert werden. Statt dessen werden sie entweder **ohne** T-Selektor, nur mit IP-Adresse und TCP-Portnummer adressiert, oder **mit** T-Selektor und TCP-Portnummer. Die TCP-Portnummer muss in diesen beiden Fällen von 102 verschieden sein. Eine solche Implementierung besitzen insbesondere CMX 3.0, CMX 4.0 und PCMX. Die aktuelle CMX-Version unterstützt die Kommunikation mit diesen Partnern.

2.2.2 TRANSDATA NEA-Architektur

Die Architektur des TRANSDATA NEA-Transportsystems ist für Telefonie-, X.21- und X.25-Weitverkehrsnetze (WAN) sowie ISDN konzipiert. Auf der Transportschicht wird das NEATE-Protokoll eingesetzt, das im Wesentlichen die Funktionen der Klassen 2 und 3 des ISO-Protokolls 8073 enthält.

Auf der Netzschicht wird das verbindungslose NEAN-Protokoll verwendet. Auf der Sicherungsschicht wird für leitungsvermittelnde Netze das HDLC-Protokoll und in paketvermittelnden Netzen das X.25-LAPB-Protokoll eingesetzt. Für NEA über ISDN können die gleichen Protokolle verwendet werden.

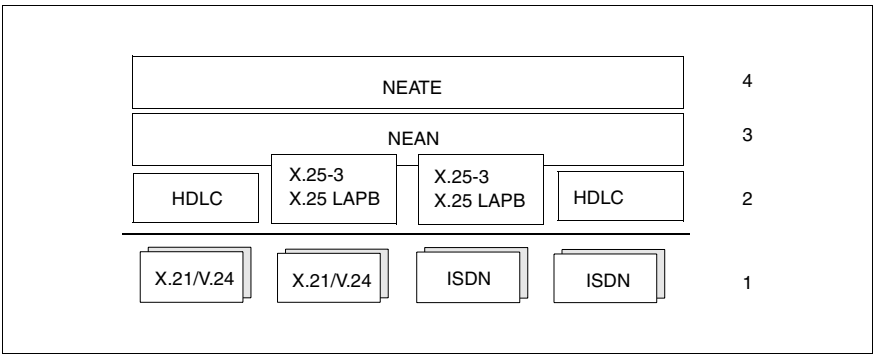


Bild 4: Architektur des TRANSDATA NEA-Transportsystems

TRANSDATA NEA-Adressen

TRANSDATA NEA-Netze können in maximal 256 Teilnetze (Regionen) gegliedert werden. Innerhalb einer Region können bis zu 256 Rechner adressiert werden. An einem Rechner wiederum lassen sich 2046 Stationen betreiben. Als Stationen werden im TRANSDATA NEA-Konzept sowohl Datenstationen als auch Anwendungen bezeichnet. Datenstationen können Drucker oder Terminals sein.

Die Hierarchie von Region und Rechner in TRANSDATA NEA-Netzen spiegelt sich bei der Adressierung wider.

Netzadresse

Mit der Netzadresse werden Rechner im TRANSDATA NEA-Netz adressiert. Sie setzt sich aus der Prozessornummer und der Regionsnummer zusammen (z. B. 1/18).

Die Prozessornummer identifiziert einen Rechner innerhalb einer Region eindeutig. Jede Region erhält eine im gesamten TRANSDATA NEA-Netz eindeutige Regionsnummer.

Prozessornummer und Regionsnummer liegen jeweils im Wertebereich zwischen 0 und 255. Innerhalb eines TRANSDATA NEA-Netzes lassen sich auf diese Weise 65536 Rechner eindeutig adressieren.

2.2.3 OSI-Architektur

Für die einzelnen WAN-Typen wurden von verschiedenen nationalen und internationalen Organisationen eine Reihe von OSI-Protokollprofilen festgelegt. Mit den CCP-Profilen werden die in der Praxis bedeutsamen angeboten.

Im WAN (und ISDN) wird in der Regel auf der Schicht 4 das verbindungsorientierte Transportprotokoll IS8073, Klasse 0 bzw. 2 verwendet, das auf dem ebenfalls verbindungsorientierten Netzdienst X.25 oder auf T.70-3 aufsetzt.

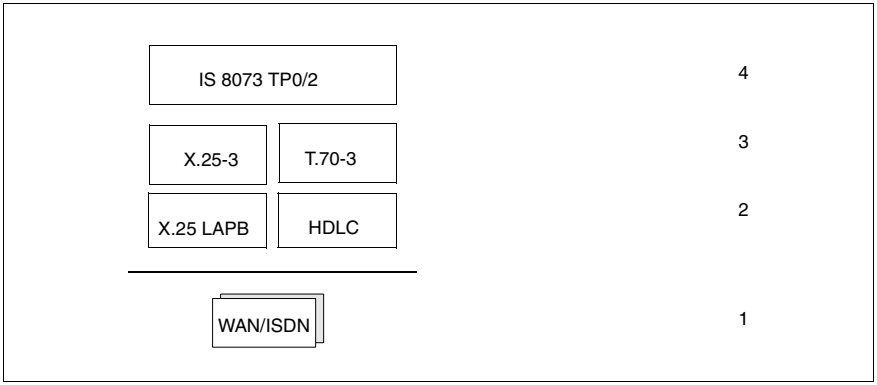


Bild 5: OSI-Protokollstacks

OSI-Transportadressen

OSI-Transportadressen bestehen aus der OSI-NSAP-Adresse und einem Transport-Selektor.

OSI-NSAP-Adresse

OSI-Adressen bestehen aus den Komponenten AFI (Authority and Format Identifier), IDI (Initial Domain Identifier) und DSP (Domain Specific Part). Nähere Informationen siehe Abschnitt „Adress-Komponenten und ihre Formate“ auf Seite 86). In vielen Fällen, insbesondere bei WAN-Anschlüssen, werden jedoch die Adressen des unterliegenden Subnetzes (z. B. X.25-Adressen) verwendet. Die OSI-NSAP-Adressen werden in diesen Fällen nicht benötigt.

Transport-Selektor

Mit dem Transport-Selektor (T-Selektor) meldet sich eine Anwendung beim OSI-Transportdienst an. Bei ankommendem Verbindungsaufbauwunsch identifiziert das Transportsystem anhand des T-Selektors die gerufene Anwendung.

2.3 Schnittstellen und Anwendungen

Im vorliegenden Handbuch werden Anwendungen, die die Dienste eines Transportsystems nutzen, unabhängig von der Rechnerplattform, auf der sie ablaufen, generell als **TS-Anwendung** bezeichnet. Die Dienste von CMX können von lokalen TS-Anwendungen über unterschiedliche Programmierschnittstellen genutzt werden. Je nachdem, über welche Programmierschnittstelle die CMX-Anwendung auf die Transportsysteme zugreift, wird zwischen ICMX-, XTI-, TLI- und NLI-Anwendung unterschieden.

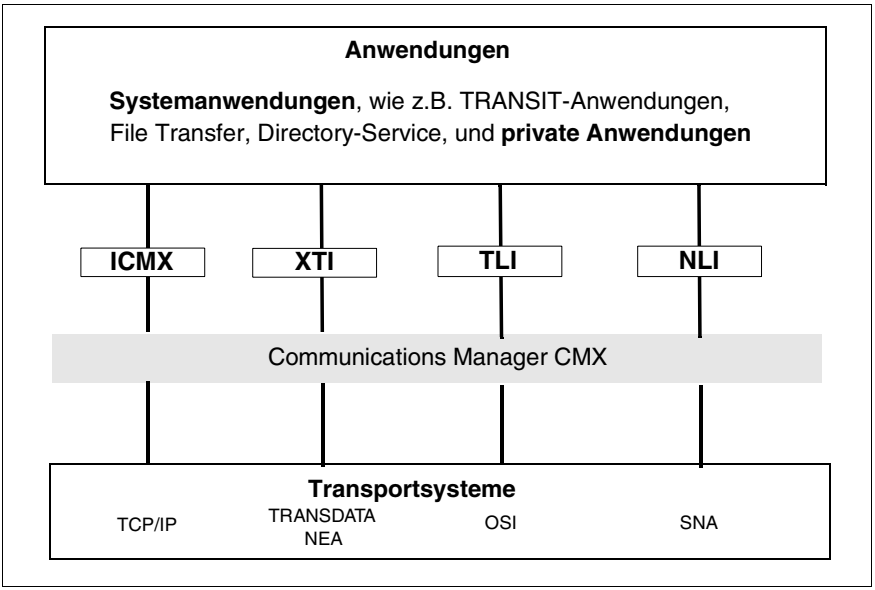


Bild 6: Programmier-Schnittstellen von CMX und Anwendungen

Die im Produkt CMX enthaltene ICMX-Schnittstelle eröffnet den Zugang zu allen Transportsystemen, die einen Transportdienst nach IS 8072 anbieten. Dazu gehören sowohl die reinen OSI-Transportsysteme über WAN, die verbreitete Variante RFC1006, die den OSI-Dienst über TCP/IP anbietet, und TRANSDATA NEA.

Die ICMX-Schnittstelle wird für alle gängigen UNIX-Systeme und im BS2000/OSD angeboten. Darüber hinaus gibt es Implementierungen für Windows. Typische Beispiele für ICMX-Anwendungen sind z. B. *open*UTM, TRANSIT oder Anwendungen, die für den Endbenutzer konzipiert sind (z. B. EMDS, *open*FT oder MAIL.X).

Die Schnittstelle X/Open Transport Interface (XTI) eröffnet den Zugang sowohl zu TCP und UDP als auch zu OSI-Transportdiensten. Die Schnittstelle wird ebenfalls in allen gängigen UNIX-Systemen angeboten. Anwendungen aus unabhängigen Software-Häusern nutzen häufig diese Schnittstelle.

Transport Layer Interface (TLI) ist eine weitere, nicht-proprietäre Schnittstelle zur Transportschicht in UNIX-Systemen. TLI -Anwendungen können über die Transportsysteme der CCPs ablaufen, sofern sie Transportdienste entsprechend IS 8072 nutzen (siehe Kapitel „Ablauf von TLI-Anwendungen“ auf Seite 363).

CMX unterstützt die Kommunikationsfunktionen der Schnittstelle Network Layer Interface (NLI), die in SUN-Systemen den direkten Zugang zu X.25-Netzen ermöglicht. Die Nutzung dieser Schnittstelle ist nur nach Sonderfreigabe erlaubt.

Darüber hinaus ist es üblich, Anwendungen durch die genutzten Transportsysteme zu kennzeichnen. Anwendungen, die das OSI-Transportsystem nutzen, werden beispielsweise häufig als OSI-Anwendungen bezeichnet.

2.4 Solaris-Kommunikationsprodukte

Der vorliegende Abschnitt zählt die durch CMX und die Kommunikationsprodukte realisierten Transportprofile auf und nennt die Funktionen des *Communication Services*.

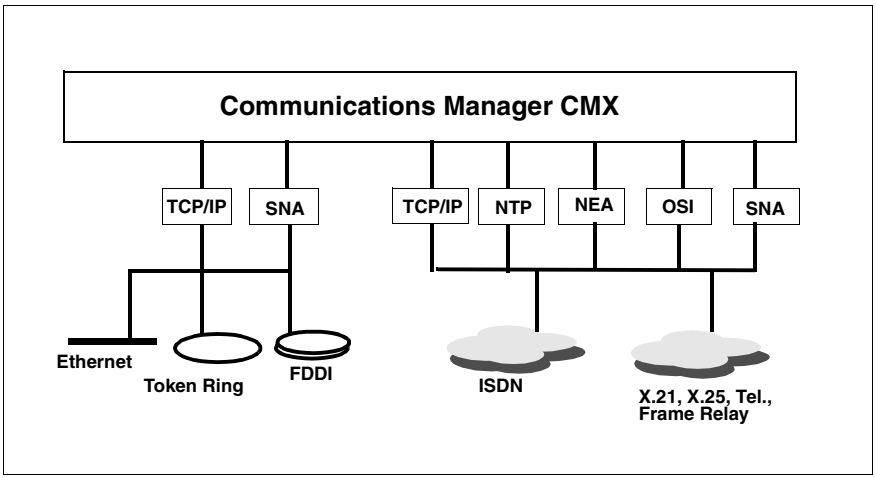


Bild 7: Netzzugänge über CMX

Die Solaris-Kommunikationsprodukte bieten im Einzelnen folgende Funktionen und Leistungen:

– Für alle wichtigen Datennetze werden die entsprechenden Transportprofile angeboten, und zwar in Form der CCP-Familie *(Communication Control Programs)*. Ein CCP kann eines oder mehrere Transportprofile bereitstellen. Wegen der Implementierung als CCPs werden Transportprofile im weiteren Text als *CCP-Profile* bezeichnet.

– Für TS-Anwendungen werden mehrere Programmschnittstellen angeboten: Die Schnittstellen ICMX(L) und ICMX(NEA), die Bestandteil von CMX sind (siehe Handbuch „CMX, Anwendungen programmieren" [1], und die herstellerunabhängige Schnittstelle XTI).

– Alle Komponenten, die zur lokalen Konfigurierung und Administration benötigt werden, sind ebenfalls im Produkt CMX enthalten (siehe Kapitel „Konfigurieren und Administrieren im Menü" auf Seite 59).

2.4.1 Transportprofile

Im Folgenden erhalten Sie eine Übersicht der Transportprofile, die von den Solaris-Kommunikationsprodukten zur Verfügung gestellt werden:

- Der TCP/IP-Transportdienst, der im Solaris-Basissystem enthalten ist.
- Die Implementierung des ISO-Transportdienstes gemäß IS 8072 Klasse 0 über TCP/IP durch das Konvergenzprotokoll RFC1006. Dieses Transportprofil ist Bestandteil des Produktes CMX und in Kapitel „Verbindungen über RFC1006 konfigurieren“ auf Seite 207 ausführlich beschrieben.
- Direkter Zugang zu X.25-Paketvermittlungsnetzen.
- TCP/IP-Transportdienste über X.25 oder Frame Relay.
- Verbindungsorientierter OSI-Transportdienst mit RFC1006 über TCP und X.25 oder Frame Relay.
- TCP/IP-Transportdienste über ISDN.
- Verbindungsorientierter OSI-Transportdienst mit RFC1006 über TCP und ISDN.
- Der SNA-Transportdienst über ISDN.
- Der OSI-Transportdienst gemäß IS 8072 Klassen 0 und 2 für OSI-Verbindungen über Weitverkehrsnetze.
- Der OSI-Transportdienst gemäß IS 8072 Klassen 0 und 2 für OSI-Verbindungen über ISDN.
- Der SNA-Transportdienst über WAN.
- Der TRANSDATA NEA-Transportdienst über ISDN, X.25, Stand- oder Wählleitungen.

Für den Einsatz sämtlicher Transportprofile ist das Produkt CMX notwendig. In der Solaris-Kommunikation wird die Implementierung eines Transportprofils als CCP-Profil bezeichnet. Die CCP-Produkte werden ausführlich in den entsprechenden Handbüchern beschrieben.

2.4.2 Routing Service

Mit dem **Routing Service** werden die Funktionen der Solaris-Kommunikation erweitert. Sie können Ihr Solaris-System zusätzlich als Router nutzen.

Durch den Einsatz von CS-ROUTE können LAN-Inseln mit TCP/IP-Protokoll über WAN/ISDN verbunden werden. CS-ROUTE kann sich mit anderen Routern mit Hilfe des Routing-Protokolls OSPF abstimmen. CS-ROUTE arbeitet über Frame Relay, X.25-Netze oder ISDN-Verbindungen und unterstützt auch das Point-to-Point-Protokoll (PPP). Der Routing Service ist auch erforderlich für die Kommunikation lokaler TCP/IP-Anwendungen über WAN/ISDN.

2.5 Architektur der CCP-Profile

Der vorliegende Abschnitt beschreibt, wie die CCP-Profile implementiert sind, die mit CMX und den CCP-Produkten angeboten werden.

Bei der Klassifizierung der CCP-Profile wird grundsätzlich zwischen **Local Area Networks (LANs)** und **Wide Area Networks (WANs)** unterschieden. Ein LAN ist ein Netz relativ geringer, von der verwendeten Technologie abhängiger Reichweite, das aber eine hohe Übertragungsgeschwindigkeit ermöglicht und damit den schnellen Austausch großer Datenmengen erlaubt. Die Reichweite ist oft auf ein Stockwerk, ein Gebäude oder einen Gebäudekomplex beschränkt. Ein LAN wird privat betrieben und verwaltet.

Ein WAN hingegen bietet zwar in der Regel eine im Vergleich zu einem LAN geringere Übertragungsgeschwindigkeit, dafür aber die Möglichkeit weltumspannender Kommunikation. Beispiele sind die X.25-Netze oder die **Integrated Services Digital Networks (ISDN)** von Netzbetreibern wie der Deutschen Telekom, British Telecom, France Télécom oder AT&T.

Ein CCP-Profil besteht aus zwei Komponenten:

– einem **Transport Service Provider (TSP)**, der die Dienste der Transportschicht und eines Teils der Netzschicht (Schicht 3c im OSI-Referenzmodell) anbietet.

– einem **Subnetz-Profil**, das die Dienste zur Unterstützung des Subnetz-Anschlusses (Schichten 1, 2 und 3a im OSI-Referenzmodell) realisiert.

Ein TSP bezeichnet die Komponenten von CCP-Profilen, die zur Steuerung von Subnetz-Profilen erforderlich sind. Zu diesen Komponenten gehört ein bestimmtes Transportprotokoll und der jeweilige Transportdienst für die unterschiedlichen Netzarchitekturen. Folgende TSPs werden angeboten:

– Der TSP RFC1006 für den OSI-Transportdienst über TCP/IP

  Der TSP RFC1006 nutzt TCP/IP als OSI-äquivalenten Netzdienst. Um diesen TSP nutzen zu können, müssen Sie Angaben zu Adressen machen, die im Transport Name Service (TNS, siehe Abschnitt „Adressierung von Transportsystem-Anwendungen im TNS“ auf Seite 37) verwaltet werden. Bei Kommunikation mit CS-ROUTE müssen Sie zusätzliche Angaben im Forwarding Support Service (FSS, siehe Abschnitt „Adressierung von Partnersystemen im FSS“ auf Seite 45) machen.

– Null Transport für ISDN- und X.25-Kommunikation

  Der TSP Null Transport (NTP) bietet den direkten Zugang auf die Dienste des Subnetzes.

– TRANSDATA NEA-TSP für TRANSDATA-Architektur

  Der TRANSDATA NEA-TSP stellt den Transportdienst im TRANSDATA-Netz zur Verfügung. Um diesen Dienst nutzen zu können, müssen Sie Angaben zu Adressen (und ggf. partnerspezifischen Dienstmerkmalen) machen, die im Forwarding Support Service (FSS, siehe Abschnitt „Adressierung von Partnersystemen im FSS“ auf Seite 45) und im Transport Name Service (TNS, siehe Abschnitt „Adressierung von Transportsystem-Anwendungen im TNS“ auf Seite 37) verwaltet werden.

– OSI TP0/2 für OSI-Kommunikation im ISDN und anderen Weitverkehrsnetzen

  OSI TP0/2 ist der TSP für ein OSI-Umfeld mit dem OSI-Transportdienst der Klassen 0 und 2. Um diesen Dienst nutzen zu können, müssen Sie Angaben zu Adressen (und ggf. partnerspezifischen Dienstmerkmalen) machen, die im Forwarding Support Service (FSS, siehe Abschnitt „Adressierung von Partnersystemen im FSS“ auf Seite 45) und im Transport Name Service (TNS, siehe Abschnitt „Adressierung von Transportsystem-Anwendungen im TNS“ auf Seite 37) verwaltet werden.

CMX definiert für jeden Transport Service Provider (TSP) einen TSP Access Point. Ein TSP Access Point definiert den Zugriffspunkt des CMX-Automaten (zentrale Komponente von CMX) zum Transport Service Provider.

An diesen Zugangspunkten bietet CMX zu den Kommunikationskomponenten eine einheitliche Sicht des Transportsystems.

Damit TS-Anwendungen über einen TSP kommunizieren können, muss der TSP betriebsbereit sein und über einen **TSP Access Point** dem CMX-Automaten Zugang gewähren. Der TSP Access Point ist insbesondere für Wartungs- und Diagnosefunktionen von Bedeutung (siehe hierzu Kapitel „Administration und Wartung“ auf Seite 227).

Ein Subnetz-Profil bezeichnet die Loadware, die auf einen Communications Controller (CC) geladen wird. Das Subnetz-Profil steuert die CCs für das jeweilige Subnetz. Die Konfigurationsdatei (KD) für das Subnetz-Profil definiert die Merkmale Ihres lokalen Subnetz-Anschlusses, z. B. Ihre eigene ISDN-Rufnummer, die beim Verbindungsaufbau einzustellenden Protokolle sowie die X.25-Merkmale des Übergangs in das X.25-Netz oder zu einem X.25-Partner am ISDN.

Beim Systemstart wird das dem CC zugewiesene Subnetz-Profil einschließlich der zugewiesenen Konfigurationsdatei auf den CC geladen. Mit einer einzigen Konfigurationsdatei können Sie das Subnetz-Profil so konfigurieren, dass Ihr System gleichzeitig unterschiedliche Transportdienste über ein und denselben CC an einem Subnetz-Anschluss nutzen kann.

Je nach Typ des CCPs sind die Komponenten TSP und Subnetz-Profil unterschiedlich ausgeführt. Diese verschiedenen Implementierungen sind in der folgenden Grafik und dem anschließenden Text dargestellt.

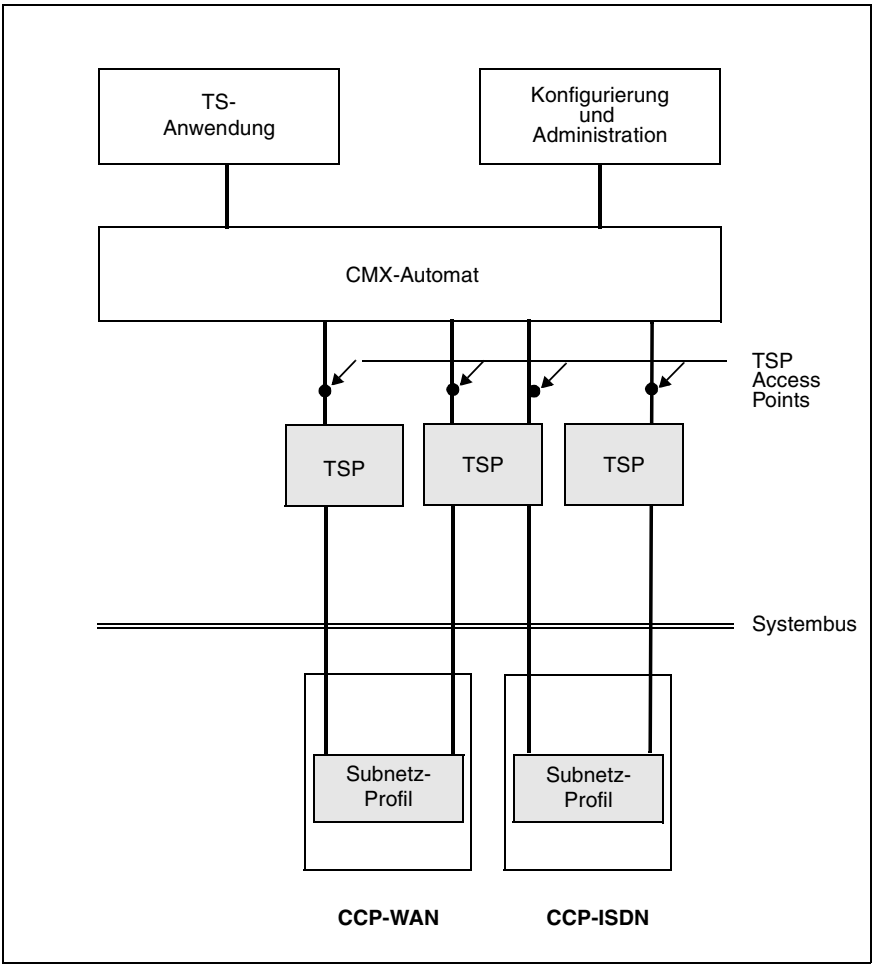


Bild 8: Implementierung der CCP-Profile

2.5.1 LAN-Profile

Der TSP RFC1006 und das TCP/IP-(Sub-)Netz sind als Kernel-Komponenten von Solaris realisiert. Der Netzzugang erfolgt über intelligente **Communication Controller (CCs)**, die durch Solaris verwaltet werden.

Diese Lösung bietet den Vorteil, dass Sie ohne zusätzliche CC-Hardware über Ihren vom Solaris-Basissystem angebotenen Anschluss gleichzeitig zwei Profile ablaufen lassen können:

– RFC1006-Konvergenzprotokoll über TCP/IP

– TCP/IP ohne Konvergenzprotokoll

Der TSP RFC1006 wird über CMX konfiguriert und administriert.

2.5.2 WAN-Profile

Die TSPs Null Transport, OSI TP0/2 und TRANSDATA NEA sind ebenfalls Kernel-Komponenten von Solaris.

Subnetz-Profile laufen auf programmierbaren CCs ab. Sowohl TSPs als auch Subnetz-Profile werden über CMX konfiguriert und administriert. Da normalerweise keine paarweise Zuordnung zwischen TSPs und Subnetz-Profilen vorliegen muss (z. B. können der TRANSDATA NEA- und der OSI TP0/2 auf dasselbe X.25-Subnetzprofil aufsetzen), werden TSPs und Subnetz-Profile unabhängig voneinander konfiguriert und administriert.

Durch die Möglichkeit, Subnetzprofile durch verschiedene TSPs zu bedienen, können Leitungen und CCs optimal ausgenutzt werden.

Das Subnetzprofil der WAN-CCPs unterstützt den Anschluss an X.25-Netze, Frame-Relay-Netze sowie analoge und digitale Wähl- und Standleitungen.

2.5.3 ISDN-Profile

Die Software-Struktur entspricht den WAN-Profilen. Auch hier können verschiedene TSPs gleichzeitig dasselbe Subnetzprofil nutzen.

2.5.4 SNA-Profile

Die SNA-Protokolle sind zum Teil in den TRANSIT-Protokollen enthalten. Für einen kompletten SNA-Anschluss ist daher neben CMX und dem jeweiligen CCP-Produkt auch das passende TRANSIT-Produkt erforderlich.

2.6 Einsatzfälle

In diesem Abschnitt finden Sie Beispiele für heterogene Netzarchitekturen mit typischen Einsatzfällen.

TCP/IP über ISDN

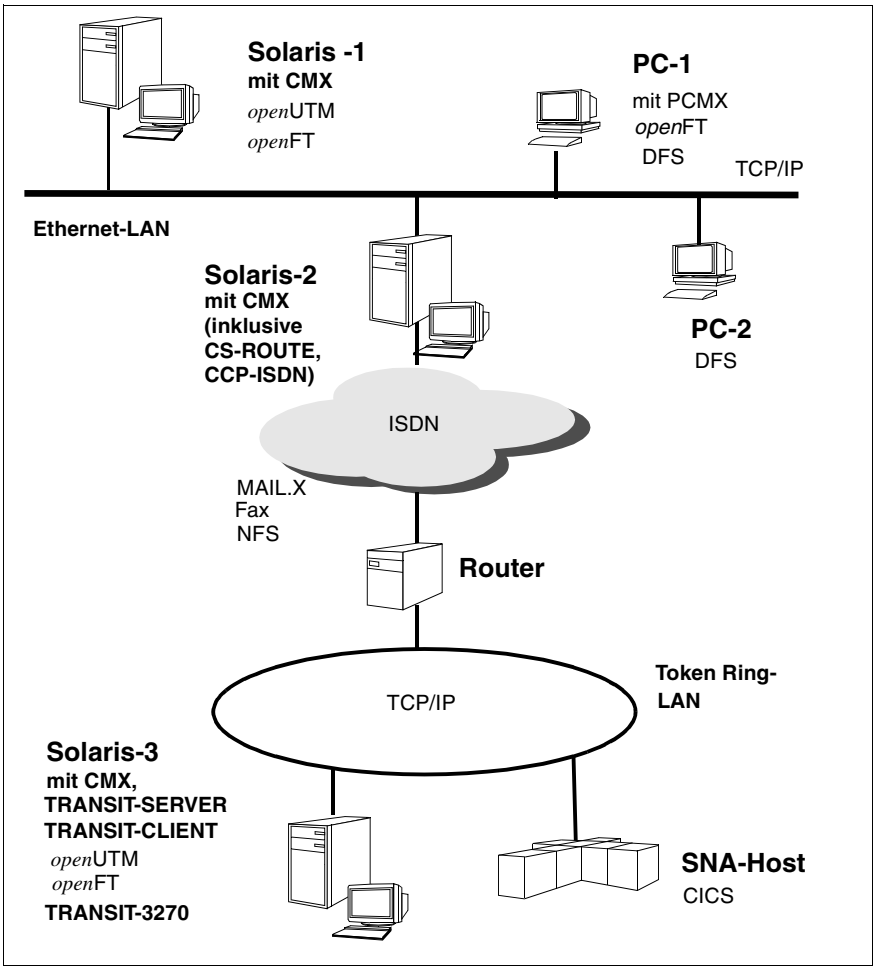


Bild 9: TCP/IP-Anwendungen über ISDN

In der dargestellten Konfiguration sind zwei lokale Netze (Ethernet und Token Ring), die jeweils TCP/IP-Protokoll fahren, über ISDN gekoppelt. Die LAN-WAN-Verbindung wird dabei wahlweise durch einen dedizierten Router oder durch ein Solaris-System mit CS-ROUTE-Software ermöglicht. Zwischen allen Anwendungen werden Daten über das IP-Protokoll ausgetauscht, so dass das ISDN-Netz für die Anwendungen unsichtbar bleibt.

Das Solaris-System 2 kann als Server für PC-Clients eingesetzt werden. Das Server-System stellt z. B. Network Services über das LAN und, mit Hilfe von CS-ROUTE, auch über ISDN zur Verfügung. Auf den PCs ist dafür die Software DFS (Distributed File Service) erforderlich. Auch MAIL-Anwendungen können Daten austauschen (hier zwischen MAIL.X auf Solaris-1 und MAIL.D auf PC-1), ohne dass über CMX hinaus Kommunikationssoftware erforderlich ist.

Am Token-Ring-LAN sind SNA-Host und Solaris-Systeme angeschlossen. Sie kommunizieren ebenfalls über TCP/IP untereinander. Mit der Software TRANSIT-SERVER und TRANSIT-CLIENT z. B. betreiben Sie Ihr Solaris-System als SNA-Terminal für Dialog- und Transaktionsbetrieb (LU6.2-Funktionalität).

*open*UTM auf dem System Solaris-3 ermöglicht verteilte Transaktionsverarbeitung (z. B. beim Zugriff auf CICS im dargestellten SNA-System oder in Kommunikation mit einer *open*UTM-Anwendung auf Solaris-1 am Ethernet-LAN).

Gleichfalls können über *open*FT File-Transfer-Anwendungen auf Solaris-1 und Solaris-3 miteinander kommunizieren. Hierfür benötigen Sie auf Solaris-1 und Solaris-3 CMX sowie auf Solaris-2 CMX und CCP-ISDN.

Eine ähnliche Konfiguration ist so wie über ISDN auch über andere Weitverkehrsnetze (X.25, X.21) denkbar. In diesem Fall muss statt eines ISDN-Produktes die Software CCP-WAN installiert werden.

Kopplung von Solaris-Systemen mit SNA-Hosts

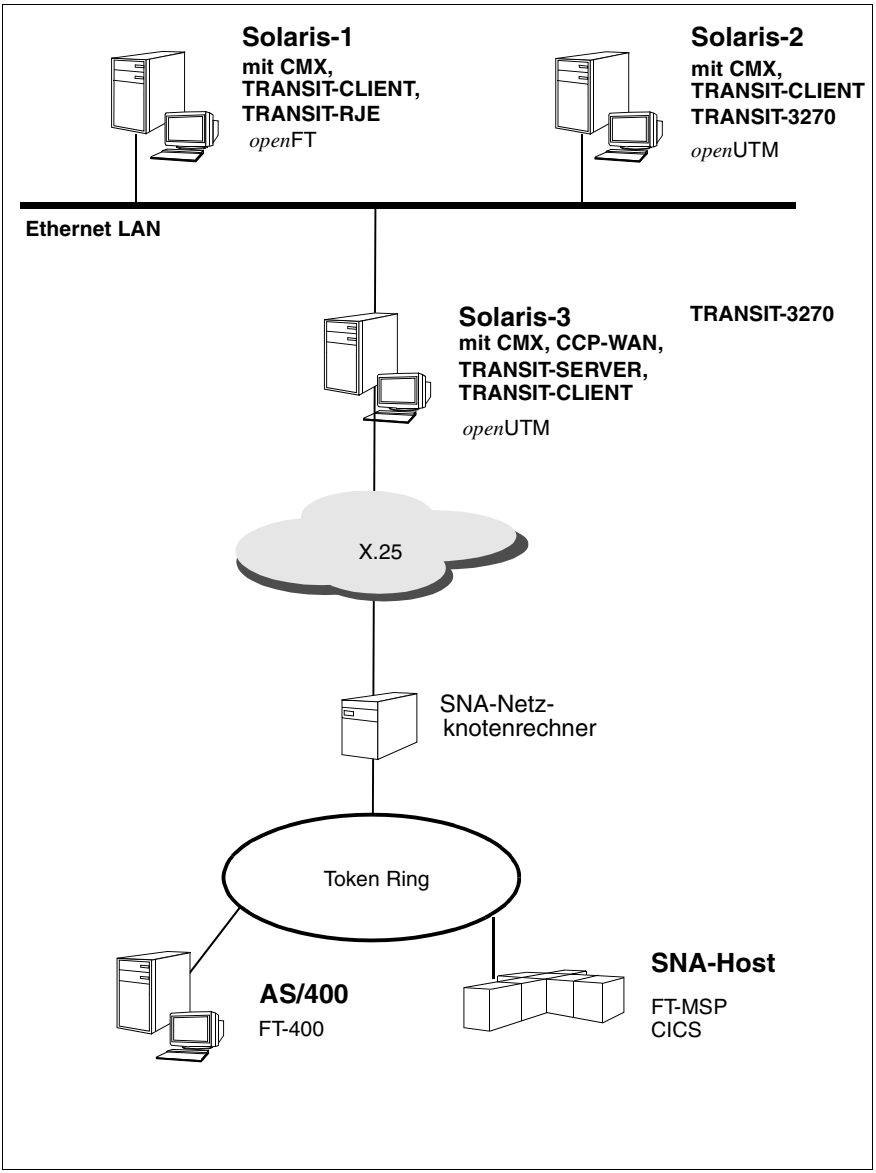


Bild 10: Kopplung von Solaris-Systemen mit der SNA-Welt

In der abgebildeten Konfiguration ist ein lokales SNA-Netz (Token Ring) über X.25-WAN mit einem Ethernet-LAN verbunden. Die Solaris-Systeme befinden sich am Ethernet-LAN.

Die Solaris-Systeme 1, 2 und 3, die in ein Ethernet-LAN integriert sind, können über unterschiedliche SNA-Protokolle (LU1, LU2, LU3, LU6.1, LU6.2) mit SNA-Systemen kommunizieren, die über X.25-Netz erreichbar sind. Das System Solaris-3 hat direkten Zugang zum X.25-Netz. Auf diesem System muss das Produkt TRANSIT-SERVER installiert sein; auf den Solaris-Systemen 1 und 2 muss TRANSIT-CLIENT (oder alternativ TRANSIT-CPIC) installiert sein.

Anwendungen wie TRANSIT-RJE (Remote Job Entry) oder TRANSIT-3270 (Nachbildung einer 3270-Datenstation) können so auf den Solaris-Systemen 1 bzw. 2 laufen.

Für File-Transfer-Anwendungen benötigen Sie auf Ihrem Solaris-System das Produkt *open*FT (hier auf Solaris-1); auf dem SNA-Host muss FT-MSP installiert sein. Für den AS/400-Rechner benötigen Sie für File-Transfer-Anwendungen das Produkt FT-400.

Über *open*UTM auf dem Solaris-System (hier: Solaris-2) kommunizieren Sie mit CICS-Anwendungen am SNA-Host. TRANSIT-CLIENT bildet die Brücke zwischen TRANSIT-SERVER einerseits und *open*FT bzw. *open*UTM andererseits.

Die gleichen Kommunikationsbeziehungen sind möglich, wenn das SNA-System nicht am Token-Ring-LAN hängt, sondern direkten Zugang zum X.25-Netz besitzt.

Kopplung von LANs über X.25-Netz

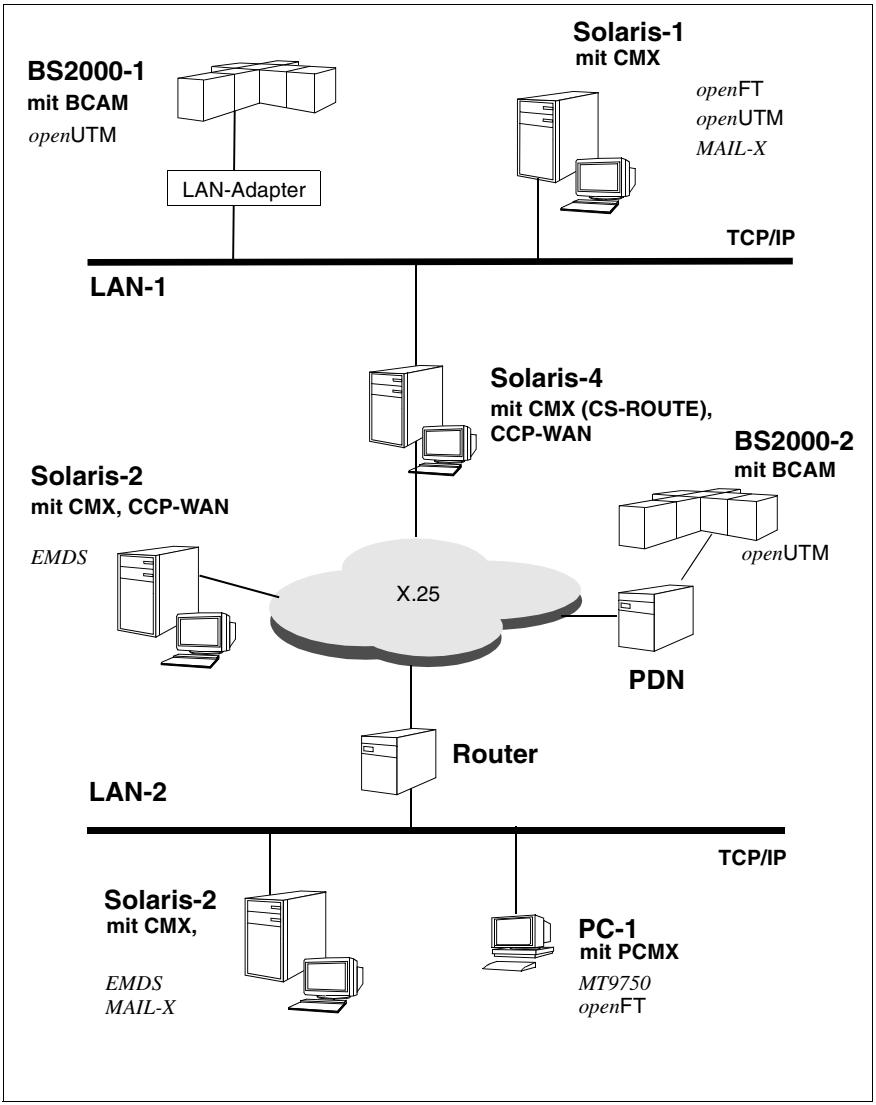


Bild 11: Kopplung von LANs über X.25

3 Die Benutzerrolle cmxadm

In Solaris ist eine Autorisierungsinfrastruktur namens RBAC (role based access control) eingeführt worden. Diese realisiert eine funktionelle Zugriffskontrolle, die gewährleistet, dass privilegierte Systemoperationen nur von hierzu berechtigten Systembenutzern ausgeführt werden. Sie ermöglicht es auch, die Ausführung solcher Operationen von der Kennung *root* des Systemverwalters zu entkoppeln und anderen Systembenutzern zuzuordnen. Damit kann auch die CMX-Administration unter einer eigenen Kennung durchgeführt werden.

3.1 Zentrale Begriffe

Zum Verständnis der nachfolgenden Ausführungen werden kurz folgende Begriffe erläutert. Detailliertere Informationen über RBAC sind in den Systemverwalterhandbüchern von Solaris 8/9 enthalten (siehe „Solaris 8/9: System Administration Guide, Volume 2" [7]).

Rolle

Eine Rolle definiert einen speziellen Benutzertyp im System, dem eine bestimmte Funktionsmenge aus dem Kommandovorrat des Systems zugeordnet ist. Bei der Ausführung einer Funktion wird von dem ausführenden Programm/Skript überprüft, ob der die Funktion initiierende Benutzer hierzu auch autorisiert ist. Eine oder mehrere Rollen können einem Systembenutzer bei dessen Einrichtung (per *useradd*) oder Änderung (per *usermod*) zugeordnet werden. Der Benutzer informiert sich mit dem Kommando *roles* über die ihm zugeordneten Rollen. Er wechselt systemlokal in eine Rolle mit dem Kommando *su*.

Autorisierung

Die Autorisierung definiert die Erlaubnis, gewisse Aktionen im System ausführen zu dürfen. Die Erlaubnis wird repräsentiert durch einen Text-String. Das die Aktion ausführende Programm/Skript überprüft, ob dem die Aktion initiierenden Benutzer die Erlaubnis zugeordnet ist. Mit dem Kommando *auths* zeigen Sie an, welche Autorisierungen einem Benutzer aktuell zugewiesen sind.

Profil

Ein Rechte-Profil beschreibt eine Menge an Funktionen, die ein Benutzer im System ausführen darf. Diese Funktionen werden implizit festgelegt durch eine Menge von Autorisierungen und durch eine Menge von zusätzlichen Kommandos, die mit bestimmten Sicherheitsattributen aus-

geführt werden dürfen (z. B. die Ausführung eines privilegierten Kommandos mit der realen/effektiven UID *root*). Mit dem Kommando *profiles* listen Sie auf, welche Profile einem Benutzer aktuell zugewiesen sind.

3.2 CMX-Installation: Erweiterung der RBAC-Datenstrukturen

Bei der Installation von CMX (präziser: mit dem Paket *SMAWcmx*) werden folgende RBAC-spezifische Objekte auf einem System eingeführt:

1. Die Autorisierung: *com.fujitsu-siemens.cmx.oam*. Die Autorisierung berechtigt die Prozesse, denen diese Autorisierung zugeordnet ist, zur Administration von CMX. Die Programme von CMX, die Betriebsparameter ändern bzw. die Komponenten in und außer Betrieb nehmen, überprüfen an Hand dieser Autorisierung, ob der initiierende Prozess hierzu auch berechtigt ist.
2. Das Rechte-Profil: *CMX Administration*. Diesem Rechte-Profil ist die Autorisierung *com.fujitsu-siemens.cmx.oam* zugeordnet. Außerdem berechtigt dieses Profil, einige privilegierte Benutzerkommandos mit definierten Sicherheitsattributen auszuführen.
3. Die Benutzerrolle *cmxadm*, der das Rechte-Profil *CMX-Administration* zugeordnet ist. Die Benutzerrolle wird ohne Kennwort installiert. Beim ersten Einloggen muss der Benutzer ein Passwort vergeben. Dieser Benutzerrolle ist die Benutzergruppe *cmxadm* und standardmäßig das Verzeichnis */var/opt/SMAWcmx/adm/log* zugeordnet.

Diese Objekte können aber auch zentral über die Namensdienste in einem Systemverbund verwaltet werden.

Die Neu-Installation von CMX V6.0 setzt voraus, dass die Benutzerkennung *cmxadm* im System noch nicht eingerichtet ist. Ist dies nicht der Fall, so wird die Installation abgebrochen. Bei der Deinstallation von CMX V6.0 bleibt die Kennung bestehen.

Um die Rolle cmxadm nutzen zu können, muss der Systemverwalter folgende Aktionen durchführen:

1. Passwortvergabe für die Benutzerrolle *cmxadm*. Die Passwortvergabe kann nach der Installation von CMX den lokalen Administrationsrichtlinien angepasst werden.

2. Definition der Benutzer, die CMX administrieren dürfen, z. B. durch die Kommandoeingabe:

   ```
   usermod -R cmxadm hugo
   ```

   die den Benutzer hugo zur CMX-Administration berechtigt.

Die an einen Benutzer vergebenen Privilegien zur CMX-Administration können ihm wieder entzogen werden.

3.3 Funktionalität der Benutzerrolle cmxadm

Neben dem Kommandovorrat, der jedem Benutzer zur Verfügung steht, sind sämtliche Aufgaben der Konfiguration, Administration und Wartung von CMX/CCP der Benutzerrolle *cmxadm* zugeordnet. Die Funktionen können sowohl über die Menü-Oberfläche CMXCUI als auch über das Command Line Interface initiiert werden.

Die Benutzerrolle *cmxadm* berechtigt nicht zur Installation der CMX/CCP-Software und berechtigt auch nicht zum System-Shutdown, um Controller auszutauschen. Dies bleibt dem übergeordneten Systemverwalter vorbehalten. Es ist jedoch möglich, das Funktionsspektrum der Benutzerrolle *cmxadm* durch die Zuordnung entsprechender Rechte-Profile zu erweitern.

Die Benutzerrolle *cmxadm* ist ein optionales Feature. Es bleibt jedem Systembetreiber überlassen, die CMX-Administration auf die Benutzerrolle *cmxadm* umzustellen. Alle OAM-Aufgaben sind auch weiterhin unter der Systemverwalterkennung *root* durchführbar. Simultan kann sogar von beiden Benutzern administriert werden.

3.4 CMX-Administration unter cmxadm

Die Administration von CMX unter der Benutzerrolle *cmxadm* kann systemlokal oder auch von remote aus erfolgen. Mit dem Kommando *su* wechselt ein autorisierter Benutzer nach erfolgreicher Authentifizierung in die Benutzerrolle *cmxadm*.

Sofern in einem Systemverbund die Benutzernamen und die zugeordneten Ids auf allen Systemen identisch sind, ist ein Einloggen über *rlogin* möglich. In dem zu administrierenden System wird überprüft, ob die sich anmeldende Benutzer-Id die angegebene Rolle einnehmen darf, bevor die Authentizität überprüft wird.

4 Adressierungskonzept

Die erforderlichen Daten, die Sie zum Konfigurieren von CCP und CMX benötigen, werden von drei verschiedenen System-Komponenten verwaltet:

– vom Transport Name Service (TNS)

– vom Forwarding Support Service (FSS)

– in den Konfigurationsdateien (CF) der Subnetz-Profile

Die Funktion dieser Komponenten bei der Adressierung wird in den folgenden Abschnitten beschrieben.

4.1 Adressierung von Transportsystem-Anwendungen im TNS

Jedes Netz und jedes Transportsystem verlangt spezielle Adress-Informationen, um die Kommunikationspartner adressieren zu können. CMX bietet den Transport Name Service TNS, mit dem Sie Namen und Adressen von TS-Anwendungen verwalten können, unabhängig davon, auf welcher Kommunikationsschnittstelle (ICMX oder XTI) diese aufsetzen.

TNS liest die Adress-Informationen aus einem Verzeichnis, dem TS-Directory (Transport Service Directory). In dem TS-Directory sind die Adress-Informationen jeder TS-Anwendung unter ihrem symbolischen Namen, dem GLOBALEN NAMEN der TS-Anwendung, abgelegt. Das TS-Directory enthält Informationen zu allen im lokalen System residierenden TS-Anwendungen und zu den potenziellen Kommunikationspartnern in fernen Systemen.

Eine TS-Anwendung arbeitet nur mit ihrem eigenen GLOBALEN NAMEN und mit den GLOBALEN NAMEN ihrer Kommunikationspartner.

TNS macht die TS-Anwendungen also von der umgebenden Konfiguration der Kommunikations-Hardware und -Software unabhängig. Diese Unabhängigkeit bezieht sich z. B. auf:

– Art und Anzahl der Communication Controller (CC), die in Ihrem Rechner installiert sind

– die Topologie des Netzes, in das Ihr Rechner integriert ist

– die CCP-Profile, die in Ihrem Rechner ablaufen

All diese Merkmale und Daten brauchen bei Verwendung des TNS innerhalb einer TS-Anwendung nicht mehr berücksichtigt zu werden, egal, ob sie am Ort ihres Ablaufes, am Ort der Partneranwendung oder auf dem Weg dorthin (Addressing, Routing) vorliegen.

Die genannten Konfigurationsabhängigkeiten verwaltet der TNS im TS-Directory. Sie sind dort als Eigenschaften zu den TS-Anwendungen abgelegt. Die Identifikation der TS-Anwendungen erfolgt über einen hierarchisch strukturierten Namen, ihren GLOBALEN NAMEN. Die Programmierschnittstellen ICMX(L) und XTI bieten Abfragefunktionen, mit denen eine TS-Anwendung auf diese Eigenschaften zugreifen kann. Auf diese Weise wird die ICMX- bzw. XTI-Anwendung von der Verarbeitung physischer Adressen entlastet. Sie ist somit leichter in neuen Umgebungen einzusetzen und unabhängig von Änderungen im Netz.

In den folgenden Abschnitten werden die für den TNS wichtigen Elemente, der GLOBALE NAME, die Eigenschaften und das TS-Directory, beschrieben. Wie ein TS-Directory verwaltet wird, erfahren Sie im Abschnitt „Adress-Verwaltung in TS-Directories“ auf Seite 38.

Feste Begriffe, wie z. B. der GLOBALE NAME oder die Namen der Eigenschaften von TS-Anwendungen, werden in Großbuchstaben angegeben, um sie innerhalb des Textes kenntlich zu machen. TS-Anwendungen, die im lokalen Endsystem residieren, werden im Folgenden als lokale TS-Anwendungen bezeichnet. Entsprechend heißen TS-Anwendungen, die in fernen Endsystemen residieren, ferne TS-Anwendungen.

4.1.1 Adress-Verwaltung in TS-Directories

Der TNS verwaltet alle Eigenschaften der TS-Anwendungen im Transport Service Directory (TS-Directory). Das TS-Directory besteht aus Einträgen, von denen jeder die Eigenschaften einer TS-Anwendung enthält. Jeder Eintrag wird durch den GLOBALEN NAMEN identifiziert. Das TS-Directory enthält Einträge für alle TS-Anwendungen im lokalen Endsystem und für alle TS-Anwendungen in fernen Endsystemen, die als Kommunikationspartner in Frage kommen.

4.1.2 Identifizierung durch GLOBALEN NAMEN

Um Kommunikationsbeziehungen herstellen zu können, müssen TS-Anwendungen in der Lage sein, sich gegenseitig zu adressieren - ähnlich, wie sich Teilnehmer im Telefonnetz anhand ihrer Telefonnummern adressieren. So wie

der Telefonnetzbetreiber den Teilnehmern Telefonnummern zuweist, so ordnen Sie den TS-Anwendungen ihre Identifikationen explizit zu: Sie geben jeder lokalen und jeder fernen TS-Anwendung einen eindeutigen GLOBALEN NAMEN. Global bedeutet hier, dass der Name unabhängig von den zu benutzenden Netzen gültig ist.

Unter einem GLOBALEN NAMEN versteht man einen hierarchisch strukturierten Anwendungsnamen. Dieser Name lässt sich in maximal 5 Teile gliedern (Namensteil 1 bis 5). Von diesen ist der Namensteil 1 in der Hierarchie der höchste, der Namensteil 5 der niedrigste. Denken Sie sich als Modell ein (weltweites!) Telefonnummernverzeichnis mit Länderkennzahl, Städtevorwahl und Telefonnummern oder ein Adressbuch mit Nationalitätskennzeichen, Postleitzahl, Zustellungsbezirk, Straße mit Hausnummer und Name des Empfängers.

Für einige Standard-Anwendungen gibt es Namenskonventionen. Beispiel:

– der Enterprise File Transfer *open*FT benutzt die GLOBALEN NAMEN *$FJAM, $FJAM001, $FJAM002* usw.

– Das Produkt EMDS benutzt für seine Anwendungen die GLOBALEN NAMEN *dss_000*, *drs_000, dss_001, drs_001* usw.

Unter einem *Netz* wird dabei eine Gesamtheit von Rechnern verstanden, die nach einem bestimmten Schema adressiert werden, wie zum Beispiel das TRANSDATA NEA-Netz (Adressierung durch Prozessor- und Regionsnummer, Beispiel: *23/355*) oder das Internet (Adressierung durch IP-Adresse, Beispiel: *139.22.195.99*). Jeder Rechner in einem Netz ist durch seine Netzadresse (Synonym: *NSAP-Adresse*) eindeutig identifiziert. Ein Rechner, der in mehrere Netze eingebunden ist, hat je eine spezifische Netzadresse für jedes dieser Netze.

4.1.3 Adress-Information im GLOBALEN NAMEN

Bei der Adressierung einer Verbindung wird zwischen lokaler (= auf dem eigenen System residierender) und ferner (= auf dem Partnersystem residierender) TS-Anwendung unterschieden. Sie werden jeweils durch einen GLOBALEN NAMEN identifiziert. Sie unterscheiden sich jedoch in den Adress-Informationen, die dem GLOBALEN NAMEN zugeordnet sind.

Sie müssen die GLOBALEN NAMEN aller lokalen TS-Anwendungen und aller TS-Anwendungen an fernen Rechnern, mit denen eine lokale TS-Anwendung kommunizieren soll, in das TS-Directory aufnehmen. Jedem dieser GLOBALEN NAMEN, d. h. jedem Blatt im Namensbaum, können Sie bestimmte Eigenschaf-

ten zuordnen. Welche Eigenschaften Sie einer TS-Anwendung zuordnen können, hängt davon ab, ob die TS-Anwendung im lokalen oder in einem fernen Endsystem residiert.

Die Eigenschaften LOKALER NAME und TRANSPORTADRESSE sind für die Kommunikation besonders wichtig. Jeder TS-Anwendung im lokalen System müssen Sie die Eigenschaft LOKALER NAME zuordnen. Jeder TS-Anwendung in einem fernen Endsystem müssen Sie die Eigenschaft TRANSPORTADRESSE zuordnen.

Unabhängig davon, ob die TS-Anwendung lokal oder fern ist, können Sie auch die Session-Komponente und die Presentation-Komponente erfassen.

Die hierarchische Struktur des GLOBALEN NAMENS bewirkt die Anordnung aller GLOBALEN NAMEN in einem Namensbaum. Ein Blatt (Leaf Entity) im Namensbaum entspricht einer TS-Anwendung. Einem Blatt kann eine Auswahl von Eigenschaften (Property) zugeordnet werden, z. B. TRANSPORTADRESSE.

Der Pfad von der Wurzel des Namensbaumes zum Blatt wird durch den GLOBALEN NAMEN der TS-Anwendung vorgegeben. Der Name kann aus bis zu 5 Namensteilen bestehen, die den Weg von der Wurzel des Baumes über die (bis zu 4) Knoten zum Blatt angeben. Es können auch Namensteile übersprungen werden. Ein „vollständiger" Namensbaum mit allen Namensteilen sieht beispielsweise wie folgt aus:

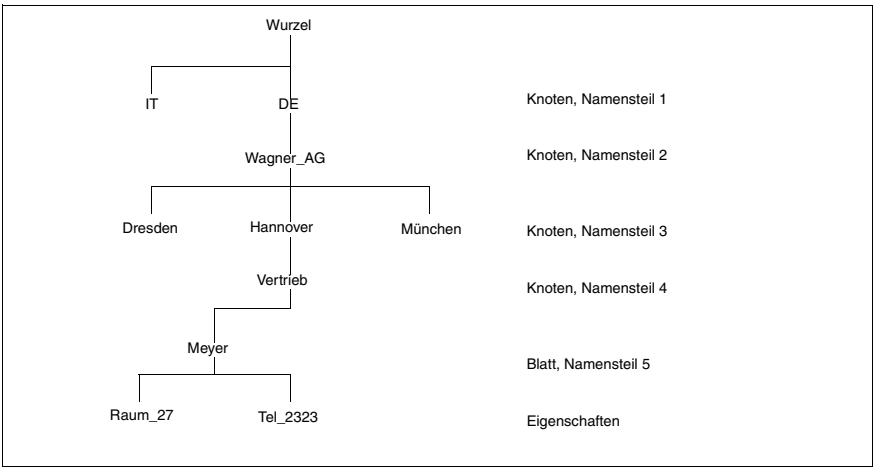


Bild 12: Beispiel für einen Namensbaum

Der zum Bild 12 korrespondierende GLOBALE NAME lautet in der Schreibweise des TNS:

```
Meyer.Vertrieb.Hannover.Wagner_AG.DE
```

Beachten Sie, dass hierbei Namensteil 5 zuerst genannt wird.

Übersicht der Merkmale eines GLOBALEN NAMENS

- Ein GLOBALER NAME ist ein Pfad im Namensbaum von der Wurzel zu einem Blatt
- Die Namensteile sind die Pfadkomponenten
- Die Festlegung von Knoten und Blatt erfolgt beim Einrichten des GLOBALEN NAMEN
- Die Namensteile 1 bis 4 können Pfadkomponenten zu einem Blatt sein
- Namensteil 5 kann nicht Pfadkomponente zu einem Knoten sein
- An einen Knoten kann ein weiterer Knoten oder ein Blatt unter Beachtung der Hierarchie angefügt werden
- Eigenschaften können nur einem Blatt zugeordnet werden

Im Namensbaum werden nur Blätter eingerichtet. Ein Knoten, an denen keine Blätter hängen, kann nicht explizit erzeugt werden. Ein Knoten wird jedoch implizit eingerichtet, wenn Blätter eingerichtet werden, und gelöscht, wenn alle Blätter gelöscht werden, die ihm zugeordnet sind.

GLOBALEN NAMEN strukturieren

Wie die Struktur des Namensbaumes angelegt wird, ob mit oder ohne Differenzierung nach Wurzel und Knoten und Blatt, bleibt dem einzelnen Anwendungsfall und der Gesamtkonfiguration aller TS-Anwendungen überlassen. Es liegt im Ermessen des Netzadministrators, wie „tief" er die Baumstruktur anlegt. Nicht zuletzt die Anzahl der TS-Anwendungen wird die Struktur prägen: Für wenige Anwendungen wird ein „flacher" Baum ohne Knoten ausreichen. Bei vielen Anwendungen sollten Sie sich der Strukturierung mit den Knoten bedienen, um deren Vorteile hinsichtlich Übersichtlichkeit, Zugriffsoptimierung etc. auszunutzen. Es empfiehlt sich, den Baum nach organisatorischen oder topologischen Gesichtspunkten zu strukturieren.

Bedeutung der Namensteile

Die Bezeichnung der Namensteile entspricht den Vorschlägen der internationalen Normungsgremien von ISO, CCITT und ECMA.

Für einen „vollständigen" Namensbaum ergibt sich folgende Zuordnung:

Namens-teil	Bezeichnung	Bedeutung	Länge in Byte	im Baum
1	TS_COUNTRY	Country	2	Knoten, Blatt
2	TS_ADMD	Administrative Domain	16	Knoten, Blatt
3	TS_PRMD	Private Domain	16	Knoten, Blatt
4	TS_OU	Organisation Unit	10	Knoten, Blatt
5	TS_PN	Personal Name	30	Blatt

Tabelle 1: Bedeutung der Namensteile

4.1.3.1 Lokale TS-Anwendung

Eine lokale *TS-Anwendung* muss sich beim Transportsystem anmelden, um Kommunikation betreiben zu können. Dabei muss sie anzeigen, welche Transport Service Provider (TSPs) sie benutzen will. Die verschiedenen TSPs werden durch unterschiedliche Adress-Formate identifiziert. In manchen Fällen bezeichnet das Adress-Format auch eine bestimmte Kombination von TSP und Transportprofil oder Adressierungsvariante.

Beispiele:

– Adress-Format WANSBKA (für TSP OSI TP0/2) für OSI-Verbindungen über WAN gemäß ISO 8072

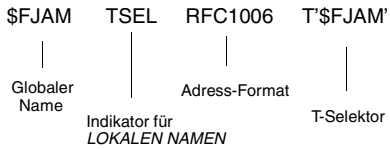
– Adress-Format RFC1006 (für TSP RFC1006) für Rechnerkopplung über TCP/IP mit RFC1006-Konvergenzprotokoll

Die bei den einzelnen Adress-Formaten anzugebende Adress-Information kann eine unterschiedliche Bedeutung haben, z. B.

– Stationsname bei einem NEA-Transportprofil
– LU-Name und LU-Nummer bei einem SNA-Transportprofil
– T-Selektor bei einem OSI-Transportprofil
– TCP-Portnummer, sofern zur Adressierung nicht die RFC1006-Standard-Portnummer 102 verwendet wird

Einträge für lokale TS-Anwendungen beginnen jeweils mit dem Indikator TSEL.

Beispiel



Dem GLOBALEN NAMEN einer lokalen TS-Anwendung ordnen Sie daher einen Satz von T-Selektoren zusammen mit den Adress-Formaten zu. Die Summe dieser Einträge wird LOKALER NAME genannt.

Beispiel: Die durch den GLOBALEN NAMEN *$FJAM* bezeichnete Anwendung soll bei den TSPs RFC1006 (Adress-Format RFC1006), TRANSDATA NEA (Adress-Format WANNEA) und dem lokalen Loopback (Adress-Format LOOPSBKA) bekannt gemacht werden. Dafür sind als LOKALER NAME drei TSEL-Einträge anzugeben:

```
$FJAM \
   TSEL RFC1006 T´$FJAM´
   TSEL WANSBKA T´$FJAM´
   TSEL LOOPSBKA T´$FJAM´
```

4.1.3.2 Ferne TS-Anwendung

Um eine *ferne TS-Anwendung* zu adressieren, werden folgende Informationen benötigt:

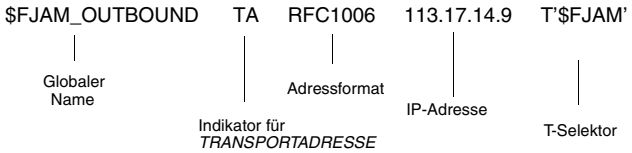
– in welchem fernen System läuft sie ab (Netzadresse)

– über welchen TSP kann sie erreicht werden (Adress-Format des Subnetzzugangs)

– wie wird sie im fernen System identifiziert (T-Selektor der fernen Anwendung)

Diese Informationen werden in CMX als TRANSPORTADRESSE bezeichnet. Die TRANSPORTADRESSE wird einem GLOBALEN NAMEN zugeordnet, durch den die ferne TS-Anwendung identifiziert wird.

Beispiel



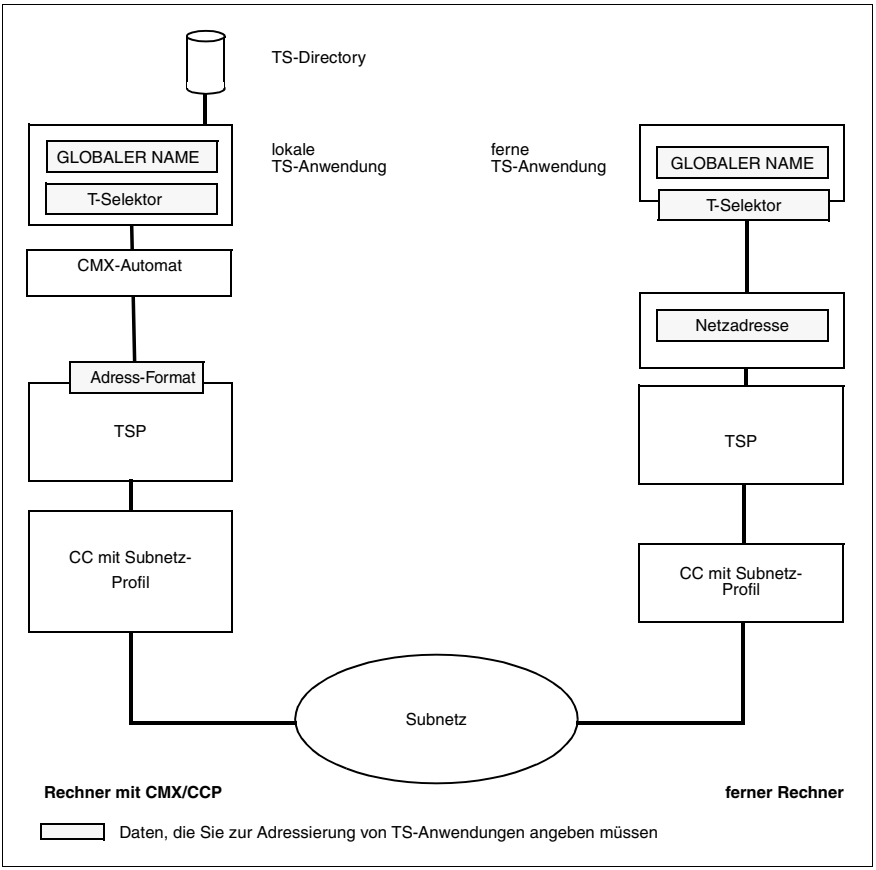


Bild 13: Adressierung im TNS

4.2 Adressierung von Partnersystemen im FSS

Der Forwarding Support Service (FSS) ergänzt die Adressierungsfunktionen des TNS. Der TNS benennt die Kommunikationspartner und stellt ihre Adressen bereit. Der FSS verwaltet Eigenschaften der Route zum Partner. Dies sind zum Beispiel

– die DTE-Adresse oder Rufnummer, über die ein NEA-Partner erreicht werden kann

– X.25-Dienstmerkmale (z. B. Gebührenumkehr), die mit einem X.25-Partner ausgehandelt werden können

Typische FSS-Einträge betreffen die eigene Netzadresse (z. B. die eigene NEA-Adresse), ferne Netzadressen (z. B. ferne NEA-Adresse), Adressen in Subnetzen (Telefonnetz, X.25-Netz usw.), über die der ferne Partner erreicht wird, sowie die physischen Leitungsanschlüsse (Subnetz-ID), die in das Subnetz führen. All diese Informationen und ihre Beziehungen untereinander werden in einer FSS-Konfiguration abgelegt. Beim Verbindungsaufbau greifen die Transport Service Provider (TSPs) auf die Informationen in der FSS-Konfiguration zu.

Jede Information, die Sie eingeben wollen, wird vom *Forwarding Support Service* Ihres Systems als Objekt einer bestimmten Klasse abgelegt. So ist eine ferne Netzadresse (NSAP-Adresse) an ein NSAP-Objekt, eine ferne Subnetzadresse (SNPA-Adresse) und die für eine ferne SNPA-Adresse in Frage kommenden Routen (d. h. die Kombination aus ferner SNPA-Adresse und dazu passenden lokalen Subnetz-Anschlüssen) an ein SNPAROUTES-Objekt geknüpft. Die Einträge des CMXCUI orientieren sich an diesen Objektklassen.

Die aktive FSS-Konfiguration erstellen und verwalten Sie mit Hilfe des CMXCUI (siehe hierzu Kapitel „Konfigurieren und Administrieren im Menü“ auf Seite 59). Eine entsprechende *Forwarding Support Konfigurationsdatei* können Sie auch im Format *fsconfig* editieren (siehe Abschnitt „FSS-Konfigurationsdatei erstellen (Format fsconfig)“ auf Seite 126). Sie sollten dies jedoch nur tun, wenn Sie eine größere Menge von Konfigurationsdaten erfassen müssen.

Folgende Übersicht zeigt die Zuordnung zwischen den am häufigsten benötigten Adress-Informationen und den FSS-Objektklassen und CMXCUI-Einträgen:

Adress-Information	**FSS-Objektklasse**	**Eintrag am CMXCUI**
Lokale Netzadresse (NSAP)	LOCNSAP	Lokal - Lokaler Rechner...
Route (Kombination von lokaler Subnetz-ID + fernem SNPA)	SNPAROUTES	Route - Routen zu fernen Subnetzanschlüssen
Ferne Netzadresse (NSAP)	NSAP	NSAPs - Ferne Rechner...

Tabelle 2: Zuordnung von Adress-Information zu Objektklassen

Im Folgenden wird dargestellt, wie Sie diese Informationen mit Hilfe der einzelnen Objektklassen verwalten können. Die möglichen Attribute für jede dieser Objektklassen finden Sie im Abschnitt „Übersicht der Objektklassen und ihrer Attribute“ auf Seite 108 ausführlich beschrieben.

4.2.1 Netzadressen

Eigene Netzadressen

Für jedes Netz (NEA, OSI, Internet), über das Ihr lokales System kommunizieren soll, muss die eigene lokale Netzadresse konfiguriert werden. In der FSS-Konfigurationsdatei geben Sie die lokalen Netzadressen über die Objektklasse LOCNSAP ein (siehe Abschnitt „Übersicht der Objektklassen und ihrer Attribute“ auf Seite 108).

Netzadressen der Partnersysteme

Sie konfigurieren Partnersysteme, mit denen Sie kommunizieren wollen, indem Sie zunächst deren Netzadressen eingeben. Dies geschieht im CMXCUI über die Option *NSAPs* und in einer FSS-Konfigurationsdatei durch Angabe von NSAP-Objekten.

4.2.2 Subnetzanschlüsse und Routen

Für den aktiven Verbindungsaufbau zu einem fernen System muss einer der Netzadressen des fernen Systems (z. B. IP-Adresse) eine Route im Subnetz (z. B. X.25-Subnetz) zugeordnet werden. Erst dann ist das ferne System erreichbar. Eine Route ist definiert durch ihren Ausgangspunkt und ihren Zielpunkt: Ausgangspunkt der Route ist ein lokaler Subnetzanschluss; Endpunkt ist der Subnetzanschluss des Partnersystems bzw. des nächsten Übergangssystems.

Routen mit gleichem Endpunkt und gleichwertigen Ausgangspunkten müssen nicht voneinander unterschieden werden, sondern werden als Bündel einmalig konfiguriert. Dies geschieht für die lokale Seite durch die Angabe der Subnetz-ID, mit der die gleichwertigen lokalen Subnetz-Anschlüsse zusammengefasst werden. Für die Partner-Seite wird als Endpunkt der Route unverändert der ferne Subnetz-Anschluss eingetragen. In der FSB werden solche gleichwertigen Routen durch ein SNPAROUTES-Objekt repräsentiert.

Vergabe der Subnetz-ID

Für jeden lokalen Subnetz-Anschluss müssen Sie eine Subnetz-ID vergeben. Führen zwei Subnetz-Anschlüsse ins gleiche Subnetz, so können Sie an beide dieselbe Subnetz-ID vergeben (dies geschieht in den Konfigurationsdateien der Subnetz-Profile, siehe Handbücher „CMX/CCP, ISDN-Kommunikation" [3] und „CMX/CCP, WAN-Kommunikation" [4]). Sie überlassen dann dem System die Auswahl des abgehenden Subnetz-Anschlusses beim aktiven Verbindungsaufbau.

Wenn Sie gezielt nur über bestimmte Subnetz-Anschlüsse Verbindungen zu Partnern aufbauen wollen, so vergeben Sie für zwei Subnetz-Anschlüsse ins gleiche Subnetz verschiedene Subnetz-IDs.

Dienstmerkmale (Facilities)

Jeder Route und jedem fernen Subnetzanschluss können Sie bestimmte Merkmale zuordnen (z. B. X.25- oder ISDN-Dienstleistungsmerkmale wie „Gebührenumkehr", „Geschlossene Benutzergruppe"). Diese Merkmale werden in einem Objekt FACIL („Facilities") definiert und einer Route oder einem fernen Subnetz-Anschluss zugeordnet. Die Konfiguration dieser Facilities im FSS erlaubt einfaches Einstellen, Abfragen und Ändern einzelner oder kombinierter Merkmale im laufenden Betrieb.

4.2.3 Routenermittlung

Beim Aufbau einer Transportverbindung ermitteln der CMX-Automat und die TSPs den Kommunikationspfad zwischen einer lokalen und einer fernen TS-Anwendung. Hierzu treffen sie Auswahlen und nehmen Abbildungen vor. In der folgenden Darstellung wird von dem Fall ausgegangen, dass noch keine Netzverbindung zu dem Rechner existiert, auf dem die ferne TS-Anwendung abläuft.

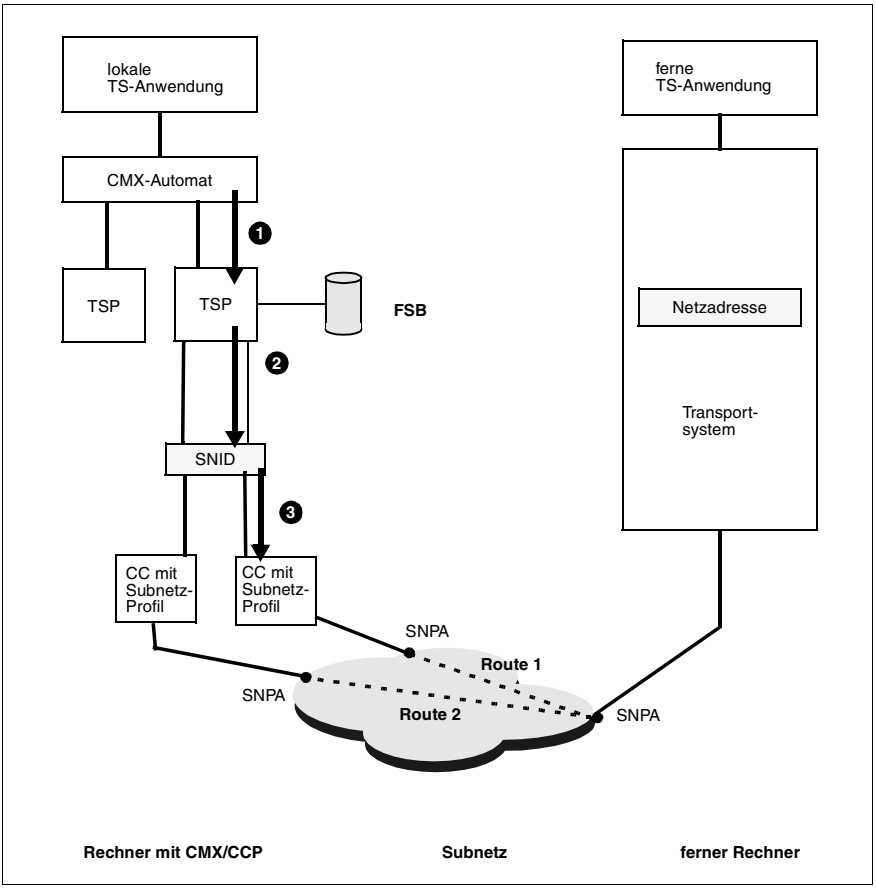


Bild 14: Konzept der Routenermittlung

Die in Bild 14 durch Pfeile symbolisierten Beziehungen sind im Folgenden beschrieben.

(1) Auswahl des TSPs

In der TRANSPORTADRESSE (TNS) ist codiert, über welches Netz (NEA, OSI, Internet) die Transportverbindung aufgebaut werden soll. Mit dieser Information wählt der CMX-Automat den geeigneten TSP aus.

Dem TSP stehen zum Aufbau einer Netzverbindung in der Regel mehrere *Subnetze* zur Verfügung. Ein Subnetz ist die Gesamtheit der Betriebsmittel (z. B. Übertragungswege, Vermittlungsstellen), die die Kommunikation innerhalb einer Gruppe von Rechnern ermöglichen, wie zum Beispiel ein Datex-P-Netz, ein Ethernet-Segment oder eine Standleitung. Zu jedem dieser Subnetze gibt es einen oder mehrere Zugangspunkte *(Subnetwork Points of Attachment, SNPA)*, die durch *SNPA-Adressen* (= Subnetz-Adressen) identifiziert werden. Bei einer SNPA-Adresse kann es sich beispielsweise um eine Rufnummer in einem Datex-P-Netz, die Adresse eines Ethernet-Anschlusses oder die Kombination aus einer CC-Identifikation und der Nummer einer Leitung dieses CC handeln.

Die lokalen Subnetz-Anschlüsse (SNPAs), die in dasselbe Subnetz führen, sind für die Anwendungen gleichwertig, da über jeden von ihnen dieselben fernen Rechner erreichbar sind. Jede derartige Gruppe lokaler SNPAs versehen Sie mit einem gemeinsamen symbolischen Bezeichner, der *Subnetz-ID (SNID)*. Die SNID beschreibt, um welche Art von Subnetz (siehe obige Beispiele) und um welche Gruppe von Zugängen zu diesem Subnetz es sich handelt. Eine SNID steht beispielsweise für zwei Anschlüsse an das öffentliche Datex-P-Netz oder für zwei ISDN-Festverbindungen, die zum gleichen fernen Rechner führen.

(2) Abbildung der fernen Netzadresse auf Subnetz-ID und ferne SNPA-Adresse

Um eine Netzverbindung zu etablieren, muss der TSP eine Route aufbauen. Dazu leitet er aus der fernen Netzadresse (die er aus der TRANSPORTADRESSE gelesen hat) die Subnetz-ID und die ferne SNPA-Adresse ab. Abhängig vom CCP-Profil beschreitet der TSP dabei einen dieser drei Wege:

– Die ferne Netzadresse enthält bereits die ferne SNPA-Adresse und Angaben, aus denen der TSP die Subnetz-ID konstruieren kann.

– Innerhalb des Netzes wird ein Routingprotokoll verwendet. Das heißt, dass jeder Rechner jedem anderen regelmäßig mitteilt, welche Netzadresse und welche SNPA-Adresse er hat. Auf diese Weise stellt sich jede Protocol Entity ein Adressbuch zusammen, mit dessen Hilfe sie beim Netzverbindungsaufbau Netzadressen auf SNPA-Adressen abbilden kann.

– Der TSP führt die Abbildung der fernen Netzadresse auf Subnetz-ID und ferne SNPA-Adresse unter Zugriff auf eine Datenbasis durch (siehe unten).

(3) Auswahl des SNPA

Bezeichnet die Subnetz-ID eine Gruppe, die aus mehr als einem lokalen SNPA besteht, so muss der TSP daraus einen lokalen SNPA auswählen. Über ihn baut er die neue Route und damit die neue Netzverbindung auf.

4.3 Ablauf eines Verbindungsaufbaus

Anhand eines Beispiels sehen Sie im Folgenden, wie die Informationen aus den verschiedenen Datenbasen zusammenfließen. Das Beispiel zeigt den Ablauf eines Verbindungsaufbaus über ein NEA-Netz. Die schwarz unterlegten Zahlen beziehen sich auf die anschließend beschriebenen Schritte beim Zugriff auf die Konfigurations-Informationen.

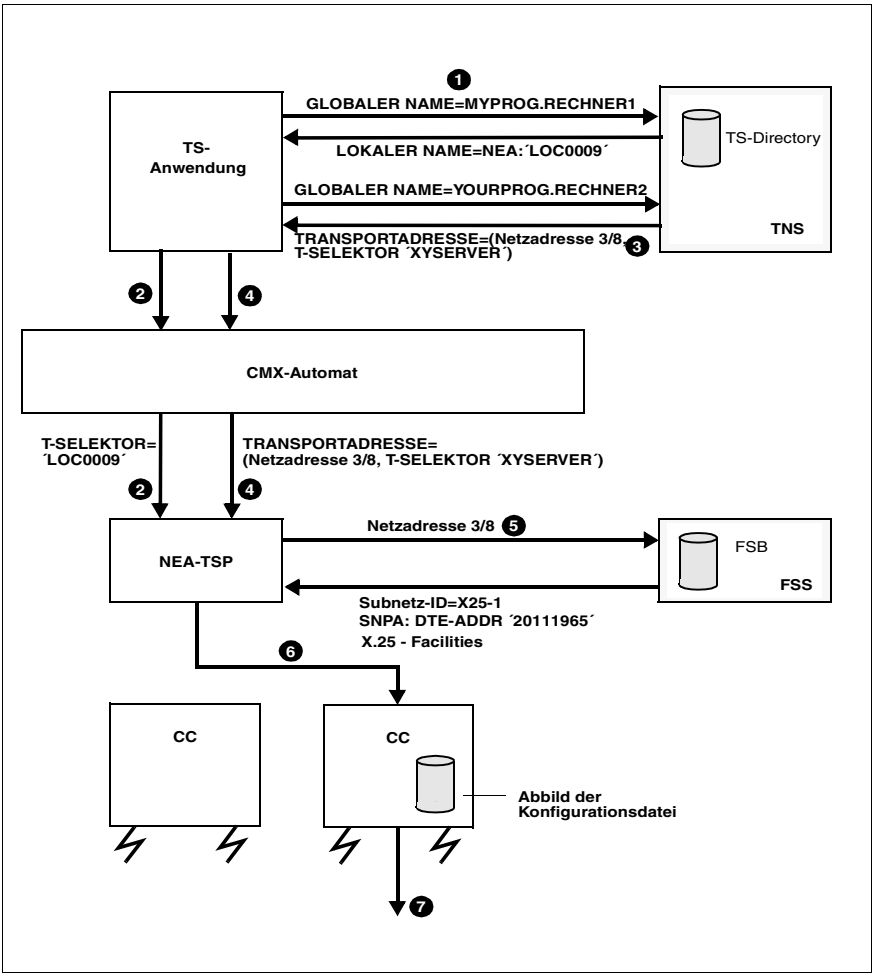


Bild 15: Aufbau einer Transportverbindung über TRANSDATA-NEA-Netz

Zugriffe auf Konfigurationsdaten

1. Die TS-Anwendung bildet mit Hilfe des TNS ihren GLOBALEN NAMEN und bildet ihn auf ihren LOKALEN NAMEN ab, der in diesem Fall nur einen T-Selektor für das NEA-Netz enthält.
2. Die TS-Anwendung meldet sich mit diesem LOKALEN NAMEN an.

3. Die TS-Anwendung erfragt vom TNS die zum GLOBALEN NAMEN der Partneranwendung gehörige TRANSPORTADRESSE. Sie besteht aus der NSAP-Adresse des fernen Systems und dem T-Selektor der Anwendung.
4. Die TS-Anwendung gibt diese TRANSPORTADRESSE bei der Verbindungsaufbau-Anforderung an den TSP weiter.
5. Es existiert noch keine Netzverbindung zu dem durch die NSAP-Adresse spezifizierten fernen Rechner. Der TSP übergibt dem FSS die Netzadresse des fernen Systems, die ein Teil der TRANSPORTADRESSE ist (siehe obige Begriffsdefinitionen), und erhält die Subnetz-ID und die ferne SNPA-Adresse, bei der es sich in diesem Fall um eine DTE-Adresse (Rufnummer in einem Datex-P-Netz) handelt. Außerdem erhält er die X.25-Facilities, die für diese ROUTE konfiguriert worden sind.
6. Der TSP stellt durch Vergleich der Subnetz-ID fest, über welchen CC die Verbindung aufgebaut werden soll. Welche Subnetz-IDs welchen CCs zugeordnet sind, erfährt der TSP beim Laden der CCs. Die Information kommt aus der Konfigurationsdatei des Subnetz-Profils. Der TSP sucht aus der durch die Subnetz-ID identifizierten Leitungsgruppe eine Leitung heraus und schickt die Verbindungsaufbau-Anforderung zum entsprechenden CC.
7. Das Subnetz-Profil auf dem CC baut eine Subnetzverbindung auf. Sobald diese existiert, baut der TSP darüber eine Transportverbindung auf.

Konfigurieren von Transportverbindungen

Über die Benutzerschnittstellen von CMX konfigurieren Sie Anwendungen und Netzpartner und stellen für Ihr eigenes System Betriebsparameter für TSPs und Controller ein. Dazu haben Sie zwei Möglichkeiten:

– Sie können mit dem zeichenorientierten Menü arbeiten (CMXCUI).

– Sie können Konfigurationsdaten über die Kommandoschnittstellen erfassen und dann dem System bekannt geben (*fssadm*, *tnsxcom*).

Konfigurierung per Editor

In einigen Einsatzfällen ist es vorteilhaft, wenn die Konfigurierungsdaten statt über das CMX-Menü über Text-Dateien in speziellen Formaten eingegeben werden können. Das ist immer dann der Fall, wenn eine große Zahl ähnlich bezeichneter Objekte erfasst werden soll (Massenkonfigurierung). Daher besteht alternativ die Möglichkeit, Konfigurierungsdaten für TNS und FSS über Dateien einzuspeisen. Dieses Vorgehen erfordert Expertenwissen und wird in Kapitel „Konfigurieren im Expertenmodus“ auf Seite 69 erläutert.

5 Installation und Inbetriebnahme

5.1 Installation von CMX

Das Produkt CMX sowie die aufsetzenden Produktkomponenten, die CCPs (Commiunication Control Programs) setzen sich aus mehreren Softwarepaketen (gemäß UNIX SVR4) zusammen.

Unter Solaris ist die Installation, Überinstallation und Deinstallation von CMX im laufenden System ohne Reboot des Systems möglich. Außerdem kann CMX per Live Upgrade (siehe Seite 54) in einem Solaris-System installiert werden.

Die Kommunikationssoftware CMX/CCP wird über *Web Start Wizard* von der CD installiert. Der Web Start Wizard bietet zwei Installationsvarianten an:

– die Installation durch Produktauswahl, bei der sämtliche Pakete des Produktes installiert werden,

– und eine kundenspezifische Installationsvariante, bei der der Kunde die zu installierenden Komponenten des Produktes seiner Kommunikationsinfrastruktur bedarfsgerecht anpassen kann.

Es können auch einzelne Produktkomponenten (nach-) installiert werden, ohne dass der laufende CMX-Betrieb bei einer Erweiterung des Kommunikationsprofile beeinträchtigt wird. Die Installation einzelner Pakete ist ebenfalls möglich, setzt aber Expertenwissen über die Abhängigkeiten zwischen den Paketen voraus.

Vor der Überinstallation oder Deinstallation müssen Sie alle CMX-Anwendungen beenden und alle betroffene Komponenten außer Betrieb nehmen. Andernfalls wird der Vorgang abgebrochen.

Nach erfolgreicher (Über-)Installation müssen Sie die Software in Betrieb nehmen, siehe Seite 56.

Genauere Informationen zu Hardware- und Software-Abhängigkeiten sowie zur Installation entnehmen Sie bitte der Freigabemitteilung.

Bei der Installation von CMX mit dem *Web Start Wizard* gehen Sie wie folgt vor:

► Legen Sie die Installations-CD in das CD-ROM-Laufwerk ein.

  Die Installations-CD wird automatisch eingehängt.

- Klicken Sie zunächst auf eines der README-Symbole, um vor der Installation die CMX-Freigabemitteilung zu lesen.
- Nach dem Lesen der CMX-Freigabemitteilungen klicken Sie auf das Symbol *Installer*.

  Sie erhalten ein Begrüßungsfenster.
- Klicken Sie auf *Next*.

  Eine Reihe weiterer Fenster führt Sie durch die Installationsprozedur. Am Schluss der Installationsprozedur öffnet sich das Fenster *Installation Summary*.
- Um ein Installationsprotokoll zu erhalten, das Informationen über den Erfolg oder Misserfolg der Installation enthält, und - im Fehlerfall - Hilfestellungen für das weitere Vorgehen und ausführliche Diagnoseinformationen, klicken Sie auf *Details*.

5.2 Live Upgrade und CMX

Solaris Live Upgrade [Sol_LU] ist ein Verfahren, das im laufenden Betrieb eine alternative Systemumgebung – mit aktualisiertem Solaris und anderer System-Software – erstellt. Dieses Verfahren der Systemaktualisierung bietet folgende Vorteile: Die Systemausfallzeit ist sehr gering, und sämtliche Systemkonfigurations- und Benutzerdaten bleiben erhalten. Hierzu kopiert Live Upgrade die aktuelle Systemumgebung vollständig in die neu zu erstellende Systemumgebung, bevor die Software in der neuen Systemumgebung aktualisiert wird.

Wenn in der aktuellen Systemumgebung das Produkt CMX installiert ist, wird es durch den Kopierprozess von Live Upgrade in die neue Umgebung übernommen. CMX ist **Live Upgrade fähig**, d.h. während des Live Upgrades kann die CMX-Version in der neuen Boot-Umgebung aktualisiert werden.

Wird in einem Live Upgrade die Solaris-Version aktualisiert, z.B. von Version 8 nach Version 9, dann müssen Sie die entsprechende CMX-Version in der aktualisierten Boot-Umgebung ebenfalls per Live Upgrade installieren. Andernfalls wird das installierte CMX nicht in Betrieb genommen.

Live Upgrade Installation von CMX

Die Installations-CD unterstützt Live Upgrade als alternative Installationsvariante zur Installation im laufenden System.

Weitere Details zu Live Upgrade Installation finden Sie in der Freigabemitteilung.

Synchronisierung der CMX-Dateien

Bei einem Live Upgrade werden im ersten Schritt die System-Dateisysteme kopiert. In der neu eingerichteten Boot-Umgebung sind daher Änderungen an diesen Dateisystemen, die nach dem Kopieren eingebracht wurden, nicht enthalten.

Beachten Sie dabei folgende Aspekte:

1. Die Konfiguration von CMX und den CCP's wird in den System-Dateisystemen abgespeichert.

   ► Damit Konfigurationsänderungen durch einen Live Upgrade nicht verlorengehen, führen Sie nach dem Start des Live Upgrades bis zum Aktivieren der neuen Boot-Umgebung keine Änderungen an der CMX-/CCP-Konfiguration durch.

2. Die In- und Außerbetriebnahme der Controller und der einzelnen Komponenten der Softwarekonfiguration CMX/CCP wird in Logging-Dateien protokolliert.

   ► Um ein lückenloses Protokoll dieser Komponenten über einen Live Upgrade sicherzustellen, geben Sie beim Aktivieren der neuen Boot-Umgebung die Synchronisierungsoption *-s* an.

   Dadurch werden Änderungen, die an der alten Boot-Umgebung durchgeführt wurden, in die neue Boot-Umgebung übernommen.

   ► Damit die Logging-Dateien auch übernommen werden, erweitern Sie die Datei */etc/lu/synchlist* um folgende zwei Einträge:

```
/var/opt/SMAWcmx/adm/log           OVERWRITE
/opt/SMAW/SMAWcmx/lib/ccp/diagfiles OVERWRITE
```

5.3 Betrieb des Produkts

Inbetriebnahme

Die installierten Komponenten werden beim Hochfahren des Systems automatisch in Betrieb genommen. Wenn Sie bei laufendem System Software nachinstalliert oder überinstalliert haben, dann können Sie diese direkt mit dem Kommando *cmx boot* in Betrieb nehmen, ohne dass das System neu gestartet werden muss.

Nach der Inbetriebnahme sind die von CMX bereitgestellten Funktionen nutzbar, die Anwendungen können gestartet werden.

Außerbetriebnahme

Sie können die Software bei laufendem System außer Betrieb nehmen, indem Sie das Kommando *cmx shutdown* eingeben. Zuvor müssen sich alle angemeldeten Anwendungen bei CMX abmelden bzw. sie müssen beendet werden.

Die Außerbetriebnahme der Software ist Voraussetzung für eine Deinstallation bzw. Updateinstallion der Kommunikationssoftware.

Grenzwerte

Die Maximalzahl der von CMX systemweit unterstützten Prozesse, Kommunikationsanwendungen und Verbindungen hängt von der Systemplattform und dem Ausgabestand von CMX ab. Die jeweils gültigen Werte finden Sie in der Freigabemitteilung.

Die in Ihrer Installation eingestellten Maximalwerte können Sie über das Kommando *cmxinfo* (siehe Abschnitt „Informationen zur CMX-Konfiguration (cmxinfo)“ auf Seite 239) abfragen. Es besteht die Möglichkeit, über das Kommando *cmxtune* (siehe Abschnitt „Grenzwerte für den CMX-Automaten ändern (cmxtune)“ auf Seite 291) die voreingestellten Werte herabzusetzen. Dies ist jedoch nur zur Speicheroptimierung im Rahmen abgestimmter Tuning-Maßnahmen zu empfehlen.

Die Maximalzahl der von jedem einzelnen TSP unterstützten Transportverbindungen hängt ebenfalls von der Systemplattform und dem Ausgabestand des entsprechenden Produkts ab. Für die TSPs RFC1006 und Null Transport (NTP) finden Sie die Werte in der Freigabemitteilung von CMX, für die anderen TSPs in der Freigabemitteilung des entsprechenden CCP-Produkts. Beachten Sie,

dass Sie möglicherweise Systemparameter in Solaris anpassen müssen, wenn die Zahl der gleichzeitig betriebenen Transportverbindungen eine bestimmte Größe überschreitet. Auch darüber gibt die Freigabemitteilung Auskunft.

Online-Manpages

CMX liefert Online-Manpages in englischer Sprache. Damit können Sie sich Informationen zu CMX-Kommandos und Dateiformaten auf dem Bildschirm anzeigen lassen.

6 Konfigurieren und Administrieren im Menü

Sie haben als Systemverwalter die Möglichkeit, Ihre Software und die gewünschten Kommunikations-Anwendungen im CMX-Menü (CMXCUI) zu konfigurieren und zu administrieren.

Es ist empfehlenswert, das System mit CMXCUI zu konfigurieren und zu verwalten.

Das zeichenorientierte Menü CMXCUI basiert auf der Schnittstelle FMLI.

Bevor Sie die Konfigurierung durchführen, müssen Sie entscheiden, welche Kommunikationswege Sie für Ihr System realisieren wollen. Sie sollten für Ihr Netz einen Adress-Plan erstellen, damit Sie bei der Konfigurierung die Adressen der betreffenden Systeme und TS-Anwendungen kennen.

6.1 Übersicht der zeichenorientierten Bedienoberfläche CMXCUI

Das CMXCUI ist eine Bedienoberfläche, mit deren Hilfe Sie Ihr System administrieren und konfigurieren sowie Informationen abfragen können. Die Bedienoberfläche wird in deutsch und englisch angeboten.

Am Beginn jedes Aufrufs wird untersucht, welche CCP-Produkte und Subnetz-Profile auf dem System installiert sind. Es erscheint das CMX-Hauptmenü, von dem aus Sie die untergeordneten Menüs erreichen können.

Sie erhalten in diesem Abschnitt eine Beschreibung der Menü-Oberfläche und eine Übersicht der Menü-Optionen.

6.1.1 Menü-Oberfläche

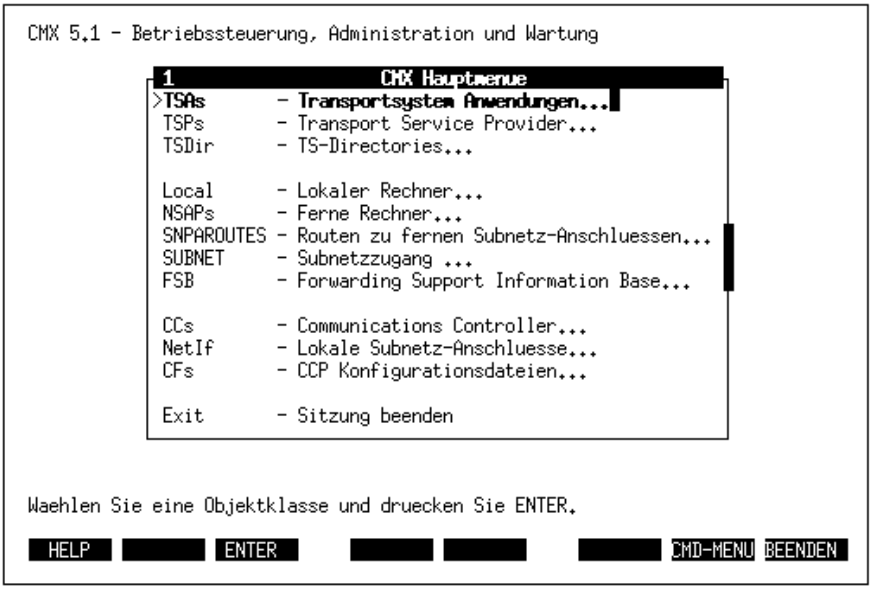


Bild 16: CMXCUI-Hauptmenü

Mit den Funktionstasten F1 bis F8 auf Ihrer Tastatur können Sie die am unteren Rand des Menüs angegebenen Funktionen bedienen. Die Tasten sind in den Untermenüs zum Teil unterschiedlich belegt. So können z. B. innerhalb von Hilfebildschirmen die Tasten F2 und F3 mit den Funktionen „Nächste Seite“ SEITE+ bzw. „Vorige Seite“ SEITE- belegt sein. Im Folgenden erhalten Sie eine kurze Übersicht der Funktionen.

i Die Funktionsfähigkeit der Funktionstasten F1 bis F8 setzt eine entsprechende Terminalemulation voraus. Sollten Sie Probleme mit der Terminalemulation haben, so können Sie sich mit den Tastenkombinationen CTRL F n behelfen.

F1 ist generell mit HILFE belegt. Über diese Funktionstaste erhalten Sie zum aktuellen Eintrag erläuternde Hinweise.

F2 ist mit AUSWAHL, MARKIEREN oder HINZUFÜGEN belegt. Damit erhalten Sie Optionen, die Sie auswählen können. In einigen Untermenüs können Sie mit dieser Taste Einträge für Mehrfachauswahl markieren. Wenn die MARKIERE-Taste erscheint, müssen Sie den gewünschten Eintrag in jedem Fall mit F2 auswählen. Ansonsten können Sie mit F3 ENTER auswählen.

Nicht für jede Aktion ist Mehrfachauswahl möglich (z. B. für „Editieren"). In diesem Fall wird die Aktion nur für die erste markierte Instanz durchgeführt.

F3 ist mit ENTER, WEITER oder SICHERN belegt. Im ersten Fall wählen Sie einen Menüeintrag aus, im zweiten bestätigen und sichern Sie Eingaben.

F4 ist in bestimmten Untermenüs mit BEARBEITEN oder NEU belegt.

F5 ist in bestimmten Untermenüs mit LÖSCHEN belegt. Der Eintrag, auf dem der Cursor positioniert ist, wird gelöscht.

F6 ist mit ABBRUCH belegt. Die Taste schließt das aktuelle Fenster und führt zum übergeordneten Menü. Die Taste bewirkt, dass vorgenommene Einträge unwirksam sind, falls sie nicht vorher gesichert wurden. Die Taste ist nur innerhalb des aktuellen Fensters wirksam. Einträge in Untermenüs werden nicht berührt und bleiben erhalten.

F7 bedeutet CMD-MENU.

F8 ist mit BEENDEN belegt.

Wenn F8 erscheint, können Sie in den alternativen bzw. in den Standard-Funktionstastensatz wechseln.

In folgenden Formularen wird automatisch auf den alternativen Funktionstastensatz umgeschaltet:

– in Formularen, in denen der Feldinhalt interaktiv verändert werden kann
– in Formularen, in denen ein Objekt, das referenziert wird, neu erzeugt werden kann
– in Formularen, die eine Liste gleichartiger Objekte enthalten

In diesen Fällen sind die Tasten folgendermaßen belegt:

Formulartyp	**F2**	**F3**	**F4**	**F5**	**F7**
Verändern	AUSWAHL	WEITER	BEARBEI-TEN		KDO-MENÜ
Hinzufügen	AUSWAHL	WEITER	NEU...		KDO-MENÜ
Liste	HINZUFÜ-GEN	WEITER	BEARBEI-TEN	LÖSCHEN	KDO-MENÜ

Tabelle 3: Alternativer Funktionstastensatz

Die Taste F1 ist immer mit HILFE belegt; die Taste F6 immer mit ABBRUCH.

Zum Anlegen eines neuen Objekts müssen Sie das Formular/Menü mit den vorhandenen Objekten mit ENTER bestätigen, ohne ein Objekt zu markieren. Sobald ein Objekt markiert ist, wird die Menüauswahl *Anlegen* inaktiv.

Aktionen, die innerhalb eines Formulars/Menüs nicht angewählt werden können, sind halbhell dargestellt.

6.1.2 Optionen im Menü

In diesem Abschnitt sind die Optionen des CMX-Menüs kurz beschrieben und die jeweils möglichen Aktionen angegeben. Das Hauptmenü ist in drei Arbeitsbereiche gegliedert:

– Konfigurierung von Anwendungen

– Konfigurierung von Partnersystemen

– Konfigurierung und Administration von Leitungen und Anschlüssen

Die Optionen zur Konfigurierung von Partnersystemen orientieren sich an den Objektklassen, die für die Forwarding Support Base (FSB) definiert sind. Jeder Eintrag in diesem Bereich entspricht einem Objekt der FSB. Beachten Sie in diesem Zusammenhang den Abschnitt „Solaris-Kommunikationsprodukte“ auf Seite 21 und den Abschnitt „Betrieb des Produkts“ auf Seite 56 in diesem Handbuch.

TSAs - Transportsystem-Anwendung verwalten

Über die Menüauswahl *Transportsystem Anwendungen* können Sie lokale und ferne TS-Anwendungen verwalten.

Über *TS-Anwendung Löschen* können Sie ausgewählte Einträge für TS-Anwendungen aus dem TS-Directory löschen.

Über *Eigenschaften anzeigen...* können Sie sich die Eigenschaften von TS-Anwendungen anzeigen lassen. Dabei ist festzulegen, ob der GLOBALE NAME „genau“ oder „mindestens“ die angegenen Namensteile enthalten soll.

In letzterem Fall werden auch TS-Anwendungen selektiert, die weitere Namensteile auf tieferen Ebenen der Namenshierarchie besitzen. Siehe hierzu Abschnitt „Adress-Information im GLOBALEN NAMEN“ auf Seite 39. Sie können als Platzhalter * oder ? eingeben.

Nachdem Sie das Formular ausgefüllt haben, werden die entsprechenden TS-Anwendungen mit ihren Eigenschaften in einem Textfenster angezeigt. Diese Information kann auch in eine Datei geschrieben werden.

Bei der Option GLOBALE NAMEN *anzeigen...* gelten die gleichen Regeln wie beim Anzeigen von Eigenschaften. Die Namen der TS-Anwendungen, die dem angegebenen Muster entsprechen, werden in einem Textfenster angezeigt.

Mit der Option LOKALEN NAMEN *zuordnen...* vergeben Sie an Ihre gewünschte lokale TS-Anwendung einen Namen.

Mit der Option TRANSPORTADRESSE *zuordnen...* weisen Sie fernen TS-Anwendungen TRANSPORTADRESSEN zu.

TSPs - Transport Service Provider

Über die Menüauswahl *Transport Service Provider* erhalten Sie eine Tabelle mit den vorhandenen TSPs, dem TSP-Typ, der Version sowie deren Status. Wählen Sie einen TSP mit der ENTER-Taste.

Sie können den ausgewählten TSP aktivieren *(Starten)* oder anhalten *(Anhalten)*.

i Bei der Aktion *Anhalten* sollten Sie bedenken, dass alle über den angehaltenen TSP laufenden Kommunikationsvorgänge abgebrochen werden.

Mit der Aktion *Neustarten* können Sie Ihren TSP nach Änderung Ihrer Konfiguration neu aktivieren.

Weiterhin können Sie den automatischen Start des ausgewählten TSP beim Hochfahren des Systems einstellen oder wieder aufheben *(Automatischen Start vorbereiten* bzw. *Automatischen Start aufheben)*.

TSDir - TS-Directories verwalten

Informationen über TS-Anwendungen werden in TS-Directories verwahrt. Jeweils eines dieser Directories kann für die aktuelle Arbeit im Menü verwendet werden. Für TS-Directories sind verschiedene Aktionen möglich, z. B. können sie auf Datenträger gesichert werden. Das Standard-Directory trägt den Namen *DIR1* (siehe hierzu Abschnitt „TS-Directories verwalten“ auf Seite 75). Es wird für den laufenden Betrieb von TS-Anwendungen verwendet.

Über die Menüauswahl *Sicherheitskopie erstellen*... können Sie ein existierendes TS-Directory sichern.

Über die Menüauswahl *Sicherheitskopie einlesen...* können Sie ein früher gesichertes TS-Directory erneut einlesen. Außerdem können Sie eine Textdarstellung der TS-Directory-Einträge im Format *tnsxfrm* (siehe Kapitel „Konfigurieren im Expertenmodus“ auf Seite 69) in ein TS-Directory umwandeln lassen.

Mit der Aktion *DIRn löschen* können Sie ein vorhandenes TS-Directory löschen.

Wenn Sie TS-Directory *DIR1* löschen, können sich CMX-Anwendungen wegen Problemen bei der Abbildung von Namen in Adressen vorzeitig beenden.

Über die Option *DIRn zum aktuellen Directory machen* können Sie für die aktuelle Sitzung auf ein alternatives TS-Directory umschalten.

Über *Vertauschen mit TS-Directory DIR1* wird für die aktuelle Sitzung ein alternatives TS-Directory, zum Beispiel *DIR3*, zum Standard-Directory *DIR1* und gleichzeitig das ursprüngliche TS-Directory *DIR1* zum alternativen TS-Directory *DIR3*.

Beachten Sie, dass Sie laufende Anwendungen durch Vertauschen der Directories abbrechen können.

Mit der Aktion *Detailinformation zeigen* können Sie Informationen über TS-Directories abrufen. Diese enthalten das Datum der letzten Änderung und statistische Angaben über die Namensteile und Eigenschaften im TS-Directory.

Local - Lokalen Rechner anzeigen und ändern

Über die Menüauswahl *Local - Lokalen Rechner anzeigen* erhalten Sie die Netzadressen des eigenen Systems angezeigt (siehe auch Abschnitt „Netzadressen“ auf Seite 46). Das Format der möglichen Adressen ist im Abschnitt „Adress-Komponenten und ihre Formate“ auf Seite 86 beschrieben.

NSAPs - Ferne Rechner eintragen

Über diese Menüauswahl können Sie die Netzadressen für Partnersysteme sowie deren Eigenschaften anlegen, ändern, löschen und sich anzeigen lassen (siehe auch Abschnitt „Netzadressen“ auf Seite 46). Jedem Partnersystem müssen Sie einen symbolischen Namen geben, über den Sie später auf die abgelegte Information zugreifen können.

SNPAROUTES - Routen zu fernen Subnetzanschlüssen festlegen

Über diese Menüauswahl können Sie Routing-Information eintragen, die zum Erreichen der Partnersysteme benötigt wird. Dazu gehört die Subnetz-ID, die die lokalen Subnetzanschlüsse bestimmt, über die das Partnersystem erreichbar ist, sowie die Subnetzadresse des Partnersystems (siehe auch Abschnitt „Subnetzanschlüsse und Routen“ auf Seite 47). Außerdem können Sie Dienstmerkmale („Facilities“) für eine Route konfigurieren. Sie können Routen neu anlegen, ändern, löschen und anzeigen lassen.

SUBNET - Subnetzzugang

Über die Menüauswahl *SUBNET - Subnetzzugang* können Sie für jeden Subnetzanschluss Ihres Systems den Zugangsschutz für ankommende Rufe festlegen. Einzelheiten sind in den Handbüchern „CMX/CCP, ISDN-Kommunikation“ [3] und „CMX/CCP, WAN-Kommunikation“ [4] beschrieben.

FSB - FSB sichern und wiedereinlesen

Über die Menüauswahl *FSB - Forwarding Support Information Base* können Sie die Informationen zu Partnersystemen und zur lokalen Netzadresse, die in der FSB gespeichert sind, sichern und wieder einlesen.

Bevor Sie Ihre Konfiguration ändern, sollten Sie Ihren Datenbestand über die Option *Sicherheitskopie erstellen...* sichern.

Wieder einlesen bedeutet, dass eine FSS-Konfigurationsdatei als neue FSB-Konfiguration eingelesen wird. Eine neue Konfigurationsnummer wird vergeben und diese neue Generierung wird als aktuelle Konfiguration definiert. Die einzu-

lesenden Daten müssen im Textformat (siehe Abschnitt „FSS-Konfigurationsdatei erstellen (Format fsconfig)“ auf Seite 126) vorliegen. Sie werden nach dem Einlesen in ein binäres Format überführt.

CCs - Communications Controller

Zugewiesen und geladen werden die Konfigurationsdateien über die Menüauswahl *CCs - Communications Controller*.

Folgende Aktionen sind möglich:

– Informationen über die CC-Hardware abrufen
– CC laden
– CC entladen
– Konfiguration ändern
– einen Speicherabzug für den CC erstellen
– in den Expertenmodus wechseln, um Diagnose-Informationen zu erhalten

Das CC wird durch einen Buchstaben (W für WAN und ISDN) und eine Zahl identifiziert.

Detailinformation zeigen

Mit der Aktion *Detailinformation zeigen* erhalten Sie Informationen über die CC-Hardware, z. B. Speichergröße, Hard- und Firmware-Version etc.

CC laden

Mit der Aktion *CC laden* wird das zugewiesene Subnetz-Profil geladen und die Konfigurationsdatei in Betrieb genommen.

Vorsicht!

Laufende Kommunikationsvorgänge werden dadurch abgebrochen.

CC entladen

Mit der Aktion *CC entladen* setzen Sie das derzeit geladene Subnetz-Profil auf dem CC außer Betrieb.

Vorsicht!

Laufende Kommunikationsvorgänge werden dadurch abgebrochen.

Konfiguration ändern...

Mit der Aktion *Konfiguration ändern* weisen Sie dem CC ein Subnetz-Profil und eine Konfigurationsdatei zu.

Speicherabzug erstellen

Mit der Aktion *Speicherabzug erstellen* erhalten Sie zunächst ein Formular, in dem Sie Parameter für die Erstellung des Speicherabzuges eingeben können. Ein Speicherabzug muss im Expertenmodus aufbereitet werden (siehe bstv-Kommando *format* in den Handbüchern „CMX/CCP, ISDN-Kommunikation" [3] und „CMX/CCP, WAN-Kommunikation" [4]).

Expertenmodus öffnen

Mit dieser Aktion gelangen Sie in einen interaktiven Modus, in dem Sie Kommandos zur Diagnose eingeben können. Die möglichen Kommandos und ihre Parameter sind vom Subnetz-Profil abhängig und sind in den jeweiligen Handbüchern beschrieben (siehe „CMX/CCP, ISDN-Kommunikation" [3], und „CMX/CCP, WAN-Kommunikation" [4]).

NetIf - Lokale Subnetz-Anschlüsse

Über die Menüauswahl *NetIf - Lokale Subnetzanschlüsse anzeigen* können Sie sich sämtliche an Ihrem System konfigurierten Subnetzanschlüsse anzeigen lassen. Sie erhalten eine Liste aller vorhandenen CC-Identifikationen und die zugehörigen konfigurierten Subnetz-Anschlüsse auf den CCs. Nach Auswahl eines Subnetz-Anschlusses mit der ENTER-Taste können Sie im folgenden Menü den Subnetz-Anschluss aktivieren oder deaktivieren.

CFs - CCP-Konfigurationsdateien

Über diese Menüauswahl können Sie das CCP-spezifische Menü erreichen, mit dem Sie Ihre Subnetz-Profile konfigurieren können. Sie können CCP-Konfigurationsdateien neu anlegen, ändern, löschen, drucken, sichern und wieder einlesen. Wie Sie dabei vorgehen müssen, entnehmen Sie den Handbüchern zu den CCP-Produkten (siehe „CMX/CCP, ISDN-Kommunikation" [3], und „CMX/CCP, WAN-Kommunikation" [4]).

Exit - Sitzung beenden

Mit dieser Auswahl verlassen Sie das CMX-Menü.

6.1.3 Vorgehensweise bei der Konfigurierung

Nach der Installation von CMX (siehe Kapitel „Installation und Inbetriebnahme“ auf Seite 53) haben Sie bereits eine Urkonfiguration auf Ihrem System. Ihre lokalen Netzadressen sind dann dem System bekannt. Bei der Erstkonfigurierung ist folgende Reihenfolge sinnvoll:

– CCP-Konfigurationsdateien erstellen
– Konfigurationsdatei und Subnetz-Profil einem CC zuweisen (Konfiguration ändern)
– CC laden
– Route zu fernen Subnetzanschlüssen festlegen
– ferne Netzadressen eintragen
– TS-Anwendungen eintragen

7 Konfigurieren im Expertenmodus

In diesem Kapitel erhalten Sie eine Darstellung der CMX-Kommandoschnittstelle zum Konfigurieren von TS-Anwendungen, von Netzpartnern und von lokalen Routen. Dabei werden die Kommandos *tnsxcom* zum Konfigurieren im Transport Name Service (TNS) und *fssadm* für den Forwarding Support Service (FSS) beschrieben.

Während Sie mit TNS die TS-Anwendungen und deren TRANSPORTADRESSE verwalten, können Sie mit FSS Netzadressen von Endsystemen sowie Routen zu diesen Systemen administrieren.

In den Konfigurationsdateien der Subnetz-Profile sowie mit Hilfe von Facilities-Einträgen im FSS (Objektklasse FACIL) werden Betriebsparameter für Protokolle der Schichten 1 bis 3a festgelegt, z. B. die Übertragungsgeschwindigkeit, Werte für Überwachungszeiten. Im FSS können solche Facilities bestimmten Routen (Objektklasse SNPAROUTES) fest zugeordnet werden.

Beachten Sie, dass in einer Konfigurationsdatei (KOGS der Communication Controller) für jeden Subnetz-Anschluss festgelegt werden muss, mit welcher Subnetz-ID er angesprochen werden kann. Die Subnetz-ID muss dann beim Konfigurieren der Routen im FSS eingetragen werden.

Wird ein Subnetz-Profil auf einen CC geladen, so wird auch der Inhalt der Konfigurationsdatei mitgegeben. Das auf dem CC ablaufende Subnetz-Profil greift auf den Inhalt dieses Abbildes der Konfigurationsdatei zu.

Eine genaue Beschreibung finden Sie in den Handbüchern „CMX/CCP, WAN-Kommunikation“ [4] und „CMX/CCP, ISDN-Kommunikation“ [3]. Der Abschnitt „Vorgehensweise bei der Konfigurierung“ erläutert die Konfigurierung von TS-Anwendungen im TNS. Der Abschnitt „Konfigurieren mit tnsxcom“ auf Seite 75 dokumentiert die Konfigurierung von Netzadressen und Routen sowie partnerspezifischer Betriebsparameter im FSS.

7.1 Vorgehensweise bei der Konfigurierung

Die folgenden zwei Abschnitte geben eine Übersicht der Arbeitsschritte beim Konfigurieren mit tnsxcom und fssadm an der Kommandoschnittstelle.

7.1.1 TNS: anwendungsspezifische Konfigurierung

Sie können eine TNS-Konfigurationsdatei im Format *tnsxfrm* mit einem beliebigen Editor erstellen und daraus im Kommandomodus oder mit der Option „Sicherheitskopie einlesen“ im CMXCUI ein TS-Directory erstellen. Die Syntax dieser Datei, die als Datenbasis für ein TS-Directory dienen soll, ist im Abschnitt „Syntax der TNS-Konfigurationsdatei“ auf Seite 77 beschrieben. Im Folgenden erhalten Sie eine Übersicht der wichtigsten Schritte beim Konfigurieren im TNS.

- Prüfen Sie die Konfigurationsdatei.

  Mit *tnsxcom -s␣file* wenden Sie eine Syntaxprüfung auf die Datei *file* an (Check-Modus). Der TNS protokolliert mögliche Syntaxfehler. Das TS-Directory wird nicht verändert.

- Erstellen Sie ein neues TS-Directory.

  Mit *tnsxcom -l␣file* füllen Sie ein bisher leeres TS-Directory mit den Einträgen aus der Datei *file* (Lade-Modus).

- Prüfen und aktualisieren Sie das TS-Directory.

  Mit *tnsxcom -S␣file* prüfen und aktualisieren Sie ein TS-Directory (check/update-Modus). Wie bei Option *-s* erfolgt in einem ersten Lauf zuerst die Syntaxprüfung auf die gesamte Datei *file*. Treten in *file* keine Syntaxfehler auf, so aktualisiert *tnsxcom* dann das TS-Directory in einem zweiten Lauf.

- TS-Directory erweitern.

  Mit *tnsxcom -u␣file* fügen Sie einem bestehenden TS-Directory die Einträge aus der Datei *file* hinzu (Update-Modus). Enthält *file* Einträge zu GLOBALEN NAMEN, die bereits im TS-Directory vorhanden sind, so werden die alten Einträge überschrieben bzw. ergänzt.

- Interaktiv administrieren.

  Anstatt die Einträge in einer Datei *file* vorzubereiten und dann an *tnsxcom* zu übergeben, können Sie nach Aufruf von *tnsxcom -i* die Konfigurationsdatei auch interaktiv bearbeiten. Abgesehen vom Erfassen neuer GLOBALER NAMEN sind im interaktiven Betrieb insbesondere folgende Eingaben möglich:

  – Eintrag für TS-Anwendung aus dem TS-Directory löschen

    Sie löschen den gesamten Eintrag einer TS-Anwendung aus dem TS-Directory mit folgendem Kommando:

    ```
    Global_name␣DEL
    ```

Alle Eigenschaften, die der TS-Anwendung *Global_name* zugeordnet sind, und der GLOBALE NAME werden aus dem TS-Directory gelöscht. Die TS-Anwendung ist dem TNS dann nicht mehr bekannt.

– Eigenschaften einer TS-Anwendung anzeigen

  Mit folgendem Kommando können Sie sich z. B. die zuvor erfassten oder geänderten Einträge einer TS-Anwendung zur Kontrolle am Bildschirm anzeigen lassen:

  ```
  Global_name␣DISP
  ```

– TS-Directory wechseln

  Sie können mit Hilfe des folgenden Kommandos in der Datei in ein anderes TS-Directory umschalten:

  ```
  DIR␣n
  ```

  Für *n* ist die Nummer des TS-Directory anzugeben, in das umgeschaltet werden soll.

  Die folgenden Eingabesätze in der Datei beziehen sich dann auf dieses TS-Directory. Es wird solange bearbeitet, bis explizit in ein anderes TS-Directory umgeschaltet oder die Eingabe mit CTRL D beendet wird.

► TS-Directory auswählen.

  TNS unterstützt bis zu 9 verschiedene TS-Directories. Alle oben aufgeführten Aktionen beziehen sich auf das Directory DIR1. Über die Option *-d␣* num (num=1,2...9) können Sie auch andere Directories bearbeiten.

7.1.2 FSS: partnerspezifische Konfigurierung

Die partnerspezifische Konfigurierung bedeutet ein Ändern bzw. Erweitern der FSB-Konfiguration, die bei der Installation von CMX erstellt wurde.

Für die Erstkonfigurierung empfiehlt es sich, eine FSS-Konfigurationsdatei zu erstellen. Die genaue Syntax der Einträge in der Konfigurationsdatei ist in Abschnitt „FSS-Konfigurationsdatei erstellen (Format fsconfig)“ auf Seite 126 beschrieben.

Aus der Konfigurationsdatei wird anschließend eine Forwarding Support Information Base (FSB) erzeugt. Es können mehrere solcher FSB-Konfigurationen erzeugt werden; genau eine FSB-Konfiguration wird vom FSS zu einem Zeitpunkt als FSB verwendet. Sie wird als die aktive FSB-Konfiguration bezeichnet.

Bevor Sie die Konfigurierung durchführen, müssen Sie entscheiden, welche Kommunikationswege Sie für Ihr System realisieren wollen. Sie sollten für Ihr Netz einen Adress-Plan erstellen, damit Sie bei der Konfigurierung die Adressen der betreffenden Systeme und TS-Anwendungen kennen.

Zum Erweitern der FSB-Konfiguration werden die beiden folgenden Vorgehensweisen empfohlen:

1. Konfigurationsdatei erzeugen, editieren und anschließend FSB-Konfiguration erzeugen und aktivieren.
2. Konfigurationseinträge direkt in der aktiven FSB-Konfiguration ändern.

Die beiden Vorgehensweisen sind in den folgenden Abschnitten ausführlich erläutert.

Konfigurationsdatei erzeugen und editieren

Im laufenden Betrieb können Sie mit dem Kommando *fssadm* aus einer bestehenden FSB-Konfiguration eine Konfigurationsdatei erzeugen. Diese Konfigurationsdatei können Sie per Editor bearbeiten und anschließend eine neue FSB-Konfiguration daraus erzeugen. Diese Konfiguration können Sie entweder sofort aktivieren oder erst zum nächsten Systemstart wirksam werden lassen.

Die ausführliche Beschreibung der Objektklassen und Attribute finden Sie im Abschnitt „Konfigurieren mit fssadm“ auf Seite 103.

Für die oben genannte Vorgehensweise führen Sie die folgenden Schritte durch:

- Fragen Sie die existierenden FSB-Konfigurationen ab mit

  ```
  fssadm get FSBGEN
  ```

- Wählen Sie die Nummer der FSB-Konfiguration aus, die Sie ändern möchten, und erzeugen Sie aus dieser FSB-Konfiguration eine Konfigurationsdatei mit

  ```
  fssadm create config-file gen-nr=wert path=wert
  ```

  gen-nr bezeichnet die ausgewählte FSB-Konfiguration; *path* ist der Pfadname, unter der die Konfigurationsdatei abgelegt wird.

- Editieren Sie die Konfigurationsdatei: Tragen Sie die gewünschten Daten ein. Beachten Sie dabei die Syntaxregeln, die als Dateiformat *fsconfig* im Abschnitt „FSS-Konfigurationsdatei erstellen (Format fsconfig)“ auf Seite 126 beschrieben sind.

► Prüfen Sie die Konfigurationsdatei auf Eingabefehler mit dem Kommando:

```
fssadm check config-file path=wert
```

path ist dabei der Pfadname der Konfigurationsdatei. Wenn ein Fehler beim Kommando *fssadm check* auftritt, korrigieren Sie die in */var/opt/SMAWcmx/adm/log/fsin_log* aufgezeichneten Fehler und geben das Kommando nochmals ein.

► Erzeugen Sie eine neue FSB-Konfiguration mit

```
fssadm create FSBGEN gen-nr=wert path=wert
```

gen-nr ist die Nummer der neuen FSB-Konfiguration.

i Wenn die vorgenommenen Änderungen keine Inkonsistenzen zu Daten bewirken, die evtl. in bestimmten Kommunikationskomponenten noch wirksam sind, und keine Gefahr des Abbruchs bestehender Netz- oder Transportverbindungen besteht, kann die FSB-Konfiguration auch sofort in Betrieb genommen werden, und zwar mit dem Kommando:

```
fssadm set FSBGEN gen-nr=wert use=ACTIVE
```

Wenn Sie beispielsweise die lokalen Netzadressen für TRANSDATA NEA und OSI TP0/2 geändert haben, müssen Sie danach diese TSPs neu starten.

► Stellen Sie die neue FSB-Konfiguration mit

```
fssadm set FSBGEN gen-nr=wert use=NEXT-ACTIVE
```

als die beim nächsten FSS- bzw. Systemstart zu aktivierende FSB-Konfiguration ein.

► Starten Sie anschließend den FSS und die geänderten Kommunikationskomponenten neu.

Konfiguration im laufenden Betrieb ändern

Sie können die partnerspezifischen Daten im laufenden Betrieb direkt in die aktive FSB-Konfiguration eintragen. Diese Konfigurationsänderungen sind sofort wirksam. Beachten Sie dabei jedoch Folgendes:

i Durch die Änderung kann eine Information, die für den Aufbau noch bestehender Netz- oder Transportverbindungen bereits verwendet worden ist und in den Protokoll-Entities noch gespeichert ist, zu der neu in der FSB eingetragenen Information inkonsistent sein. Dies kann, abhängig vom betroffenen Objekt und Attribut, entweder dazu führen, dass

eine Netz- oder Transportverbindung nicht aufgebaut wird, oder die neue Information kann für eine gewisse Zeit oder bis zum nächsten Systemstart unberücksichtigt bleiben. Ein Beispiel wäre die lokale Netzadresse für NEA (LOCNSAP-Attribut nea-addr) oder für OSI TP0/2 (LOCNSAP-Attribut osi-addr). Wenn Sie eine dieser Adressen geändert haben, müssen Sie anschließend den entsprechenden TSP neu starten.

- Zum Ändern von konfigurierten Objekten geben Sie ein:

  `fssadm set` *objektklasse attribut*

  Beispiel:

  ```
  fssadm set SNPAROUTES name=route1 subnet=X25-1 dte-addr=23456
  ```

- Zum Erzeugen neuer Objekte geben Sie ein:

  `fssadm create` *objektklasse attribut*

  Beispiel:

  ```
  fssadm create NSAP name=partner1  internet-addr=129.22.11.8 \
     net=INTERNET snpa-list=route1
  ```

- Zum Löschen von konfigurierten Objekten geben Sie ein:

  `fssadm delete` *objektklasse attribut*

  Beispiel:

  ```
  fssadm delete NSAP name=partner1
  ```

  Wenn bei der Ausführung des Kommandos *fssadm* ein Fehler auftritt, so werten Sie die Fehlermeldung aus und setzen das korrigierte Kommando erneut ab. Sobald *fssadm* fehlerfrei ausgeführt wurde, sind auch die eingegebenen Konfigurationsänderungen wirksam.

Die relevanten Objektklassen und Attribute sind im Abschnitt „Übersicht der Objektklassen und ihrer Attribute“ auf Seite 108 zusammengestellt. Im Abschnitt „Konfigurieren mit fssadm“ auf Seite 103 erhalten Sie Detailinformation zum Kommando *fssadm* und zum Dateiformat *fsconfig*.

7.2 Konfigurieren mit tnsxcom

Auf Shell-Ebene erstellt, aktualisiert und liest man TS-Directories mit Hilfe des TNS-Compilers *tnsxcom*. *tnsxcom* übersetzt Eingabesätze, die Sie im Format *tnsxfrm* an den Compiler übergeben, in das Format des TS-Directory und trägt die erzeugten Einträge in das TS-Directory ein. Ebenso liest der *tnsxcom* die Einträge der TS-Directories und übersetzt sie in ein abdruckbares Format. Der *tnsxcom* wird mit dem Kommando *tnsxcom* aufgerufen. Die Syntax von *tnsxcom* ist im Kommandokatalog (Abschnitt „TS-Directory erstellen, aktualisieren, lesen (tnsxcom)“ auf Seite 308) beschrieben.

In diesem Abschnitt wird beschrieben:

– welche Aktionen Sie mit dem *tnsxcom* durchführen können

– wie Sie die Eigenschaften erfassen müssen, die ins TS-Directory aufgenommen werden sollen (Format der Eingaben für den *tnsxcom*)

– in welchem Format die Adressen und T-Selektoren für die verschiedenen Transportsysteme angegeben werden müssen

– wie Sie GLOBALE NAMEN in den Eingabesätzen pauschal erweitern können, z. B. wenn Sie Änderungen an einem Zweig des Namensbaums vornehmen wollen (siehe Abschnitt „Namen mit gleichen höherwertigen Namensteilen“ auf Seite 96).

– wie der *tnsxcom* arbeitet

7.2.1 TS-Directories verwalten

Der TNS unterstützt gleichzeitig bis zu 9 verschiedene TS-Directories mit den Identifikationen 1-9. Die TS-Directories sind im Dateisystem als Dateiverzeichnisse *DIR1, DIR2, ... DIR9* abgelegt. Das TS-Directory *DIR1* ist das Standard-Directory, auf das TS-Anwendungen grundsätzlich zugreifen. Die anderen TS-Directories können Sie zum Beispiel als Sicherungskopien oder als experimentelle Bestände verwenden.

Wie Sie ein TS-Directory erstellen, ändern und Informationen dazu abfragen können, erfahren Sie im Abschnitt „Erfassungsregeln für TNS-Dateien“ auf Seite 95.

Folgende Aktionen können Sie mit dem *tnsxcom* ausführen:

– neue TS-Directories erzeugen.

  Mit dem *tnsxcom* können Sie neue TS-Directories DIR<n> (<n> = 1,...,9) erzeugen. Dazu erstellen Sie mit einem beliebigen Editor eine Datei. In dieser Datei erfassen Sie alle TS-Anwendungen mit ihren Eigenschaften, die in dieses TS-Directory eingetragen werden sollen, im Format *tnsxfrm* des *tnsxcom*. Dann rufen Sie den *tnsxcom* im LADE-Modus auf (*tsnxcom -l datei*). Er erzeugt aus den Sätzen der Datei die Einträge für das TS-Directory und schreibt diese in das zuvor leere TS-Directory. Dieses TS-Directory (DIR<n>) darf vorher nicht existieren. Sie dürfen es insbesondere nicht mit *mkdir* anlegen. Das neue TS-Directory DIR<n> wird implizit vom *tnsxcom* angelegt. Die Dateien des TS-Directory werden von ihm erzeugt.

– TS-Directories aktualisieren.

  Mit dem *tnsxcom* können Sie neue TS-Anwendungen in ein bestehendes TS-Directory aufnehmen bzw. bestehende Einträge zu TS-Anwendungen aus dem TS-Directory löschen. Sie können TS-Anwendungen neue Eigenschaften zuordnen, Eigenschaften ändern und löschen.

  Sie können bei der Aktualisierung eines TS-Directory genauso vorgehen wie bei der Erzeugung eines neuen TS-Directory und die Änderungen in einer Datei erfassen. Sie müssen *tnsxcom* dazu im UPDATE-Modus aufrufen (siehe Abschnitt „Syntax der TNS-Konfigurationsdatei“ auf Seite 77). Sie können ein TS-Directory aber auch aktualisieren, indem Sie dem *tnsxcom* die Änderungen interaktiv über die Standardeingabe übergeben. Dazu rufen Sie *tnsxcom* im INTERAKTIV-Modus auf (siehe Abschnitt „TS-Directory“ auf Seite 99). Das Format der Eingaben ist in beiden Modi gleich. Das Format der Eingaben ist auch unabhängig davon, ob Sie ein neues TS-Directory erstellen oder ein bestehendes aktualisieren wollen. Der *tnsxcom* muss lediglich im entsprechenden Modus aufgerufen werden.

– TS-Directories lesen.

  Ein TS-Directory besteht hauptsächlich aus nicht abdruckbaren Zeichen. Wollen Sie ein TS-Directory lesen, so können Sie es sich vom *tnsxcom* in eine abdruckbare Form aufbereiten und in eine Datei schreiben lassen. Diese Datei können Sie auch wieder als Eingabe für den *tnsxcom* verwenden.

  Mit dieser Funktion können Sie ein TS-Directory eines anderen Rechners auf Ihren Rechner portieren. Sie müssen es an dem fremden Rechner mit dem *tnsxcom* in eine Datei schreiben und diese Datei an Ihrem Rechner einlesen. Hier übersetzen Sie das TS-Directory erneut mit dem *tnsxcom*.

Prüfen Sie vor dem Übersetzen, ob in den Einträgen des TS-Directory an Ihrem Rechner residierende TS-Anwendungen als lokale TS-Anwendungen erfasst sind. Entsprechend müssen die TS-Anwendungen, die an dem fremden Rechner residieren, als ferne TS-Anwendungen erfasst sein.

Das am fremden Rechner erzeugte TS-Directory können Sie auch in ein TS-Directory einfügen, das an Ihrem Rechner bereits existiert. Das TS-Directory hat dann die richtige Struktur für den TNS, auch wenn das Ausgangs-TS-Directory mit einer älteren TNS-Version erstellt wurde.

Welche dieser Aktionen der *tnsxcom* ausführen soll, bestimmen Sie durch die Angabe bestimmter Optionen beim Kommando *tnsxcom*.

7.2.2 Syntax der TNS-Konfigurationsdatei

Alle Einträge, die ins TS-Directory aufgenommen werden sollen, müssen in Form der folgenden Eingabesätze übergeben werden.

```
[Global_name␣]type[␣data-Felder]
```

Global_name, *type* und *data* bezeichnen Felder. Eckige Klammern zeigen an, dass es sich um optionale Felder handelt.

Die Bedeutung der einzelnen Felder innerhalb eines Eingabesatzes wird im Folgenden erläutert.

7.2.2.1 GLOBALER NAME

Im Feld *Global_name* geben Sie den GLOBALEN NAMEN der TS-Anwendung an, dem die in den folgenden Feldern beschriebene Eigenschaft zugeordnet werden soll. Wenn Sie für eine TS-Anwendung mehrere Sätze erfassen, so müssen Sie den GLOBALEN NAMEN nur im ersten Satz angeben. In den folgenden Sätzen können Sie das *name*-Feld leer lassen. Die Sätze müssen jedoch unmittelbar hintereinander stehen. D. h. ist in einem Satz das *name*-Feld leer, so gilt der zuletzt in einem Satz angegebene Wert.

Der GLOBALE NAME besteht aus 1 bis 5 hierarchisch angeordneten Namensteilen *Npi* (i = 1, 2, 3, 4, 5). *Np1* ist der Namensteil der höchsten Hierarchiestufe (TS_COUNTRY), *Np5* der der niedrigsten (TS_PN). Siehe hierzu auch Abschnitt „Adress-Information im GLOBALEN NAMEN“ auf Seite 39.

Groß- und Kleinbuchstaben haben unterschiedliche Bedeutung („case sensitivity"). Enthalten die Namensteile Sonderzeichen (siehe Abschnitt „Zeichen mit Sonderbedeutung“ auf Seite 95), deren Sonderbedeutung eine Mehrdeutigkeit der Syntax verursachen würde, so müssen diese Sonderzeichen mit \ (Gegenschrägstrich) oder durch Apostrophierung entwertet werden.

Im Zweifelsfall sollten Sie jedes Sonderzeichen entwerten. Ist die Entwertung überflüssig, so wird sie ignoriert. Die maximalen Längen für *Npi* sind:

Namensteil Npi	1	2	3	4	5
Länge in byte	2	16	16	10	30

Tabelle 4: Maximale Länge der Namensteile

Die Anordnung der *Npi* in *name* erfolgt mit von links nach rechts ansteigender Hierarchie, wobei die *Npi* durch . (Punkt) getrennt werden (Np5.Np4.Np3.Np2.Np1). Leere *Npi* (i = 1, 2, 3, 4, 5) sind zulässig, das folgende Trennzeichen . (Punkt) muss jedoch angegeben werden. Beispiel:
.Np4..Np2.Np1

Endet ein GLOBALER NAME mit mindestens einem . (Punkt), so ist der GLOBALE NAME absolut, d. h. er ist direkt unter ROOT des Namensbaumes eingeordnet. Endet er dagegen nicht mit . (Punkt), so ist er relativ. Relativ angegebene GLOBALE NAMEN werden um einen Ursprung erweitert, sofern Sie einen Ursprung (Origin) definiert haben (siehe auch Abschnitt „Namen mit gleichen höherwertigen Namensteilen“ auf Seite 96).

Gültige Beispiele für die Angabe des GLOBALEN NAMENS sind:

Np5
: nur Namensteil 5, relativ (eventuell zu ROOT)

Np5.
: nur Namensteil 5, absolut

Np5.Np4
: nur Namensteile 5 und 4, relativ (eventuell zu ROOT)

Np5....Np1
: nur Namensteile 5 und 1, absolut

..Np3
: nur Namensteil 3, relativ (eventuell zu ROOT)

.Np4..Np2.
: nur Namensteile 4 und 2, absolut

7.2.2.2 Typ der Anwendung

Der Eintrag für den Typ bzw. die Eigenschaften der Anwendung hat folgende Syntax:

```
type[␣data-Felder]
```

Der Wert für *type* bestimmt den Typ des Eintrags (auch *Eigenschaft* genannt); d. h. *type* gibt an, ob es sich um einen LOKALEN NAMEN oder eine TRANSPORTADRESSE, eine *Session-* oder eine *Presentation-Komponente* handelt. In den *data*-Feldern ist der Wert der Eigenschaft anzugeben, z. B. die TRANSPORTADRESSE. Die einzelnen *data*-Felder sind durch Zwischenraum voneinander zu trennen.

Mögliche Werte für *type* sind:

- TSEL für Transport-Selektor-Eintrag einer lokalen Anwendung
- TA für TRANSPORTADRESSE einer fernen Anwendung
- PSEL für Presentation Selector
- SSEL für Session Selector

Sätze, mit denen Sie neue Einträge für TS-Anwendungen erzeugen wollen, und Sätze, mit denen Sie bestehende Einträge verändern oder erweitern wollen, haben die gleiche Form. Ist eine TS-Anwendung oder eine Eigenschaft einer TS-Anwendung im TS-Directory noch nicht vorhanden, so wird aus den Angaben des Satzes ein neuer Eintrag erzeugt. Ist bereits ein Eintrag vorhanden, so wird er entsprechend den Angaben in dem Satz geändert.

Sie müssen für jede Eigenschaft, die Sie erfassen wollen, mindestens einen Satz übergeben. Jeder TSEL-Eintrag eines LOKALEN NAMENS muss z. B. in einem eigenen Satz übergeben werden. Jeder Satz entspricht einer logischen Zeile. Ist es notwendig, dass sich ein Satz über mehrere Zeilen erstreckt, so muss das Zeilenende durch \ (Gegenschrägstrich) entwertet werden oder die Angaben müssen in () (runde Klammern) eingeschlossen werden.

Es ist nicht möglich, nur den GLOBALEN NAMEN einer TS-Anwendung ins TS-Directory aufzunehmen, ohne ihm eine Eigenschaft zuzuordnen.

Wie Sie die einzelnen Eigenschaften erfassen, ist im Folgenden beschrieben.

7.2.3 LOKALER NAME

Der LOKALE NAME einer TS-Anwendung besteht aus einem oder mehreren TSEL-Einträgen (einem TSEL-Eintrag pro Transportsystem, über das die TS-Anwendung kommunizieren soll). Für jeden T-Selektor, den Sie der lokalen TS-Anwendung *Global_name* zuordnen wollen, müssen Sie einen Satz übergeben. Der Satz muss wie folgt aufgebaut sein:

```
[Global_name␣]TSEL[␣addrform[␣data-Feld mit T-Selektor]]
```

Für *addrform* sind dieselben Angaben wie beim Erfassen der TRANSPORTADRESSE zulässig. Der im *data*-Feld angegebene Wert wird als T-Selektor für das Transportsystem *addrform* in das TS-Directory aufgenommen. Ist im TS-Directory für dieses Transportsystem bereits ein T-Selektor im LOKALEN NAMEN enthalten, wird dieser mit dem neuen Wert überschrieben. Enthält der LOKALE NAME bisher noch keinen T-Selektor für dieses Transportsystem, so wird dieser T-Selektor zum bisherigen LOKALEN NAMEN hinzugenommen. Ist das *data*-Feld leer, so wird ein bereits vorhandener T-Selektor für das entsprechende Transportsystem aus dem Eintrag im TS-Directory gelöscht. Hierbei erfolgt weder eine Warnung noch eine Fehlermeldung, wenn ein solcher T-Selektor nicht im LOKALEN NAMEN vorhanden ist. Ist der zu löschende T-Selektor die einzige Komponente des LOKALEN NAMENS, so wird die Eigenschaft LOKALER NAME für diese TS-Anwendung gelöscht.

Die beschränkte Länge des LOKALEN NAMENS erlaubt die Aufnahme von höchstens 8 verschiedenen T-Selektoren. Für mehrere Transportsysteme gleichlautende T-Selektoren beanspruchen dabei nur einen der 8 Speicherplätze. Hiervon ausgenommen sind die Transportsysteme mit *addrform* LANINET und EMSNA, d. h. die zu diesen Transportsystemen gehörigen T-Selektoren gelten unabhängig von ihrem Wert immer als von anderen verschieden.

T-Selektoren können in verschiedenen Formaten angegeben werden (siehe folgende Beispiele sowie Abschnitt „Adress-Komponenten und ihre Formate“ auf Seite 86). Ihre Länge ist auf 10 Zeichen beschränkt (im TRANSDATA-Format: 8 Zeichen). Die Sonderzeichen ’ (Apostroph) und \ (Gegenschrägstrich) müssen durch \ (Gegenschrägstrich) entwertet werden, falls sie zum T-Selektor gehören sollen.

Die folgende Tabelle enthält die erlaubten Angaben für T-Selektoren bei den verschiedenen Adress-Formaten. Die Bedeutung der Adress-Formate und T-Selektor-Formate sowie die Darstellungsformate für die T-Selektoren finden Sie ab Seite 84.

Adress-Format	**T-Selektor-Format**
EMSNA	LU-Name, LU-Nummer
LANINET	Portnummer
LOOPSBKA	T-Selektor
RFC1006	T-Selektor
SDLCSBKA	Stationsname
TRSNA	T-Selektor
WANNEA	Stationsname
WANSBKA	T-Selektor
WAN3SBKA	T-Selektor

Tabelle 5: Adress- und T-Selektor-Formate

Beispiel für LOKALE NAMEN

```
Global_name  type addrform T-Selektor
-------------------------------------------------------------------
loopleer     TSEL LOOPSBKA V''          ; leerer T-Selektor
laninet      TSEL LANINET  A'4712'      ; dezimale Portnummer
rfc1006      TSEL RFC1006  A'Cologne'
iso          TSEL WANSBKA  E'wansbka'   ; in EBCDIC abzulegen
x28          TSEL WAN3SBKA A'WAN3'      ; in ASCII abzulegen
wan1         TSEL WANNEA                ; Stationsname löschen
```

7.2.4 TRANSPORTADRESSE

Sätze, mit denen Sie die TRANSPORTADRESSE einer fernen TS-Anwendung erfassen, haben das folgende Format:

```
[Global_name␣]TA[␣addrform[␣data-Felder mit Adress-
Komponenten]]
```

TA ist der Indikator für eine TRANSPORTADRESSE.

Mit dem Adress-Format *addrform* geben Sie den Typ des verwendeten Transportsystems an. Beim Erfassen einer TRANSPORTADRESSE erzeugt TNS stets auch einen Eintrag für das Transportsystem. In den folgenden *data*-Feldern übergeben Sie die Adress-Komponenten.

Wollen Sie eine TRANSPORTADRESSE ändern, die bereits ins TS-Directory eingetragen ist, so geben Sie in den *data*-Feldern die neuen Adress-Komponenten an. Der Eintrag im TS-Directory wird dann mit der neuen TRANSPORTADRESSE überschrieben.

Wollen Sie eine TRANSPORTADRESSE aus dem TS-Directory löschen, so entfernen Sie die Einträge für die Adress-Komponenten. Die TRANSPORTADRESSE wird dann für diese TS-Anwendung aus dem TS-Directory gelöscht.

Im Folgenden sind die erlaubten Angaben für das Adress-Format *addrform* und die zu *addrform* anzugebenden Adress-Komponenten aufgelistet. In eckigen Klammern eingeschlossene Adress-Komponenten sind optional. Die Bedeutung der Adress-Formate und der Adress-Komponenten sowie das Format, in dem Sie die einzelnen Adress-Komponenten übergeben müssen, sind im Abschnitt „Adress-Komponenten und ihre Formate“ auf Seite 86 beschrieben.

addrform	**Adress-Komponenten**
EMSNA	LU-Name, Rechner-/Regionsnummer
LANINET	IP-Adresse oder *HOST* Hostname, Portnummer
LOOPSBKA	T-Selektor
RFC1006	IP-Adresse oder *HOST* Hostname, [*PORT* Portnummer], T-Selektor
SDLCSBKA	Rufnummer, [WAN CC/Leitungskennzeichen]
TRSNA	Sym-Dest-Name, T-Selektor
WANNEA	Stationsname, Rechner-/Regionsnummer, [WAN CC/Leitungskennzeichen]
WANSBKA	OSI-NSAP, T-Selektor [TPI] [TPC] [WAN CC/Leitungskennzeichen]
WAN3SBKA	SNPA-Information [T-Selektor] [WAN CC/Leitungskennz.]

Tabelle 6: Adress-Formate und zugehörige Adress-Komponenten

Beispiel für TRANSPORTADRESSEN

```
Global_name  type addrform  Adress-Komponenten
-------------------------------------------------------------
neate     TA   WANNEA    T'$DIALOG' 1/18 WAN 1:1 2:3
X25      (TA   WANSBKA   X.121 4589004033 ; DTE-Adresse (IDI)
                         A'dtxp-33-01'    ; T-Selektor
                         2/0              ; TPC)
tcp/ip    TA   LANINET   128.0.1.23 A'4711'
rfc1006   TA   RFC1006   HOST D018B016 A'Cologne'
```

7.2.5 Session-Komponente

Die Session-Komponente erweitert die TRANSPORTADRESSE einer fernen TS-Anwendung zu einer SSAP-Adresse bzw. den LOKALEN NAMEN einer lokalen TS-Anwendung um einen S-Selektor. Die SSAP-Adresse ist die Adresse einer TS-Anwendung in der Kommunikationssteuerschicht (Session Layer, Schicht 5 des OSI-Referenzmodells).

Die Session-Komponente einer TS-Anwendung wird in folgendem Satz übergeben:

```
[Global_name␣]SSEL[␣data-Feld mit S-Selektor]
```

Falls das *data*-Feld leer ist, so wird die Session-Komponente aus der Eigenschaft TRANSPORTADRESSE bzw. LOKALER NAME entfernt. Ist bereits ein S-Selektor für diese TS-Anwendung im TS-Directory vorhanden, so wird dieser „alte" Wert durch den angegebenen Wert überschrieben. Es ist nicht notwendig, dass der TS-Anwendung vor dem Eintrag des S-Selektors bereits eine TRANSPORTADRESSE bzw. ein LOKALER NAME zugeordnet wurde.

Die folgende Tabelle enthält die erlaubten Angaben für den S-Selektor.

S-Selektor	**Bedeutung**
SSEL	SSEL-Eintrag löschen
SSEL V"	Leereintrag für S-Selektor
SSEL *ssel*	ssel-Darstellungsformate: A'string': string von maximal 16 Zeichen; Ablage in ASCII E'string': string von maximal 16 Zeichen; Ablage in EBCDIC X'string': gerade Anzahl von Hexziffern in string; max.32 T'string': nach TRANSDATA-Konventionen

Tabelle 7: Erlaubte Angaben für den S-Selektor

7.2.6 Presentation-Komponente

Die Presentation-Komponente erweitert die TRANSPORTADRESSE bzw. die SSAP-Adresse einer fernen TS-Anwendung zu einer PSAP-Adresse. Bei einer lokalen TS-Anwendung erweitert die Presentation-Komponente den LOKALEN NAMEN um einen P-Selektor.

Die PSAP-Adresse ist die Adresse einer TS-Anwendung in der Darstellungsschicht (Presentation Layer, Schicht 6 des OSI-Referenzmodells).

Die Presentation-Komponente einer TS-Anwendung wird in folgendem Satz übergeben:

```
[name␣]PSEL[␣data-Feld mit P-Selektor]
```

Falls das *data*-Feld leer ist, so wird die Presentation-Komponente aus der Eigenschaft TRANSPORTADRESSE bzw. LOKALER NAME entfernt. Ist bereits eine Presentation-Komponente für die angegebene TS-Anwendung im TS-Directory vorhanden, so wird dieser „alte" Wert durch den angegebenen Wert überschrieben. Es ist nicht notwendig, dass der TS-Anwendung vor dem Eintrag des P-Selektors bereits eine TRANSPORTADRESSE bzw. ein LOKALER NAME zugeordnet wurde. Falls in der TRANSPORTADRESSE bzw. in dem LOKALEN NAMEN noch keine Session-Komponente enthalten ist, so wird für die Session-Komponente beim Eintrag der Presentation-Komponente automatisch der Leereintrag (SSEL V'') erzeugt.

Die folgende Tabelle enthält die erlaubten Angaben für den P-Selektor.

P-Selektor	**Bedeutung**
PSEL	PSEL-Eintrag löschen
PSEL V''	Leereintrag für P-Selektor
PSEL *psel*	psel-Darstellungsformate: A'string': string von maximal 16 Zeichen; Ablage in ASCII E'string': string von maximal 16 Zeichen; Ablage in EBCDIC X'string': gerade Anzahl von Hexziffern in string; max.32 T'string': nach TRANSDATA-Konventionen

Tabelle 8: Erlaubte Angaben für den P-Selektor

7.2.7 Adress-Formate

Beim Erfassen der TRANSPORTADRESSEN und der TSEL-Einträge des LOKALEN NAMENS müssen Sie das Adress-Format *addrform* des Transportsystems übergeben, auf das sich die folgenden Angaben für die Adress-Komponenten bzw. für den T-Selektor beziehen.

In der folgenden Tabelle sind die verschiedenen erlaubten Werte für *addrform*, das zugehörige Transportsystem, der entsprechende CCP-Name im CMX-Menü und alle zugehörigen Adress-Komponenten aufgeführt.

Beachten Sie bitte die Erläuterungen im Abschnitt „Syntax der TNS-Konfigurationsdatei“ auf Seite 77. Dort ist angegeben, welche Adress-Komponenten Sie im Einzelfall angeben müssen.

Im Anschluss an die Tabelle sind die Bedeutung der Adress-Komponenten und ihre Darstellungsformate beschrieben. Die Adress-Komponenten sind in alphabetischer Reihenfolge aufgelistet.

addrform	**Transportsystem für**	**CCP-Profil im Menü**	**Adress-Komponenten**
EMSNA	Kopplung mit TRANSIT über SNA-Backbone	TRANSIT LU0	LU-Name Rechner/Region
LANINET	Rechnerkopplung über TCP/IP	TCP/IP RFC1006	IP-Adresse oder *HOST* Hostname Portnummer
LOOPSBKA	Interprozess-Kommunikation mit CMX	CMX-LOCAL	T-Selektor
RFC1006	Rechnerkopplung über TCP/IP mit RFC1006-Konvergenzprotokoll	TCP/IP RFC1006	IP-Adresse oder *HOST* Hostname [*PORT* Portnummer] T-Selektor
SDLCSBKA	Stationskopplung an SNA-Netze über SDLC	WAN-SDLC	Rufnummer WAN CC/Leitungskennzeichen
TRSNA	Rechnerkopplung über ein transparentes SNA-Netz	TRANSIT LU6.2	Sym-Dest-Name T-Selektor
WANNEA	Rechnerkopplung im WAN und ISDN mit Protokoll NEATE, NEAN	WAN-NEA WAN-NX25 ISDN-NEA ISDN-NX25	Stationsname Rechner/Region WAN CC/Leitungskennzeichen

Tabelle 9: Adress-Formate, Transportsysteme und Adress-Komponenten

addrform	Transportsys-tem für	CCP-Profil im Menü	Adress-Komponenten
WANSBKA	Rechnerkopp-lung im WAN und ISDN mit ISO-Protokollen der Klasse 0 u. 2 (SBKA-Profil)	WAN-CONS ISDN-CONS	SNPA-Information oder OSI-NSAP T-Selektor TPI TPC WAN CC/Leitungskenn-zeichen
WAN3SBKA	heterogene Kopplung im X.25-Netz ohne Transportproto-koll	WAN-X.25 ISDN-X.25	SNPA-Information T-Selektor WAN CC/Leitungskenn-zeichen

Tabelle 9: Adress-Formate, Transportsysteme und Adress-Komponenten

Die Adress-Komponenten zur Beschreibung der Netzadresse eines Endsystems (Ethernet-, IP-Adresse, Hostname, Rechner, Region, Sym-Dest-Name) werden als Zeichenkette aus alphanumerischen Zeichen übergeben.

Die Adress-Komponenten, die die Adresse einer TS-Anwendung innerhalb des eigenen Systems beschreiben (T-Selektor, Stationsname usw.), werden eingeschlossen in Hochkommata als Zeichenkette übergeben. Vor dem T-Selektor ist ein Formatindikator anzugeben (siehe folgender Abschnitt).

7.2.7.1 Adress-Komponenten und ihre Formate

In diesem Abschnitt sind die Adress-Komponenten und ihre Darstellungsformate erläutert, die Sie beim Erfassen von TS-Anwendungen angeben.

Ethernet-Adresse

Geben Sie die Ethernet-Adresse des Endsystems an, in dem sich die ferne TS-Anwendung befindet.

Darstellungsformat:

Geben Sie genau 12 Hexadezimalziffern [0-9,A-F,a-f] an.

IP-Adresse

Geben Sie die IPv4- oder IPv6-Adresse des fernen Endsystems an.

Darstellungsformat:

Im Fall IPv4 geben Sie genau 4 Dezimalzahlen zwischen 0 und 255 an. Diese Zahlen müssen mit dem Sonderzeichen . (Punkt) getrennt werden.

Beispiel: 123.0.3.98

Im Fall IPv6 geben Sie die 128 Bit lange Adresse als bis zu 8 Sedezimalzahlen an, die jeweils 16 Bit umfassen und mit dem Sonderzeichen : (Doppelpunkt) getrennt werden. Benachbarte Sedezimalzahlen mit Wert 0 können einmal pro Adresse mit 2 aufeinanderfolgenden Doppelpunkten (::) abgekürzt werden. Bei IPv6-Adressen, die aus IPv4-Adressen abgeleitet wurden, kann der IPv4-Adressanteil alternativ in der oben definierten IPv4-Schreibweise dargestellt werden.

Beispiele:

fe80::280:17ff:fe28:7b08
::ffff:123.0.3.98

Hostname

Geben Sie den Hostnamen (mit dem Schlüsselwort HOST vorangestellt) des fernen Endsystems an.

Darstellungsformat:

Geben Sie maximal 60 Zeichen im ASCII-Format an.

Beispiele: PGTW1339 oder V116.mch.sni.de

LU-Name

Geben Sie für eine TS-Anwendung in einem SNA-System den VTAM-Applikationsnamen der SNA-Anwendung an. Für eine TS-Anwendung in einem TRANSDATA-Rechner, die über ein SNA-System erreichbar ist, geben Sie den Stationsnamen der TS-Anwendung an.

Darstellungsformat:

Der LU-Name ist als String ('LU-Name') zu übergeben. Es ist nur das TRANSDATA-Format zugelassen, das Sie mit dem Formatindikator T angeben. Der Stationsname darf maximal 8 Zeichen lang sein. Kürzere Eingaben werden automatisch durch Leerzeichen ergänzt. Längere Eingaben werden als Fehler abgewiesen.

LU-Nummer

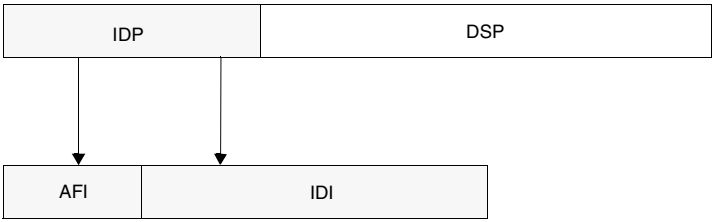
Die LU-Nummer können Sie alternativ zum LU-Namen angeben. Geben Sie die LU-Nummer (Locaddr) an, die in der TRANSIT-Konfiguration für den TRANSIT-Anschluss der TS-Anwendung angegeben ist.

Darstellungsformat:

Geben Sie eine Dezimalzahl zwischen 1 und 255 ein.

OSI-NSAP

Der Aufbau der hier anzugebenden OSI-Netzadressen ist in ISO 8348/Add.2 beschrieben. Er wird hier in Kurzform wiedergegeben:



IDP (Initial Domain Part)

Besteht aus zwei Teilen: dem AFI und dem IDI.

AFI (Authority and Format Identifier)

Die Kennziffer des AFI legt die Struktur und die Länge des IDP fest.

Es werden festgelegt:

– das IDI-Format
– die Institution, die die IDI-Werte festlegt
– welche Füllziffern bei der Codierung des IDP verwendet werden
– die abstrakte Syntax des Domain Specific Part (DSP)

IDI (Initial Domain Identifier)

Der IDI beschreibt den Adressierungsbereich und die Instanz für die Vergabe des DSP.

Es werden festgelegt:

– der Adressierungsbereich, aus dem die DSP-Werte stammen
– die Institution, die für die Vergabe der DSP-Werte in diesem Bereich zuständig ist

DSP (Domain Specific Part)

Der DSP gibt die Möglichkeit einer weiteren Detaillierung an. Die Semantik des DSP wird durch die über den IDI bezeichnete Institution festgelegt. Die abstrakte Syntax wird durch den AFI festgelegt. Die nachste-

hende Tabelle gibt die minimale/maximale Stellenzahl für den IDP (d. h. 2-stelliger AFI und zugehöriger IDI) und den DSP an. Zu beachten ist, dass für die binäre DSP-Syntax prinzipiell nur gerade DSP-Stellenanzahlen erlaubt sind (auch wenn der Maximalwert nicht erreicht wird!).

IDI-Format	AFI	IDP min.	IDP max.	DSP-Syntax	DSP max.
X.121	36	3	16	dec	24 Dezimalziffern
X.121	37	3	16	bin	12 x 2 Hexadezimalziffern
X.121	52	3	16	dec	24 Dezimalziffern
X.121	53	3	16	bin	12 x 2 Hexadezimalziffern
ISO_DCC	38	5	5	dec	35 Dezimalziffern
ISO_DCC	39	5	5	bin	17 x 2 Hexadezimalziffern
F.69	40	3	10	dec	30 Dezimalziffern
F.69	41	3	10	bin	15 x 2 Hexadezimalziffern
F.69	54	3	10	dec	30 Dezimalziffern
F.69	55	3	10	bin	15 x 2 Hexadezimalziffern
E.163	42	3	14	dec	26 Dezimalziffern
E.163	43	3	14	bin	13 x 2 Hexadezimalziffern
E.163	56	3	14	dec	26 Dezimalziffern
E.163	57	3	14	bin	13 x 2 Hexadezimalziffern
E.164	44	3	17	dec	23 Dezimalziffern
E.164	45	3	17	bin	11 x 2 Hexadezimalziffern
E.164	58	3	17	dec	23 Dezimalziffer
E.164	59	3	17	bin	11 x 2 Hexadezimalziffern
ISO_ICD	46	6	6	dec	34 Dezimalziffern
ISO_ICD	47	6	6	bin	17 x 2 Hexadezimalziffern
Local	48	2	2	dec	38 Dezimalziffern
Local	49	2	2	bin	19 x 2 Hexadezimalziffern
Local	50	2	2	bin	19 x 2 Hexadezimalziffern
Local	51	2	2	bin	19 x 2 Hexadezimalziffern

Tabelle 10: Minimale/maximale Stellenzahl für IDP und DSP
bin = binäre DSP-Syntax, dec = dezimale DSP-Syntax

AFI und IDI werden direkt hintereinander eingetragen. IDP und DSP werden durch '+' voneinander getrennt.

Beispiel: 49+1234569876 oder 38123+556678

Portnummer

Geben Sie die TCP-Portnummer für die TS-Anwendung an.

Darstellungsformat:

Geben Sie eine Dezimalzahl zwischen 1 und 32767 an. Die Portnummer müssen Sie bei LANINET als String mit Formatindikator A (ASCII) übergeben.

Beispiel:

A'4712'

Rechner

Geben Sie die Rechnernummer des Kommunikations- oder Verarbeitungsrechners an, in dem die ferne TS-Anwendung residiert.

Darstellungsformat:

Geben Sie bei WANNEA eine Dezimalzahl zwischen 0 und 255 ein, bei EMSNA eine Dezimalzahl zwischen 0 und 31. Die Rechnernummer wird zusammen mit der Regionsnummer in der Form Rechnernummer/Regionsnummer (z. B. 7/16) angegeben.

Region

Geben Sie die Regionsnummer des Kommunikations- oder Verarbeitungsrechners an, in dem die ferne TS-Anwendung residiert.

Darstellungsformat:

Geben Sie eine Dezimalzahl zwischen 0 und 255 ein. Die Regionsnummer wird zusammen mit der Rechnernummer in der Form Rechnernummer/Regionsnummer (z. B. 7/16) angegeben.

Rufnummer

Geben Sie die Rufnummer an, unter der Sie das Partnersystem erreichen.

Darstellungsformat:

Geben Sie maximal 17 Dezimalziffern an.

SNPA-Information

Die Abkürzung SNPA steht für Subnetwork Point of Attachment und bezeichnet den Zugangspunkt zu einem Subnetz. Sie geben hier die Rufnummer, DTE-Adresse oder PVC-Nummer an, über die Sie den Partner erreichen.

Darstellungsformat:

Die SNPA-Information enthält eine Anschluss-Nummer, angeführt von einem subnetzspezifischen Schlüsselwort.

- für Rechnerkopplung im WAN oder ISDN mit ISO-Protokollen der Kl. 0/2 (Adress-Format WANSBKA):

  E.164 <ISDN-Nummer>
  20-stellige ISDN-Nummer

  E.163 <Telephonnummer>
  24-stellige Telephonnummer

  X.121 <IDI>
  17-stellige X.25-DTE-Adresse

  PVC <PVC-Nummer>
  X.25-PVC-Nummer(1 - 4095)

  X.31 <ISDN-Nummer> X.121 <IDI>
  Zweistufenwahl mit 20-stelliger ISDN-Nummer und 17-stelliger X.25-DTE-Adresse

  X.32 <Rufnummer> X.121 <IDI>
  Zweistufenwahl mit 24-stelliger Telefonnummer und 17-stelliger X.25-DTE-Adresse

- für heterogene Rechnerkopplung im X.25-Netz ohne Transportprotokoll (Adress-Format WAN3SBKA):

  E.164 <ISDN-Nummer>
  20-stellige ISDN-Nummer

  X.121 <IDI>
  17-stellige X.25-DTE-Adresse

  PVC <PVC-Nummer>
  X.25-PVC-Nummer (1 - 4095)

  X.31 <ISDN-Nummer> X.121 <IDI>
  Zweistufenwahl mit 20-stelliger ISDN-Nummer und 17-stelliger X.25-DTE-Adresse

X.32 <Rufnummer> X.121 <IDI>
Zweistufenwahl mit 24-stelliger Telefonnummer und 17-stelliger X.25-DTE-Adresse

Beispiele:

```
E.163 08963641625
X.121 123456
PVC 123
```

Stationsname
Geben Sie den Stationsnamen (T-Selektor) aus der NEA-Adresse an. Mit dem Stationsnamen meldet sich die TS-Anwendung im Endsystem, auf dem sie residiert, beim Transportsystem an.

Darstellungsformat:

Der Stationsname ist als String 'Stationsname' mit vorangestelltem Formatindikator zu übergeben. Es sind die Formatindikatoren T, A, E und X zugelassen. Beim T-Selektor des Adress-Formats SDLCSBKA ist jedoch nur der Formatindikator T erlaubt. Der Formatindikator bestimmt das im String anzugebende Format (siehe Abschnitt „Formatindikatoren:“ auf Seite 94).

Der Stationsname darf bei allen Formaten maximal 8 Zeichen lang sein. Das entspricht beim Formatindikator X 16 Hexadezimalziffern.

Kürzere Eingaben werden beim TRANSDATA-Format durch Leerzeichen, ansonsten durch NIL ergänzt. Längere Eingaben werden als Fehler abgewiesen.

Sym-Dest-Name
Geben Sie den Symbolic Destination Name aus der TRANSIT-Konfiguration an. Der Sym-Dest-Name bezeichnet das LU6.2-Programm (LU = Logical Unit) für TRANSIT-LU6.2 auf der Partner-LU.

Darstellungsformat:

Der Sym-Dest-Name ist als Zeichenkette mit genau 8 Zeichen zu übergeben. Die Zeichenkette darf nur Großbuchstaben [A-Z] und Ziffern [0-9] enthalten.

TPI (optional)
Geben Sie bei der TRANSPORTADRESSE mit Adress-Format WANSBKA die Transportprotokoll-Identifikation (TPI) an, wenn diese beim Verbindungsaufbau zu der fernen TS-Anwendung erwartet wird.

CCP-WAN bewertet den TPI beim T-CONNECT.request nicht, sondern entnimmt den Wert in diesem Fall ungeprüft der TRANSPORTADRESSE und trägt ihn in das „call user data"-Feld des Call Request Packet ein.

Darstellungsformat:

Den TPI müssen Sie als String mit Formatindikator X (Hexadezimalformat) übergeben. Der Wert muss eine gerade Anzahl von Hexadezimalziffern enthalten. Maximal dürfen Sie 32 Hexadezimalziffern angeben.

TPC (optional)

Mit der Eingabe der Transportprotokollklasse (TPC) können Sie die Aushandlung der Transportprotokollklasse gemäß ISO 8073 beim T-CONNECT.request steuern. Wenn Sie kein TPC eingeben, gilt der durch die CCP-Konfiguration eingestellte Standardwert (2/2).

Darstellungsformat:

Für TPC können Sie *2/0, 2/2, 0/0* oder *0/-* angeben. Die Werte bedeuten:

- 2/0 bevorzugt Klasse 2, alternativ Klasse 0
- 2/2 bevorzugt Klasse 2, alternativ Klasse 2
- 0/0 bevorzugte Klasse 0, alternative Klasse 0
- 0/- nur Klasse 0, keine Alternative

T-Selektor

Mit dem T-Selektor meldet sich die TS-Anwendung im Endsystem, auf dem sie residiert, beim Transportsystem an.

Darstellungsformat:

Der T-Selektor muss als String 'T-Selektor' mit vorangestelltem Formatindikator T, A, E, X oder V angegeben werden. Der Formatindikator bestimmt das im String anzugebende Format bzw. dessen Codierung. Im Hexadezimalformat (Formatindikator X) ist eine gerade Anzahl von Ziffern anzugeben, maximal 20 (in der TRANSPORTADRESSE maximal 64). Im ASCII-Format (A) und im EBCDIC-Format (E) dürfen Sie maximal 10 (in der TRANSPORTADRESSE 32) Zeichen angeben. Im TRANSDATA-Format (T) dürfen Sie maximal 8 Zeichen angeben.

Geben Sie den Formatindikator V an, so wird die folgende Angabe für den T-Selektor ignoriert. In das TS-Directory wird ein leerer Eintrag für den T-Selektor aufgenommen.

Formatindikatoren:

Die verschiedenen Formatindikatoren haben folgende Bedeutung:

T (TRANSDATA-Format)
: Der T-Selektor wird im TRANSDATA-Format für Stationsnamen angegeben. D. h. der String darf nur aus Großbuchstaben, Ziffern und den Sonderzeichen '$', '#' und '@' bestehen, höchstens 8 Zeichen lang sein und nicht mit einer Ziffer beginnen. Der T-Selektor wird dann intern in EBCDIC.DF.03 (Internationale/Deutsche DF-Version 03) abgelegt und mit Leerzeichen auf 8 Stellen ergänzt.

A (ASCII-Zeichenformat)
: Jedes eingegebene Zeichen wird im ISO-7-Bit-Code abgelegt. Die Zeichenkette darf maximal 10 (in der TRANSPORTADRESSE 32) oder 8 Zeichen lang sein, je nach Wahl des Transportsystems.

E (EBCDIC-Zeichenformat)
: Jedes eingegebene Zeichen wird im EBCDIC-Code EBCDIC.DF.03 (Internationale/Deutsche DF-Version 03) abgelegt. Die Zeichenkette darf maximal 10 (in der TRANSPORTADRESSE 32) bzw. 8 Zeichen lang sein, je nach Wahl des Transportsystems.

X (Hexadezimalformat)
: Der T-Selektor wird als Hexadezimalstring übergeben. Der String muss eine gerade Anzahl von Hexadezimalziffern [0-9,A-F,a-f] enthalten. Je ein Ziffernpaar wird als ein Byte (Zeichen) abgelegt, wobei die 1. Ziffer den Wert der höherwertigen und die 2. Ziffer den der niederwertigen Bits beschreibt. X'3a' entspricht zum Beispiel der Bitdarstellung '0011 1010' (höchstwertiges Bit am weitesten links).

V (Leerformat)
: Mit diesem Formatindikator können Sie einen Leereintrag erzeugen. Der entsprechende T-Selektor existiert dann, hat aber keinen Wert.

  Einen Leereintrag erzeugen Sie durch Angabe von V''. Geben Sie nach V einen nicht leeren String an, dann wird dieser String ignoriert.

WAN CC/Leitungskennzeichen

CCs und Leitungen, die für die Verbindung genutzt werden können.

Darstellungsformat:

Liste von (durch Leerzeichen getrennten) CC-Nummern. Zu jeder CC-Nummer kann optional, durch Doppelpunkt getrennt, eine Liste von Komma-separierten Leitungsnummern angegeben werden. Eine Leitungsnummer kennzeichnet einen Leitungsanschluss auf dem CC.

Zulässig sind die Leitungsnummern 0, 1, 2, 3, 4, 32, 33 und 34 und CC-Nummern von 1 bis 255. Welche Kombinationen sinnvoll sind, hängt von Ihrer Systemkonfiguration ab. Beachten Sie auch die Hinweise in den Handbüchern „CMX/CCP, WAN-Kommunikation“ [4] und „CMX/CCP, ISDN-Kommunikation“ [3] und den Freigabemitteilungen. Die Liste wird von dem Schlüsselwort WAN angeführt.

Beispiel: WAN 1:1,2 2:33

7.2.8 Erfassungsregeln für TNS-Dateien

7.2.8.1 Zeichen mit Sonderbedeutung

Außer dem Leerzeichen haben folgende Zeichen eine Sonderbedeutung:

$ Dollar leitet eine INCLUDE-, ORIGIN- oder VERSION-Anweisung ein.

$ muss entwertet werden, wenn $INCLUDE, $ORIGIN oder $VERSION als GLOBALE NAMEN definiert werden.

; Semikolon leitet einen Kommentar ein, der Rest der laufenden Zeile wird ignoriert.

() Runde Klammern können verwendet werden, um Felder über Zeilengrenzen hinaus zu einem Eingabesatz zusammenzufassen. Insbesondere können damit Felder an beliebiger Position innerhalb des Eingabesatzes mit einem Kommentar versehen werden (ein Kommentar zeigt i.a. das Zeilenende an).

Folgendes Beispiel beschreibt *einen* Eingabesatz zur Angabe der TRANSPORTADRESSE der TS-Anwendung „X.25“.

```
X\.25 (  TA WANSBKA
X.121 45890040033    ; DTE-Adresse
A'dtxp-33-01'        ; T-Selektor
2/0                  ; Transport Protocol Class (TPC)
)
```

\ Gegenschrägstrich dient zur Entwertung der Sonderbedeutung des nachfolgenden Zeichens. Falls das dem \ folgende Zeichen keine Sonderbedeutung hat, wird \ ignoriert.

. Punkt dient zur Trennung von Namensteilen bei der Angabe des GLOBALEN NAMENS.

' Apostroph; innerhalb eines von Apostrophen eingeschlossenen Strings ist die Sonderbedeutung der Zeichen $; ( ) . * @ und des Leerzeichens aufgehoben. Die Zeichen repräsentieren dort sich selbst.

Strings für T-, S- und P-Selektoren sind immer mit Apostrophen einzuschließen.

* Stern ist reserviert für zukünftige Zwecke. Es ist nicht erlaubt, * als einziges Zeichen eines Namensteils des GLOBALEN NAMENS anzugeben. In diesem Fall kann * auch nicht entwertet werden.

@ Kommerzielles AT ist reserviert für zukünftige Zwecke.

Die Sonderbedeutung eines Zeichens wird durch \ (Gegenschrägstrich) oder durch Apostrophierung aufgehoben. Der Zeilentrenner wird ignoriert, falls ihm ein \ vorangeht oder falls er mit ( ) (runden Klammern) entwertet ist.

7.2.8.2 Namen mit gleichen höherwertigen Namensteilen

Wollen Sie Einträge für mehrere TS-Anwendungen erfassen, die zu Blättern an einem Ast des Namensbaums gehören, so müssen Sie die gemeinsamen Namensteile der zugehörigen GLOBALEN NAMEN in den Angaben für *name* nicht laufend wiederholen. Sie können die gemeinsamen Namensteile als *origin* vorgeben. Alle relativ angegebenen Namen in den *name*-Feldern werden dann um . (Punkt) und den Wert von *origin* erweitert. Sie müssen dann in den *name*-Feldern also nur noch die Namensteile angeben, die nicht im *origin* enthalten sind. Der relativ angegebene Wert von *name* muss zusammen mit dem Wert von *origin* einen syntaktisch korrekten GLOBALEN NAMEN bilden.

Ein GLOBALER NAME im Feld *name* ist relativ zu einem *origin*, wenn er *nicht* mit . (Punkt) endet. Endet er mit einem . (Punkt), so ist er absolut (relativ zu ROOT). Ein absolut angegebener GLOBALER NAME wird nicht erweitert, auch wenn Sie einen *origin* vorgeben. Ist kein *origin* vorgegeben, so ist jede Angabe für einen GLOBALEN NAMEN absolut (relativ zu ROOT).

Beispiel

```
Feldinhalt       Wert                resultierender
von name         von origin          GLOBALER NAME
-------------------------------------------------
myappl           myhost.sttz.Mch-P.D
myappl .myhost.  sttz.Mch-P.D
np5.np4          .np2                np5.np4..np2
```

Den Wert für *origin* können Sie wie folgt durch eine Steuerzeile in der Eingabedatei festlegen:

$ORIGIN[␣origin]

Für *origin* geben Sie den Ursprung an, um den alle relativen Namensangaben erweitert werden sollen. *$ORIGIN␣origin* muss der einzige Inhalt des entsprechenden Satzes sein. Wird für *origin* kein Wert angegeben, so werden die GLOBALEN NAMEN der folgenden Eingabesätze nicht erweitert. Eine $ORIGIN-Anweisung ändert die Festlegung für *origin* bei der Kommando-Eingabe im Format *tnsxfrm*.

Die Ursprungsdefinition mit $ORIGIN gilt nur bis zur nächsten $ORIGIN-Anweisung oder bis zum Ende dieser Datei.

7.2.8.3 Eingabedateien verschachteln

Sie können Ihre Eingaben für den TNS in mehrere Dateien aufteilen (z. B. zur produktspezifischen Trennung der TNS-Einträge).

Durch $INCLUDE-Anweisungen können Sie die Dateien ineinander verschachteln. Eine $INCLUDE-Anweisung innerhalb einer Eingabedatei wird durch den Inhalt der angegebenen Datei ersetzt.

Eine $INCLUDE-Anweisung ist ein Satz mit dem einzigen Inhalt:

$INCLUDE␣file

Für *file* ist der Name der Datei anzugeben, die eingefügt werden soll. *file* muss aus Einträgen im Format *tnsxfrm* bestehen. Die Datei *file* darf weitere $INCLUDE-Anweisungen enthalten. Diese dürfen aber keine direkte oder indirekte Rekursion auslösen. Maximal können 10 $INCLUDE-Anweisungen geschachtelt werden.

Die Einstellung des Ursprungs ($ORIGIN-Anweisung) wird an die untergeordnete INCLUDE-Stufe weiter vererbt. Bei Rückkehr in die übergeordnete INCLUDE-Stufe wird der ursprüngliche Wert für den Ursprung dieser Stufe wieder eingestellt.

7.2.8.4 Versionsangabe für Format und Syntax

Eine $VERSION-Anweisung ist ein Satz mit dem einzigen Inhalt:

$VERSION␣version

Für *version* ist die Versionsnummer 6.0 anzugeben.

7.2.8.5 Migration

Dieser Abschnitt ist nur von Interesse, wenn Sie Ihre TNS-Konfiguration von einem Reliant-UNIX-Rechner mit älteren CMX-Versionen in den Solaris-Rechner integrieren wollen.

Die Syntax des Formats *tnsxfrm* hat sich von CMX V3.0 auf V4.0 in einigen Adress-Formaten geändert, von CMX 4.0 auf 5.x gab es jedoch keine inkompatiblen Änderungen. TNS bietet für Dateien, die unter CMX V3.0 erstellt wurden, eine Migration in das Format von CMX V5.x. Diese Migration wird automatisch angestoßen. TNS erkennt die Version des Dateiformats anhand des $VERSION-Satzes (siehe Abschnitt „Versionsangabe für Format und Syntax“ oben). Nachfolgend die Gegenüberstellung der betroffenen Adress-Formate in der CMX V3.0- und der CMX V5.x-Syntax. Von CMX V4.0 zu CMX V5.x ist keine Migration nötig.

Wenn Sie mit RFC1006 arbeiten, beachten Sie auch die Hinweise im Kapitel „Verbindungen über RFC1006 konfigurieren“ auf Seite 207.

```
3.0: TA ISDNSBKA idi tsel [tpi] [tpc]
5.0: TA WANSBKA E.164 idi tsel [tpi] [tpc]

3.0: TA WANSBKA idi tsel [tpi] [tpc]
5.0: TA WANSBKA X.121 idi tsel [tpi] [tpc]

3.0: TA WAN3SBKA idi
5.0: TA WAN3SBKA X.121 idi

3.0: TA ISDNSBKA LNR line tsel [tpi] [tpc] CC Wijk
5.0: TA WANSBKA tsel [tpi] [tpc] WAN i:line j:line k:line

3.0: TA WANSBKA PVC pvcNumber LNR line tsel [tpi] [tpc] CC Wijk
5.0: TA WANSBKA PVC pvcNumber tsel [tpi] [tpc] WAN i:line j:line
k:line

3.0: TA WANSBKA LNR line tsel [tpi] [tpc] CC Wijk
5.0: TA WANSBKA tsel [tpi] [tpc] WAN i:line j:line k:line

3.0: TA WAN3SBKA PVC pvcNumber LNR line CC Wijk
```

```
5.0: TA WAN3SBKA PVC pvcNumber WAN i:line j:line k:line

3.0: (MSA ISDNSBKA idi1
     TA WANSBKA idi2 tsel)
5.0: TA WANSBKA X.31 idi1 X.121 idi2 tsel
```

7.2.9 TS-Directory

Mit dem Kommando *tnsxcom* können Sie Dateien des Formates *tnsxfrm* in TS-Directories überführen. Dabei können Sie verschiedene Modi einstellen für Funktionen wie Syntaxprüfung, Aktualisierung oder Neuerstellung des TS-Directories. Das Kommando hat folgende Syntax (verkürzte Darstellung, weitere Details und Optionen finden Sie auf Seite 308):

tnsxcom [␣**-d**␣num] [␣modus] [␣file ...]

Die Optionen haben folgende Bedeutung:

-d␣num
: Nummer des TS-Directory, das bearbeitet werden soll. Sie können die Nummern 1 bis 9 angeben. Ohne Angabe wird 1 (entspricht DIR1) eingestellt.

modus
: Für *modus* sind folgende Angaben möglich:

  -l LOAD-Modus

  tnsxcom nimmt die Einträge einzeln aus der Datei *file* und füllt das (bisher leere) TS-Directory mit den syntaktisch korrekten Einträgen.

  -s CHECK-Modus

  tnsxcom wendet nur die Syntaxprüfung auf die Datei *file* an und protokolliert mögliche Syntaxfehler. Das TS-Directory wird nicht verändert.

  -S CHECK_UPD-Modus

  Wie bei Option *s* erfolgt in einem ersten Lauf zuerst die Syntaxprüfung auf die gesamte Datei *file*. Treten in *file* keine Syntaxfehler auf, so aktualisiert *tnsxcom* dann das TS-Directory in einem zweiten Lauf.

-u UPDATE-Modus

tnsxcom nimmt die Einträge einzeln aus der editierbaren Datei *file* und mischt die syntaktisch richtigen Einträge in das TS-Directory durch Erfassung bisher nicht vorhandener oder Aktualisierung existierender Einträge (Option *u* ist der Standardwert von *option*).

-i INTERAKTIV-Modus

tnsxcom liest Einträge im Format *tnsxfrm* von *stdin*, nachdem er durch Ausgabe eines Promptzeichens seine Eingabebereitschaft angezeigt hat, und mischt sie in das TS-Directory. Bisher im TS-Directory noch nicht vorhandene Einträge werden eingefügt, existierende Einträge werden aktualisiert.

file ...

Name der Datei mit Einträgen im Format *tnsxfrm*, die im Fall *option* = *l*, *s*, *S* oder *u* von *tnsxcom* ausgewertet werden soll. Es können mehrere Dateien angegeben werden.

Im Fall *option* = *d* ist der Name der Datei anzugeben, in die *tnsxcom* den Inhalt des TS-Directory aufbereiten soll.

Beispiel

Der folgende Aufruf überführt die Einträge aus der Datei *input.dir* in das bisher leere TS-Directory 2:

```
tnsxcom -d 2 -l input.dir
```

7.2.9.1 Eintrag für TS-Anwendung aus dem TS-Directory löschen

Wollen Sie den gesamten Eintrag einer TS-Anwendung aus dem TS-Directory löschen, so übergeben Sie dem *tnsxcom* folgenden Satz.

```
[name␣]DEL
```

Der *tnsxcom* löscht dann alle Eigenschaften, die der TS-Anwendung *name* zugeordnet sind, und den GLOBALEN NAMEN aus dem TS-Directory.

Die TS-Anwendung ist dem TNS dann nicht mehr bekannt. Siehe hierzu auch Kommando *tnsxdel* im Abschnitt „TNS-Einträge löschen (tnsxdel)“ auf Seite 312.

7.2.9.2 Eigenschaften einer TS-Anwendung anzeigen

Wenn Sie Ihre Eingaben für den *tnsxcom* interaktiv vornehmen, dann können Sie sich die aktuell im TS-Directory eingetragenen Eigenschaften einer TS-Anwendung am Bildschirm anzeigen lassen. Dazu übergeben Sie einen Satz mit folgendem Format:

```
[name␣]DISP
```

Damit können Sie sich z. B. die zuvor erfassten oder geänderten Einträge einer TS-Anwendung zur Kontrolle anzeigen lassen.

Geben Sie einen Satz mit dem obigen Format in einer Datei an, die Sie dann an den *tnsxcom* übergeben, so wird beim Compilieren eine Warnung ausgegeben, und der Eintrag wird vom *tnsxcom* ignoriert.

7.2.9.3 Angabe des TS-Directory

Sie können mit Hilfe eines Eingabesatzes in der Datei in ein anderes TS-Directory umschalten. Der Satz muss folgendes Format haben:

```
DIR␣n
```

Für *n* ist die Nummer des TS-Directory anzugeben, in das umgeschaltet werden soll.

Die folgenden Sätze beziehen sich dann auf dieses TS-Directory. Es wird solange bearbeitet, bis explizit in ein anderes TS-Directory umgeschaltet oder die Eingabe beendet wird.

7.2.9.4 Beispiel für tnsxcom-Einträge

Folgende Beispieldatei soll die Syntax von *tnsxcom* verdeutlichen:

```
; RFC1006-Transportadresse einer über IP-Adresse 10.25.1.27
; erreichbaren Anwendung mit T-Selektor im TRANSDATA-Format
;
; name   type data
rfcanw01 TA   RFC1006 10.25.1.27 PORT 102 T'RFCANW01'
;
; Der NEA-Partner $DIALOG im Rechner 1/18,
;
;name   type data
wan1anw TA   WANNEA T'$DIALOG' 1/18
;
; Zwei Anwendungen, die über Interprozeßkommunikation
; miteinander kommunizieren
```

```
;
;name   type data
ipclok  TA   LOOPSBKA A'IPC-LOK'
        TSEL LOOPSBKA A'IPC-LOK'
ipcrem  TA   LOOPSBKA A'IPC-REM'
        TSEL LOOPSBKA A'IPC-REM'
;
;Transportadresse eines WAN-Partners, der über OSI-
;Transportprotokoll und über DTE-Adresse 123456 erreicht wird.
;
;name   type data
wananw01 TA   WANSBKA X.121 123456 A'ANW01'
```

7.2.9.5 Sonderfälle bei TNS-Einträgen

Zweistufenwahl (X.25-Zugang über ISDN- oder Telefonnetz)

Der TNS-Eintrag enthält für den abgehenden Ruf die X.25-Adresse und die (ISDN- bzw. Telefon-) Rufnummer. Ankommend wird oft nur die X.25-Adresse übermittelt. Für Anwendungen, die den rufenden Partner über einen TNS-Aufruf identifizieren, muss daher ein zweiter (Dummy-) Eintrag erstellt werden.

Beispiel

Eintrag für abgehenden Ruf mit Telefonnummer:

```
tel_out TA WANSBKA X.32 23456 X.121 65432 A'remote_appl'
```

Eintrag für abgehenden Ruf mit ISDN-Nummer:

```
isdn_out TA WANSBKA X.31 23456 X.121 65432 A'remote_appl'
```

Eintrag für ankommende Rufe:

```
tel/isdn_in TA WANSBKA X.121 65432 A'remote_appl'
```

7.3 Konfigurieren mit fssadm

Im vorliegenden Abschnitt wird die Konfigurierung des Forwarding Support Service (FSS) über die Kommandoschnittstelle (CLI) beschrieben. Zum Erzeugen von FSS-Objekten können Sie auch das zeichenorientierte Menü CMXCUI benutzen (siehe Abschnitt „Übersicht der zeichenorientierten Bedienoberfläche CMXCUI“ auf Seite 59).

Der *fssadm*-Kommandomodus stellt einen Expertenmodus dar und sollte nur mit entsprechenden Kenntnissen genutzt werden. Lesen Sie zum Verständnis in jedem Fall die Hinweise zum FSS-Adressierungskonzept im Abschnitt „Adressierung von Partnersystemen im FSS“ auf Seite 45. Hinweise zum methodischen Vorgehen beim Konfigurieren finden Sie im Abschnitt „Vorgehensweise bei der Konfigurierung“ auf Seite 69.

In einer FSS-Konfiguration werden Daten in Form von Objekten abgelegt (z. B. Routen, Netzadressen, Betriebsparameter, siehe Abschnitt „Adressierung von Partnersystemen im FSS“ auf Seite 45), denen bestimmte Attribute zugeordnet sind.

Die Beschreibung einer Konfiguration wird als FSB-Konfiguration in der Datenbasis des FSS, der Forwarding Support Information Base (FSB), abgelegt.

Konkrete Einträge in die Datenbasis werden erzeugt, indem Objekte zu den vorgegebenen Objektklassen angelegt oder geändert werden. Jeder Objektklasse ist eine Reihe von Attributen zugeordnet. Beim Erzeugen oder Ändern von Objekten müssen diese Attribute mit den aktuellen Werten versorgt werden.

Neben der Erzeugung von Objekten mit dem *fssadm*-Kommando haben Sie die Möglichkeit, eine Konfiguration als Datei im Format *fsconfig* zu erzeugen (siehe Abschnitt „FSS-Konfigurationsdatei erstellen (Format fsconfig)“ auf Seite 126). Eine bestehende Konfiguration können Sie mit dem *fssadm*-Kommando verändern (siehe hierzu Abschnitt „Vorgehensweise bei der Konfigurierung“ auf Seite 69).

Aktionen

Sie können mit dem *fssadm*-Kommando verschiedene Aktionen ausführen:

– mit *create* erzeugen Sie ein Objekt.

Beispiel:

```
fssadm create GNSAP name=NEA_REG12 nea-addr-pattern=*/12 \
   snpa-list=route1
```

– mit *get* lassen Sie sich ein Objekt anzeigen. Wenn Sie Attributwerte angeben, wählt *fssadm* nur Objekte mit diesen Attributwerten zur Anzeige aus.

Beispiel:

```
fssadm get NSAP nea-addr=1/18
```

– mit *set* verändern Sie ein konfiguriertes Objekt. Dabei ist das Objekt entweder von sich aus eindeutig oder es wird durch die angegebenen Attributwerte syntaktisch eindeutig bestimmt.

Beispiel:

```
fssadm set NSAP name=BS2000-2 nea-addr=2/14 net=NEA \
   snpa-list=route2
```

– mit *delete* löschen Sie ein Objekt. Es muss durch die angegebenen Attribute identifizierbar sein.

Beispiel:

```
fssadm delete NSAP name=NEAHOST1
```

Die Angabe von mehreren Attributen, von denen eines ein Objekt bereits eindeutig identifiziert, wird von *fssadm* abgelehnt.

– mit *check* überprüfen Sie die Gültigkeit der FSB-Konfiguration oder der FSS-Konfigurationsdatei.

Objektklassen

Aus der folgenden Tabelle ist ersichtlich, welche Aktionen im Kommando *fssadm* auf eine bestimmte Objektklasse angewendet werden können und welche Attribute der Objektklasse zugeordnet sind.

Objektklasse	Aktionen	Attribute
FSBGEN	create | delete | set | get | check	gen-nr, path, id, version, date-time, print, use, replace
config-file	create | check	gen-nr, path,
LOCNSAP	set | get	gen-nr, name, nea-addr, osi-addr, internet-addr
NSAP	create | delete | set | get	gen-nr, name, nea-addr, osi-addr, internet-addr, net, r6-impl, r6-tpdusize, r6-drtpdu, r6-aktpdu, access, hop-nsap, snpa-list, type, subnet
GNSAP	create | delete | set | get	gen-nr, name, nea-addr-pattern, net, access, snpa-list, type, subnet
FACIL	create | delete | set | get	gen-nr, name, short-id, facil, admit, npid, compress, ppp-accm, ppp-profile, ppp-auth-params, ppp-auth-protocol, t70-profile, isdn-cug, isdn-throughput, isdn-ra, isdn-partner-prot, x25-octet-string, x25-throughput, x25-window-size, x25-packet-size, x25-cug, x25-cug-oa, x25-bcug, x25-revch, x25-transit-delay, x25-rpoa-selection, x25-fast-select, x25-nui, x31min-svc-to-Bchan, x25-description, fr-prio, fr-cir, fr-cbs, fr-ebs, fr-encaps, fr-max-transit-delay, in-max-idle, out-max-idle
SUBNET	create | delete | set | get	subnet,incoming-call, facil, osi-nsap-address
SNPAROUTES	create | delete | set | get	gen-nr, name, short-id, type, facil, subnet, dte-addr, pvc-nr, dial-nr, line-nr, isdn-nr, nailed-up-isdn, phone-nr, x31-dte-addr, x32-phone-nr, x31-pvc-nr, x31-msa, fr-pvc

Tabelle 11: Aktionen, Objektklassen und Attribute

Objektklasse	Aktionen	Attribute
PPPAUTH	create | delete | set | get	gen-nr, name, short-id, loc-id, peer-id, pap-loc-pwd, pap-peer-pwd, chap-loc-secret, chap-peer-secret
logging-params	set | get	sent-records, got-records, fssd-kbytes, fssadm-kbytes
statistics	set | get	date-time, seconds, searches, hits, compares, stores
trace	set | get	level

Tabelle 11: Aktionen, Objektklassen und Attribute

Attribute

Bei den Attributen ist zwischen Pflicht- und optionalen Attributen zu unterscheiden. Insgesamt können folgende Gruppen von Attributen gebildet werden:

1. Pflichtangabe bei *fssadm create* und beim Editieren einer fsconfig-Datei.
2. Optionale Angabe bei *fssadm create* und beim Editieren einer *fsconfig*-Datei.
3. Optionale Angabe in *fsconfig*-Datei
4. Optionale Angabe bei *fssadm create*
5. Attribut innerhalb einer Gruppe, aus der mindestens eines angegeben werden muss.
6. Attribut innerhalb einer Gruppe von Attributen, die sich gegenseitig ausschließen.
7. Attribut innerhalb einer Gruppe von Attributen, die sich gegenseitig teilweise ausschließen.
8. Redundantes Attribut, das zwar von *fssadm create* und *fsconfig* akzeptiert wird, jedoch weggelassen werden kann, weil es aus anderen Attributen abzuleiten ist. Ein solches Attribut kann als Filter beim *get*-Kommando nützlich sein.
9. Attribut, das nur als Filter beim *get*-Kommando angegeben werden kann.
10. Attribut, das bei der Ausgabe auf ein *get*-Kommando angezeigt wird, aber nicht eingegeben werden kann.
11. Attribut, das bei Objekterzeugung automatisch vergeben wird und nur mit den Kommandos *delete, set* und *get* eingegeben werden kann.

In den folgenden Tabellen ab Seite 108 sind die Attribute jeweils mit den entsprechenden Ziffern gekennzeichnet.

Hilfe zur fssadm-Syntax

Die im Folgenden beschriebene Hilfefunktion bezieht sich nur auf die Syntax von *fssadm*. Es ist möglich, dass eine von der Hilfe angebotene Syntax nach der semantischen Prüfung abgelehnt wird. Ebenso kann der Fall eintreten, dass syntaktisch korrekte Werte oder Attributkombinationen keinen Sinn ergeben, z. B. weil eine auswertende Funktion nicht installiert oder nicht freigegeben ist.

Sie erhalten Informationen zur Syntax des *fssadm*-Kommandos mit folgenden Kommando-Eingaben:

– *fssadm ?* gibt eine allgemeine Beschreibung der Syntax von *fssadm* und Hinweise zur Hilfe-Funktion aus.

– *fssadm aktion ?* gibt aus, für welche Objektklasse eine Aktion möglich ist.

– *fssadm aktion objektklasse [[attributname=] attributwert ...] ?* vervollständigt das Kommando um die Attribute, die zum angegebenen Kontext passen. Dabei gilt die Einschränkung, dass der Kontext nur für diejenigen Attribute berücksichtigt ist, die dem Kontext in der Ausgabe folgen. Beispiel: Die Eingabe *fssadm create snparoutes type=isdn-nc ?* liefert zurück:

```
fssadm create SNPAROUTES <name> [<subnet>] type=ISDN-NC \
{<remsnpa> <nailed-up-isdn> } (min=0,max=1) [<facil>]
```

– *fssadm aktion objektklasse [[attributname=] attributwert ...] attributname= ?* gibt die Syntax des angegebenen Attributs im angegebenen Kontext aus. Vom angegebenen Kontext werden nur solche Attribute berücksichtigt, die dem gefragten Attribut vorausgehen. Beispiel: Die Eingabe *fssadm create snparoutes subnet=isdn-1 type=?* liefert zurück:

```
<type>: ISDN | ISDN-[NC] | X31-M[SA] | X31-S[VC] | X31-P[VC]
```

Bei der Eingabe des Fragezeichens (?) ist die Sonderbedeutung für die Shell zu beachten. Das Zeichen muss ggf. durch Gegenschrägstrich (\) entwertet werden.

7.3.1 Übersicht der Objektklassen und ihrer Attribute

Die folgenden Tabellen enthalten Übersichten zu den konfigurierbaren Parametern für die verschiedenen Objektklassen und beschreiben ihre Bedeutung. Die Objektklassen sind nach vier Kategorien geordnet:

- Übergeordnete Objektklassen, die die FSB-Konfiguration bzw. die FSS-Konfigurationsdatei betreffen. Zu dieser Gruppe gehören: config-file, FSBGEN
- Objektklassen, die für jede Konfiguration relevant sind. Dazu gehören: FACIL, SNPAROUTES, LOCNSAP, NSAP, GNSAP, SUBNET
- Objektklasse, die nur bei Verwendung von CS-ROUTE von Bedeutung ist. Dazu gehört: PPPAUTH
- Objektklassen, die zu Wartungs- und Diagnosezwecke dienen. Dazu gehören: logging-params, statistics, trace

Objektklasse config-file

Ein Objekt der Klasse *config-file* bezeichnet eine FSS-Konfigurationsdatei. Zur Bedeutung siehe Abschnitt „FSS-Konfigurationsdatei erstellen (Format fsconfig)“ auf Seite 126.

Die Nummern nach den Attributnamen weisen auf Informationen hin, die Sie im Abschnitt „Attribute“ auf Seite 106 finden.

Attribut	Format	Bedeutung
gen-nr 2)	Dezimalzahl zwischen 1 und 9999	Nummer der FSB-Konfiguration, aus der eine Konfigurationsdatei erzeugt werden soll.
path 1)	max. 63 Zeichen	Pfadname der Konfigurationsdatei. Pflichtattribut bei *check*.

Tabelle 12: Aktionen und Attribute der Objektklasse config-file

Objektklasse FSBGEN: FSB-Konfiguration

Objekte der Klasse *FSBGEN* bezeichnen eine FSB-Konfiguration, die aus einer FSS-Konfigurationsdatei erstellt wurde.

Attribut	**Format**	**Bedeutung**
id	Zeichenkette (max. 64 Zeichen) *	Identifikationstext. Ausgabe bei *get*. Eingabe nur im *fsconfig*-Format möglich.
gen-nr 4)	Dezimalzahl zwischen 1 und 9999 bzw. *NEXT-GEN-NR*	Nummer der FSB-Konfiguration. Pflichtattribut bei *set* und *check*. Die gleichzeitige Angabe von *gen-nr* und *use* ist bei *get* nicht erlaubt. **
path	max. 63 Zeichen	Pfadname der Konfigurationsdatei (nur bei *fssadm create*).
replace	YES | NO	Nur bei *create*: gibt an, ob eine evtl. bereits vorhandene FSB-Konfiguration durch die neue FSB-Konfiguration ersetzt werden soll oder nicht; auch die aktive FSB-Konfiguration kann ersetzt werden. Optional, Voreinstellung: NO
use	ACTIVE | NEXT-ACTIVE	Bei *set* Pflicht; bei *get* optional. Die gleichzeitige Angabe von *gen-nr* und *use* ist bei *get* nicht erlaubt. ***
version 10)	6-stellige Hexadezimal-Zeichenkette	Version der FSB-Konfiguration (Ausgabe bei *get*)
date-time 10)	Monat Tag hh:mm:ss Jahr	Datum und Uhrzeit der Erzeugung (Ausgabe bei *get*).
print	MINIMUM | VERBOSE	Umfang der Ausgabe des Kommandos *fssadm check FSBGEN* (optionales Attribut).

Tabelle 13: Attribute der Objektklasse FSBGEN

* Falls der Text Leerzeichen enthalten soll, muss die Zeichenkette durch doppelte Anführungszeichen (“) eingeschlossen werden.

** NEXT-GEN-NR bedeutet „nächste noch nicht belegte Nummer“. Dieser Wert ist nur bei *create* erlaubt und gilt dort als Voreinstellung.

*** use=ACTIVE bedeutet: der FSS befindet sich im Aktiv-Zustand und verwendet diese FSS-Konfiguration, solange dieser Zustand anhält.

use=NEXT-ACTIVE bedeutet: der FSS wird diese Konfiguration verwenden, wenn er das nächste Mal aktiviert wird.

Objektklasse FACIL: Dienstmerkmale (Facilities)

Jeder Route (Objektklasse SNPAROUTES) können Sie bestimmte Merkmale zuordnen (z. B. Gebührenumkehr, Durchsatzrate). Diese Merkmale werden in einem Objekt FACIL („Facilities“) definiert.

Die Attribute isdn-* und x25-* können nicht mit den Attributen fr-* kombiniert werden.

Die Nummern nach den Attributnamen weisen auf Informationen hin, die Sie im Abschnitt „Attribute“ auf Seite 106 finden.

Attribut	**Format**	**Bedeutung**
gen-nr 9)	Dezimalzahl zwischen 1 und 9999	Nummer der FSB-Konfiguration
name 1)	1-15 Zeichen: Buchstaben, Ziffern, die Sonderzeichen ’_’ und ’#’. Groß- und Kleinbuchstaben werden unterschieden. Das 1. Zeichen darf keine Ziffer sein und kein Unterstrich (_) sein.	Name des FACIL-Objekts.
short-id 11)	Dezimalzahl zwischen 1 und 9999	Implizit vergebene Kurzbezeichnung
facil 2)	siehe *name*	Name eines weiteren FACIL-Objekts
admit 2)	BOTH_IN_AND_OUT | OUTGOING ONLY | INCOMING_ONLY | NEITHER_IN_NOR_OUT	Zugangsschutz auf Subnetzebene

Tabelle 14: Attribute der Objektklasse FACIL

Attribut	Format	Bedeutung
npid 2)	OSI-CONS | INTERNET | NEA | SNA/FR | FAX2/3 | PRIVATE	Netzprotokoll-Kennung. Das Attribut ist unwirksam, wenn ppp-profile= STANDARD eingestellt ist.
compress 2)	TCP/IP | NO	Van-Jacobsen-Header-Compression
ppp-accm 2)	ALL_CNTRL_CHARS | NO_MAPPING | *	PPP mit asynchronem Verfahren über ISDN
ppp-profile 2)	STANDARD | NO | GSM	Nutzung des Point-to-Point-Protokolls.
ppp-auth-params 2)	siehe *name*	Name eines PPPAUTH-Objekts.
ppp-auth-protocol 2)	NO | PAP | CHAP	Authentifizierungsprotokoll
t70-profile 2)	YES | NO	Nutzung der Protokollvariante T.70 des Profils CCP-WAN-CONS
isdn-cug 2), 7)	Dezimalzahl zwischen 0 und 65535	Geschlossene Benutzergruppe
isdn-throughput 2), 7)	9,6 | 64 | 128	Durchsatz
isdn-ra 2), 7)	X30/V110-SYN | V110-ASYN	Anpassung der Übertragungsrate
isdn-partner-prot 2), 7)	1TR6/TYPE1 | 1TR6/TYPE1A | SIMPLE	Anpassung der ISDN-Signalisierung
fr-encaps 2), 7)	YES | NO	Protokoll-Encapsulation gemäß RFC1490
fr-cir 2), 7)	0 bis 2048 KBit pro Sekunde	Committed Information Rate

Tabelle 14: Attribute der Objektklasse FACIL

Attribut	Format	Bedeutung
fr-cbs 2), 7)	0 bis 2048 KBit	Committed Burst Size
fr-ebs 2), 7)	0 bis 2048 KBit	Exceeded Burst Size
fr-prio 2), 7)	1 | 2 | 3 (1 = höchste Priorität)	Priorität
fr-max-transit-delay 2), 7)	1-65535 Zehntelsekunden	Maximale Übertragungsdauer
x25-octet-string 2), 7)	1-109 Oktette im Hex-Format	DTE-Facilities gemäß CCITT X.25 Annex G (IS8208)
x25-packet-size 2), 7)	Senderichtung[/Empfangsrichtung] mit den Werten für S/E: 16 | 32 | 64 |128 | 256 | 512 | 1024 | 2048. Ohne Angabe von E gilt E=S.	Paketgröße
x25-window-size 2), 7)	Senderichtung[/Empfangsrichtung] mit den Werten für S/E: 1-127	Fenstergröße
x25-throughput 2), 7)	Senderichtung[/Empfangsrichtung] mit den Werten für S/E in KBit/s: 2,4 | 4,8 | 9,6 | 19,2 | 48 | 64	Durchsatzklasse
x25-cug 2), 7)	0-9999. Führende Nullen werden bewertet: 1-2-stellige Eingabe bedeutet 'basic format', 3-4-stellige Eingabe bedeutet 'extended format'.	Auswahl einer geschlossenen Benutzergruppe
x25-cug-oa 2), 7)	0-9999. Siehe x25-cug.	Auswahl einer geschlossenen Benutzergruppe mit uneingeschränktem abgehendem Ruf
x25-bcug 2), 7)	0-9999. Führende Nullen werden nicht bewertet; es gilt immer das 'extended format'.	Auswahl einer bilateral geschlossenen Benutzergruppe

Tabelle 14: Attribute der Objektklasse FACIL

Attribut	Format	Bedeutung
x25-revch 2), 7)	BOTH_REQ_AND_ACC | REQUEST_ONLY | ACCEPT_ONLY | NEITHER_REQ_NOR_ACC	Gebührenüber-nahme anfordern bzw. Anforderung der Gebührenüber-nahme annehmen
x25-transit-delay 2), 7)	0-65534 Millisekunden	Gewünschte Über-tragungszeit
x25-fast-select 2), 7)	NO-RESTRICTION | RESTRICTION	Fast Select (Kurzdia-log durch Verwen-dung des Call User Data-Feldes)
x25-rpoa 2), 7)	DNIC[+DNIC...] mit maximal 12 Elementen	Auswahl einer Route über einen (oder mehrere) private Netzbetreiber, die durch ihren DNIC (Data Network Identi-fication Code) identi-fiziert werden
x25-nui 2), 7)	max. 16 abdruckbare Zeichen (ASCII, EBCDIC) oder max. 16 Hexadezimalziffernpaare: Format: *formind:nui-wert* formind = A | E | X	Network User Identi-fication
x31min-svc-to-Bchan	n-TO-EACH | MAX_TO_EACH | MAX-TO-ONLY-ONE n-TO-EACH: n={1...127}	Belegung der B-Kanäle zu einem ISDN-Partner durch SVC. Nur bei X.25-Minimumintegration oder DTE-DTE-Kopplung
x25-description 2), 7)	Name eines XZSTW-Makros (TRANSDATA-Konventionen)	Auswahl einer vorde-finierten Beschrei-bung des X.25-Zugangs

Tabelle 14: Attribute der Objektklasse FACIL

* Mit dem Parameter *ppp-accm* legen Sie fest, welche Steuerzeichen beim Verbindungsaufbau über Mobilfunk transparent übertragen werden sollen. Neben den beiden angegebenen Werten (ALL_CNTRL_CHARS und NO_MAPPING) können Sie einzelne Steuerzeichen als Kürzel oder als Hexadezimalstring eingeben. Die genaue Syntax erhalten Sie mit dem Hilfe-Kommando *fssadm create facil ppp-accm=?*

Die Steuerzeichen finden Sie in der folgenden Tabelle:

Steuerzeichen	**Hex**	**Bedeutung**
NUL	00	Keine Operation
SOH	01	Vorspannanfang
STX	02	Textanfang
ETX	03	Textende
EOT	04	Übertragungsende
ENQ	05	Stationsanruf
ACK	06	Bestätigung
BEL	07	Klingel
BS	08	Korrekturtaste
HT	09	Tabulatorzeichen
LF	0A	Zeilenvorschub, neue Zeile
VT	0B	Vertikaler Tabulator
FF	0C	Formularvorschub
CR	0D	Wagenrücklauf
SO	0E	Umschalten Zeichensatz
SI	0F	Zurückschalten Zeichensatz
DLE	10	Austritt aus Datenverbindung
DC1	11	Gerätestrg. 1, Ausgabe starten (XON)
DC2	12	Gerätestrg. 2
DC3	13	Gerätestrg. 3, Ausgabe anhalten (XOFF)
DC4	14	Gerätestrg. 4
NAK	15	Fehlermeldung

Tabelle 15: Steuerzeichen bei asynchronem PPP

Steuerzeichen	Hex	Bedeutung
SYN	16	Synchronisierung
ETB	17	Datenblockende
CAN	18	ungültig, Zeilenlöscher
EM	19	Datenträgerende, quit (Signal3)
SUB	1A	Zeichen ersetzen
ESC	1B	Rücksprung
FS	1C	Dateitrennung
GS	1D	Gruppentrennung
RS	1E	Satztrennung
US	1F	Einheitentrennung
SP	20	Leerzeichen

Tabelle 15: Steuerzeichen bei asynchronem PPP

Objektklasse LOCNSAP: Lokale NSAP-Adressen

Mit der Objektklasse LOCNSAP ist die Adresse des eigenen Systems definiert.

Attribut	Format	Bedeutung
gen-nr 9)	Dezimalzahl zwischen 1 und 9999	Nummer der FSB-Konfiguration.
name 1)	1-32 abdruckbare sichtbare Zeichen	Name des LOCNSAP-Objekts
nea-addr 5)	*p/r* mit Dezimalzahlen *p* und *r*(0 ... 255)	NEA-Adresse: Prozessor/ Regionsnummer
osi-addr 5)	gemäß ISO 8348 Ad 2, siehe Seite 88	OSI-NSAP-Adresse

Tabelle 16: Attribute der Objektklasse LOCNSAP

Attribut	Format	Bedeutung
internet-addr 5)	kanonische Darstellung einer IPv4- oder IPv6-Adresse, siehe Seite 15	IP-Adresse, die bei den Adress-Abbildungsfunktionen von CS-GATE das lokale System repräsentiert. Wenn keine spezielle Address-Abbildungsfunktion mit einem bestimmten IP-Interface konfiguriert werden soll, geben Sie hier 0.0.0.0 angeben.

Tabelle 16: Attribute der Objektklasse LOCNSAP

Objektklasse NSAP: Ferner NSAP oder ferne Netz-Entity

Jedes Endsystem oder Übergangssystem, für das Transportverbindungen aufgebaut werden sollen, wird durch ein NSAP-Objekt repräsentiert.

Die Nummern nach den Attributnamen weisen auf Informationen hin, die Sie im Abschnitt „Attribute“ auf Seite 106 finden.

Attribut	Format	Bedeutung
gen-nr 9)	Dezimalzahl zwischen 1 und 9999	Nummer der FSB-Konfiguration
name 1)	1-32 abdruckbare sichtbare Zeichen	Name des NSAP-Objekts
nea-addr 5)	*p/r* mit *p* und *r* (0 ... 255)	NEA-Adresse: Prozessor-/Regionsnummer
osi-addr 5)	gemäß ISO 8348 Ad 2, siehe Seite 88	OSI-NSAP-Adresse
internet-addr 5)	kanonische Darstellung einer IPv4- oder IPv6-Adresse	IPv4- oder IPv6-Adresse eines fernen NSAP
net 1) oder 8)	NEA | INTERNET | OSI-CONS	Netz, das vom lokalen System zum Erreichen des NSAP verwendet wird.
access 8)	DIRECT | DYNAMIC | HOP | NSAP-ADDR	Zugriff auf die SNPA-Adresse, über die der NSAP erreicht werden kann.

Tabelle 17: Attribute der Objektklasse NSAP

Attribut	Format	Bedeutung
snpa-list 6) und bedingt 1)	*snpa+snpa+..+snpa* mit max. 20 Listenelementen. *snpa:* name | name/weight *name*: siehe Attribut *name* bei SNPAROUTES *weight:* Ziffer von 1-20. *	Liste der SNPAROUTES-Objekte, die zum Erreichen dieses NSAP alternativ benutzt werden können. Priorität kann mit einem Wert für *weight* angegeben werden (20 ist die höchste Priorität)..
hop-nsap 6) und bedingt 1)	siehe *name*	Name des NSAP-Objekts, dessen snpa-list verwendet werden soll.

Tabelle 17: Attribute der Objektklasse NSAP

* Zur besseren Übersichtlichkeit dürfen vor oder hinter dem Zeichen „+“ Leerzeichen und Neue-Zeile-Steuerzeichen stehen. Dazu muss der gesamte Ausdruck in doppelte Anführungszeichen (“) eingeschlossen werden.

Beispiel

```
snpa-list=“Route1 + Route2“
```

Zusätzliche Filterattribute bei *get NSAP*

Die folgende Tabelle enthält weitere Attribute, die nur bei get als Filter verwendet werden können.

Die Nummern nach den Attributnamen weisen auf Informationen hin, die Sie im Abschnitt „Attribute“ auf Seite 106 finden.

Attribut	Format	Bedeutung
subnet 9)	X25-*n* | X21-*n* | PT-*n* | FR-1...128 | PP-*n* | ISDN-*n* *n*= 1, .., 32	Subnetz-ID
type 9)	X25 | PVC | X21 | PP | ISDN | X21DIRECT | X31-MSA | PT| X31-SVC | X31-PVC | X32-PTMSA | FR | ISDN-NC	SNPA-Adress-Typ

Tabelle 18: Zusätzliche Filterkriterien bei get NSAP

Es werden nur diejenigen NSAP-Objekte ausgegeben, denen Routen vom angegebenen SNPA-Adress-Typ bzw. mit der angegebenen Subnetz-ID zugeordnet sind. In der Anzeige des Attributs *snpa-list* sind nur die Routen aufgeführt, die den Filterkriterien entsprechen. Falls noch weitere Routen zugeordnet sind, die den Filterkriterien nicht genügen, werden diese summarisch durch „+*“ angezeigt.

Objektklasse SUBNET: Lokaler Subnetzanschluss

Ein Objekt der Klasse SUBNET repräsentiert einen lokalen Subnetzanschluss, der eindeutig durch eine Subnetz-ID identifiziert wird oder eine Gruppe gleichartiger lokaler Subnetzanschlüsse, die durch die diesen Anschlüssen gemeinsame Subnetz-ID (Attribut subnet) identifiziert wird.

Dem Objekt werden Werte zugeordnet, die für die Einstellung der X.25-Minimumintegration bei Anrufen unbekannter ISDN-Partner bzw. der X.32-Wahl bei Telefonanrufen sowie beim Aktivieren und Deaktivieren des Zugangsschutzes nötig sind.

Attribut	**Format**	**Bedeutung**
subnet	X25-*n* | X21-*n* | PT-*n* | ISDN-*n* *n*= 1, .., 32	Subnetz-ID
incoming-call	NONE | RESTRICTED | ALL	Schalter zum Ein- und Ausschalten des Zugangsschutzes
x25-description	Name eines XZSTW-Makros (TRANSDATA-Konventionen)	Auswahl einer vordefinierten Beschreibung des X.25-Zugangs
facil	siehe *name*	Name eines weiteren FACIL-Objekts
osi-nsap-address	gemäß ISO 8348 Ad 2, siehe auch Seite 86	OSI-NSAP-Adresse

Tabelle 19: Attribute der Objektklasse SUBNET

Objektklasse SNPAROUTES: Route

Mit einem SNPAROUTES-Objekt konfigurieren Sie eine Route innerhalb eines Subnetzes. Diese wird durch ihren Anfangs- und ihren Endpunkt definiert. Ausgangspunkt der Route ist ein lokaler Subnetzanschluss, Endpunkt ist der Subnetzanschluss des fernen Systems. Dabei können lokal mehrere Anschlüsse

zusammengefasst werden, wenn sie in dasselbe Subnetz führen. Der Anfangspunkt der Route wird dann durch eine Subnetz-ID definiert, unter der die gewünschten Subnetzanschlüsse zusammengefasst werden.

Die verschiedenen Subnetzadressen werden folgendermaßen den Subnetz-IDs zugeordnet:

SNPA-Adress-Typ	**Subnetz-ID**
X25 | PVC	X25-x
X21 | X21DIRECT	X21-x
PP (Punkt-zu-Punkt)	PP-x
PT (Public Telephone) | X32-PTMSA	PT-x
ISDN | ISDN-NC | X31-PVC | X31-SVC | X31-MSA	ISDN-x
FR	FR-x

Tabelle 20: Zuordnung von Subnetz-ID zum SNPA-Adress-Typ

Bei ISDN-Festverbindungen kann die Subnetz-ID sowohl einem Anschluss als auch einem einzelnen Kanal (B- oder D-Kanal) zugeordnet werden.

Die Nummern nach den Attributnamen weisen auf Informationen hin, die Sie im Abschnitt „Attribute“ auf Seite 106 finden.

Attribut	**Format**	**Bedeutung**
gen-nr 9)	Dezimalzahl zwischen 1 und 9999	Nummer der FSB-Konfiguration
name 1)	1-15 Zeichen: Buchstaben, Ziffern, die Sonderzeichen ’_’ und ’#’. Groß- und Kleinbuchstaben werden unterschieden. Das 1. Zeichen darf keine Ziffer und kein Unterstrich (_) sein.	Name des SNPAROUTES-Objekts
short-id 11)	Dezimalzahl zwischen 1 und 9999	Implizit vergebene Kurzbezeichnung
subnet 1)	X25-*n* | X21-*n* | PT-*n* | FR-1...128 | PP-*n* | ISDN-*n* *n*= 1, .., 32	Subnetz-ID

Tabelle 21: Attribute der Objektklasse SNPAROUTES

Attribut	Format	Bedeutung
type 8) oder 1)	X25 | PVC | X21 | PP | ISDN | X21DIRECT | X31-MSA | PT| X31-SVC | X31-PVC | FR-PVC | X32-PTMSA | ISDN-NC	SNPA-Adress-Typ
facil 2)	Siehe Attribut *name*	Name eines FACIL-Objekts, auf das verwiesen wird.
dte-addr 5) und 6)	1-17 Dezimalziffern	Adresse der fernen X.25-DTE
pvc-nr 5) und 6)	*pvc/dte* *pvc:*␣Dezimalzahl (0 ... 4095) *dte:*␣1-17 Dezimalziffern	X.25-PVC-Nummer und zugehörige eigene DTE-Adresse.
dial-nr 5) und 6)	*dial-nr* | DIRECT / *dial-nr* *dial-nr:* 1-24 Dezimalziffern oder 1-24 beliebige sichtbare Zeichen eingeschlossen zwischen einfachem Apostroph (')	Ferne X.21-Rufnummer. Im Falle von „Direct Mode“: lokale X.21-Rufnummer.
phone-nr 5) und 6)	1-24 Dezimalziffern oder 1-24 beliebige sichtbare Zeichen eingeschlossen zwischen einfachem Apostroph (')	Telefonnummer
line-nr 2) und 6)	[CC-Nr/] line-nr line-nr: 1 | 2 | 3 | 4 CC-Nr: 1-256	Optionales Attribut: Leitungsnummer für Standleitung (KOGS-Parameter LPUFADR)
x31-dte-addr 5) und 6)	rem-dte-addr[/loc-dte-addr]	X.31 Maximum Integration: Adresse der fernen X.25-DTE und wahlweise die Adresse der lokalen X.25-DTE
x31-msa 5) und 6)	isdn-nr/dte-addr	Zweistufenwahl: ferne ISDN-Rufnummer/ferne X.25-DTE-Adresse

Tabelle 21: Attribute der Objektklasse SNPAROUTES

Attribut	Format	Bedeutung
x32-phone-nr 5) und 6)	phone-nr/x25-dte-addr	X.32 über Telefonnetz
fr-pvc 5) und 6)	CC-Nr/line/pvc CC-Nr: 1-256 line: 0-255 pvc: 1-65535	Frame Relay-PVC
nailed-up-isdn 2) und 6)	CC-Nr./line-nr	Optionales Attribut: ISDN-Festverbindung
isdn-nr 5) und 6)	1-20 Dezimalziffern	ISDN-Wählverbindung
x31-pvc-nr 5) und 6)	pvc-nr[/loc-dte-addr]	X.31 Maximum Integration: PVC-Nummer und wahlweise die Adresse der entsprechenden lokalen X.25-DTE

Tabelle 21: Attribute der Objektklasse SNPAROUTES

Objektklasse GNSAP: Generalisierter NSAP

Ein GNSAP-Objekt repräsentiert eine Gruppe von NEA-Rechnern, deren NEA-Adressen einem bestimmten Muster entsprechen.

Die Nummern nach den Attributnamen weisen auf Informationen hin, die Sie im Abschnitt „Attribute“ auf Seite 106 finden.

Attribut	Format	Bedeutung
gen-nr 9)	Dezimalzahl zwischen 1 und 9999	Nummer der FSB-Konfiguration
name 1)	1-32 abdruckbare sichtbare Zeichen	Name des GNSAP-Objekts
nea-addr-pattern 1)	*/r* mit *r* (0 ... 255 | *)	NEA-Adresse: Prozessor-/Regionsnummer.

Tabelle 22: Attribute der Objektklasse GNSAP

Attribut	Format	Bedeutung
snpa-list 1)	*snpa+snpa+..+snpa* mit max. 20 Listenelementen. *snpa:* name | name/weight *name*: siehe Attribut *name* bei SNPAROUTES *weight:* Ziffer von 1-20. Siehe auch NSAP.	Liste der Routen, die zum Erreichen von NEA-Rechnern benutzt werden können, die durch diesen GNSAP repräsentiert werden. Priorität kann mit einem Wert für *weight* angegeben werden (20 ist die höchste Priorität).

Tabelle 22: Attribute der Objektklasse GNSAP

Bei der Arbeit mit dem *get*-Kommando stehen zusätzlich die Attribute „gen-nr", „type" und „subnet" zur Verfügung (wie bei der Objektklasse NSAP).

Objektklasse PPPAUTH: Lokale Identifikation bei PPP

Diese Objektklasse wird nur für die Konfigurierung von CS-ROUTE im Falle einer Kommunikation über TCP/IP mittels PPP (Point-to-Point-Protokoll) benötigt. Ein PPPAUTH-Objekt enthält Informationen zum Zugangsschutz durch PAP (Password Authentication Protocol) und CHAP (Challenged Handshake Authentication Protocol).

Die Nummern nach den Attributnamen weisen auf Informationen hin, die Sie im Abschnitt „Attribute“ auf Seite 106 finden.

Attribut	Format	Bedeutung
gen-nr 9)	Dezimalzahl zwischen 1 und 9999	Nummer der FSB-Konfiguration.
name 1)	1-15 Zeichen: Buchstaben, Ziffern, die Sonderzeichen '_' und '#'. Groß- und Kleinbuchstaben werden unterschieden. Das 1. Zeichen darf keine Ziffer und kein Unterstrich (_) sein.	Name des PPPAUTH-Objekts.
short-id 11)	Dezimalzahl zwischen 1 und 9999	Implizit vergebene Kurzbezeichnung.

Tabelle 23: Attribute der Objektklasse PPPAUTH

Attribut	Format	Bedeutung
loc-id 2)	1-32 abdruckbare sichtbare Zeichen	Lokale Identifikation.
peer-id 2)	1-32 abdruckbare sichtbare Zeichen	Partner-Identifikation.
pap-loc-pwd 2)	Name einer Datei (max. 63 Zeichen), in der das Passwort (max. 32 Zeichen) im Klartext abgelegt ist.	PAP-Passwort für das lokale System.
pap-peer-pwd 2)	Name einer Datei (max. 63 Zeichen), in der das Passwort (max. 32 Zeichen) im Klartext abgelegt ist.	PAP-Passwort für das Partnersystem.
chap-loc-secret 2)	Name einer Datei (max. 63 Zeichen), in der das CHAP-Secret (max. 255 Zeichen) im Klartext abgelegt ist.	CHAP-Secret für das lokale System.
chap-peer-secret 2)	Name einer Datei (max. 63 Zeichen), in der das CHAP-Secret (max. 255 Zeichen) im Klartext abgelegt ist.	CHAP-Secret für das Partnersystem.

Tabelle 23: Attribute der Objektklasse PPPAUTH

Objektklasse statistics

Die Objektklasse *statistics* dient zur Ausgabe und zum Rücksetzen der Cache-Statistiken des FSS.

Die Nummern nach den Attributnamen weisen auf Informationen hin, die Sie im Abschnitt „Attribute“ auf Seite 106 finden.

Attribut	Format	Bedeutung
date-time 10)	hh:mm:ss	Datum und Zeit des Starts von Statistiken
seconds 10)	Dezimalzahl	Zeitraum, in dem Statistiken gesammelt wurden

Tabelle 24: Attribute der Objektklasse statistics

Attribut	Format	Bedeutung
searches 10)	Dezimalzahl	Anzahl der Suchfragen nach einem Objekt im Cache
hits 10)	Dezimalzahl	Anzahl der Treffer
compares 10)	Dezimalzahl	Anzahl der Vergleiche mit Objekten im Cache, die während einer Suche durchgeführt wurden
stores 10)	Dezimalzahl	Gibt an, wie oft ein Objekt im Cache gespeichert wurde

Tabelle 24: Attribute der Objektklasse statistics

Objektklasse logging-params

Die Objektklasse *logging-params* dient zur Anzeige und zur Änderung der Parameter für das Logging des *fssadm* und des FSS-Dämons. Nur die Aktionen *set* und *get* sind möglich.

Der FSS-Dämon protokolliert abwechselnd in die Dateien *fssd_log.A* und *fssd_log.B*. *fssadm* protokolliert abwechselnd in die Dateien *fssadm_log.A* und *fssadm_log.B*.

Die Nummern nach den Attributnamen weisen auf Informationen hin, die Sie im Abschnitt „Attribute“ auf Seite 106 finden.

Attribut	Format	Bedeutung
sent-records 2)	YES | NO	Logging von Datensätzen, die der FSS-Dämon in den Cache lädt.
got-records 2)	YES | NO	Logging von Datensätzen, die *fssadm* beim *get*-Kommando ausgibt.
fssd-kbytes 2)	1...9999 KByte	Wechsel der Protokolldatei des FSS-Dämons nach ... KBytes.
fssadm-kbytes 2)	1...9999 KByte	Wechsel der Protokolldatei für *fssadm* nach ... KBytes.

Tabelle 25: Attribute der Objektklasse logging-params

FSS-Protokolldateien

fssadm_log.A, fssadm_log.B
enthält folgende Angaben:

- Datum und Zeit.
- eingegebenes Kommando.
- Fehlermeldung, falls das Kommando abgelehnt wurde oder ein Fehler auftrat.
- Falls Sie im Objekt *logging-params* das Attribut *got-records=YES* gesetzt haben, werden zusätzlich alle Datensätze protokolliert, die an *stdout* ausgegeben wurden.

Mit einem Objekt *logging_params* bestimmen Sie die Ausgabe von *fssadm_log*.

fsin_log
enthält Logging-Einträge zu den Kommandos *fssadm check config-file* sowie *fssadm create FSBGEN*. Die Datei enthält Fehlermeldungen und Warnungen mit Zeilenangabe sowie Hinweise auf automatisch erzeugte Objekte.

fsin_acc
enthält alle akzeptierten Einträge der FSS-Konfigurationsdatei im ausgeschriebenen Standardformat.

fssd_log.A, fssd_log.B
enthält Angaben zu Start und Stop des FSS, Dämonen, Kernelspeicherbedarf der FSB-Objekte (siehe auch *logging_params*). Abwechselnd wird in Datei A und B geschrieben.

Die Dateien befinden sich unter */var/opt/SMAWcmx/adm/log*.

7.3.2 FSS-Konfigurationsdatei erstellen (Format fsconfig)

Eine Konfigurationsdatei besteht aus Anweisungen, mit denen Objektklasse und Attribute eines Objektes angegeben bzw. erzeugt werden. Außerdem können $INCLUDE-Anweisungen verwendet werden. Eine $INCLUDE-Anweisung gibt an, dass eine weitere Konfigurationsdatei eingefügt werden soll. Die Verschachtelung von $INCLUDE-Anweisungen ist bis zu einer Tiefe von 10 erlaubt.

Innerhalb oder zwischen den Anweisungen dürfen Kommentare angegeben werden. Ein Kommentar beginnt mit ';' und wird mit einem Dateiendezeichen oder einem Neue-Zeile-Zeichen beendet.

Ein Feld innerhalb einer Anweisung wird durch ein oder mehrere Neue-Zeile-Zeichen, Leerzeichen oder Tabulatorzeichen beendet. Anweisungen können also in Spaltenform angeordnet werden.

Eine Anweisung wird durch ein Neue-Zeile-Zeichen beendet. Ein \ am Ende einer Zeile macht ein Neue-Zeile-Zeichen unwirksam. Sie können auch eine Anweisung in runde Klammern einschließen, falls die Anweisung sehr lang ist oder Kommentare eingefügt werden sollen, und deshalb Neue-Zeile-Zeichen unwirksam gemacht werden müssen. Werden mehr Felder als erforderlich eingegeben, so wird dies als Fehler zurückgewiesen.

Die Grundzüge der Syntax von Objektklassen, Attributnamen und -werten sind wie folgt definiert:

– Die Objektklasse, der Attributname und der symbolische Wert eines Attributs und $INCLUDE können in Groß- oder Kleinbuchstaben angegeben werden.

– Attribute können explizit durch *Attribut-name*=*Wert* angegeben werden.

fsconfig beschreibt die Syntax von Anweisungen in einer Konfigurationsdatei des Forwarding Support Service. Die Konfigurationsdatei können Sie mit einem beliebigen Editor schreiben und daraus eine FSB-Konfiguration erstellen (siehe Abschnitt „Vorgehensweise bei der Konfigurierung“ auf Seite 69).

Ein Eintrag in eine FSS-Konfigurationsdatei hat folgendes Grundformat:

```
Objektklasse [Attributname=]Attributwert  ...
```

Objektklasse bezeichnet die definierten Objektklassen, z. B. *LOCNSAP* und *NSAP*.

Attributname bezeichnet die definierten Attribute zu den Objektklassen. Eine vollständige Beschreibung der Objektklassen und ihrer Attribute finden Sie im Abschnitt „Konfigurieren mit fssadm“ auf Seite 103.

Innerhalb einer FSS-Konfigurationsdatei dürfen Beziehungen nur zu vorausgegangenen Attributen geknüpft werden. Verweise auf nachfolgende Einträge sind nicht erlaubt. Sinnvollerweise sollte daher beim Erstellen der Datei folgende Reihenfolge ingehalten werden:

– FSBGEN
– PPPAUTH
– FACIL
– SNPAROUTES
– LOCNSAP
– NSAP
– GNSAP
– SUBNET

Jede FSS-Konfigurationsdatei muss einen LOCNSAP-Eintrag enthalten (LOCNSAP ist der einzige obligatorische Eintrag in der Datei).

Für folgende Objektklassen wird bei fehlender Angabe ein Default-Eintrag erzeugt:

– FSBGEN mit Attribut *id*,
– tune-nsaps

Sie können mit $INCLUDE-Anweisungen andere Dateien in Ihre Konfigurationsdatei einbinden:

`$INCLUDE` *Pfadname/Dateiname*

Der folgende Abschnitt enthält beispielhafte Einträge für die FSS-Konfigurationsdatei.

Beispiele

NSAP-Adresse des lokalen Systems, das über TRANSDATA NEA kommuniziert:

```
LOCNSAP  ( name=DO18S265 nea-addr=1/18 )
```

Definition eines Facilities-Objekts mit dem Merkmal „Gebührenübernahme nur bei abgehenden X.25-Verbindungen“:

```
FACIL name=charging x25-revch=REQUEST_ONLY
```

Route zum fernen System mit der DTE-Adresse 1930000 über den lokalen Subnetz-Anschluss mit der Identifikation X25-1; Zuordnung des Merkmals „Gebührenübernahme“:

```
SNPAROUTES name=x25_prv subnet=X25-1  dte-addr=1930000 \
    facil=charging
```

Route zum fernen System über Leitungsnummer 4 über lokalen Subnetz-Anschluss mit der Identifikation PP-11:

```
SNPAROUTES name=ddv subnet=PP-11  [ line-nr=4 ]
```

Beschreibung eines NEA-Partners mit der NEA-Adresse 28/5, der über die Route *x25_prv* erreicht wird, die im SNPAROUTES-Objekt festgelegt wurde:

```
NSAP  name=PGTR0039 nea-addr=28/5 net=NEA access=DIRECT \
    snpa-list=x25_prv
```

Beschreibung eines NEA-Partners mit der NEA-Adresse 19/5, der über die Route *ddv* erreicht wird, die im SNPAROUTES-Objekt festgelegt wurde:

```
NSAP  name=D018V019 nea-addr=19/5 \
    net=NEA access=DIRECT snpa-list=ddv
```

7.4 Beispielkonfiguration

Damit Ihre Anwendungen, die das Transportsystem nutzen, miteinander kommunizieren können, müssen Sie CMX und CCP konfigurieren. Dazu müssen Sie die folgenden Fragestellungen klären und entsprechende Maßnahmen ergreifen:

– Welche TS-Anwendungen sollen miteinander kommunizieren?

  Sie benennen TS-Anwendungen, die auf dem lokalen und auf fernen Rechnern ablaufen. Die erforderlichen Einträge machen Sie in der CMX-Komponente TNS.

– Wie erreichen sich diese TS-Anwendungen gegenseitig?

  Sie legen für die möglichen Kommunikationsbeziehungen fest, über welche Betriebsmittel (wie z. B. Leitungen) sie hergestellt werden sollen. Die dazu erforderlichen Informationen erfahren Sie vom Netzbetreiber. Sie tragen diese Angaben in die Konfigurationsdateien der Subnetzprofile ein.

Adressen von Partnersystemen und Routen zu diesen Systemen tragen Sie in die CMX-Komponente FSS ein, wenn es sich um Übergangssysteme handelt oder wenn die Verbindung über TRANSDATA NEA oder TCP/IP über WAN aufgebaut wird. Für andere Profile nehmen Sie die Adress-Einträge in der Regel ausschließlich im TNS vor.

– Welche Leistungsmerkmale müssen für die gewünschten Verbindungen eingestellt werden?

  Sie stellen auf Ihrem Rechner die Betriebsparameter der Subnetz-Profile so ein, dass sich diese mit ihren Partnerinstanzen in anderen Rechnern korrekt verständigen können. Auch diese Angaben erhalten Sie beim Netzbetreiber. Wie Sie bei der Konfiguration der Subnetzprofile vorgehen, ist in den Handbüchern „CMX/CCP, ISDN-Kommunikation“ [3] und „CMX/CCP, WAN-Kommunikation“ [4]) beschrieben. Partnerspezifische Betriebsparameter stellen Sie im FSS ein.

Beim Konfigurieren von CMX sind folglich diese Aufgaben zu erledigen:

– Anwendungen konfigurieren
– Route konfigurieren (über NEA und TCP/IP-Profile über WAN)
– Facilities konfigurieren
– Partnersysteme konfigurieren
– Konfigurationsdateien für WAN- bzw. ISDN-Subnetzprofile erzeugen

Außerdem müssen Sie für alle beteiligten Systeme die WAN-Interfaces für IP konfigurieren.

Als Beispiel soll folgende Konfiguration dienen, bei der Solaris-1 das lokale und Solaris-2 das ferne Solaris-System darstellt. Beide Solaris-Systeme befinden sich jeweils an einem lokalen TCP/IP-Netz. Die beiden LANs sind über X.25-Netz und Routing-Systeme miteinander verbunden. LAGER und EINKAUF sind die Namen der CMX-Anwendungen, die miteinander kommunizieren sollen.

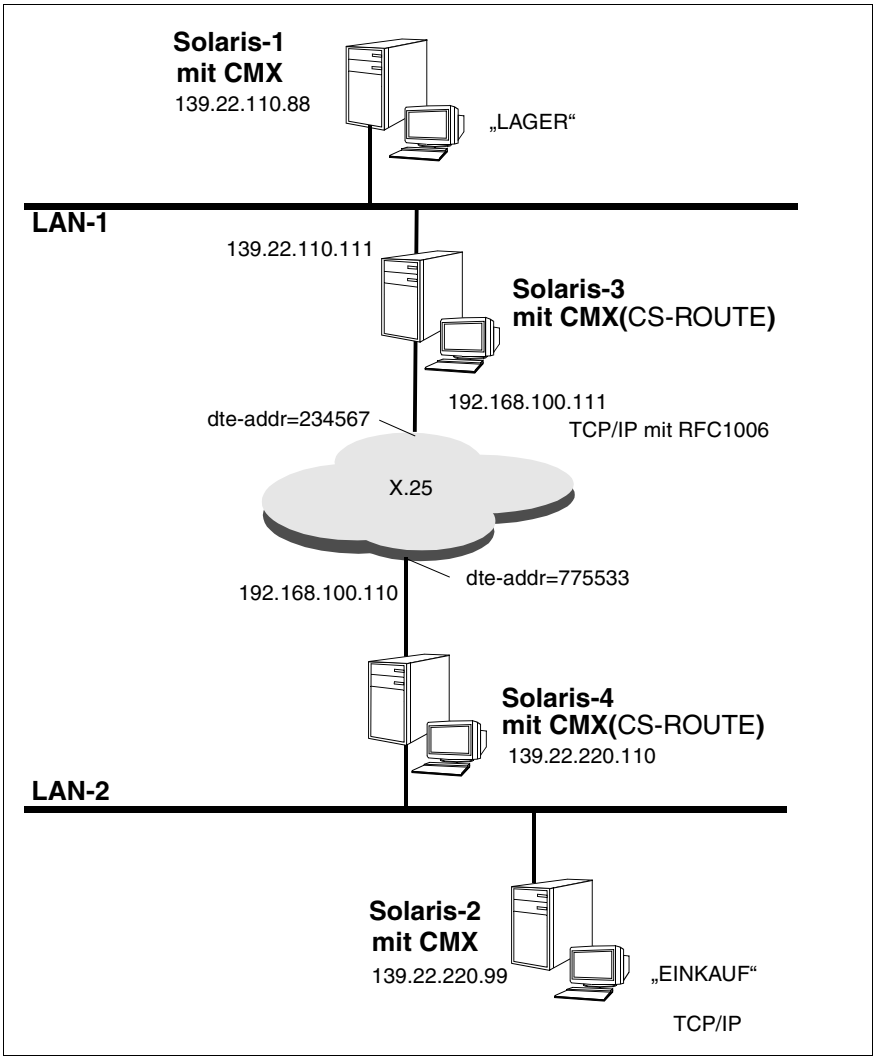


Bild 17: Konfigurationsbeispiel für LAN-Kopplung über X.25

7.4.1 Anwendungen konfigurieren

Die GLOBALEN NAMEN der CMX-Anwendungen sollen folgendermaßen definiert werden:

Namensteil	**1**	**2**	**3**	**4**	**5**
für die CMX-Anwendung im Rechner Solaris-1			Meier&Co	Solaris-1	LAGER
für die CMX-Anwendung im Rechner Solaris-2			Meier&Co	Solaris-2	EINKAUF

Tabelle 26: GLOBALE NAMEN der Beispielanwendungen

Damit die CMX-Anwendung LAGER mit der CMX-Anwendung EINKAUF kommunizieren kann, ist von den Systemverwaltern der Endsysteme Solaris-1 und Solaris-2 folgendermaßen vorzugehen.

Arbeitsschritte am System Solaris-1:

1. Die CMX-Anwendung LAGER muss ins TS-Directory des Systems Solaris-1 eingetragen werden. LAGER residiert im lokalen System Solaris-1, daher ist zum GLOBALEN NAMEN der LOKALE NAME in Solaris-1 anzugeben. Folgenden Eintrag müssen Sie in eine Datei schreiben:

   LAGER TSEL RFC1006 A'LAGER'

   ► Nehmen Sie den erstellten Eintrag mit folgendem Kommando ins TS-Directory auf (Standard ist DIR1):

   `tnsxcom -u` *Dateiname*

   Der Systemverwalter des Systems Solaris-1 muss nun den Systemverwalter des Systems Solaris-2 veranlassen, die CMX-Anwendung LAGER als TS-Anwendung im fernen Endsystem in das TS-Directory des Systems Solaris-2 einzutragen. Der T-Selektor LAGER muss am System Solaris-2 in der TRANSPORTADRESSE von LAGER erscheinen (siehe hierzu Arbeitsschritt 2 für fernes System).

2. Damit eine TS-Anwendung am System Solaris-1 mit der CMX-Anwendung EINKAUF an Solaris-2 Verbindung aufnehmen kann, muss die CMX-Anwendung EINKAUF in das TS-Directory am System Solaris-1 eingetragen werden. EINKAUF residiert aus der Sicht von Solaris-1 im fernen System Solaris-2, daher sind der GLOBALE NAME von EINKAUF, das bei der

Kommunikation zu verwendende Transportsystem (CCP-Profil) und die TRANSPORTADRESSE im System Solaris-2 anzugeben. Folgende Einträge müssen in eine Datei geschrieben werden:

EINKAUF TA RFC1006 139.22.220.99 A'EINKAUF'

- Nehmen Sie den erstellten Eintrag mit folgendem Kommando ins TS-Directory auf (Standard ist DIR1):

  `tnsxcom -u` *Dateiname*

Arbeitsschritte am System Solaris-2:

Im System Solaris-2 muss die CMX-Anwendung EINKAUF als TS-Anwendung im lokalen Endsystem erfasst werden und die CMX-Anwendung LAGER als TS-Anwendung im fernen Endsystem.

1. Die Einträge für EINKAUF sind wie folgt aufzunehmen:

   EINKAUF TSEL RFC1006 A'EINKAUF'

   - Nehmen Sie den erstellten Eintrag mit folgendem Kommando ins TS-Directory auf (Standard ist DIR1):

     `tnsxcom -u` *Dateiname*

2. Die CMX-Anwendung LAGER, die im System Solaris-1 residiert, muss im System Solaris-2 als ferne Anwendung konfiguriert werden. Folgenden Eintrag müssen Sie in eine Datei schreiben:

   LAGER TA RFC1006 139.22.110.88 A'LAGER'

   - Nehmen Sie den erstellten Eintrag mit folgendem Kommando ins TS-Directory auf (Standard ist DIR1):

     `tnsxcom -u` *Dateiname*

Der T-Selektor LAGER muss mit dem T-Selektor des LOKALEN NAMENS in der CMX-Anwendung LAGER im System Solaris-1 übereinstimmen.

7.4.2 Routen konfigurieren

Für die TCP/IP-Kopplung zweier Anwendungen über WAN (hier: X.25) müssen Sie für die beteiligten Übergangssysteme (hier: Solaris-3 und Solaris-4) im FSS Adress-Einträge vornehmen und Routen konfigurieren. Diese Vorgehensweise gilt immer, wenn Routing-Funktionen durch den Routing Service (CS-ROUTE) von CMX realisiert werden.

Um vom System Solaris-1 das ferne System Solaris-2 zu erreichen, müssen Sie in den Übergangssystemen Solaris-3 und Solaris-4 Routen eintragen, um die Erreichbarkeit von Solaris-1 und Solaris-2 sicherzustellen. Der notwendige Eintrag enthält neben dem Namen der Route eine Subnetz-ID für das zu benutzende Subnetz (hier: X25-1), die Sie vergeben müssen, sowie die DTE-Adresse des anderen Übergangssystems.

Arbeitsschritte am System Solaris-3:

- Geben Sie eine Route zum Übergangssystem Solaris-4 an:

```
fssadm create SNPAROUTES name=route4 subnet=X25-1 dte-addr=
775533
```

Arbeitsschritte am System Solaris-4:

- Geben Sie eine Route zum Übergangssystem Solaris-3 an:

```
fssadm create SNPAROUTES name=route3 subnet=X25-1 dte-addr=
234567
```

7.4.3 Facilities einstellen

Jeder Route und jedem fernen Subnetz-Anschluss können bestimmte Merkmale (Facilities) zugeordnet werden. So können Sie z. B. Gebührenumkehr vereinbaren, um die Verbindungskosten vom Partnersystem abrechnen zu lassen. Der Eintrag für dieses Merkmal hat folgende Form:

```
fssadm create FACIL name=charging x25-revch=REQUEST_ONLY
```

Sie ordnen anschließend dieses Merkmal der konfigurierten Route zu:

```
fssadm set SNPAROUTES name=route3 facil=charging
```

Für die andere Route können Sie dieses Merkmal ebenfalls zuordnen oder beliebige andere Facilities definieren.

Beachten Sie dabei, dass Mehrfachzuordnungen von Facilities zu Routen nur im Kommandomodus möglich sind.

7.4.4 Ferne Systeme konfigurieren

Ferne Systeme, die Sie mit dem TCP/IP-Protokoll über WAN erreichen wollen, müssen Sie in den FSS eintragen. Nachdem Sie die Route eingetragen haben, über die das ferne Subnetz erreichbar ist, tragen Sie im FSS die Netzadresse des jeweils anderen Übergangssystems ein.

Arbeitsschritte am System Solaris-3

- Tragen Sie ein NSAP-Objekt ein, das Solaris-4 repräsentiert. Das Objekt enthält die IP-Adresse von Solaris-4 sowie einen Verweis auf die Route, die verwendet werden soll:

```
fssadm create NSAP name=Solaris4 internet-addr=
\192.168.100.110 snpa-list=route4
```

Arbeitsschritte am System Solaris-4

- Tragen Sie analog ein NSAP-Objekt ein, das Solaris-3 repräsentiert:

```
fssadm create NSAP name=Solaris3 internet-addr=
\192.168.100.111 snpa-list=route3
```

7.4.5 WAN-Interfaces für IP konfigurieren

Jeder WAN-Anschluss, der für TCP/IP genutzt werden soll, muss dem Solaris-System bekannt gemacht werden. Sie tragen dazu eine WAN-IP-Adresse ein.

Arbeitsschritte am System Solaris-3

- Das WAN-Interface erhält einen eindeutigen Namen und die zugehörige IP-Adresse:

```
csr create if name=clwip0 ipaddr=192.168.100.111
```

  Vom System wird eine Datei namens *clw.routes.clwip0* eingerichtet.

- Der IP-Entity muss bekannt gemacht werden, dass alle Pakete zum Subnetz 139.22.220 über den lokalen Anschluss 192.168.100.111 geroutet werden sollen:

```
route add net 139.22.220 192.168.100.111 1
```

Arbeitsschritte am System Solaris-4

- Das WAN-Interface erhält einen eindeutigen Namen und die zugehörige IP-Adresse:

  ```
  csr create if name=clwip0 ipaddr=192.168.100.110
  ```

  Vom System wird eine Datei namens *clw.routes.clwip0* eingerichtet.

- Der IP-Entity muss bekannt gemacht werden, dass alle Pakete zum Subnetz 139.22.110 über den lokalen Anschluss 192.168.100.110 geroutet werden sollen:

  ```
  route add net 139.22.110. 192.168.100.110 1
  ```

Arbeitsschritte am System Solaris-1

- Konfigurieren Sie die Route:

  ```
  route add net 139.22.220.0 139.22.110.111 1
  ```

  Alternativ können Sie das System Solaris-3 in der Datei */etc/defaultrouter* als Default-Router angeben. Tragen Sie ein:

  139.22.110.111

Arbeitsschritte am System Solaris-2

- Route konfigurieren:

  ```
  route add net 139.22.110.0 139.22.220.110 1
  ```

  Alternativ können Sie das System Solaris-3 in der Datei */etc/defaultrouter* als Default-Router angeben. Tragen Sie ein:

  139.22.220.110

8 Web-basierte CMX-Administration

Mit CMX wird das Paket *SMAWwca* (web-basierte CMX-Administration) ausgeliefert. *SMAWwca* ermöglicht den Zugang zur CMX-Administration über das PRIMEPOWER-ServerView Management GUI. Die Administrationsclients können auf Solaris-Rechnern (LAN-Konsole oder System Management Konsole) oder auf Windows-Rechnern laufen.

Um die Sicherheit zu erhöhen, kann die Verbindung zwischen Administrationsclient und Kommunkationsserver mittels SSL/TLS verschlüsselt werden. Dazu wird das Zusatzpaket *SMAMswca* zur Verfügung gestellt, das auf einem separaten Administrationsserver installiert werden kann.

Das folgende Bild veranschaulicht die verschiedenen Administrationsvarianten, und zeigt, welche Pakete auf welchen Rechnern installiert sein müssen.

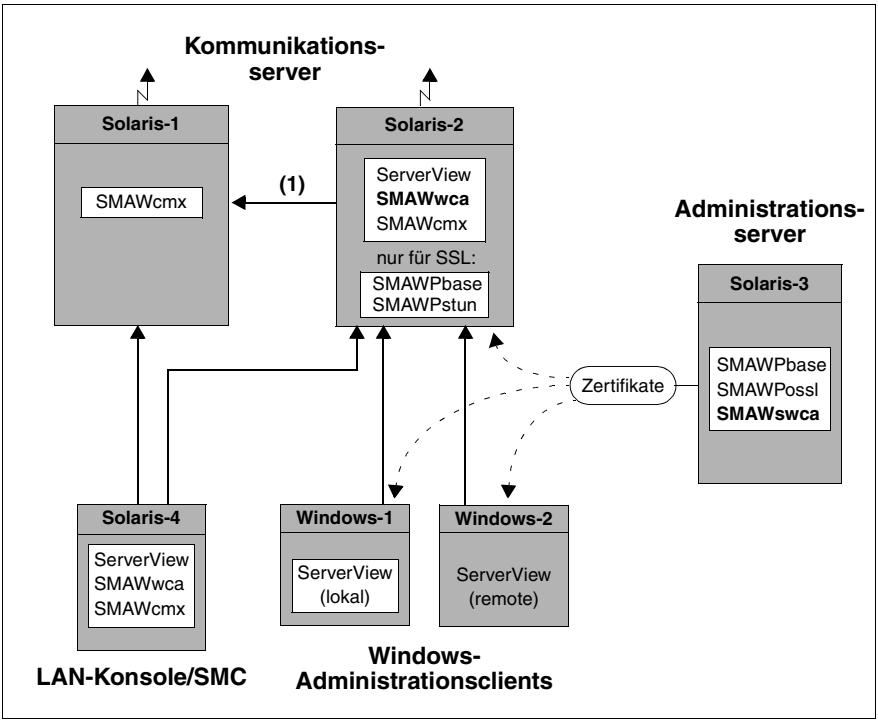


Bild 18: Web-basierte CMX-Administration

Der Rechner Solaris-1 enthält nur das Paket *SMAWcmx*. Auf dem Rechner Solaris-2 sind dagegen alle verfügbaren Pakete für eine gesicherte web-basierte CMX-Administration installiert. Damit ergeben sich folgende Möglichkeiten:

– Solaris-2 kann von allen Clients direkt über das Web administriert werden.

– Die Verbindung zwischen Solaris-2 und den Windows-Clients Windows-1, Windows-2 kann per SSL/TLS gesichert werden. Dazu muss auf dem Administrationsserver Solaris-3 ein Server-Zertifikat erzeugt und auf Solaris-2 kopiert werden. Falls selbstsignierte Root-Zertifikate verwendet werden, müssen Sie diese auf die beiden Windows-Administrationsclients kopieren.

– Solaris-1 kann nur von der LAN-Konsole/SMC (Solaris-4) aus direkt administriert werden, da er kein *SMAWwca* enthält.

  Die Windows-Administrationsclients können Solaris-1 nur indirekt über die Java-Anwendung auf Solaris-2 administrieren; auf dieser Verbindung (1) ist keine Verschlüsselung über SSL/TLS möglich.

8.1 Installation

Bevor Sie die Kommunikationsserver administrieren können, müssen Sie die nachfolgend beschriebenen Komponenten auf den Kommunikationsservern und auf den Administrationsclients installieren. Gehen Sie dabei in der angegebenen Reihenfolge vor.

Kommunikationsserver ohne SMAWwca

Dieser Server kann nur über die LAN-Konsole/SMC direkt administriert werden.

► Installieren Sie das Paket *SMAWcmx*.

Kommunikationsserver mit SMAWwca

Dieser Server kann sowohl über die LAN-Konsole/SMC als auch über Windows-Clients administriert werden. Gehen Sie wie folgt vor:

► Installieren Sie ServerView ab V2.2 von der Control DVD.

► Installieren Sie das Paket *SMAWcmx*.

► Installieren Sie das Paket *SMAWwca*.
SMAWwca muss im Rahmen der kundenspezifischen Installation explizit ausgewählt werden (Auswahl *custom*).

► Erweitern Sie die ServerView-Konfigurationsdatei mit Hilfe des Kommandos *add_cmxadm*, siehe Seite 196.

Falls Sie SSL/TLS einsetzen möchten, sind zusätzliche Schritte notwendig:

► Installieren Sie *SMAWPbase* und *SMAWPstun* von der Control-DVD.

► Kopieren Sie das Server-Zertifikat nach */opt/SMAW/SMAWcmx/wca/stunnel/certs*, siehe Seite 160.

Falls Sie IPSec einsetzen möchten, gehen Sie vor wie in Abschnitt „Verschlüsselung mit IPSec“ auf Seite 165 beschrieben.

Administrationsserver

Dieser Server wird nur benötigt, wenn Sie SSL/TLS einsetzen möchten. Er wird dabei ausschließlich für das Zertifikatsmanagement eingesetzt. Diese Funktion kann auch von einem Kommunikationsserver übernommen werden. Gehen Sie wie folgt vor:

► Installieren Sie nacheinander *SMAWPbase* und *SMAWPossl* von der Control-DVD.

► Installieren Sie das Paket *SMAWswca*.
SMAWswca muss im Rahmen der kundenspezifischen Installation explizit ausgewählt werden (Auswahl *custom*).

LAN-Konsole / System Management Konsole

Über die LAN-Konsole bzw. der System Management Konsole können Sie alle Kommunikationsserver direkt administrieren. Gehen Sie wie folgt vor:

► Installieren Sie ServerView ab V2.2 von der Control DVD.

► Installieren Sie das Paket *SMAWcmx*.

► Installieren Sie das Paket *SMAWwca*.
SMAWwca muss im Rahmen der kundenspezifischen Installation explizit ausgewählt werden (Auswahl *custom*). *SMAWwca* muss immer nach ServerView und *SMAWcmx* installiert werden.

► Erweitern Sie die ServerView-Konfigurationsdatei mit Hilfe des Kommandos *add_cmxadm*, siehe Seite 196.

Windows-Client mit lokalem ServerView

Auf diesem Administrationsclient soll ServerView als lokale win32-Anwendung gestartet werden. Gehen Sie wie folgt vor:

- Installieren Sie JRE (Java Runtime Environment) mit Version 1.4.2 oder höher. Damit wird automatisch Java Web Start installiert. Nach der Installation des JRE können Sie den Java Web Start Anwendungsmanager unter Verwendung des Befehls *Ausführen* im Startmenü aufrufen. Geben Sie dazu im Fenster *Ausführen* das Kommando *javaws* ein.

  Sie können JRE über die ServerView-Download-Seite (Port 8883) vom Server herunterladen, wenn ServerView dort installiert ist. Siehe auch Bild 21 auf Seite 149.

- Installieren Sie ServerView als win32-Anwendung.

  Die selbstextrahierende exe-Datei können Sie ebenfalls über die ServerView-Download-Seite (Port 8883) vom Server herunterladen, siehe Bild 21 auf Seite 149. Nach Ausführen der exe-Datei wird ServerView in das Windows-Startmenü eingetragen.

- Erstellen bzw. erweitern Sie die Datei *WSAConfig* wie auf Seite 146 beschrieben.

- Erweitern Sie die Policy-Datei, siehe Seite 144.

- Falls Sie SSL/TLS einsetzen möchten:
  Importieren Sie das Root-Zertifikat, siehe Seite 161ff.

Windows-Client mit remote ServerView

Auf diesem Administrationsclient soll ServerView als remote Anwendung gestartet werden. Gehen Sie wie folgt vor:

- Installieren Sie JRE (Java Runtime Environment) mit Version 1.4.2 oder höher. Damit wird automatisch Java Web Start installiert. Nach der Installation des JRE können Sie den Java Web Start Anwendungsmanager unter Verwendung des Befehls *Ausführen* im Startmenü aufrufen. Geben Sie dazu im Fenster *Ausführen* das Kommando *javaws* ein.

  Sie können JRE über die ServerView-Download-Seite (Port 8883) vom Server herunterladen, wenn ServerView dort installiert ist. Siehe auch Bild 21 auf Seite 149.

- Erweitern Sie die Policy-Datei, siehe Seite 144.

► Falls Sie SSL/TLS einsetzen möchten:
Importieren Sie das Root-Zertifikat, siehe Seite 161ff.

Die ServerView-Anwendung wird über Eingabe der URL gestartet, siehe Abschnitt „ServerView starten“ auf Seite 148.

Es wird empfohlen, die angegebene Installationsreihenfolge einzuhalten. Das Produkt ServerView sowie die Pakete *SMAWwca*, *SMAWswca*, *SMAWPbase*, *SMAWPstun* und *SMAWPossl* können jedoch auch zu einem späteren Zeitpunkt installiert werden.

Readme-Dateien

Die Readme-Dateien von *SMAWwca* liegen auf dem Server, auf dem *SMAWwca* installiert wurde. Sie können wie folgt auf diese Readme-Dateien zugreifen:

– Im Client:
Rufen Sie im Browser die CMX_ADM-Download-Seite auf und klicken Sie auf die *README* Option, siehe Bild 19 auf Seite 142.

– Auf dem administrierten Kommunikationsserver:
Geben Sie folgendes Kommando ein:
`pg /opt/SMAW/SMAWcmx/wca/README`.

8.2 Client konfigurieren

ServerView und die web-basierte CMX-Administration nutzen die Java™ Web Start-Technologie. Um den Ablauf der web-basierten CMX-Administration in dieser Umgebung zu ermöglichen, müssen Sie die Konfigurationsdatei *.java.policy* erzeugen. Wenn ServerView lokal als win32-Anwendung gestartet werden soll, müssen Sie zusätzlich die lokale ServerView-Konfigurationsdatei *WSAConfig* erstellen.

Dafür werden mit dem Paket *SMAWwca* Template-Dateien ausgeliefert, die Sie vom Server herunterladen können. Geben Sie dazu im Browser des Administrationsclients nach der Installation folgende URL ein:
`http://`*comm-server*`:8881/CMX/CMX_ADM_download.htm`
(*comm-server* = Kommunikationsserver, auf dem *SMAWwca* installiert ist).

Sie erhalten folgende Seite:

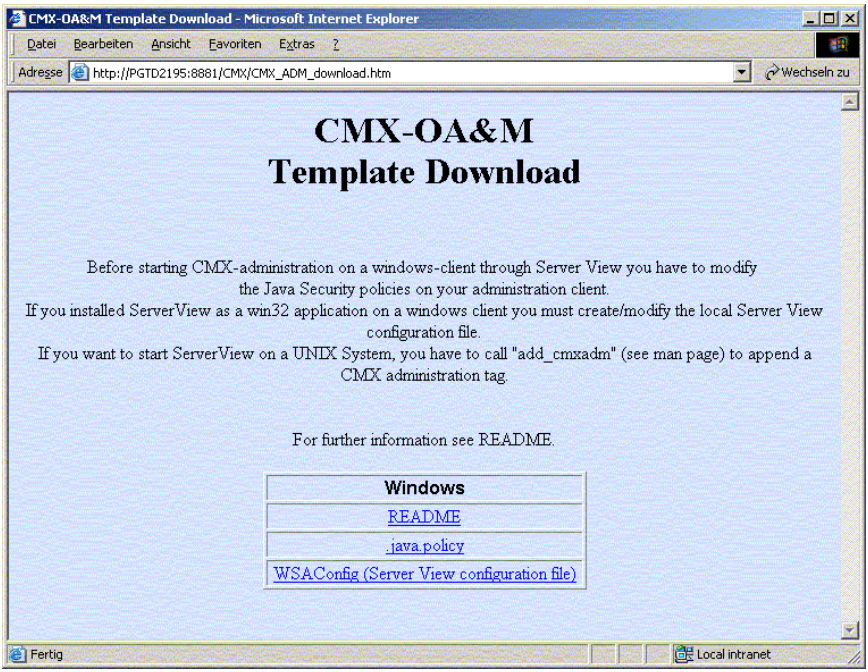


Bild 19: CMX_ADM-Download-Seite

8.2.1 Java-Sicherheitseinstellungen

Beim Start von ServerView über Java Web Start besitzt ServerView nur eingeschränkte Rechte, u.a. darf Java Web Start nicht auf eine entfernte jnlp-Datei zugreifen. Deshalb ist es notwendig, die Java-Sicherheitseinstellungen auf allen Administrationsclients zu erweitern.

Die Java-Sicherheitseinstellungen werden mit Hilfe von Policy-Dateien definiert. Damit Sie die web-basierte CMX-Administration nutzen können, müssen Sie deshalb die benutzerbezogene Datei *.java.policy* erzeugen bzw. erweitern. Template-Dateien mit den neuen/geänderten Sicherheitseinstellungen können Sie von der CMX_ADM-Download-Seite (siehe oben) laden.

Sicherheitseinstellungen ändern

Zum Ändern der Sicherheitseinstellungen stehen die folgenden drei Möglichkeiten zur Verfügung:

- Die Datei *.java.policy* nach *%USERPROFILE%/.java.policy* kopieren, falls diese noch nicht existiert, oder
- eine bereits vorhandene, private Policy-Datei *%USERPROFILE%/.java.policy* um die Einträge in der Template-Datei *.java.policy* ergänzen, oder
- die Sicherheitseinstellungen mit Hilfe des Tools *policytool* ändern (MS-DOS-Fenster) .

%USERPROFILE% bezeichnet dabei der Wert der Variablen USERPROFILE, siehe unten.

Vorbereitungen

In allen genannten Fällen müssen Sie zuerst folgende Daten ermitteln:

1. Den Wert der Variablen *USERPROFILE*.

   Dazu verwenden Sie das Kommando *set* oder das Kommando *echo %USERPROFILE%* (im MS-DOS-Fenster)

2. Den Wert der Variablen *USERNAME*.

   Dazu verwenden Sie das Kommando *set* oder das Kommando *echo %USERNAME%* (im MS-DOS-Fenster)

3. Den Pfad des Java Web Start Verzeichnisses (cache file).

   Dieser Pfad hängt von der jeweils installierten JRE-Version ab. Für JRE 1.5.0 ist dies in der Regel folgender Pfad:

   file:c/Profiles/%USERNAME%/Application Data/Sun/Java/Deployment/cache/javaws/http/-

   Dieser Pfad ist im Java Policy Template voreingestellt, Sie müssen nur *%USERNAME%* durch Ihren Benutzernamen ersetzen und ggf. den Laufwerkbuchstaben anpassen. Überprüfen Sie, ob dieser Pfad auch Ihrer Installation entspricht. Den in Ihrer Java Installation verwendeten Pfad können Sie den Java Web Start Einstellungen entnehmen. Dazu rufen Sie Java Web Start (javaws) über das Windows Startmenü, Befehl *Ausführen* auf. Sie erhalten folgendes Dialogfeld:

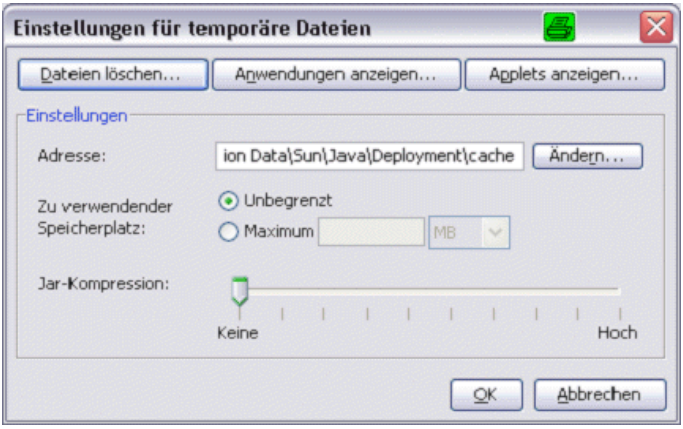


Bild 20: Einstellung von Java Web Start

Für den Eintrag in der Policy-Datei ersetzen Sie „\“ durch „/“ und ergänzen den Namen des Anwendungsordners um *javaws/http/-*.

Datei .java.policy von der CMX_ADM-Download-Seite speichern

- Wählen Sie mit der rechten Maustaste die Datei *.java.policy* auf der Download-Seite aus, siehe Bild 19 auf Seite 142.
- Wählen Sie *Save as* / *Ziel speichern unter*.
- Wählen Sie den Pfadnamen *USERPROFILE* aus und setzen Sie *type* auf *all files* bzw. *Dateityp* auf *alle Dateien*.
- Wählen Sie *Save* / *Speichern*.
- Überprüfen Sie den Eintrag für das Java Web Start Cache File. Sie müssen auf jeden Fall *USERNAME* durch Ihren eigenen Benutzernamen ersetzen. Dazu können Sie einen beliebigen Texteditor verwenden.

Eine bereits existierende Datei .java.policy erweitern

Die Datei *.java.policy* ist eine ASCII-Datei, d.h. sie kann mit einem beliebigen Editor editiert werden.

- Kopieren Sie die Einträge aus der Template-Datei *.java.policy* in die private Datei *%USERPROFILE%/.java.policy*.

► Überprüfen Sie den Eintrag für das Java Web Start Cache File. Sie müssen auf jeden Fall *USERNAME* durch Ihren eigenen Benutzernamen ersetzen. Dazu können Sie einen beliebigen Texteditor verwenden.

Das Tool policytool auf dem Client nutzen

► Starten Sie das Programm *JREHOME\bin\policytool*, z.B. in einem MS-DOS-Fenster oder über das Menü *Start* -> *Ausführen*, Eingabe *JREHOME*`/bin/policytool`.

Standard für *JREHOME* ist *C:/Program Files/Java/j**.

► Öffnen Sie über das Menü *File* -> *Open* die entsprechende Datei *.java.policy*.

► Wählen Sie *Add Policy Entry* aus:

- ► Legen Sie die CodeBase fest, d.h. den Verweis auf den Java Web Start Anwendungscache, siehe oben.
- ► Klicken Sie auf die Schaltfläche *Add permission*.
- ► Wählen Sie den Eintrag *File Permission.*
- ► Wählen Sie bei *Target Name* den Wert <<*ALL FILES*>> aus.
- ► Geben Sie als *action* den Eintrag `execute` ein.
- ► Klicken Sie auf die Schaltflächen *OK* und dann *Done*, um Ihre Angaben zu quittieren.

► Sichern Sie die Konfiguration über Menü *File* -> *Save*.

► Schließen Sie *policytool*.

8.2.2 Datei WSAConfig

Damit sich ServerView als win32-Anwendung auf dem Administrationsclient starten lässt, müssen Sie die lokale ServerView-Konfigurationsdatei *WSAConfig* von der CMX_ADM-Download-Seite laden.

Die mit dem Paket *SMAWwca* bereitgestellte Konfigurationsdatei ist eine Erweiterung der von ServerView ausgelieferten Datei. Sie enthält neben den Einträgen für den AlarmService und ARMTech einen Eintrag für die CMX-Administration. Falls keine weiteren Fremdapplikationen unterstützt werden, dann können Sie die Datei *WSAConfig* direkt in das Verzeichnis *%ProgramFiles%/Fujitsu Siemens Computers/WebSysAdmin* kopieren. Andernfalls erweitern Sie bitte die existierende Datei um den CMXADM-Tag, siehe geladene *WSAConfig*-Datei: *[CMXADM].....#CMXADM_END*

WSAConfig enthält u.a. einen Verweis auf die Java Web Start Anwendung, Voreinstellung ist der Standardpfad der englischen JRE Installation. Falls das JRE in einem anderen Verzeichnis installiert ist, müssen Sie *WSAConfig* modifizieren. Gehen Sie wie folgt vor:

- Wählen Sie auf der CMX_ADM-Download-Seite die Datei *WSAConfig* durch Klicken mit der rechten Maustaste aus, siehe Bild 19 auf Seite 142.
- Wählen Sie *Save as* bzw. *Speichern unter*
- Wählen Sie den Pfadnamen SYSTEM(C:) und setzen Sie *type=all files* bzw. *Dateityp=alle*.
- Wählen Sie *Save* bzw. *Speichern*.
- Öffnen Sie die Datei *WSAConfig* mit einem beliebigen Texteditor, überprüfen Sie den Eintrag für *javaws.exe* und ändern Sie ihn falls erforderlich.

8.3 Kommunikationsserver konfigurieren

Um ServerView den Zugriff auf die CMX-Administration zu ermöglichen, müssen Sie die ServerView-Konfigurationsdatei *WSAConfig* auf dem Kommunikationsserver erweitern. Dazu verwenden Sie das Kommando *add_cmxadm* wie auf Seite 196 beschrieben.

Mit dem Kommando *del_cmxadm* können Sie einzelne Einträge entfernen, siehe Seite 197.

Achtung!
Bei Deinstallation von *SMAWwca* werden alle CMX-spezifischen Einträge aus der ServerView Konfigurationsdatei *WSAConfig* gelöscht.

8.4 SMAWwca aktivieren und deaktivieren

Nach der Installation von *SMAWwca* und ggf. *SMAWstunnel* aktivieren Sie die Administrationssoftware auf dem Kommunikationsserver wie folgt:

- Rufen Sie das Kommando *wca_init start* auf (siehe Seite 201). Dieses Kommando bewirkt, dass alle von *SMAWwca* benötigten Konfigurationsdateien bereit gestellt werden. Falls *Stunnel* bereits installiert und die benötigten Zertifikate erzeugt wurden, dann wird *Stunnel* ebenfalls gestartet. Andernfalls können Sie *Stunnel* zu einem beliebigen späteren Zeitpunkt über das Kommando *wca_stunnel start* starten (siehe Seite 201).
- Geben Sie das Kommando *add_cmxadm* ein (siehe Seite 196). Damit wird der Zugriff auf die CMX-Administration über ServerView freigeschaltet.

SMAWwca deaktivieren

Vor der Deinstallation des Paktes *SMAWwca* sollten Sie *SMAWwca* deaktivieren. Verwenden Sie hierzu das Kommando *wca_init stop*. Damit werden die durch den Start der web-basierten CMX-Administration erfolgten Änderungen im System rückgängig gemacht.

8.5 ServerView starten

Das Starten von ServerView hängt von der Installationsart und der Plattform ab:

- ServerView in Windows als remote Anwendung:

  Dazu haben Sie 2 Möglichkeiten:

  – Starten über die URL *http://comm-server:8881*

  – Starten über die URL *http://comm-server:8883* und Auswahl der PRIMEPOWER ServerView Suite (remote Application using Java Web Start), siehe Bild 21 auf Seite 149.

  Dabei ist *comm-server* der Name des Kommunikationsservers, auf dem ServerView installiert ist.

- ServerView in Windows als lokale win32-Anwendung:

  Rufen Sie ServerView über die Schaltfläche im Windows Startmenü auf.

- ServerView als Solaris-Anwendung
  (Grafische Konsole auf GP7000F, PRIMEPOWER oder PRIMESTATIION)

  Starten Sie das ServerView Management GUI über folgendes shell-Skript:

  `/opt/SMAW/bin/wsa` [`sync`] [*hostname*]

  Es wird die ServerView-Startseite angezeigt, siehe Bild 22 auf Seite 150. Bitte beachten Sie, dass die DISPLAY-Variable des Xwindows-Systems gesetzt sein muss, z.B. *DISPLAY=rechner-adresse:0.0*.

Weitere Informationen zu ServerView entnehmen Sie bitte dem Handbuch „PRIME POWER ServerView Suite2.2“.

ServerView Download-Seite

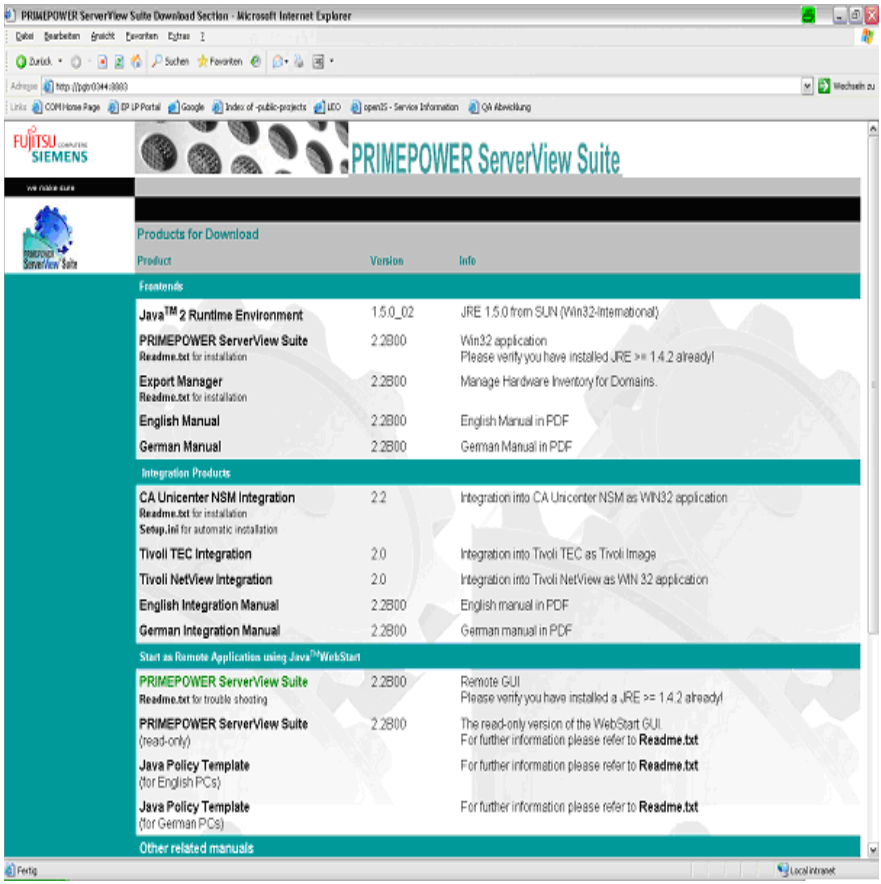


Bild 21: ServerView Download-Seite

ServerView Startseite

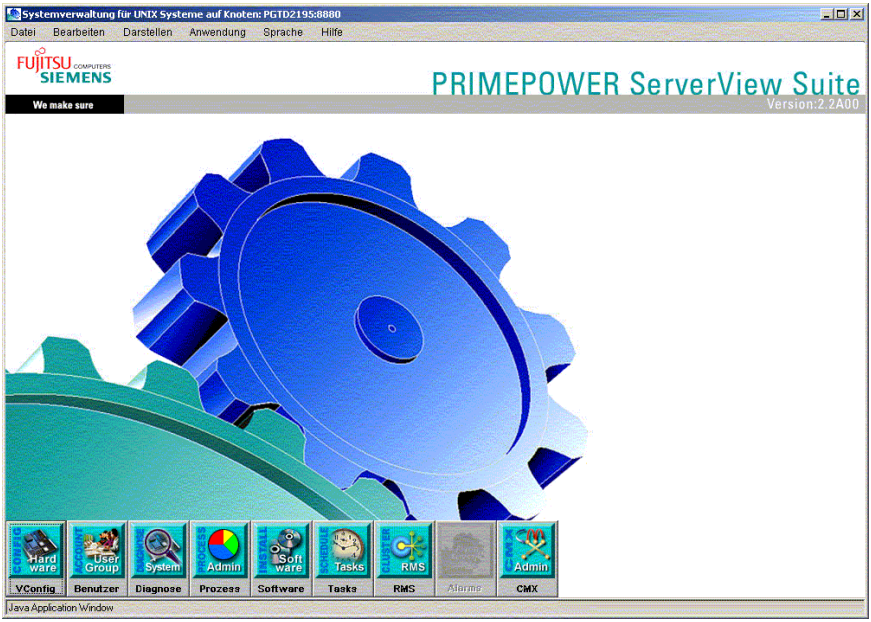


Bild 22: ServerView Startseite

8.6 CMX-Administrationsoberfläche starten

Nach der Installation von ServerView und *SMAWwca* auf dem Kommunikationsserver haben Sie folgende Möglichkeiten, die CMX-Administration (CLI oder CUI) auf dem Administrations-Client zu starten:

- Starten Sie ServerView und klicken Sie auf der Startseite auf die Schaltfläche *CMXADM,*
- oder starten Sie ServerView und wählen Sie das Menü *Anwendung* -> *CMXADM* aus,
- oder doppelklicken Sie auf das CMXADM-Symbol (wenn schon vorhanden).

> **i** Die Telnet-Anwendung, die die CMX-Administration startet, ist als Java Web Start Anwendung implementiert. Daher erzeugt Java Web Start automatisch Verknüpfungen auf dem Windows-Desktop und im Startmenü. Standardmäßig fragt Java Web Start beim zweiten Start einer Anwendung, ob eine Verknüpfung erstellt werden soll. Über das Einstellungspanel kann dies geändert werden. Anschließend kann die CMX-Administration direkt durch Doppelklick auf das Symbol der Anwendung gestartet werden.

Die CMXADM-Startseite wird im Browser angezeigt.

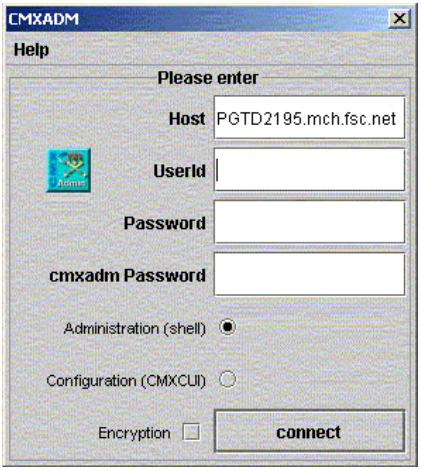


Bild 23: CMXADM-Startseite

- Geben Sie im Feld *Host* den Namen des Kommunikationsservers ein. Auf diesem Server muss *SMAWcmx* installiert sein.
  Voreinstellung: Name des Servers, auf dem *SMAWwca* installiert ist.
- Geben Sie Benutzerkennung, zugehöriges Kennwort und Administrationskennwort (Rolle *cmxadm*) des Kommunikationsservers ein. Das Kennwort der Rolle *cmxadm* ist optional und nur dann von Bedeutung, wenn von einer Kennung aus administriert werden soll, der die Rolle *cmxadm* zugeordnet ist.
- Wählen Sie auf der Startseite entweder *Administration (shell)* oder *Configuration (CMXCUI)* aus.
- Falls die Daten mittels SSL/TLS verschlüsselt werden sollen, aktivieren Sie die Checkbox *Encryption*.
- Drücken Sie die Schaltfläche *Connect*.

Eine Shell Session oder das CMXCUI wird gestartet.

Aus Kompatibilitätsgründen wird die Eingabe der Kennung *root* mit zugehörigem Passwort akzeptiert.

8.7 Sicherheit

Die CMX-Administration erfolgt über eine Telnet-Verbindung zwischen Windows-Client und dem Kommunikationsserver. Telnet ist ein TCP/IP-Protokoll, das keinerlei Sicherheit bietet, d.h. die Daten werden unverschlüsselt durch das Netz übertragen. Dies gilt insbesondere für Benutzerkennung und zugehöriges Kennwort.

Zur Erhöhung der Sicherheit wird der Einsatz von SSL/TLS oder IPSec empfohlen, insbesondere falls Administrationsclient und Kommunikationsserver nicht in einem privaten LAN liegen.

8.7.1 Verschlüsselung mittels SSL/TLS

Das Protokoll Secure Sockets Layer (SSL) wurde in den Jahren 1994 und 1995 von Netscape Communications Corporation in den beiden Versionen SSLv2 und SSLv3 veröffentlicht. Auf dieser Basis definierte die Internet Engineering Task Force (IETF) den Internet-Standard Transport Layer Security Protocol Version 1 (TLS 1.0), der im Januar 1999 als RFC2246 herausgegeben und im Juni 2003 mit RFC3546 ergänzt wurde.

Die häufigste Anwendung findet SSL/TLS im World Wide Web als Protokollschicht zwischen TCP und HTTP, um den Datenverkehr zwischen Web-Servern und Web-Browsern zu **verschlüsseln** und diese beiden Kommunikationspartner zu **authentifizieren**.

In Rahmen der CMX-Administration wird SSL/TLS eingesetzt, um die Kommunikation über eine Telnet-Verbindung abzusichern. SSL/TLS wird mit Hilfe der Komponenten *OpenSSL* und *Stunnel* realisiert, da die Telnet-Anwendung das SSL/TLS-Protokoll nicht unterstützt:

– *OpenSSL* ist eine frei verfügbare Implementierung des SSL/TLS-Protokolls. In der SSL-Bibliothek (*libssl*) sind alle Protokollversionen von SSLv2, SSLv3 und TLS 1.0 implementiert. Die kryptographische Bibliothek (*libcrypto*) stellt die gängigen Algorithmen für die Kryptographie bereit und unterstützt zusätzlich das Zertifikats- und Key-Management. *SMAWswca* nutzt die entsprechenden *OpenSSL*-Funktionen, um Zertifikate zu erstellen.

– *Stunnel* ist wie *OpenSSL* eine frei verfügbare Software. *Stunnel* wird verwendet, um die Netzwerk-Kommunikation von Diensten zu verschlüsseln, die keine Kryptographie-Funktionen unterstützen, wie etwa Telnet. *Stunnel* arbeitet als SSL-Wrapper.

Weitere Informationen zu *OpenSSL* und *Stunnel* erhalten Sie unter folgenden URLs:

http://www.openssl.org
http://www.stunnel.org
http://www.tldp.org/HOWTO/SSL-Certificates-HOWTO/index.html

8.7.1.1 Funktionsweise von SSL/TLS

Die folgende Abbildung veranschaulicht den Datenfluss zwischen Administrationsclient und Kommunikationsserver, wenn mittels SSL/TLS verschlüsselt wird.

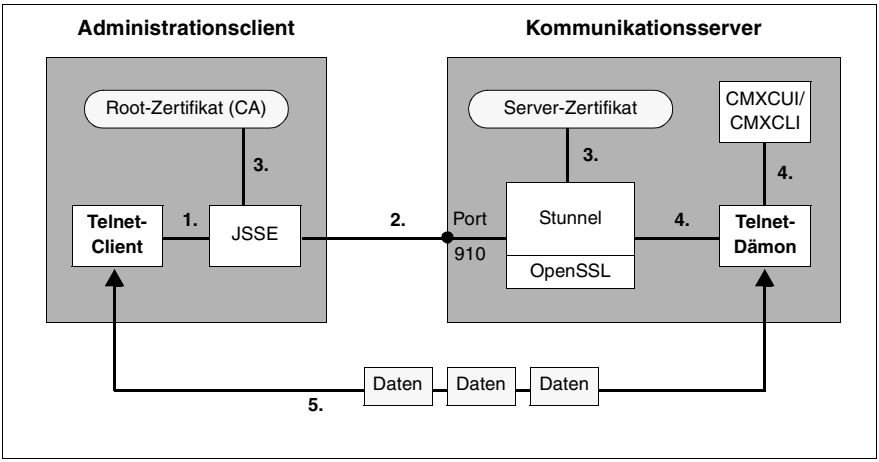


Bild 24: Verbindungsaufbau und Datenfluss bei Einsatz von SSL/TLS

Erläuterung:

1. Durch Auswahl von *Encryption* auf der CMXADM-Startseite initiiert der Telnet Client über JSSE eine Verbindung zum Kommunikationsserver. JSSE (Java Secure Socket Extension) ist in Java JRE ab 1.4 enthalten und stellt eine Java-Version des SSL/TLS-Protokolls zur Verfügung. JSSE umfasst u.a. Funktionen zur Authentifizierung und Verschlüsselung.
2. Es wird zunächst über Port 910 (Standard) eine Verbindung zum SSL-Wrapper *Stunnel* aufgebaut. Diese Portnummer wird in den *Stunnel*-Konfigurationsdateien festgelegt, siehe Abschnitt „Einsatz von Stunnel“ auf Seite 164. Für die CMX-Administration wird die Portnummer standardmäßig auf 910 gesetzt.

Über diese Verbindung werden SSL-Version und Zertifikate ausgetauscht. Dabei wird das SSL-Handshake Protokoll verwendet.

3. Die Zertifikate werden geprüft. Danach ist eine SSL/TLS-Verbindung eingerichtet. Über diese können die aufgesetzten Protokolle gesichert arbeiten.
4. *Stunnel* baut eine Verbindung zum Telnet-Dämon auf, der wiederum das CMXCUI bzw. CMXCLI startet.
5. Die Daten zwischen Administrationsclient und CMXCUI bzw. CMXCLI auf dem Kommunikationsserver werden nun über eine gesicherte SSL/TLS-Verbindung ausgetauscht.

8.7.1.2 Voraussetzungen für den Einsatz von SSL/TLS

Wenn die Informationen zwischen Clients und Kommunikationsservern über eine gesicherte SSL-Verbindung ausgetauscht werden sollen, sind zunächst folgende Schritte notwendig:

- Auf dem Administrationserver
  - Installation von *SMAWPbase* und *SMAWPossl* von der Control-DVD
  - Installation von *SMAWswca*
  - Erzeugen eines Server-Zertifikats für den Einsatz mit *Stunnel*
- Auf allen Kommunikationsservern
  - Installation von *SMAWPbase* und *SMAWPstun* von der Control-DVD
  - Kopieren des Server Zertifikats nach */opt/SMAW/SMAWcmc/wca/stunnel/certs*
  - Starten von *Stunnel* für die CMX-Administration
- Auf allen Administrationsclients
  - Importieren des Public Key

8.7.1.3 Zertifikate erzeugen mit Stunnel

Um die Funktionen von *Stunnel* in Verbindung mit *SMAWwca* nutzen zu können, müssen Sie ein Zertifikat/Schlüsselpaar in einem genau festgelegten Format erzeugen. Hierbei ist darauf zu achten, dass der private Schlüssel (private key) nicht verschlüsselt wird, da *Stunnel* keine Möglichkeit bietet, das Passwort für den Schlüssel vom Benutzer zu erfragen.

Ein Zertifikat erhalten Sie bei einer beliebigen Zertifizierungsstelle (Certificate Authority, kurz **CA** genannt), z.B. VeriSign, TC TrustCenter oder einer anderen offiziellen CA. Dazu müssen Sie ein Certificate Signing Request (CSR) an die Zertifizierungsstelle senden. Der CSR enthält den öffentlichen Schlüssel (public key) und die Daten des Antragstellers. Der CSR wird von der Zertifizierungsstelle nach Prüfung der Angaben des Antragstellers signiert und ein passendes Zertifikat erzeugt. Der public key der CA steht öffentlich zur Verfügung.

In einigen JRE-Versionen sind Zertifikate von CAs vorinstalliert. Dadurch wird gewährleistet, dass die von den vorinstallierten CAs ausgestellten und signierten Zertifikate verifiziert werden können. Zu Testzwecken oder wenn Sie Ihren Kommunikationsserver nur im Intranet einsetzen, können Sie auch eine private CA erzeugen und sich damit ein Zertifikat erstellen.

Um Ihnen den Umgang mit den Zertifikaten zu vereinfachen, werden mit *SMAWswca* Skripten installiert, mit denen Sie selbstsignierte Zertifikate, einen Zertifikatsantrag und das von *Stunnel* erwartete Server-Zertifikat erzeugen können. Dazu sind folgende Schritte notwendig:

1. Demo CA erzeugen mit dem Kommando *manage_cert -newca*
2. Privaten Schlüssel erzeugen mit *manage_cert -newkey*
3. Zertifikatanforderung erzeugen mit *manage_cert -newreq*
4. Zertifikatanforderung mit der privaten CA signieren (*manage_cert -sign*)
5. Server-Zertifikat erstellen mit *manage_cert -finish*

Die Schritte 1 und 4 entfallen, falls Ihr Zertifikat von einer offiziellen CA ausgestellt wird.

Die einzelnen Schritte werden im Folgenden näher erläutert. Details zu dem Kommando *manage_cert* entnehmen Sie bitte dem Abschnitt „manage_cert - Zertifikate verwalten“ auf Seite 198.

Schritt 1: Demo CA erzeugen

Mit dem Kommando *manage_cert -newca* erzeugen Sie eine private Certificate Authority namens **Snakeoil-CA**. Sie können Ihren Zertifikatsantrag mit dieser CA signieren. Beachten Sie, dass das so signierte Zertifikat nicht vertrauenswürdig ist und nur zu Testzwecken eingesetzt werden sollte.

Nach Aufruf des Kommandos *manage_cert -newca* werden Sie aufgefordert, eine Reihe von Fragen zu beantworten. Dabei werden Ihnen die in der Konfigurationsdatei voreingestellten Werte angezeigt. Wenn Sie diese Standardwerte übernehmen möchten, geben Sie ENTER ein.

Als Ergebnis werden mehrere Verzeichnisse und Dateien erstellt; u. a. die Dateien für den öffentlichen und den privaten Schlüssel:
/opt/SMAW/SMAWswca/PrivateCA/certs/wca_cacert.pem (public key)
/opt/SMAW/SMAWswca/PrivateCA/private/wca_cakey.pem (private key)

Es wird empfohlen, die erzeugten Dateien zu sichern, da Sie sonst ein ausgestelltes Zertifikat nicht verlängern können.

Beispiel

```
# manage_cert -newca
CA certificate filename (or enter to create)  <enter>

Making CA certificate ...
Generating a 1024 bit RSA private key
....++++++
.................................................++++++
writing new private key to
'/opt/SMAW/SMAWswca/PrivateCA/private/wca_cakey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XY]:                                )  <enter>
State or Province Name (full name) [Snake Desert]:                )  <enter>
Locality Name (eg, city) [Snake Town]:                            )  <enter>
Organization Name (eg, company) [Snake Oil, Ltd]:                 )  <enter>
Organizational Unit Name (eg, section) [Certificate Authority]:   )  <enter>
Common Name (eg, CA name) [Snake Oil CA]:                         )  <enter>
Email Address [ca@snakeoil.dom]:                                  )  <enter>
```

Das Zertifikat ist standardmäßig 365 Tage gültig. Wenn Sie eine andere Gültigkeitsdauer wünschen, dann geben Sie diese mit der Zusatzoption *-days number* an, siehe Abschnitt „manage_cert - Zertifikate verwalten“ auf Seite 198.

Schritt 2: Privaten Schlüssel erzeugen

Mit dem Kommando *manage_cert -newkey* erzeugen Sie einen privaten Schlüssel. Es wird empfohlen, eine Sicherheitskopie des Schlüssels zu erstellen, da im Falle eines Verlustes das Zertifikat seine Gültigkeit verliert.

Als Ergebnis erhalten Sie Ihren privaten Schlüssel unter
/opt/SMAW/SMAWswca/ssl/private/wca_privkey.pem

Beispiel

```
# manage_cert -newkey
Create private key
Generating RSA private key, 2048 bit long modulus
.....+++

.........................+++
e is 65537 (0x10001)
Private key is in /opt/SMAW/SMAWswca/ssl/private/wca_privkey.pem
```

Schritt 3: Zertifikatsanforderung erzeugen

Mit dem Kommando *manage_cert -newreq comm-server* erzeugen Sie eine Zertifikatsanfrage für den Kommunikationsservers *comm-server*. Sie können diese Anfrage an eine offizielle Zertifizierungsstelle senden oder mit Ihrer privaten CA signieren.

Nach Aufruf des Kommandos *manage_cert -newreq* werden Sie aufgefordert, eine Reihe von Fragen zu beantworten. Dabei werden Ihnen die in der Konfigurationsdatei voreingestellten Werte angezeigt. Wenn Sie diese Werte übernehmen, antworten Sie mit ENTER. Dabei ist *Common Name* der Name des Rechners, auf dem *Stunnel* läuft. Sie müssen bei *Common Name* immer den vollständigen Rechnernamen angeben (siehe Ausgabe des Kommandos *nslookup rechner-name*).

Als Ergebnis erhalten Sie die Zertifikatsanforderung unter */opt/SMAW/SMAWswca/ssl/req/comm-server.newreq.pem*

Beispiel

```
# manage_cert -newreq myHost
Create a certificate request
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XY]:
State or Province Name (full name) [Snake Desert]:
Locality Name (eg, city) [Snake Town]:
Organization Name (eg, company) [Snake Oil, Ltd]:
Organizational Unit Name (eg, section) [Development]:
Common Name (hostname of the machine) [local host]:myHost.aaa.bbb.ccc
Email Address [ca@snakeoil.dom]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password ****
Request is in /opt/SMAW/SMAWswca/ssl/req/myHost.newreq.pem
```

Schritt 4: Zertifikatsanforderung durch private CA signieren

Mit dem Kommando *manage_cert -sign comm-server* signieren Sie Ihre Zertifikatsanforderung mit dem privaten Schlüssel Ihrer CA (*comm-server* = Name des Kommunikationsservers). Das Kommando nutzt implizit die Datei */opt/SMAW/SMAWswca/ssl/req/comm-server.newreq.pem* als Eingabe.

Das Zertifikat wird ausgegeben unter:
/opt/SMAW/SMAWswca/ssl/certs/comm-server.newcert.pem.

Beispiel

```
# manage_cert -sign myHost
Using configuration from /opt/SMAW/SMAWswca/conf/openssl_ca.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Jul 19 10:15:12 2004 GMT
            Not After : Jul 19 10:15:12 2005 GMT
        Subject:
            countryName               = XY
            stateOrProvinceName       = Snake Desert
            localityName              = Snake Town
            organizationName          = Snake Oil, Ltd
            organizationalUnitName    = Development
            commonName                = myHost
            emailAddress              = ca@snakeoil.dom
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                4D:B1:EC:9D:53:C7:EC:3E:A2:1A:41:1B:DC:A6:2F:04:9A:8E:B5:54
            X509v3 Authority Key Identifier:

keyid:E3:75:EE:63:28:B7:9A:1B:FB:77:6B:94:B4:4E:FC:6D:E9:97:21:62
                DirName:/C=XY/ST=Snake Desert/L=Snake Town/O=Snake Oil,
Ltd/OU=Certificate Authority/CN=Snake Oil CA/emailAddress=ca@snakeoil.dom
                serial:00

Certificate is to be certified until Jul 19 10:15:12 2005 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y

Certificate:
.....


-----END CERTIFICATE-----
Signed certificate is in /opt/SMAW/SMAWswca/ssl/certs/myHost.newcert.pem
```

Schritt 5: Server-Zertifikat für Stunnel erstellen

Das Server-Zertifikat kann auf jedem Administrationsserver erzeugt werden, auf dem die Pakete *SMAWswca*, *SMAWPbase* und *SMAWPossl* installiert wurden.

Stunnel erwartet die Daten in einem genau festgelegten Format. Das endgültige Zertifikat muss den privaten Schlüssel, die Zertifikatsanforderung und das signierte Zertifikat enthalten. Sie erstellen die zugehörige Datei mit Hilfe von *manage_cert -finish comm-server* (*comm-server* = Name des Kommunikationsservers).

Als Ergebnis erhalten Sie die Input-Datei für *Stunnel* unter */opt/SMAW/SMAWswca/ssl/certs/comm-server.pem*.

Beispiel

```
# manage_cert –finish myHost
Final certificate is in /opt/SMAW/SMAWswca/ssl/certs/myHost.pem
```

Wenn Sie Ihren Request von einer offiziellen CA signieren, entfallen die Schritte 1 und 4. Für Schritt 5 ist es erforderlich, dass Sie das von der CA signierte Zertifikat kopieren und unter folgendem Namen speichern: */opt/SMAW/SMAWswca/ssl/certs/comm-server.newcert.pem*.

8.7.1.4 Server-Zertifikat auf Kommunikationsserver kopieren

Sie müssen das in Schritt 5 erzeugte Zertifikat auf den Kommunikationsserver kopieren und dort im folgendem Verzeichnis ablegen:
/opt/SMAW/SMAWcmx/wca/stunnel/certs

8.7.1.5 Root-Zertifikat auf Administrationsclient importieren

Das Vorgehen hängt davon ab, ob Sie ein privates Root-Zertifikat oder ein offizielles Root-Zertifikat verwenden.

Privates Root-Zertifikat importieren

Wenn Sie Ihren Zertifikatsantrag mit Ihrer eigenen CA signiert haben, muss das private Root-Zertifikat auf jedem Administrationsclient importiert werden.

- Kopieren Sie dazu das Zertifikat
  /opt/SMAW/SMAWswca/PrivateCA/certs/wca_cacert.pem
  von Ihrem Administrationsserver auf den Client.
- Importieren Sie dieses Zertifikat mit dem Kommando *keytool -import*, siehe Syntaxbeschreibung unten.

  Geben Sie dabei die Option *-keystore cacerts* an. Das Standard-Passwort für die Keystore-Datei *cacerts* lautet *changeit*. Es wird bei jedem Zugriff auf diese Datei abgefragt. Außerdem müssen Sie die Option *-alias* ... angeben, um den Eintrag eindeutig zu identifizieren.

  Syntax des Kommandos *keytool -import* laut JSSE:

  keytool ␣**-import** {␣**-alias**␣alias} {␣**-file**␣cert_file} [␣**-keypass**␣keypass]
  {␣**-noprompt**} {␣**-trustcacerts**} {␣**-storetype**␣storetype}
  {␣**-keystore**␣keystore} [␣**-storepass**␣storepass]
  [␣**-provider**␣provider_class_name] {␣**-v**} {␣**-J**javaoption}

  Die Syntaxdarstellung gemäß JSSE unterscheidet sich von der Darstellung im übrigen Handbuch:

  {...} bedeutet, dass ein Standardwert angenommen wird, falls die Option nicht angegeben wurde.

  [...] bedeutet, dass der Wert erfragt wird, falls er nicht angegeben wurde und nicht im security properties file definiert ist.

Das Kommando *keytool* liegt im Allgemeinen nicht im Pfad der Pfadvariablen. Um sich die Eingabe zu erleichtern, erweitern Sie die Pfadvariable um den JRE-Pfad *C:\Program Files\Java\jre1.5.0_02\bin*, z.B. über die Funktionen des Start-Menüs:
Start -> Einstellungen -> Systemsteuerung -> System -> Erweitert -> Umgebungsvariablen.

Beispiel 1: CA-Zertifikat importieren

Sie importieren das CA Zertifikat *wca_cacert.pem* mit Hilfe des Kommandos *keytool -import*, siehe Bild 25. Das Zertifikat wurde vorher nach *P:\rg\wca_cacert.pem* kopiert. Als Keystore-Datei muss *C:\Program Files\Java\jre1.5.0_02\lib\security\cacerts* gewählt werden. Ggf. müssen Sie vor Aufruf des Kommandos in das Verzeichnis *C:\Program Files\Jaca\jre1.5.0_02\lib\security* wechseln, da die Keystore-Datei immer im aktuellen Verzeichnis erzeugt/erwartet wird.

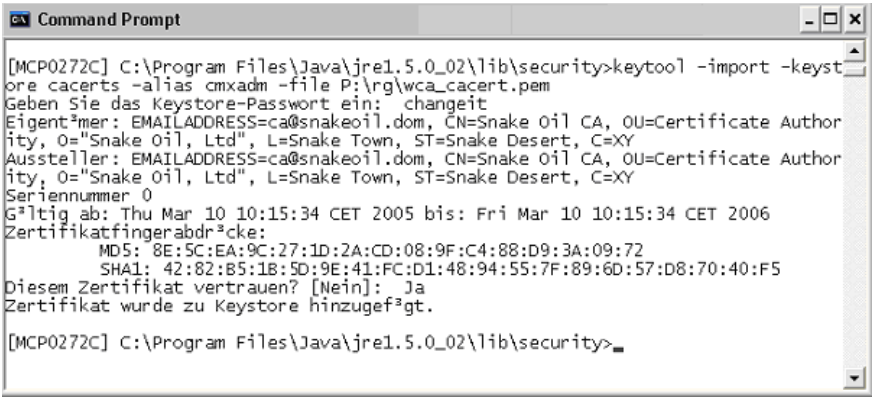


Bild 25: Importieren eines Zertifikats mit keytool -import

Beispiel 2: Löschen des Root-Zertifikats

Wenn Sie das Root-Zertifikat nicht mehr benötigen, löschen Sie es mit Hilfe des Kommandos *keytool -delete* aus der Keystore-Datei:

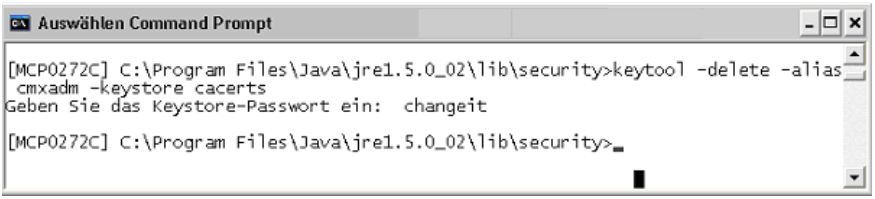


Bild 26: Löschen eines Zertifikats mit keytool -delete

Offizielles Root-Zertifikat importieren

Mit dem Java Runtime Environment werden eine Reihe von Zertifikaten bekannter Zertifizierungsstellen installiert. Wurde Ihr Request von einer dieser CAs signiert (offizielles Root-Zertifikat), sind auf der Client-Seite keine Änderungen erforderlich. Eine Liste der bekannten Root-Zertifikate können Sie über das Kommando *keytool -list* ermitteln:

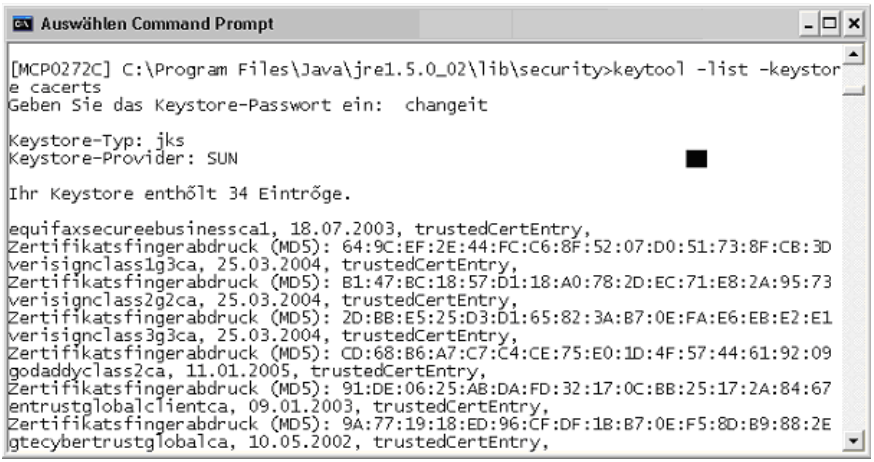


Bild 27: Zertifizierungsinformation über keytool -list ausgeben

Wenn Sie Ihren Zertifikat-Request durch eine CA signieren lassen, die in der Java-Umgebung auf Ihrem Administrationsclient nicht bekannt ist, dann müssen Sie den öffentlichen Schlüssel von der Zertifizierungsstelle erfragen, die sie gewählt haben. Anschließend importieren Sie diesen Schlüssel wie unter Schritt 1 auf Seite 156 beschrieben.

8.7.1.6 Einsatz von Stunnel

Damit die Daten zwischen Client und Kommunikationsserver über eine gesicherte SSL-Verbindung ausgetauscht werden können, muss *Stunnel* auf dem Kommunikationsserver konfiguriert und gestartet werden. Außerdem muss ein signiertes Zertifikat in */opt/SMAW/SMAWcmx/wca/stunnel/certs/comm-server.pem* vorhanden sein, siehe „Schritt 5: Server-Zertifikat für Stunnel erstellen“ auf Seite 160.

Stunnel konfigurieren

Beim Aktivieren der web-basierten CMX-Administration mittels *wca_init start* wird eine Konfigurationsdatei mit folgendem Namen erzeugt: */opt/SMAW/SMAWcmx/wca/stunnel/conf/stunnel_wca.conf*

Diese Datei enthält alle für den Start von *Stunnel* relevanten Informationen:
– Verzeichnis und Name der Server-Zertifikatsdatei
– verwendete Portnummer
– den Namen und die Argumente des Programms, das gestartet wird

Gegebenenfalls müssen Sie *stunnel_wca.conf* anpassen. *Stunnel* erwartet das Server Zertifikat unter */opt/SMAW/SMAWcmx/wca/stunnel/certs/comm-server.pem*. Weiterhin horcht *Stunnel* standardmäßig auf den Port 910. Sollte dieser Port in Ihrer Systemumgebung bereits belegt sein, müssen Sie einen freien Port wählen und die Portnummer in allen Konfigurationsdateien eintragen. Verwenden Sie dazu das Kommando *set_port*, siehe Seite 200.

Stunnel starten und stoppen

Stunnel wird nach erfolgreicher Inbetriebnahme von *SMAWwca* und nach dem Booten des Kommunikationsservers automatisch gestartet, wenn das in der Konfigurationsdatei angegebene Server-Zertifikat existiert und wenn Port 910 nicht durch eine andere Anwendung belegt ist.

Sie können *Stunnel* zu einem späteren Zeitpunkt auch mit dem Kommando *wca_stunnel start* starten und mit *wca_stunnel stop* anhalten, siehe Seite 201.

Bitte beachten Sie, dass *Stunnel* nach jeder Zertifikatsänderung neu gestartet werden muss.

8.7.2 Verschlüsselung mit IPSec

Zum Schutz von IP-Datagrammen können die IPSec-Protokolle Encapsulating Security Payload (ESP) und Authentication Header (AH) verwendet werden: AH bietet Daten-Authentifizierung, Datenintegrität und Schutz vor dem wiederholten Senden von Paketen. Das ESP-Protokoll bietet zusätzlich eine Vertraulichkeit des Verkehrsflusses.

Wenn Sie IPSec zur Absicherung der Kommunikation über die Telnet-Verbindung einsetzen wollen, müssen die IPSec-Einstellungen auf Server- und Client-Seite aufeinander abgestimmt werden.

IPSec arbeitet im Transport-Modus und verwendet sowohl das AH (Authentifizierungsalgorithmus MD5) als auch das ESP (Verschlüsselungsalgorithmus DES) -Protokoll, d.h. das gesamte IP-Datagramm wird authentifiziert und die Daten werden verschlüsselt.

Der von IPSec bereitgestellte Sicherheitsdienst verlangt die Verwendung von gemeinsamen Schlüsseln für die Ausführung von Authentifizierung und/oder Vertraulichkeit. Für die Verwaltung der Schlüssel wird das Internet Key Exchange-Protokoll (IKE) eingesetzt (ab Solaris Version 9).

Als Authorisierungsmethode wird "preshared keys" verwendet. Dabei müssen die verwendeten Schlüssel auf beiden Seiten übereinstimmen.

Im Folgenden wird die IPSec-Konfiguration für den Datentransfer zwischen einem Administrations-Server mit Solaris V9 und einem Administrations-Client mit Windows >2000 anhand eines Beispiels erläutert. In diesem Beispiel wird der Datenverkehr über den Telnet-Port (Port 23) mittels IPSec geschützt.

8.7.2.1 Server-Konfiguration (Solaris V9)

Die Konfiguration von IPSec unter Solaris V9 gliedert sich in folgende Arbeitsschritte:

- Sicherheitsdatenbank Security Policy Database (SPD) erstellen bzw. erweitern
- IKE-Sicherheitseinstellungen erstellen bzw. erweitern
  - Konfigurationsdatei (IKE Policy File)
  - Schlüssel für die IKE-Authentifizierung (ike.preshared-Datei)
- SPD laden
- IKE-Dämon starten

Security Policy Database (SPD) – /etc/inet/ipsecinit.conf

In der Datei */etc/inet/ipsecinit.conf* werden die IPSec-Sicherheitseinstellungen festgelegt, die angeben, wie der Datenverkehr überwacht werden soll.

Falls die Datei */etc/inet/ipsecinit.conf* auf Ihrem Rechner noch nicht existiert, muss sie erzeugt werden.

Falls die Datei bereits existiert, ergänzen Sie sie um folgenden Eintrag:

```
# telnet Verkehr,  AH-Authentifizierung: md5, ESP-Verschluesselung: des,
# ESP-Authentifizierung: md5, shared association
#
{lport 23} ipsec {auth_algs md5 encr_algs des encr_auth_algs md5 sa
shared}
```

Beispiel: /etc/inet/ipsecinit.conf

```
#
#ident"@(#)ipsecinit.sample1.601/10/29 SMI"
#
# Copyright (c) 1999,2001 by Sun Microsystems, Inc.
# All rights reserved.
#
# This file should be copied to /etc/inet/ipsecinit.conf to enable IPsec
# systemwide policy (and as a side-effect, load IPsec kernel modules).
# Even if this file has no entries, IPsec will be loaded if
# /etc/inet/ipsecinit.conf exists.
#
# Add entries to protect the traffic using IPSEC. The entries in this
# file are currently configured using ipsecconf from inetinit script
# after /usr is mounted.
#
# For example,
#
# {rport 23} ipsec {encr_algs des encr_auth_algs md5}
#
# Or, in the older (but still usable) syntax
#
#       {dport 23} apply {encr_algs des encr_auth_algs md5 sa shared}
#       {sport 23} permit {encr_algs des encr_auth_algs md5}
#
# will protect the telnet traffic originating from the host with ESP using
# DES and MD5.  Also:
#
# {raddr 10.5.5.0/24} ipsec {auth_algs any}
#
# Or, in the older (but still usable) syntax
#
#       {daddr 10.5.5.0/24} apply {auth_algs any sa shared}
#       {saddr 10.5.5.0/24} permit {auth_algs any}
#
# will protect traffic to/from the 10.5.5.0 subnet with AH using any
available
# algorithm.
#
# To do basic filtering, a drop rule may be used. For example:
#
```

```
# {lport 23 dir in} drop {}
# {lport 23 dir out} drop {}
#
# will disallow any remote system from telnetting in.
#
# WARNING:This file is read before default routes are established, and
#before any naming services have been started.  The
#ipsecconf(1M) command attempts to resolve names, but it will
#fail unless the machine uses files, or DNS and the DNS server
#is reachable via routing information before ipsecconf(1m)
#invocation.  (E.g. the DNS server is on-subnet, or DHCP
#has loaded up the default router already.)
#
#It is suggested that for this file, use hostnames only if
#they are in /etc/hosts, or use numeric IP addresses.
#
#If DNS gets used, the DNS server is implicitly trusted, which
#could lead to compromise of this machine if the DNS server
#has been compromised.
##################################################################
#
# telnet Verkehr,  AH-Authentifizierung: md5, ESP-Verschluesselung: des,
# ESP-Authentifizierung: md5, shared association
#

{lport 23} ipsec {auth_algs md5 encr_algs des encr_auth_algs md5 sa shared}
```

IKE Policy-Datei – /etc/inet/ike/config

In der Datei */etc/inet/ike/config* werden die Regeln für IKE-Anfragen festgelegt.

Folgende Einträge müssen aufgenommen werden:

```
p1_lifetime_secs 28800
p1_nonce_len 20

## Werte der p1_xform-Parameter müssen mit den Eintragen in
## in /etc/inet/ipsecinit.conf übereinstimmen!!!
p1_xform { auth_method preshared oakley_group 2 auth_alg md5 encr_alg des }
p2_pfs 2

### Regeln (für jeden Administrations-Client muss eine eigene
### Regel erstellt werden):

{
   label "<string>"
   local_id_type ip
   local_addr <eigene IP-Adresse>
   remote_addr <IP-Adresse eines Administrations-Clients>
}
```

Beispiel: /etc/inet/ike/config

```
#
#ident"@(#)config.sample1.201/12/06 SMI"
#
# Copyright (c) 2001 by Sun Microsystems, Inc.
# All rights reserved.
```

```
##
## This file should be copied into /etc/inet/ike/config to enable the
## launch of the IKE daemon, in.iked(1m), at boot time.  You can also
## launch the IKE daemon after creating this file without rebooting by
## invoking /usr/lib/inet/in.iked with a root shell.
##

# Consult the ike.config(4) man page for further details.  Here is a small
# example from the man page.

### BEGINNING OF FILE

### First some global parameters...

## certificate parameters...

# Root certificates.  I SHOULD use a full Distinguished Name.
# I MUST have this certificate in my local filesystem, see ikecert(1m).
#cert_root    "C=US, O=Sun Microsystems\\, Inc., CN=Sun CA"

# Explicitly trusted certs that need no signatures, or perhaps self-signed
# ones.  Like root certificates, use full DNs for them for now.
#cert_trust    "EMAIL=root@domain.org"

# Where do I send LDAP requests?
#ldap_server   "ldap1.domain.org,ldap2.domain.org:389"

# Some PKI-specific tweaks...
# If you wish to ignore CRLs, uncomment this:
#ignore_crls
# If you wish to use HTTP (with name resolution) for URLs inside certs,
# uncomment this:
#use_http
# HTTP proxy and socks URLs should also be indicated if needed...
#socks "socks://socks-relay.domain.org"
#proxy "http://http-proxy.domain.org:8080"

## Phase 1 transform defaults...

p1_lifetime_secs 28800
p1_nonce_len 20

## Parameters that may also show up in rules.

p1_xform { auth_method preshared oakley_group 2 auth_alg md5 encr_alg des }
p2_pfs 2

### Now some rules...

{
   label "Client1"
   local_id_type ip
   local_addr 172.25.124.140
   remote_addr 172.25.123.64
}
{
   label "Client2"
   local_id_type ip
   local_addr 172.25.124.140
```

```
      remote_addr 172.25.123.153
   }
```

IKE preshared Datei – /etc/inet/secret/ike.preshared

Die Datei */etc/inet/secret/ike.preshared* enthält den Schlüssel für die IKE-Authentifizierung.

Für jeden Administrations-Client ist ein Eintrag der folgenden Form erforderlich:

```
{
          localidtype IP
          localid <eigene IP-Adresse>
          remoteidtype IP
          remoteid <IP-Adresse eines Administrations-Clients>
          key <hexstring, 16-Zeichen, abgestimmt mit Partnerkonfiguration>
}
```

Dabei können Sie für jeden Administrations-Client einen eigenen Schlüssel wählen.

Beispiel: /etc/inet/secret/ike.preshared

```
   #
   #ident"@(#)ike.preshared1.101/09/28 SMI"
   #
   # Copyright (c) 2001 by Sun Microsystems, Inc.
   # All rights reserved.
   #

   # ike.preshared - Pre-shared secrets for IKE authentication.
   #
   # Entries are of the form:
   #
   # {
   #<attribute> <value>
   #...
   # }
   #
   # Consult the man page for ike.preshared(4) for details.
   {
             localidtype IP
             localid 172.25.124.140
             remoteidtype IP
             remoteid 172.25.123.64
             key 31313131313131313131313131313131
   }
   {
             localidtype IP
             localid 172.25.124.140
             remoteidtype IP
             remoteid 172.25.123.153
             key 31323334313233343132333431323334
   }
```

SPD laden

Wenn die Datei */etc/inet/ipsecinit.conf* existiert, wird die Security Policy Database automatisch beim Booten des Systems geladen.

Zum Laden der Datenbank im laufendem Betrieb verwenden Sie das Kommando *ipsecconf*. Rufen Sie das Kommando zunächst mit der Option *-f* und anschließend mit der Option *-a* auf:

- Flush Policies:

```
/usr/sbin/ipsecconf -f
```

- SPD laden:

```
/usr/sbin/ipsecconf -a /etc/inet/ipsecinit.conf
```

Um die Einträge zu überprüfen, rufen Sie das Kommando *ipsecconf* ohne Optionen auf. Das System zeigt alle Policy-Einträge an.

Beispiel

```
# /usr/sbin/ipsecconf -f
# /usr/sbin/ipsecconf
# /usr/sbin/ipsecconf -a /etc/inet/ipsecinit.conf
        WARNING : New policy entries that are being added may
        affect the existing connections. Existing connections
        that are not subjected to policy constraints, may be
        subjected to policy constraints because of the new
        policy. This can disrupt the communication of the
        existing connections.
# /usr/sbin/ipsecconf
#INDEX 74
{lport 23} ipsec {auth_algs md5 encr_algs des encr_auth_algs md5 sa shared}
```

IKE-Dämon – in.iked

Wenn die Datei */etc/inet/ike/config* existiert, wird der IKE-Dämon automatisch beim Booten des Systems gestartet.

Wenn Sie die Änderungen an der IKE Policy-Datei im laufendem Betrieb übernehmen wollen, stoppen Sie den IKE-Dämon und starten Sie ihn anschließend mittels */usr/lib/inet/in.iked* neu.

Beispiel

```
# ps -ef |grep iked
    root 20866     1  0 08:44:15 ?        0:00 /usr/lib/inet/in.iked
    root 20868 27027  0 08:44:20 pts/4    0:00 grep iked
# kill -9 20866
# /usr/lib/inet/in.iked
```

8.7.2.2 Client-Konfiguration (Windows 2000)

Die Konfiguration von IPSec unter Windows 2000 gliedert sich in folgende Arbeitsschritte:

- IPSec-Policy konfigurieren/erstellen
  - Konfiguration/Erstellung einer Sicherheitsregel
  - Konfiguration der IPSec-Authentifizierungsregel
  - Konfiguration/Erstellung einer IPSec-Filterliste
  - Konfiguration/Erstellung einer IPSec-Filteraktion
- Die zu verwendenden Authentifizierungs- und Verschlüsselungsalgorithmen bei IKE festlegen
- Neue Sicherheitsrichtlinie zuweisen

Sie konfigurieren die Sicherheitsrichtlinien unter Windows 2000 über das *Local Security Policy*-Tool.

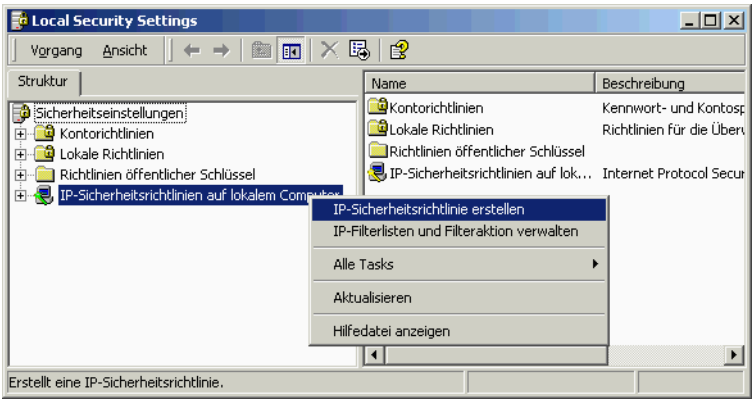
Die einzelnen Arbeitsschritte werden im Folgenden anhand von Bildschirmabzügen erläutert.

IPSec-Policy konfigurieren/erstellen

1. Neue IPSec-Policy einrichten
   - Lokale Sicherheitseinstellungen öffnen:

     Wählen Sie Menü *Start* -> *Programme* -> *Administrative Tools* -> *Local Security Policy*.

► IP-Sicherheitsrichtlinie erstellen:

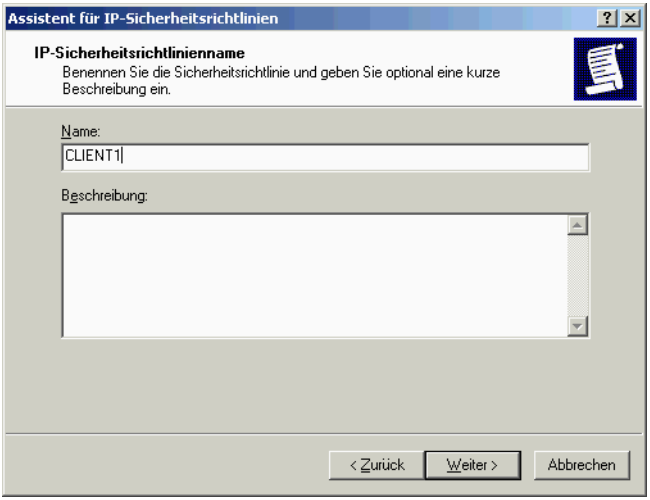


Klicken Sie mit der rechten Maustaste auf *IP-Sicherheitsrichtlinie auf lokalem Computer* und wählen Sie im Kontextmenü *IP-Sicherheitsrichtlinie erstellen* aus.

Der Assistent für IP-Sicherheitsrichtlinien startet.

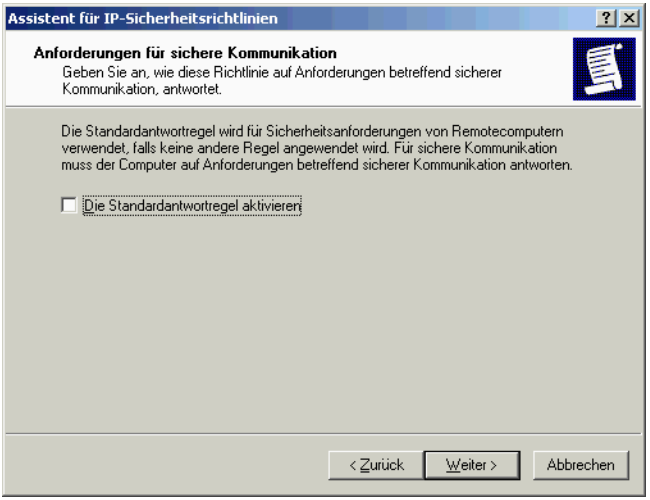
Klicken Sie auf die Schaltfläche *Weiter >*.

► Namen der IP-Sicherheitsrichtlinie festlegen:



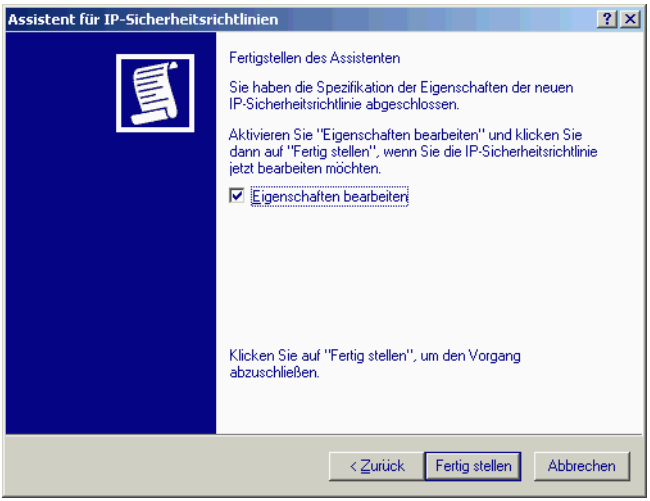
Geben Sie den Namen in das Feld *Name* ein und klicken Sie auf *Weiter >*.

► Standardantwortregel deaktivieren:



Deaktivieren Sie die Option *Die Standardantwortregel aktivieren* und klicken Sie auf die Schaltfläche *Weiter >*.

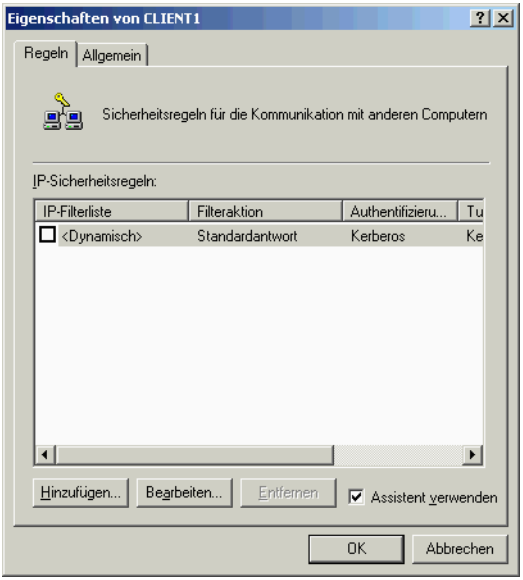
► IP-Sicherheitsrichtlinie fertig stellen:



Aktivieren Sie die Option *Eigenschaften bearbeiten* und klicken Sie auf die Schaltfläche *Fertig stellen*.

2. Sicherheitsregel für die neue Policy erstellen

   ► Sicherheitsregel erstellen:

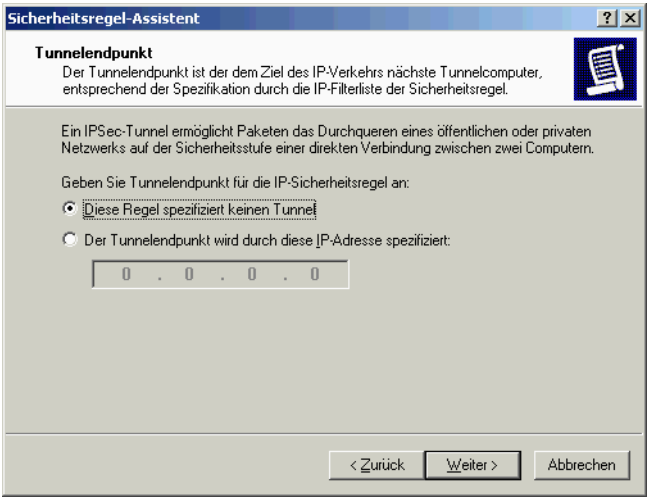


   Aktivieren Sie in der Dialogbox die Option *Assistent verwenden* und klicken Sie anschließend auf die Schaltfläche *Hinzufügen...*

   Der Sicherheitsregel-Assistent wird geöffnet.

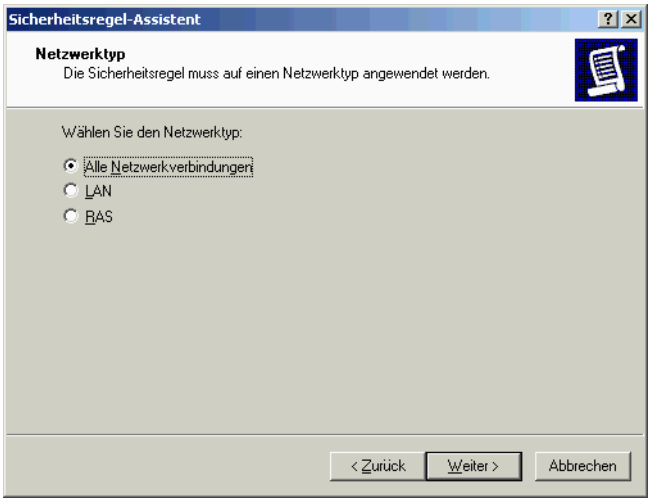
   Klicken Sie auf die Schaltfläche *Weiter >*.

► IPSec-Modus festlegen:



Wählen Sie die Option *Diese Regel spezifiziert keinen Tunnel* und klicken Sie auf die Schaltfläche *Weiter >*.

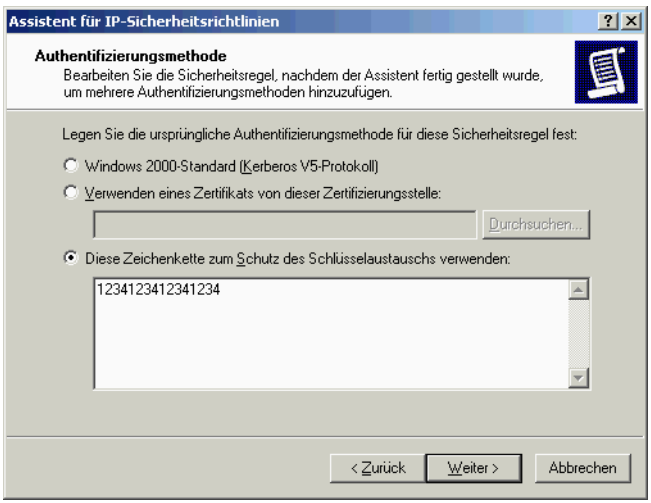
► Netzwerktyp festlegen:



Wählen Sie die Option *Alle Netzwerkverbindungen* und klicken Sie auf die Schaltfläche *Weiter >*.

3. Konfiguration der IPSec-Authentifizierungsregel

   – Authentifizierungsmethode Pre-shared Keys festlegen:



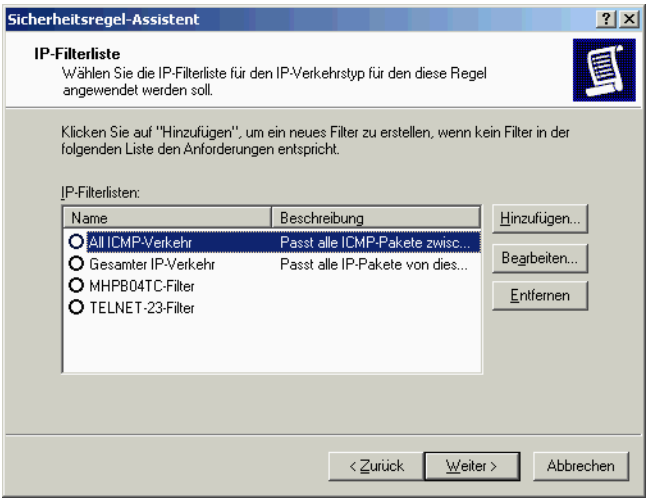
Wählen Sie die Option *Diese Zeichenkette zum Schutz...* aus und geben Sie den Pre-Shared Key (String, 16 Zeichen) ein.

Achtung!

Auf dem Administrations-Server muss der gleiche Key als Hexadezimal-String eingegeben werden.

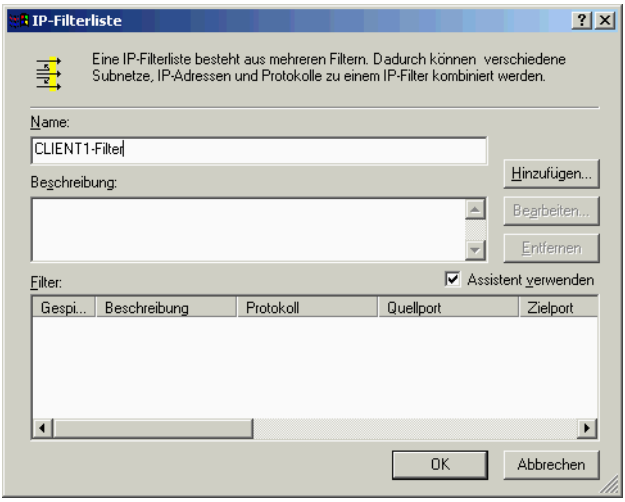
4. Filterliste konfigurieren

   ► Filterregel für die Policy-Regel erstellen:



Klicken Sie auf die Schaltfläche *Hinzufügen...*

► Namen der Filter(liste) festlegen:

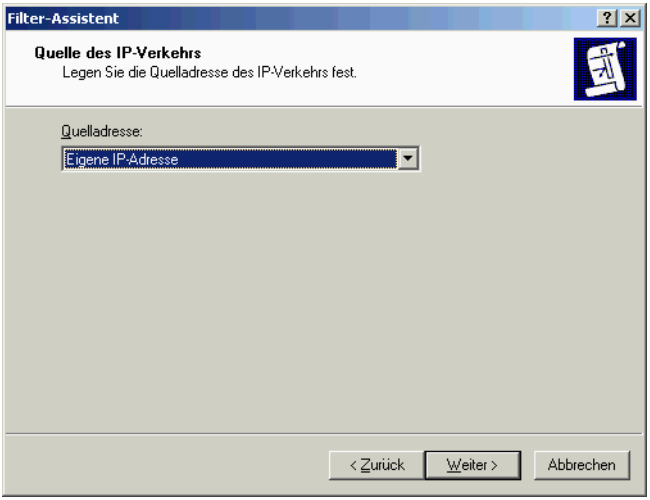


Geben Sie in das Feld *Name* den Namen ein, aktivieren Sie die Option *Assistent verwenden* und klicken Sie auf die Schaltfläche *Hinzufügen*.

Der IP-Filter-Assistent wird geöffnet.

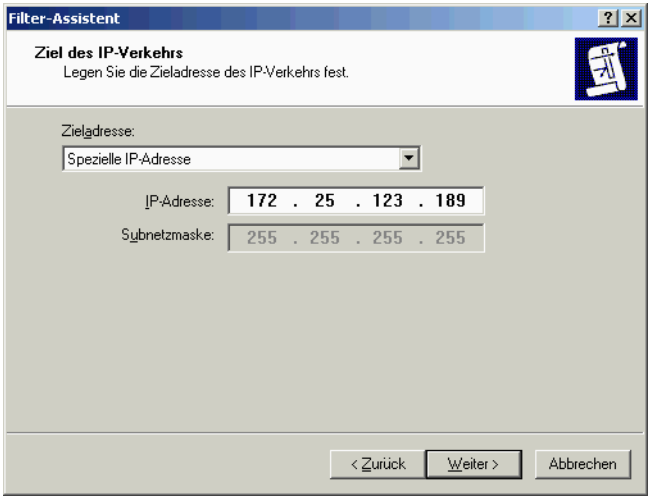
Klicken Sie auf die Schaltfläche *Weiter >*.

► Quelle des IP-Verkehrs festlegen:



Wählen Sie in der Liste *Quelladresse* die Option *Eigene IP-Adresse* aus und klicken Sie auf die Schaltfläche *Weiter >*.

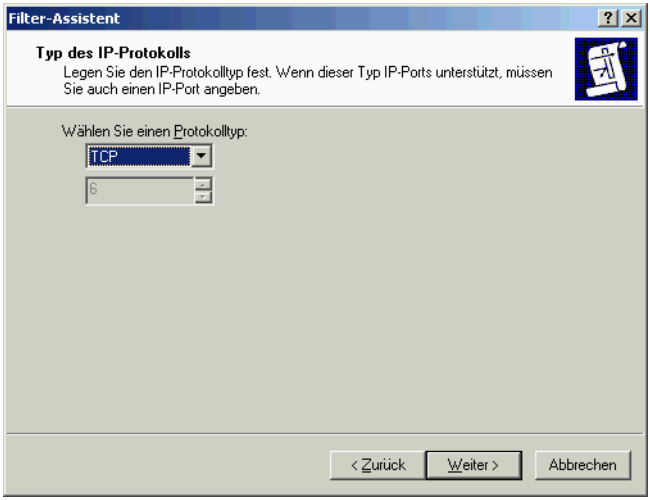
► Ziele des IP-Verkehrs festlegen:



Wählen Sie in der Liste *Zieladresse* die Option *Spezielle IP-Adresse* aus und geben Sie in das Feld *IP-Adresse* die Adresse des Administrations-Servers ein.

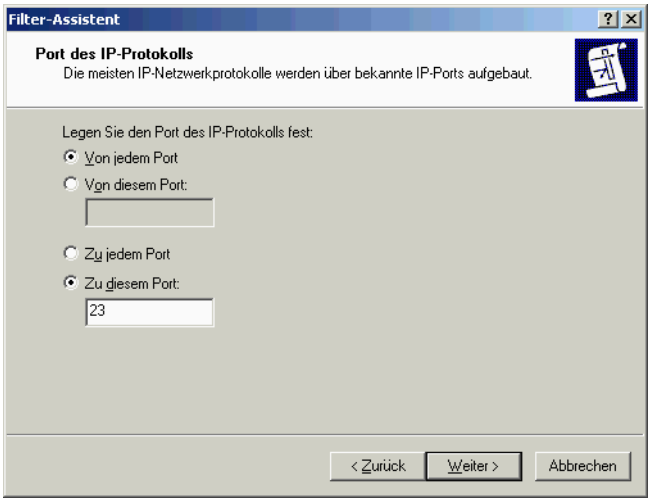
Klicken Sie auf die Schaltfläche *Weiter >*.

► Typ des IP-Protokolls festlegen:



Wählen Sie in der Liste den Protokolltyp *TCP* aus und klicken Sie auf *Weiter >*.

► Quell- und Zielport festlegen:

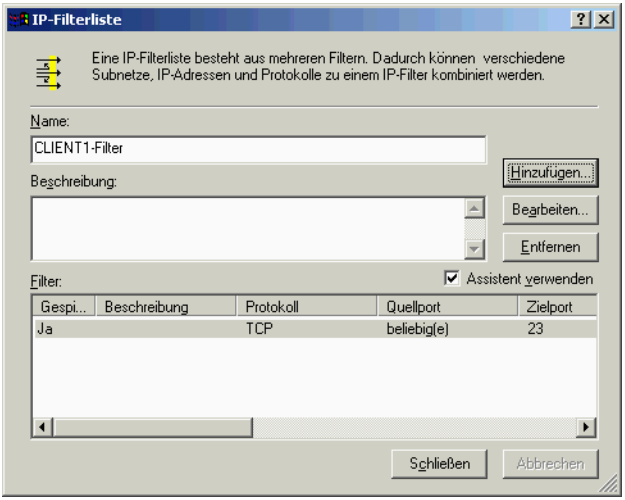


Wählen Sie die Optionen *Von jedem Port* und *Zu diesem Port*, geben Sie als Portnummer 23 ein und klicken Sie auf die Schaltfläche *Weiter >*.

► Filterliste fertigstellen:

Deaktivieren Sie die Option *Eigenschaften bearbeiten* und klicken Sie auf die Schaltfläche *Fertig stellen*.

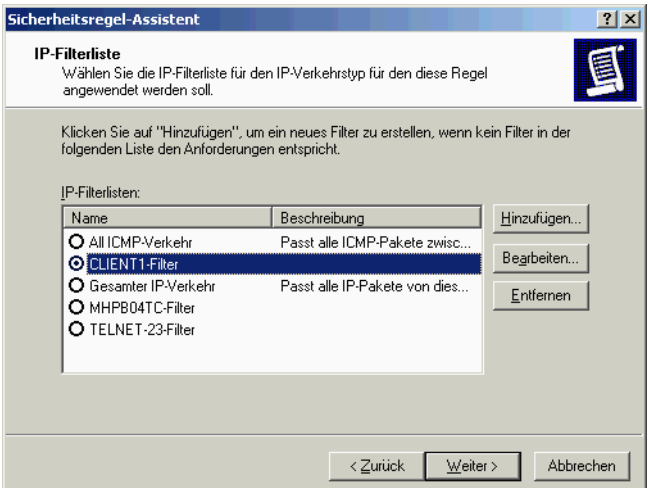
► IP-Filterliste schließen:



Klicken Sie auf die Schaltfläche *Schließen*.

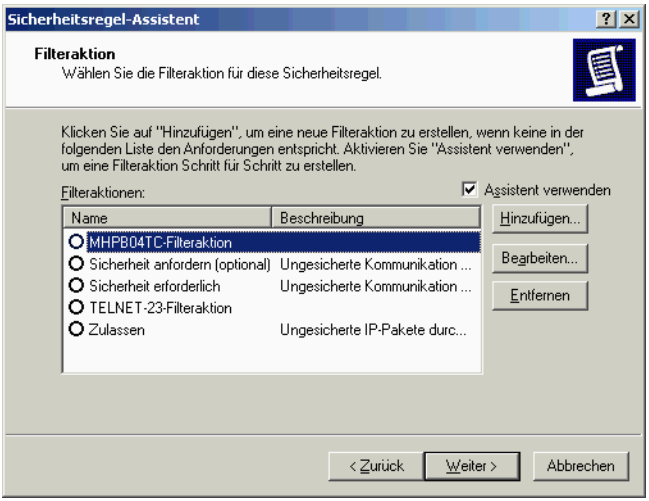
5. Filteraktion konfigurieren/erstellen

► Filter auswählen:



Wählen Sie den neu erstellten Filter `CLIENT1-Filter` aus und klicken Sie auf die Schaltfläche *Weiter >*.

► Filteraktion erstellen:

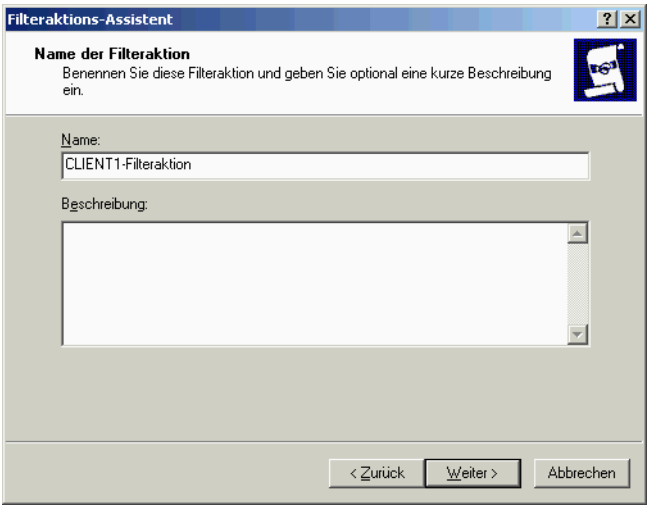


Aktivieren Sie die Option *Assistent verwenden* und klicken Sie auf die Schaltfläche *Hinzufügen...*

Der Filteraktions-Assistent wird geöffnet.

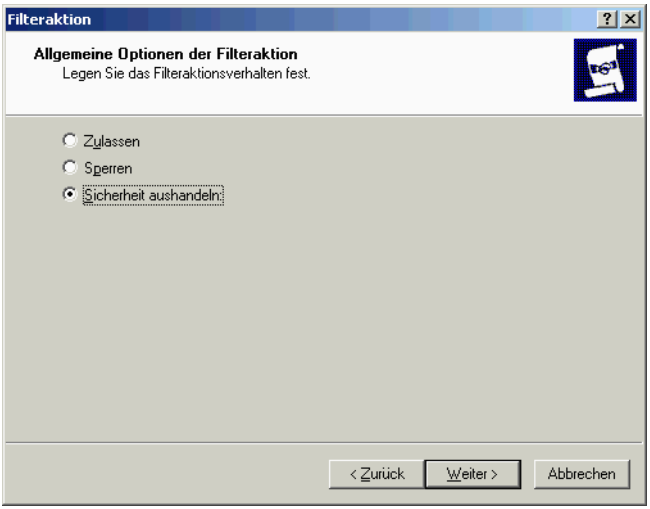
Klicken Sie auf die Schaltfläche *Weiter >*.

► Namen der Filteraktion festlegen:



Geben Sie in das Feld *Name* den Namen ein und klicken Sie auf die Schaltfläche *Weiter* >.

► Filteraktion festlegen:

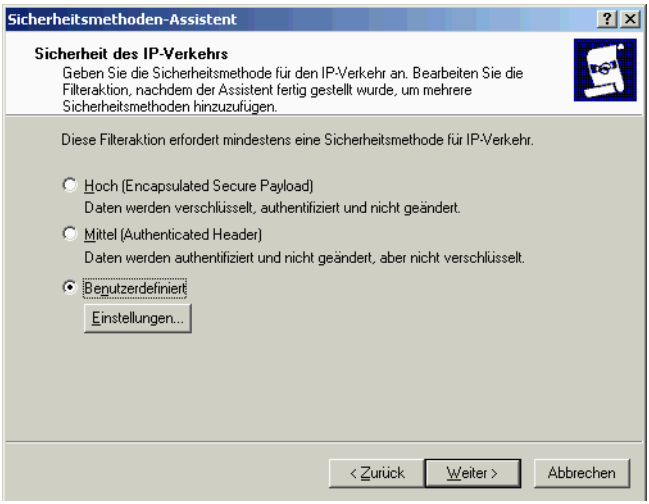


Wählen Sie die Option *Sicherheit aushandeln* und klicken Sie auf die Schaltfläche *Weiter* >.

► Kommunikation mit Computern, die kein IPSec unterstützen, festlegen:

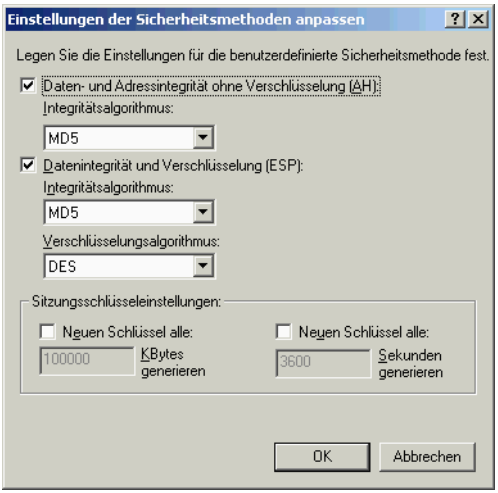
Wählen Sie die Option *Keine Kommunikation mit Computern zulassen, die IPSec nicht unterstützen* und klicken Sie auf die Schaltfläche *Weiter* >.

► Sicherheit des IP-Verkehrs festlegen:



Wählen Sie die Option *Benutzerdefiniert* und klicken Sie auf die Schaltfläche *Einstellungen...*

► Einstellungen für die Sicherheitsmethode festlegen:



Wählen Sie Folgendes aus:

- in der Liste *Integritätsalgorithmus (AH)* die Option *MD5*
- in der Liste *Integritätsalgorithmus (ESP)* die Option *MD5*
- in der Liste *Verschlüsselungsalgorithmus* die Option *DES*

Diese Werte müssen mit der Server-Konfiguration abgestimmt sein.

Klicken Sie auf die Schaltfläche *OK*.

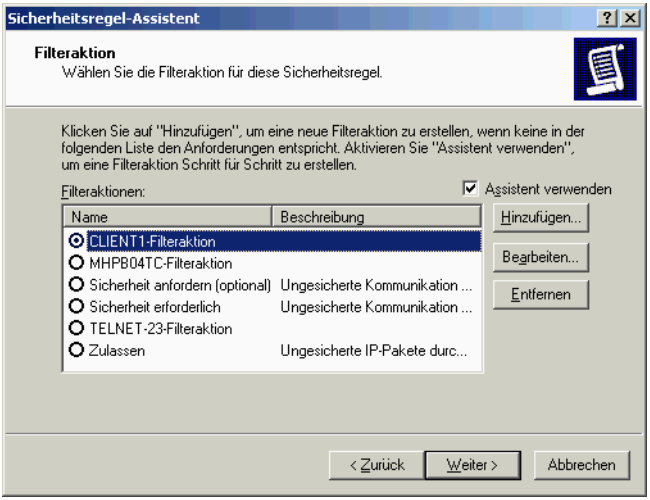
Der Sicherheitsmethoden-Assistent wird erneut angezeigt.

Klicken Sie auf die Schaltfläche *Weiter >*.

► Filteraktion fertigstellen:

Deaktivieren Sie die Option *Eigenschaften bearbeiten* und klicken Sie auf die Schaltfläche *Fertig stellen*.

► Filterregel für die Sicherheitsregel auswählen:

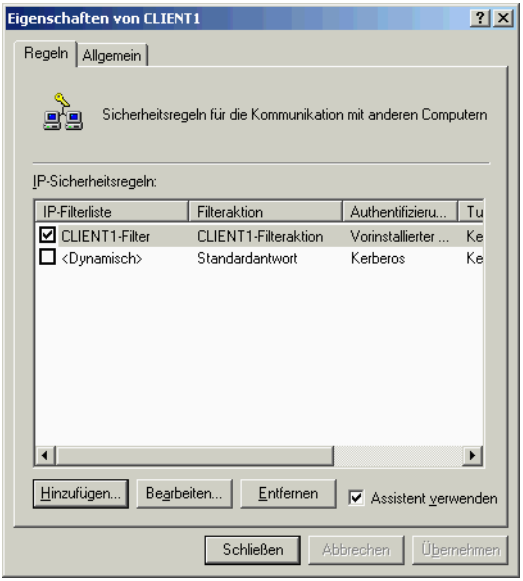


Wählen Sie die neu erstellte Filteraktion `CLIENT1-Filteraktion` aus und klicken Sie auf die Schaltfläche *Weiter >*.

► Regel fertigstellen:

Deaktivieren Sie die Option *Eigenschaften bearbeiten* und klicken Sie dann auf die Schaltfläche *Fertig stellen*.

► Sicherheitsregel abschließen:



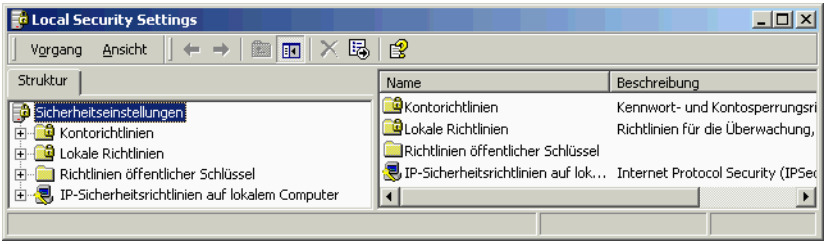
Klicken Sie auf die Schaltfläche *Schließen*.

IKE-Einstellungen

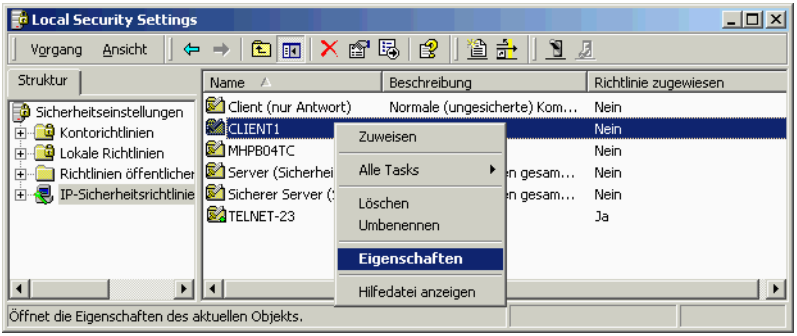
► Lokale Sicherheitseinstellungen öffnen:

Wählen Sie Menü *Start* -> *Programme* -> *Administrative Tools* -> *Local Security Policy*.

► Wählen Sie *IP-Sicherheitsrichtlinien auf lokalem Computer*.

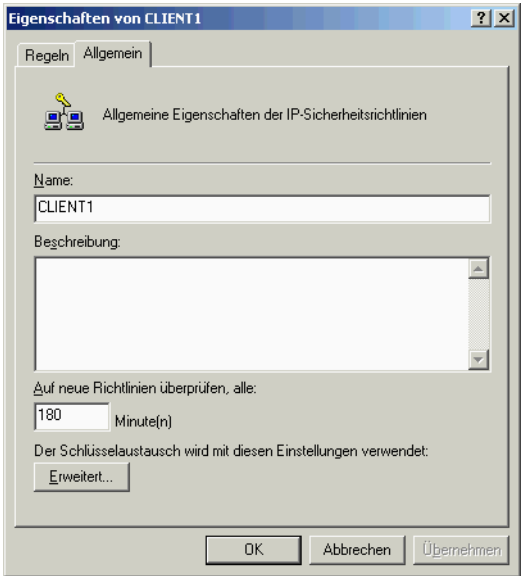


► Eigenschaften der gewünschten Sicherheitsrichtlinie auswählen:

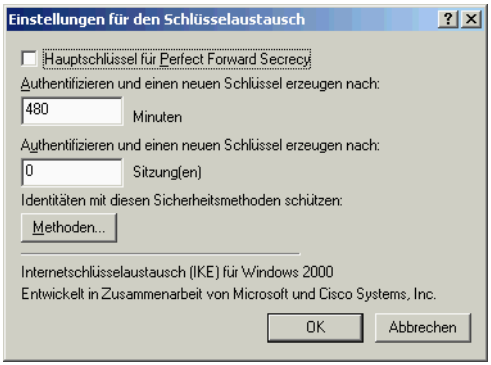


Klicken Sie mit der rechten Maustaste auf die Sicherheitsrichtlinie `CLIENT1` und wählen Sie im Kontextmenü die Option *Eigenschaften*.

► Wählen Sie im Fenster *Eigenschaften* das Register *Allgemein* aus und klicken Sie auf die Schaltfläche *Erweitert...*

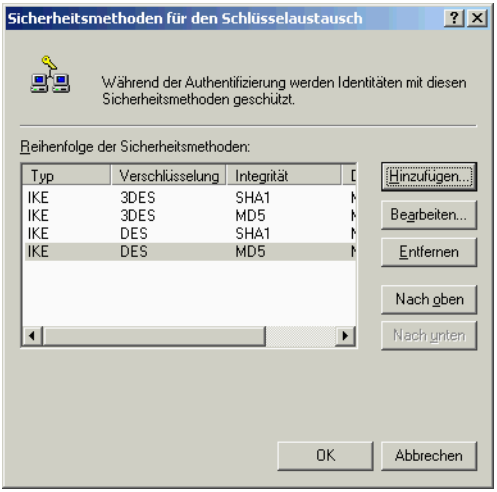


► Klicken Sie im Fenster *Einstellungen für den Schlüsselaustausch* auf die Schaltfläche *Methoden...*

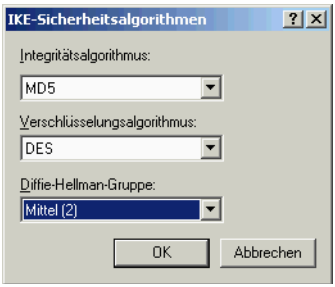


► Löschen Sie im Fenster *Sicherheitsmethoden...* die voreingestellten Methoden und klicken Sie auf die Schaltfläche *Hinzufügen...*

Wählen Sie die Einstellungen im Fenster *Sicherheitsmethoden...*so, dass sie denen der Serverseite entsprechen.



► Sicherheitsmethoden hinzufügen:



Wählen Sie im Fenster *IKE-Sicherheitsalgorithmen* Folgendes aus:

– in der Liste *Integritätsalgorithmus* die Option *MD5*
– in der Liste *Verschlüsselungsalgorithmus* die Option *DES*
– in der Liste *Diffie-Hellman-Gruppe* die Option *Mittel (2)*

Klicken Sie auf die Schaltfläche *OK*.

Das Fenster *Sicherheitsmethoden für den Schlüsselaustausch* wird erneut angezeigt.

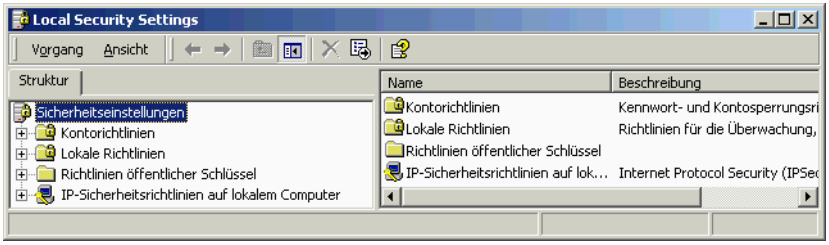
Klicken Sie auf die Schaltfläche *OK*.

Neue Sicherheitsrichtlinie zuweisen

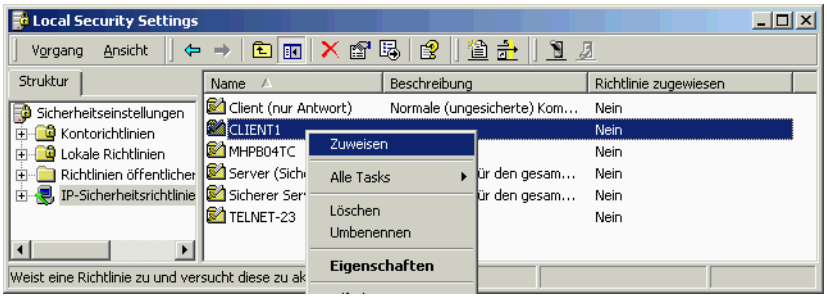
► Lokale Sicherheitseinstellungen öffnen:

Wählen Sie das Menü *Start* -> *Programme* -> *Administrative Tools* -> *Local Security Policy*

► Wählen Sie im Fenster *Local Security Settings* die Option *IP-Sicherheitsrichtlinien auf lokalem Computer* aus.



► Klicken Sie mit der rechten Maustaste auf die Richtlinie `CLIENT1` und wählen Sie im Kontextmenü die Option *Zuweisen*.



8.8 Kommandosschnittstelle

Die web-basierte CMX-Administration stellt Ihnen folgende Kommandos zur Verfügung:

- **add_cmxadm**
  ServerView-Konfigurationsdatei erweitern
- **del_cmxadm**
  Eintrag aus der ServerView-Konfigurationsdatei löschen
- **manage_cert**
  Zertifikate auf dem Administrationsserver verwalten
- **set_port**
  Portnummer in Konfigurationsdateien ändern
- **wca_init**
  SMAWwca aktivieren und deaktivieren
- **wca_stunnel**
  Stunnel starten und anhalten

Das Kommando *manage_cert* kann nur auf dem Administrationsserver aufgerufen werden, die anderen Kommandos sind nur auf den Kommunikationsservern verfügbar.

Die folgenden Abschnitte beschreiben die Kommandos in alphabetischer Reihenfolge.

8.8.1 add_cmxadm - ServerView-Konfigurationsdatei erweitern

Das Kommando *add_cmxadm* dient dazu, alle von ServerView für den Start der CMX-Administration benötigten Informationen in die ServerView-Konfigurationsdatei einzutragen. Dieser Eintrag bewirkt außerdem, dass auf der ServerView-Startseite eine Schaltfläche zum Start der CMX-Administration eingerichtet wird. Damit die Änderungen wirksam werden, muss ServerView erneut gestartet werden.

Syntax

/opt/SMAW/bin/add_cmxadm ␣applname␣{ pathname | **win** | **sol** }

applname
: Name der Anwendung, die hinzugefügt werden soll. Dieser Name wird unten auf der Anwendungsschaltfläche angezeigt (maximal 8 Zeichen).

pathname
: Vollständiger Pfadname der Java Web Start Anwendung auf dem System, auf dem *javaws* gestartet wird. Sie müssen den Pfadnamen nur angeben, wenn er vom Standardpfad abweicht, ansonsten geben Sie die Option *win* oder die Option *sol* an.

  Enthält der Pfadname für den Windows-Client Sonderzeichen wie Leerzeichen, dann muss der String in Anführungsstrich gefolgt von Hochkomma gebettet werden.

  Beispiel für JRE 1.4.2_04:
  '" C:/Program Files/Java/j2re1.4.2_04/javaws/javaws.exe "'

  Beispiel für JRE 1.5.0_02:
  '" c:/Program Files/Java/jre1.5.0_02/bin/javaws.exe "'

win
: Es wird der Standardpfad für JRE 1.5.0_02 auf Windows-Systemen verwendet:
  c:/Program Files/Java/jre1.5.0_02/bin/javaws.exe

sol
: Es wird der Pfad für JRE 1.5.0_02 auf Solaris-Systemen verwendet:
  /opt/SMAW/SMAWj2rt/jre/solaris/jre1.5.0_02/javaws/javaws

8.8.2 del_cmxadm - Eintrag aus ServerView-Konfigurationsdatei löschen

Das Kommando *del_cmxadm* löscht alle von ServerView für den Start der CMX-Administration benötigten Informationen aus der ServerView -Konfigurationsdatei.

Syntax

/opt/SMAW/bin/del_cmxadm applname

applname
: Name der Anwendung, die gelöscht werden soll. Dieser Name wird unten auf der Anwendungsschaltfläche angezeigt (maximal 8 Zeichen).

8.8.3 manage_cert - Zertifikate verwalten

Das Skript *manage_cert* stellt eine benutzerfreundliche Schnittstelle zu den *OpenSSL* Zertifizierungsprogrammen zur Verfügung. Dabei beschränkt es sich auf die Argumente, die im Zusammenhang mit der CMX-Administration relevant sind. *manage_cert* greift auf die CMX-administrationsspezifischen Standardwerte zu, die in den *OpenSSL* Konfigurationsdateien festgelegt sind.

Das Kommando wird auf dem Administrationsserver aufgerufen.

Syntax

/opt/SMAW/SMAWcmx/bin/manage_cert
[␣**?** | ␣**-h** | ␣**-help**]
{ ␣**-newca** [␣**-days** number] |
␣**-newkey** |
␣**-newreq** [␣comm-server] |
␣**-sign** [␣comm-server] |
␣**-finish** [␣comm-server] |
␣**-verify** |
␣**-print**␣certificate }

? | **-h** | **-help**
Gibt die Kommandosyntax aus.

-newca [␣**-days** number]
Erzeugt eine neue CA, falls diese noch nicht existiert. Nach dem Drücken der ENTER-Taste werden Details zur CA erfragt. Sie können die vorgegebenen Standardwerte durch Eingabe von ENTER übernehmen. Alle relevanten Dateien werden in dem Verzeichnis */opt/SMAW/SMAWswca/PrivateCA* erzeugt.

number gibt die Gültigkeitsdauer des Zertifikats in Tagen an.
Standard: 365

-newkey
Erzeugt einen neuen privaten Schlüssel speziell für Ihren Server (*/opt/SMAW/SMAWswca/ssl/private/wca_privkey.pem*). Beachten Sie bitte, dass dieser private Schlüssel nicht verschlüsselt wird, da *Stunnel* keine Möglichkeit bietet, das Passwort für den Schlüssel vom Benutzer zu erfragen.

-newreq [␣comm-server]
Erzeugt einen Zertifikatsantrag (Certificate Signing Request CSR). Dieser wird in */opt/SMAW/SMAWswca/ssl/req/comm-server.newreq.pem* gespeichert.

comm-server ist der Name des Kommunikationsservers, für den das Server-Zertifikat erstellt werden soll.
Standard: *`uname -n`*

-sign [␣comm-server]
Aufruf des *OpenSSL* CA-Programms zum Signieren eines Zertifikatantrags. Das Programm erwartet den Antrag in */opt/SMAW/SMAWswca/ssl/req/comm-server.newreq.pem*.
Das signierte Zertifikat wird nach */opt/SMAW/SMAWswca/ssl/certs/comm-server.newcert.pem* geschrieben.

comm-server ist der Name des Kommunikationsservers, für den das Server-Zertifikat erstellt werden soll.
Standard: *`uname -n`*

-finish [␣comm-server]
Erzeugt das Server Zertifikat (*/opt/SMAW/SMAWswca/ssl/certs/comm-server.pem*) in dem Format, in dem es von *Stunnel* benötigt wird. *manage_cert* erwartet das signierte Zertifikat in */opt/SMAW/SMAWswca/ssl/certs/comm-server.newcert.pem*, den zugehörigen Request in */opt/SMAW/SMAWswca/ssl/req/comm-server.newreq.pem* und den privaten Schlüssel in */opt/SMAW/SMAWswca/ssl/private/wca_privkey.pem*.

comm-server ist der Name des Kommunikationsservers, für den das Server- Zertifikat erstellt werden soll.
Standard: *`uname -n`*

-verify
verifiziert alle Zertifikate, die sich im Verzeichnis */opt/SMAW/SMAWswca/ssl/certs* befinden.

-print␣certificate
Zeigt den Inhalt des mit *certificate* angegebenen Zertifikats an.

Ende-Status

0 erfolgreiche Beendigung

1 keine oder fehlerhafte Argumente

2 Ausgabedatei existiert bereits

3 Eingabedatei fehlt

x bezeichnet den *OpenSSL*-Returncode

Dateien

/opt/SMAW/SMAWswca/conf/openssl_wca.cnf
OpenSSL-Konfigurationsdatei mit *Stunnel*-spezifischen Standardwerten.

/opt/SMAW/SMAWswca/conf/openssl_ca.cnf
OpenSSL-Konfigurationsdatei mit CA-spezifischen Standardwerten.

8.8.4 set_port - Portnummer ändern

Mit dem Kommando *set_port* können Sie die Portnummer in allen Konfigurationsdateien ändern, die die CMX-Administration betreffen. Dies ist immer dann erforderlich, wenn die voreingestellte Portnummer 910 bereits durch eine andere Anwendung belegt ist.

Syntax

/opt/SMAW/bin/setport ␣new-port [␣old-port]

new-port
Neue Portnummer, die in die Konfigurationsdateien eingetragen werden soll.

old-port
Bisherige Portnummer
Voreinstellung: 910.

Dateien

/opt/SMAW/SMAWcmx/wca/stunnel/conf/stunnel_wca.conf

/opt/SMAW/SMAWcmx/wca/telnet.cfg

/opt/SMAW/SMAWcmx/wca/telnet-windows.cfg

8.8.5 wca_init - SMAWwca aktivieren und deaktivieren

Mit *wca_init* aktivieren und deaktivieren Sie in einer aktiven Bootumgebung die web-basierte CMX-Administration. Beim Aktivieren werden die relevanten Konfigurationsdateien erzeugt und beim Deaktivieren wieder aus dem System entfernt. Falls *Stunnel* installiert ist, wird *Stunnel* beim Aktivieren ebenfalls gestartet und beim Deaktivieren beendet.

Zusätzlich können Sie mit dem Kommando *wca_init status* den aktuellen Status der Administrationssoftware erfragen.

Syntax

/opt/SMAW/bin/wca_init ␣{ **start** | **stop** | **status** }

start
Die web-basierte CMX-Administration wird aktiviert.

stop
Die web-basierte CMX-Administration wird deaktiviert.

status
Der aktuelle Status wird ausgegeben.

Beispiel

```
# wca_init status
SMAWwca started
stunnel(/opt/SMAW/SMAWcmx/wca/stunnel/conf/stunnel_wca.c
onf) is running
```

8.8.6 wca_stunnel - Stunnel starten und anhalten

Mit *wca_stunnel start* starten Sie den *Stunnel*-Dämon mit einer CMX-administrationsspezifischen Konfigurationsdatei. *Stunnel* arbeitet in diesem Fall als SSL-Wrapper für die Telnet-Verbindung zwischen Administrationsclient und Kommunikationsserver.

Das Server Zertifikat wird in *opt/SMAW/SMAWcmx/wca/stunnel/certs/comm-server.pem* erwartet. *Stunnel* akzeptiert Verbindungen standardmäßig über Port 910.

Mit *wca_stunnel stop* wird dieser Dämon-Prozess beendet.

Syntax

/opt/SMAW/SMAWcmx/bin/wca_stunnel ␣{ **start** | **stop** }

start
Startet den *Stunnel*-Dämon mit einer CMX-administrationsspezifischen Konfigurationsdatei.

stop
Beendet den *Stunnel*-Dämon für die CMX-Administration.

Ende-Status

0 erfolgreiche Ausführung

1 falsches Argument

2 *Stunnel* ist bereits gestartet

3 Port 910 ist belegt

4 *Stunnel* konnte nicht gestartet werden

5 Das angegebene Zertifikat existiert nicht

6 *SMAWPstun* ist nicht installiert

7 *Stunnel* ist nicht gestartet (bei *wca_stunnel stop*)

8 no autorization

9 kill wca_stunnel failed

Datei

/opt/SMAW/SMAWcmx/wca/stunnel/conf/stunnel_wca.conf
Stunnel-Konfigurationsdatei

8.9 Problemfälle lösen

Beim Starten von ServerView können verschiedene Probleme auftreten. Im Folgenden sind einige Problemfälle benannt und Umgehungen/Lösungen aufgezeigt.

1. Seite wurde nicht gefunden

   Anzeige:

   ```
   Die Seite kann nicht angezeigt werden.

   Die gewünschte Seite ist zur Zeit nicht verfügbar. Möglicherweise sind technische Schwierigkeiten aufgetreten oder Sie sollten die Browsereinstellungen überprüfen.
   ```

   Fehlerbehebung:

   - Prüfen Sie mit dem Kommando *ps -ef | grep http*, ob der Apache-Server läuft.
   - Wenn das Ergebnis negativ ist, starten Sie den Apache-Server mit folgenden Kommandos:

     1. cd /etc/rc2.d
     2. sh S97SIapache

2. JAVA WebStart Download-Fehler

   Anzeige im Detail-Bildschirm:

   ```
   Beim Starten/Ausführen der Anwendung ist ein Fehler aufgetreten.
   Titel: Serverview(MHPBO4TC)
   Hersteller: Fujitsu Siemens Computers
   Kategorie: Download-Fehler

   Server lieferte fehlerhaften MIME-Typ beim Zugriff auf Ressource: http://MHPBO4TC:8881/wsa.jnlp - text/html
   ```

   Fehlerbehebung:

   Ändern Sie die WebStart-Konfiguration wie folgt:

   - Wählen Sie Menü *Start* -> *Programme* -> *Java Web Start* -> *Datei* -> *Einstellungen*.
   - Wählen Sie für *Proxy-Server* die Option *Keine*.

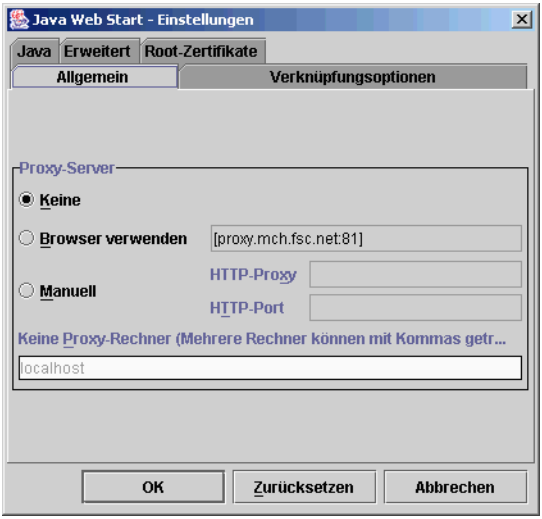


Bild 28: Java Web Start Einstellungen

3. Keine Reaktion beim Starten von ServerView (nur kurzes Flackern)

   In diesem Fall wird Java Web Start nicht gestartet, da für jnlp-Dateien fehlerhafte Ordneroptionen eingestellt sind.

   Fehlerbehebung:

   - Wählen Sie im Windows Explorer das Menü *Extras* -> *Ordneroptionen* -> *Dateitypen*.
   - Unter *Öffnen mit* muss *javaws* angegeben sein.
     Klicken Sie gegebenenfalls auf die Schaltfläche *Ändern...*, um diesen Eintrag zu ändern.

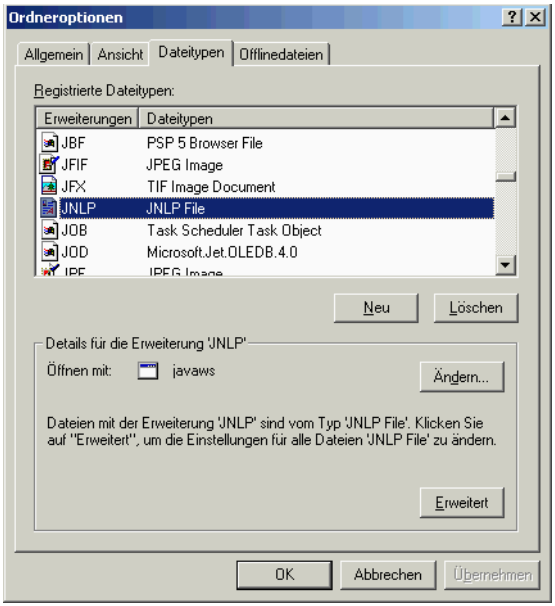


Bild 29: Ordneroptionen in Windows

4. Keine Reaktion beim Starten der CMX-Administration durch Klicken auf die Schaltfläche CMXADM

   In diesem Fall wird Java Web Start nicht gestartet. Dies kann an einem fehlerhaften Eintrag in der ServerView-Konfigurationsdatei liegen. Bitte überprüfen Sie den Pfad zur Java Web Start Anwendung, den Sie mit Hilfe des Kommandos *add_cmxadm* in die Konfigurationsdatei */opt/SMAW/public_html/WSAConfig* eingetragen haben.

5. Fehler beim Starten der CMX-Administration über SSL/TLS

   Hierbei können die zwei folgenden Meldungsboxen ausgegeben werden:

Bild 30: Fehlermeldung beim Start von CMXADM (1)

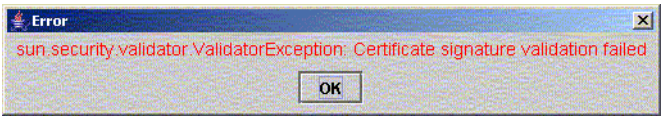


Bild 31: Fehlermeldung beim Start von CMXADM (2)

Im Fall (1) wurde *Stunnel* noch nicht gestartet. Starten Sie *Stunnel* mit dem Kommando *wca_stunnel start*.

Im Fall (2) haben sich möglicherweise Ihre Zertifikate und/oder Schlüssel geändert, ohne dass *Stunnel* erneut gestartet wurde. Beenden Sie deshalb *Stunnel* mit *wca_stunnel stop* und starten Sie *Stunnel* erneut mit *wca_stunnel start*.

9 Verbindungen über RFC1006 konfigurieren

Der Internet *Request for Comments* RFC1006 wurde durch das Internet Architecture Board (IAB) herausgegeben, um oberhalb von TCP/IP einen OSI-Transportdienst gemäß ISO 8072 zu definieren.

Das Transmission Control Protocol (TCP) ist verbindungsorientiert, d. h., vor der eigentlichen Datenübertragung wird eine logische Verbindung aufgebaut. TCP sorgt dafür, dass Daten in der richtigen Reihenfolge zur Anwendung gelangen und dass keine Daten verlorengehen oder verfälscht werden.

Beim Transportdienst gemäß ISO 8072 werden Nachrichten durch Endemarken voneinander abgegrenzt; der Datenfluss ist nachrichtenorientiert. Beim TCP gibt es solche Endemarken nicht; ein kontinuierlicher Datenstrom wird übertragen. TCP-Anwendungen können nur anhand des Dateninhalts erkennen, wo eine logische Nachricht zu Ende ist und die nächste beginnt.

Ein weiterer Unterschied zwischen TCP/IP und dem ISO-Transportdienst besteht beim Verbindungsabbau: TCP garantiert, dass beim Verbindungsabbau alle vorher gesendeten Daten den Empfänger erreichen (orderly release). Gemäß ISO 8073 liegt die Verantwortung dafür, dass Verbindungen erst nach erfolgtem Datenaustausch beendet werden (abortive release), nicht beim Transportsystem, sondern bei den Anwendungen selbst.

RFC1006 ist in CMX als Transport Service Provider (TSP) implementiert. Der TSP RFC1006 bietet seine Dienste über das von The Open Group standardisierte Transport-Provider-Interface (TPI) an, auf das werden die Programmierschnittstellen ICMX, XTI und TLI intern abgebildet werden.

Der TSP RFC1006 wird automatisch mit CMX gestartet. Sie können Tuning-Parameter einstellen und sich Status und Statistiken anzeigen lassen.

9.1 Übersicht der Konfigurationsdaten

Die folgende Abbildung gibt einen Überblick der Konfigurationsdaten, die für den Verbindungsaufbau über RFC1006 von Bedeutung sind.

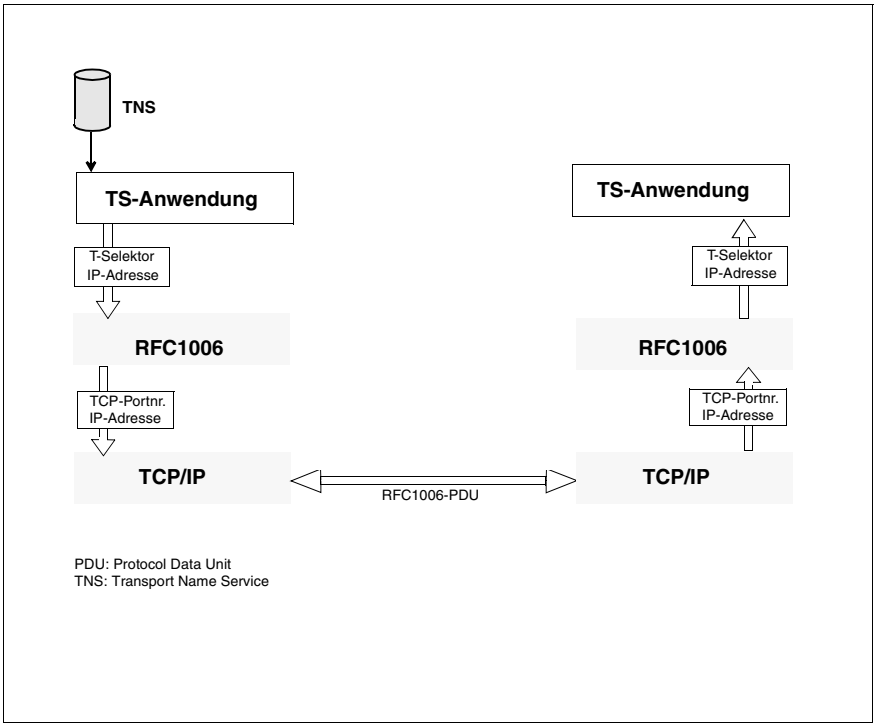


Bild 32: Verbindungsaufbau über RFC1006 - Überblick

Eine Transportverbindung über den RFC1006-TSP ist eindeutig einer TCP-Verbindung zugeordnet. Zum Aufbau der TCP-Verbindung werden IP-Adresse und Portnummer benötigt. Das Paar IP-Adresse und Portnummer wird im Folgenden als TCP-Adresse bezeichnet. Eine TCP-Adresse identifiziert eindeutig einen TCP-Verbindungsendpunkt. Somit wird eine TCP-Verbindung eindeutig durch ein Paar von TCP-Adressen, nämlich des lokalen und fernen Endpunkts, identifiziert. Eine TS-Anwendung meldet sich mit einem sogenannten Transport-Selektor, kurz T-Selektor genannt, beim RFC1006-TSP an. Sie wird auf der Protokollebene des RFC1006-TSP über diesen T-Selektor adressiert.

Ankommende TCP-Verbindungsanforderungen zeigt TCP dem RFC1006-TSP über einen sogenannten „TCP listening stream“, kurz TCP-Listener, an. Wenn der RFC1006-TSP gestartet wird, richtet er einen TCP-Listener mit der TCP-Adresse ’*.iso-tsap’ ein. Hierbei steht ’*’ für die IP-Adresse 0.0.0.0, die die Gesamtheit der lokalen Netzzugangspunkte bezeichnet, und ’iso-tsap’ steht für die Portnummer 102, die gemäß RFC1006 für den ISO-Transport-Dienst reserviert ist. Das bedeutet, dass alle an einem beliebigen lokalen Netzzugangspunkt für Portnummer 102 eintreffenden TCP-Verbindungsanforderungen dem RFC1006-TSP angezeigt werden. Darüberhinaus können TS-Anwendungen durch die Verwendung spezieller Adressen und Optionen den RFC1006-TSP dazu veranlassen, weitere TCP-Listener einzurichten. Über diese werden die Verbindungsanforderungen zugestellt, die an einem ganz bestimmten lokalen Netzzugangspunkt angekommen sind, oder die für eine andere Portnummer als 102 bestimmt sind.

Bei einer abgehenden TCP-Verbindungsanforderung wird für den fernen Endpunkt stets seine genaue TCP-Adresse, gewöhnlich mit Portnummer 102, angegeben. Der lokale Endpunkt dagegen wird offen gelassen, indem dafür die TCP-Adresse ’*.*’ (alle Adress-Bits auf 0 gesetzt) angegeben wird. TCP wählt dann für den lokalen Endpunkt eine zu diesem Zeitpunkt unbenutzte Portnummer aus, während IP einen geeigneten Netzzugangspunkt auswählt. Auch hier kann die TS-Anwendung, die die Verbindung aktiv aufbaut, durch die Verwendung spezieller Adressen und Optionen den RFC1006-TSP dazu veranlassen, für den fernen Endunkt eine andere Portnummer als 102 und für den lokalen Endunkt eine bestimmte IP-Adresse zu verwenden. Die Portnummer des lokalen Endpunkts bleibt jedoch stets unbestimmt. Gewöhnlich meldet sich die TS-Anwendung für den aktiven Verbindungsaufbau nur mit einem T-Selektor an und gibt als Partneradresse nur IP-Adresse und T-Selektor an.

Eine TNS-Anwendung entnimmt die für den Verbindungsaufbau benötigten Adressen der Datenbasis des TNS.

9.1.1 Konfigurationsdaten für lokale TS-Anwendungen

Um eine Transport-Verbindung über den TSP RFC1006 aufzubauen, definieren Sie im Regelfall eine Transportsystem-Anwendung auf Ihrem eigenen System (im Folgenden lokale TS-Anwendung genannt) und eine weitere für jede Partner-Anwendung, die Sie erreichen wollen (im Folgenden ferne TS-Anwendung genannt).

Um eine lokale TS-Anwendung im TNS zu konfigurieren, müssen Sie im Regelfall folgende Daten angeben:

- Einen GLOBALEN NAMEN, mit dem die Anwendung im TNS identifiziert werden kann. Zu Struktur und Merkmalen des GLOBALEN NAMENS siehe Abschnitt „GLOBALER NAME“ auf Seite 77.
- Das Adress-Format, das den TSP identifiziert, über den die Verbindung aufgebaut werden soll. Der TSP RFC1006 wird über die Formate *RFC1006* oder *LANINET* identifiziert.
- Beim Adress-Format LANINET eine Portnummer, für die der RFC1006-TSP gegebenenfalls einen TCP-Listener für den passiven Verbindungsaufbau einrichtet.

  Beispiel:

  ```
  TSEL LANINET A'1100'
  ```

  Wenn sich die TS-Anwendung für passiven Verbindungsaufbau anmeldet, richtet der RFC1006-TSP gegebenenfalls einen TCP-Listener mit Portnummer 1100 ein.

  Diese Angabe wird nur dann benötigt, wenn ein Partnersystem **nicht** die RFC1006-Standard-TCP-Portnummer 102 als Endpunkt der TCP-Verbindung adressiert. Das ist zum Beispiel bei RFC1006-Implementierungen älterer CMX-Versionen als CMX V5.0 der Fall, und in *open*FT-Anwendungen, die auf Rechnern anderer Hersteller ablaufen.
- Beim Adress-Format RFC1006 den T-Selektor,
  - über den ferne TS-Anwendungen die lokale TS-Anwendung, die für passiven Verbindungsaufbau angemeldet ist, adressieren und
  - der beim aktiven Verbindungsaufbau dem Partnersystem als Adressinformation im RFC1006-Protokoll in Form des Parameters *calling TSAP-ID* mitgeteilt wird.

  Zum Beispiel verwenden *open*FT-Anwendungen für den passiven Verbindungsaufbau den T-Selektor *T'$FJAM'* und für den aktiven Verbindungsaufbau die T-Selektoren *T'$FJAM001'*, *T'$FJAM002'*, usw.

Sie können die Adress-Formate LANINET und RFC1006 jeweils allein oder beide zusammen angeben.

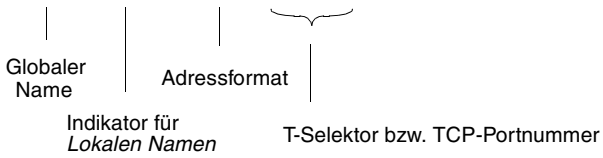
Wenn Sie nur das Adress-Format LANINET angegeben haben, handelt es sich um eine sogenannte reine LANINET-Anwendung. Wenn eine solche TS-Anwendung für passiven Verbindungsaufbau angemeldet ist, beansprucht sie einen eigenen TCP-Listener, d.h. keine andere TS-Anwendung kann sich mit dieser Portnummer für passiven Verbindungsaufbau anmelden. Umgekehrt scheitert die Anmeldung der reinen LANINET-Anwendung, falls bereits ein TCP-Listener mit dieser Portnummer existiert. Beim aktiven Verbindungsaufbau einer reinen LANINET-Anwendung wird die angegebene Portnummer dem Partnersystem als Adressinformation über das RFC1006-Protokoll in Form des Parameters *calling TSAP-ID* mitgeteilt.

Wenn Sie beide Adress-Formate angegeben haben, können sich mehrere TS-Anwendungen denselben TCP-Listener und die dazugehörige Portnummer teilen. Der mit dem Adress-Format RFC1006 angegebene T-Selektor muss jedoch eindeutig sein und kann nur genau einer für passiven Verbindungsaufbau angemeldeten TS-Anwendung angehören. Sowohl nur mit Adress-Format RFC1006 als auch für beide Adress-Formate angemeldete TS-Anwendungen können Verbindungsanforderungen über den TCP-Listener mit Adresse '*.isotsap' empfangen.

T-Selektor und Portnummer mit den entsprechenden Adress-Format-Angaben stehen bei lokalen TS-Anwendungen im LOKALEN NAMEN, der mit dem Indikator TSEL vor jeder Adress-Format-Angabe gekennzeichnet wird. Der LOKALE NAME ist einem GLOBALEN NAMEN zugeordnet.

Beispiel-Eintrag für eine lokale TS-Anwendung

```
$FJAM TSEL RFC1006 T´$FJAM´
          TSEL LANINET A´1100´
```



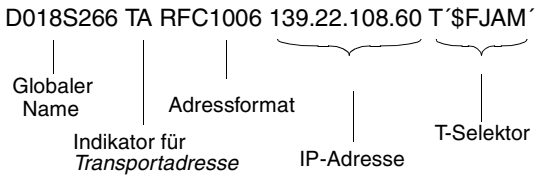
9.1.2 Konfigurationsdaten für ferne TS-Anwendungen

Um eine ferne TS-Anwendung im TNS zu konfigurieren, müssen Sie folgende Daten angeben:

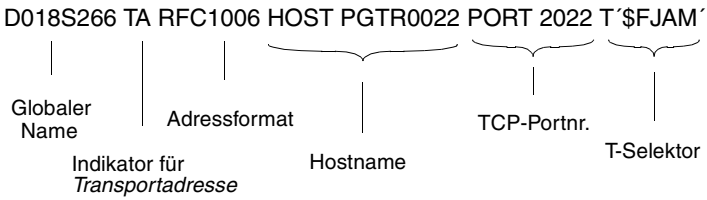
- Den GLOBALEN NAMEN, mit dem die Anwendung im TNS identifiziert werden kann. Zu Struktur und Merkmalen des GLOBALEN NAMENS siehe Abschnitt „GLOBALER NAME“ auf Seite 77.
- Die IP-Adresse des fernen Systems (oder alternativ den Hostnamen).
- Das Adress-Format, das den Transport Service Provider identifiziert, über den das ferne System erreicht werden soll. Der TSP RFC1006 wird über die Formate *RFC1006* oder *LANINET* identifiziert.
- Beim Adress-Format RFC1006
  - den T-Selektor, der zur Identifizierung der TS-Anwendung im fernen System dient und der dem Partnersystem als Adressinformation über das RFC1006-Protokoll in Form des Parameters *called TSAP-ID* mitgeteilt wird
  - optional eine Portnummer, die den fernen Endpunkt der TCP-Verbindung identifiziert. Ohne die Portnummer-Angabe wird Portnummer 102 adressiert.
- Beim Adress-Format LANINET die Portnummer, die den fernen Endpunkt der TCP-Verbindung identifiziert und die zusätzlich als Adressinformation im RFC1006-Protokoll in Form des Parameters *called TSAP-ID* übertragen wird.

IP-Adresse oder Hostname, Format, evtl. die TCP-Portnummer sowie der T-Selektor stehen bei fernen TS-Anwendungen in der TRANSPORTADRESSE, die mit dem Indikator TA gekennzeichnet wird. Die TRANSPORTADRESSE ist einem GLOBALEN NAMEN zugeordnet.

Beispiel-Einträge für eine ferne TS-Anwendung



Im obigen Beispiel wird die Anwendung über die Standardportnummer 102 adressiert.



9.2 Verbindung zu fernem Partnersystem aufbauen

In diesem Abschnitt finden Sie Informationen zu den Abläufen beim Verbindungsaufbau und zur Vorgehensweise bei der Konfigurierung am Beispiel eines Verbindungsaufbaus zwischen zwei Systemen. Beide Systeme nutzen eine CMX-Version ab V5.0. Die Abläufe im aktiven und passiven System werden jeweils aus der Sicht des betreffenden Systems in einem eigenen Abschnitt beschrieben.

9.2.1 Aktiver Verbindungsaufbau

Zunächst ermittelt die CMX-Anwendung über den TNS ihren LOKALEN NAMEN und meldet sich mit diesem für den aktiven Verbindungsaufbau bei CMX an. Danach liest die CMX-Anwendung die TRANSPORTADRESSE (IP-Adresse oder Hostname, T-Selektor) der Partner-Anwendung aus der TNS-Datenbasis. Ist in der TNS-Datenbasis der Hostname angegeben, so wird dieser dynamisch beim Abruf der Adresse in die IP-Adresse übersetzt.

Die CMX-Anwendung übergibt diese Information an RFC1006-TSP. Dieser baut mittels IP-Adresse und Standard-Portnummer 102 eine TCP-Verbindung zum RFC1006-TCP im Partnersystem auf. Über diese sendet dann der RFC1006-TSP den T-Selektor der lokalen CMX-Anwendung in Form des Parameters *calling TSAP-ID* und den T-Selektor der fernen CMX-Anwendung in Form des Parameters *called TSAP-ID* in einem RFC1006-Protokoll-Element (CR TPDU) an den fernen RFC1006-TSP. Der weitere Ablauf ist im folgenden Abschnitt aus der Sicht des passiven Systems beschrieben.

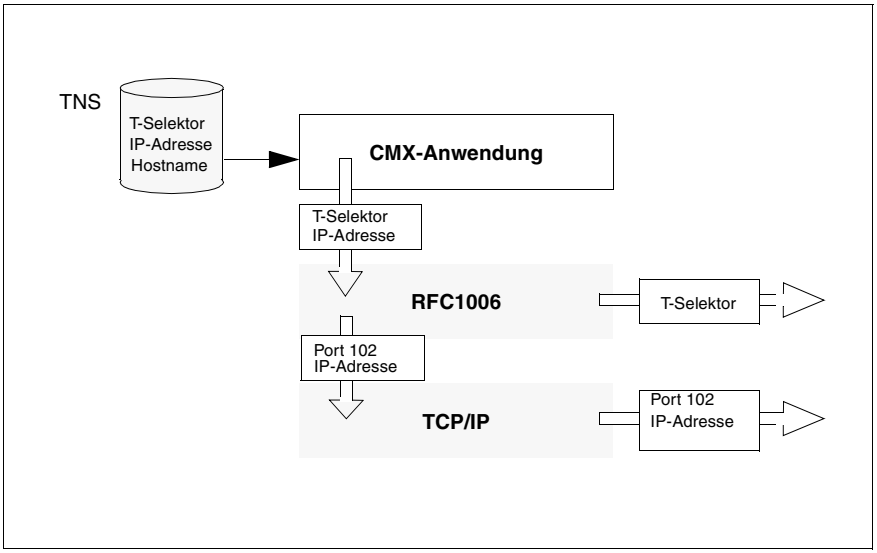


Bild 33: Aktiver Verbindungsaufbau zwischen Systemen mit CMX V5

Angenommen, das ferne System mit CMX ab V5.0 hätte den Hostnamen PGRTV009 und die IP-Adresse 76.3.13.11, dann wäre für die ferne Anwendung einer der beiden folgenden Einträge erforderlich:

```
rloginrem TA RFC1006 HOST PGRTV009 T'rlogin'
oder
rloginrem TA RFC1006 76.3.13.11 T'rlogin'
```

9.2.2 Passiver Verbindungsaufbau

Die CMX-Anwendung liest vor dem Verbindungsaufbau ihren LOKALEN NAMEN aus dem TNS und meldet sich mit diesem für passiven Verbindungsaufbau bei CMX an. Unabhängig von der Anmeldung der CMX-Anwendung hat der RFC1006-TSP den TCP-Listener '*.iso-tsap' eingerichtet, und „hört" damit auf alle an einem beliebigen lokalen Netzzugangspunkt für Portnummer 102 ankommenden TCP-Verbindungsanforderungen.

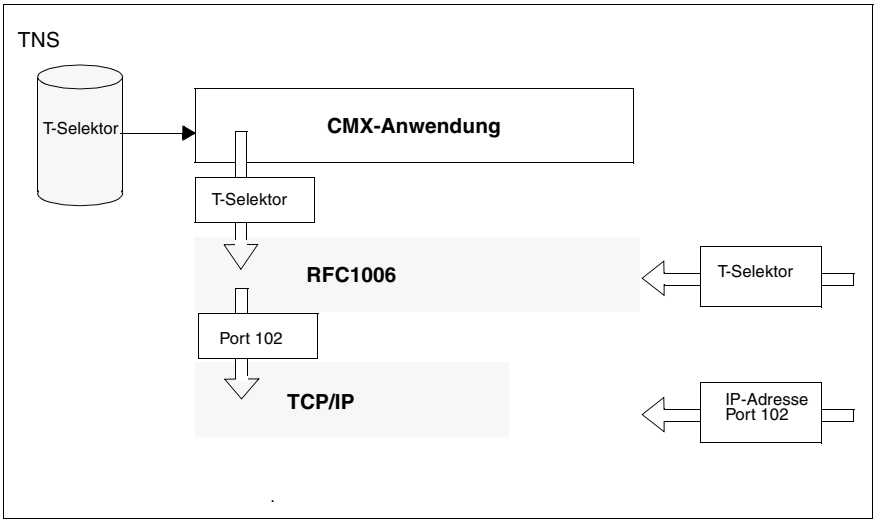


Bild 34: Passiver Verbindungsaufbau zu Partnersystemen mit CMX V5

Ausgelöst durch die Initiative der Partner-Anwendung erhält der RFC1006-TSP über den obengenannten TCP-Listener eine Verbindungsanforderung mit der Quell-Adresse (IP-Adresse und Portnummer) des fernen TCP-Verbindungsendpunkts und der Ziel-adresse '*.iso-tsap'. Nachdem diese TCP-Verbindung etabliert ist, empfängt der RFC1006-TSP über diese ein RFC1006-Protokoll-Element (CR TPDU). Dieses Protokoll-Element enthält in Form des Parameters *calling TSAP-ID* den T-Selektor der rufenden CMX-Anwendung und in Form des Parameters *called TSAP-ID* den T-Selektor der gerufenen CMX-Anwendung, nämlich der angemeldeten lokalen CMX-Anwendung. Dieser CMX-Anwendung wird dann eine Verbindungsanforderung zugestellt und dabei die IP-Adresse des fernen TCP-Verbindungsendpunkts und der T-Selektor der rufenden CMX-Anwendung in Form der
TRANSPORTADRESSE mitgeteilt. Die lokale CMX-Anwendung hat dann die Möglichkeit über den TNS mittels dieser TRANSPORTADRESSE den GLOBALEN NAMEN der Partner-Anwendung zu ermitteln.

Ihr CMX V5-System muss im TNS in folgender Weise konfiguriert werden (lokale Anwendung, für Verbindungsaufbau ankommend von der CMX V5-Partneranwendung):

```
rloginlocal TSEL RFC1006 T´rlogin´
```

9.3 Status/Statistiken des RFC1006-TSP abfragen (rfc1006stat)

Mit diesem Kommando können Sie Status und Statistiken des RFC1006-TSP abfragen.

rfc1006stat ␣ [**-d**] ␣ [**-s**]

rfc1006stat ␣ [**-a**] ␣ [**-c**] ␣ [**-d**] ␣ [**-l**] ␣ [**-n**]

rfc1006stat ␣ **-z**

rfc1006stat gibt den Inhalt verschiedener Datenstrukturen des RFC1006-TSP aus, der den ISO-8072-Transport-Dienst über TCP/IP entsprechend RFC1006 zur Verfügung stellt.

Es gibt folgende drei Modi des Kommandos (siehe oben):

- Im ersten Modus – ohne die Optionen *-a*, *-c* und *-z* – zeigt *rfc1006stat* folgende Inhalte an:
  - TSP-Status
  - maximale und aktuelle Anzahl von Transport-Verbindungen
  - maximale und aktuelle Anzahl von TSAPs, die für passiven Verbindungsaufbau angemeldet sind (im Folgenden kurz als Listener-TSAPs bezeichnet)
  - maximale TIDU-Größe
  - bevorzugte TPDU-Größe
  - maximale TIDU-Größe zum TCP
  - die für den Nutzer des Transport-Dienstes wichtigsten Statistiken

  Wenn Sie die Option *-s* angegeben haben, werden zusätzlich spezielle Einstellungen, auf den Pool der freien TCP-Streams bezogene Zähler und Statistiken sowie der Mangel an diversen STREAMS-Ressourcen angezeigt.

  Wenn Sie die Option *-d* angegeben haben, werden weitere Statistiken für Debugzwecke ausgegeben.
- Im zweiten Modus – Option *-a* und/oder *-c* – erhalten Sie detaillierte Informtionen über die aktuell vorhandenen TCEPs und/oder Listener-TSAPs. Das Ausgabeformat wird im Abschnitt „TSAP/TCEP Ausgabeformat“ auf Seite 218 beschrieben.

Mit der Option *-d* werden alle geöffneten RFC1006-Streams mit den dazugehörigen TCP Streams unabhängig von ihrem TPI-Zustand angezeigt. Wie im ersten Modus beginnt die Ausgabe mit der Basisinformation über den TSP-Status, die erlaubte und maximale Anzahl von TSAPs und TCEPs usw. Statistiken werden jedoch keine ausgegeben.

- Der dritte Modus setzt die Statistiken zurück ohne den RFC1006-TSP neu zu starten.

Optionen

-a access points

gibt detaillierte Informationen aus über Listener-TSAPs und über TCEPs, die nicht den TPI-Zustand TS_DATA_XFER haben. Solange der RFC1006-TSP im betriebsfähigen Zustand ist, ist ein Listener-TSAP aktiv, der an den Kontroll-Stream (UPPER = CONTROL) gebunden ist.

-c connection end points

gibt detaillierte Informationen über TCEPs im TPI-Zustand TS_DATA_XFER aus.

-d internal details

gibt zusätzliche Informationen über interne Details für Debugzwecke aus.

-l long line format

verbindet die beiden Zeilen des TSAP-/TCEP-Ausgabeformates miteinander.

-n number

gibt IP-Adressen und Portnummern als Zahl aus statt als symbolische Namen.

-s special

gibt spezielle Zusatzinformation wie Einstellungen, Zähler und Statistik bezüglich Anforderung, Freigabe und Verfügbarkeit von TCP-Streams und Engpässen von Streamsressourcen aus.

-z zero

setzt alle Zähler der Statistiken zurück.

TSAP/TCEP Ausgabeformat

Die TSAP- und die TCEP-Details werden im gleichen Format ausgegeben. Dieses Format besteht aus 13 Feldern, die in einer Doppelzeile angeordnet sind (ohne Option *-l*).

Die Felder in der ersten Zeile beziehen sich auf das obere TPI zwischen dem ISO-8072-Transport-Dienst-Benutzer und dem RFC1006-TSP.
Die zweite Zeile bezieht sich auf das untere TPI zwischen dem RFC1006-TSP als Transport-Dienst-Benutzer und TCP als Transport-Dienst-Erbringer.

Bezeichnung und Bedeutung der Felder in der ersten Zeile:

UPPER
: Upper stream ID, d.h. die Nummer zur Identifikation des Streams zwischen dem TPI-Adapter und dem RFC1006-TSP.

  Der Kontroll-Stream */dev/SMAWcmx/rfc1006lm* hat die Nummer 0. Er wird symbolisch mit CONTROL angezeigt.
  Der Administrations-Stream */dev/SMAWcmx/rfc1006adm* hat die Nummer 1. Er wird symbolisch mit ADMIN angezeigt.
  Alle anderen (Klons von */dev/SMAWcmx/rfc1006*) haben Nummern zwischen 2 und *r6-max-conns-and-saps*. *r6-max-conns-and-saps* ist in der Datei */usr/kernel/drv/SMAWr6.conf* festgelegt.

TP0-STATE
: TPI-Zustand bezogen auf den ISO-8072-Transport-Dienst-Benutzer.

CIND(MAX)
: Momentane und maximal zugelassene (in Klammern) Anzahl ausstehender Verbindungsanzeigen. Außer für Listener-Streams immer „-“.

TREF
: Transport-Referenz der CMX-Anwendung

PV
: RFC1006-Protokoll-Variante:
  ’3.0’ falls das CMX-Adress-Format LANINET ist.
  ’-’ falls das CMX-Adress-Format RFC1006 ist.

TPDUSZ
: Maximale TPDU-Größe für diese Verbindung.

OWN-TSEL
: Kodierungsschema
'A:' ASCII, 'E:' EBCDIC, 'T:' TRANSDATA, 'V:' void, 'X:' hexadezimal und Wert des T-Selektors der mit Adress-Format RFC1006 lokal angemeldeten Anwendung

PARTNER-TSEL
: Kodierungsschema
'A:' ASCII, 'E:' EBCDIC, 'T:' TRANSDATA, 'V:' void, 'X:' hexadezimal und Wert des T-Selektors der Partner-Anwendung

Bezeichnung und Bedeutung der Felder in der zweiten Zeile:

LOWER
: Lower stream ID, d.h. die Identifikationsnummer des Streams zwischen RFC1006-TSP und TCP

TCP-STATE
: TPI-Zustand bezogen auf TCP als Transport-Dienst-Erbringer

CIND(MAX)
: Momentane und maximal zugelassene (in Klammern) Anzahl ausstehender Verbindungsanzeigen. Außer für Listener-Streams immer „-".

OWN-TCP-ADDR
: TCP-Adresse in der Form *host.port* oder **.port*.
* steht hier für die IPv4-Adresse 0.0.0.0 bzw. die IPv6-Adresse ::, die die Gesamtheit der lokalen Netzzugangspunkte identifiziert.

PARTNER-TCP-ADDR
: TCP-Adresse in der Form *host.port*. *host* ist der Hostname ohne die Default-Domäne. Falls die Option *-n* angegeben ist, wird *host* ersetzt durch die IP-Adresse im kanonischen IPv4- oder IPv6-Darstellungsformat.

Besonderheiten beim Ausgabeformat für Listener-TSAPs

Mit T_PASSIVE und Adress-Format RFC1006 angemeldete CMX-Anwendungen teilen sich den TCP-Listener '*.iso-tsap' bzw. den TCP-Listener '*host*.iso-tsap', falls T_OPTA7 mit dem Hostnamen *host* spezifiziert worden ist. Entsprechendes gilt für CMX-Anwendungen, die zusätzlich mit Adress-Format LANINET und zwar für dieselbe Portnummer *port* angemeldet sind (d.h. der LOKALE NAME enthält sowohl `TSEL RFC1006` *T-Selektor* als auch `TSEL LANINET` *port*). In diesem Fall nutzen sie zusätzlich den TCP-Listener '*.*port*' bzw. '*host.port*' gemeinsam.

Alle gemeinsam genutzten TCP-Listener tauchen in der Ausgabe der Listener-TSAPs nur einmal auf:

- '*.iso-tsap' ist immer dem Kontroll-Stream (UPPER = CONTROL) zugeordnet.
- '*.*port*' ist ebenso wie '*host.port*' irgendeinem passenden Listener-TSAP zugeordnet.
- '*host*.iso-tsap' ist ebenfalls einem passenden Listener-TSAP zugeordnet, wird aber in einer zusätzlichen, eigenen Doppelzeile dargestellt, wobei dem Feld UPPER ein '*' (Stern) vorangestellt ist.

Listener-TSAPs, die sowohl für 'iso-tsap' (Portnummer 102) als auch für eine eigene Portnummer angemeldet sind (d.h. der LOKALE NAME enthält sowohl `TSEL RFC1006` ***T-Selektor*** als auch `TSEL LANINET` *Portnummer*), werden mit der eigenen Portnummer im Feld OWN-TCP-ADDR dargestellt. Dennoch gilt gleichzeitig die Anmeldung für 'iso-tsap'.

Im Gegensatz dazu sind reine LANINET-Listener (d.h. der LOKALE NAME enthält nur `TSEL LANINET` *Portnummer*) niemals für 'iso-tsap' angemeldet und sie teilen ihren TCP-Listener niemals mit anderen Listener-TSAPs. Solche reinen LANINET-Listener sind gekennzeichnet durch '`3.0`' im Feld PV.

Siehe auch

rfc1006tune

9.4 Betriebsparameter für RFC1006-TSP setzen (rfc1006tune)

Mit *rfc1006tune* setzen Sie Betriebsparameter für den RFC1006-TSP.

Vorsicht!

Das Tuning sollten nur erfahrene Systemverwalter mit ausgezeichneten Kenntnissen in UNIX-Systemen, Netzwerken und Protokollen durchführen.
Wenn Tuning-Maßnahmen falsch oder nur teilweise durchgeführt werden, kann sich das Systemverhalten möglicherweise verschlechtern oder das System inoperabel werden.

rfc1006tune [␣**-a**␣maxsaps] [␣**-c**␣maxconns] [␣**-e**␣secs] [␣**-h**␣high] [␣**-l**␣low] [␣**-p**␣port] [␣**-r**␣response] [␣**-t**␣tidusize] [␣**-u**␣unrelated]

rfc1006tune␣**-d**

Das Kommando *rfc1006tune* erlaubt das Tuning diverser Einstellungen des RFC1006-TSP.

Parameterwerte, die sich von den Default-Werten unterscheiden, werden automatisch in der Konfigurationsdatei */opt/SMAW/SMAWcmx/lib/rfc1006/rfc1006.conf* gesichert. Diese Konfigurationsdatei wird beim Start des RFC1006-TSP vom Dämon */opt/SMAW/SMAWcmx/etc/rfc1006d* gelesen.

Für das *rfc1006tune*-Kommando gibt es folgende zwei Modi:

- Modus eins setzt die angegebenen Parameter auf die spezifizierten Werte. Mittels leerer Zeichenfolge werden die angegebenen Parameter auf deren Default-Wert gesetzt. Nicht angegebene Parameter behalten ihre Werte.
- Modus zwei setzt alle Parameter auf die Default-Werte zurück.

Jedes erfolgreich abgesetzte *rfc1006tune*-Kommando gibt folgende Werte aus:

– derzeit verwendete Parameterwerte
– Parameterwerte, die nach dem nächsten Restart verwendet werden
– Default-Parameterwerte
– Wertebereiche

Verwenden Sie Kommandomodus eins ohne Angabe von Optionen, um sich über die Parameterwerte zu infomieren.

Optionen

-a maxsaps
: setzt die zulässige Anzahl gleichzeitiger Listener-TSAPs auf den Wert *maxsaps*.
Sie müssen den Parameter *maxsaps* als Dezimalzahl angeben oder als leere Zeichenfolge, wodurch der Default-Wert initiiert wird. Das zugehörige Schlüsselwort in der Konfigurationsdatei lautet *MAXSAPS*.

Beachten Sie, dass ein Listener-TSAP implizit vom Kontroll-Stream verwendet wird (siehe in der Ausgabe des Kommandos *rfc1006stat -a* die Doppelzeile mit UPPER = CONTROL).

1 <= maxsaps <= r6-max-conns-and-saps - maxconns
wobei *r6-max-conns-and-saps* in der Datei */usr/kernel/drv/SMAWr6.conf* festgelegt ist.

Der Default-Wert für *maxsaps* beträgt *(r6-max-conns-and-saps +1) / 2*.

-c maxconns
: setzt die maximal zulässige Anzahl gleichzeitig eröffneter Transportverbindungen auf den Wert *maxconns*.
Sie müssen den Parameter *maxconns* entweder als Dezimalzahl angeben oder als leere Zeichenfolge, wodurch der Default-Wert initiiert wird. Das zugehörige Schlüsselwort in der Konfigurationsdatei lautet *MAXCONNS*.

0 <= maxconns <= r6-max-conns-and-saps - maxsaps
wobei *r6-max-conns-and-saps* in der Datei */usr/kernel/drv/SMAWr6.conf* festgelegt ist.

Der Default-Wert für *maxconns* beträgt *r6-max-conns-and-saps / 2*.

-d default
: setzt alle Werte auf die Voreinstellung (Default-Werte) zurück.

-e secs
: setzt die Zeiteinheit für Timer (d.h. die Anzahl Sekunden zwischen zwei Alarm-Ticks) auf *secs*.

Bei jedem Alarm-Tick erniedrigt und prüft der RFC1006-TSP

- die Restlaufzeit der Timer für nicht-zugeordnete (unrelated) TCP-Verbindungen (Nicht-zugeordnete TCP-Verbindungen sind passiv aufgebaute TCP-Verbindungen, auf denen noch keine CR TPDU angekommen ist, siehe Option *-u*.)
- den Timer für Antwort auf abgehende Verbindungsanforderungen (Warten auf CC TPDU, siehe Option *-r*)

Geben Sie *secs* entweder als Dezimalzahl an oder als leere Zeichenfolge, wodurch der Default-Wert initiiert wird. Der neue Wert wird sofort wirksam. Das entsprechende Schlüsselwort in der Konfigurationsdatei lautet *PERIOD*.

6 <= secs <= 50000

Der Default-Wert für *secs* beträgt 30.

-h high

setzt das Maximum freier TCP Streams im Pool auf *high*.

Sie müssen *high* entweder als Dezimalzahl angeben oder als leere Zeichenfolge, wodurch der Default-Wert initiiert wird. Das entsprechende Schlüsselwort in der Konfigurationsdatei lautet *HIGHPOOL*.

low <= high <= r6-max-conns-and-saps /4
wobei *r6-max-conns-and-saps* in der Datei */usr/kernel/drv/SMAWr6.conf* festgelegt ist.

Der Default-Wert für *high* beträgt 64.

-l low

setzt das Minimum freier TCP Streams im Pool auf *low*.

Sie müssen *low* entweder als Dezimalzahl angeben oder als leere Zeichenfolge, wodurch der Default-Wert initiiert wird. Das entsprechende Schlüsselwort in der Konfigurationsdatei lautet *LOWPOOL*.

0 <= low <= min(high, r6-max-conns-and-saps / 4)
wobei *r6-max-conns-and-saps* in der Datei */usr/kernel/drv/SMAWr6.conf* festgelegt ist.

Der Default-Wert für *low* beträgt 32.

-p port

setzt die Portnummer für den TCP-Listener, der dem Kontroll-Stream zugeordnet ist, auf den Wert *port*.

Sie müssen *port* entweder als Dezimalzahl angeben oder als leere Zeichenfolge, wodurch der Default-Wert initiiert wird. Die neue Portnummer wird ab dem nächsten Neustart des RFC1006-TSP wirksam. Das entsprechende Schlüsselwort in der Konfigurationsdatei lautet *LISTEN-PORT*.

1 <= port <= 32767

Der Default-Wert für *port* beträgt 102.

-r response

setzt die maximale Wartezeit für Antwort auf abgehende Verbindungsanforderungen (CR TPDU) auf den Wert *response*. *response* ist die Anzahl Alarm-Ticks, siehe Option *-e*. Wenn innerhalb dieses Zeitraums keine Bestätigung (CC TPDU) oder Ablehnung (DR TPDU) vom Partnersystem eintrifft, baut der RFC1006-TSP die Verbindung ab. Die lokale CMX-Anwendung erhält eine Verbindungsabbauanzeige mit Abbaugrund T_RLNORESP.

Geben Sie *response* entweder als Dezimalzahl an oder als leere Zeichenfolge, wodurch der Default-Wert initiiert wird. Der neue Wert wird nur für solche Verbindungsanforderungen wirksam, die nach dem Änderungszeitpunkt erfolgen. Das entsprechende Schlüsselwort in der Konfigurationsdatei lautet *RESPONSEPRD*.

2 <= response <= 50000

Der Default-Wert für *response* beträgt 6.

-t tidusize

setzt die maximale TIDU-Größe auf den Wert *tidusize*.

Sie müssen *tidusize* entweder als Dezimalzahl angeben oder als leere Zeichenfolge, wodurch der Default-Wert initiiert wird. Die neue *tidusize* wird ab dem nächsten Neustart des RFC1006-TSP wirksam. CMX-Applikationen erhalten diesen Wert beim Aufruf der Funktion *t_info()*. Die maximale TIDU-Größe beeinflusst die Aushandlung der TPDU-Größe des RFC1006-Protokoll. Das entsprechende Schlüsselwort in der Konfigurationsdatei lautet *TIDUSIZE*.

4089 <= port <= 65273

Der Default-Wert für *tidusize* beträgt 4089.

-u unrelated

setzt die maximale Lebenszeit von nicht-zugeordneten (unrelated) TCP-Verbindungen auf den Wert *unrelated*. *unrelated* ist die Anzahl Alarm-Ticks, siehe Option *-e*. Wenn innerhalb dieses Zeitraums auf einer vom Partnersystem initiierten TCP-Verbindung keine RFC1006-Verbindungsanforderung (CR TPDU) eintrifft, baut der RFC1006-TSP die TCP-Verbindung ab. Geben Sie *unrelated* entweder als Dezimalzahl an oder als leere Zeichenfolge, wodurch der Default-Wert initiiert wird. Der neue Wert wird nur für solche TCP-Verbindungen wirksam, die nach dem Änderungszeitpunkt passiv aufgebaut werden.

Das Schlüsselwort in der Konfigurationsdatei lautet *UNRELTCPPRD*.

2 <= unrelated <= 50000

Der Default-Wert für *unrelated* beträgt 2.

Dateien

optionale Konfigurationsdatei
/opt/SMAW/SMAWcmx/lib/rfc1006/rfc1006.conf

Vorlage für die Konfigurationsdatei
/opt/SMAW/SMAWcmx/lib/rf1006/rfc1006.template

Siehe auch

rfc1006stat

10 Administration und Wartung

Dieses Kapitel enthält die Beschreibung der Wartungs- und Administrationsfunktionen, wie sie über die Kommandoschnittstelle aktiviert werden können. Sie finden die CMX-Kommandos in alphabetischer Reihenfolge.

Man Pages

Mit dem Produkt CMX werden *Man Pages* ausgeliefert. Sie umfassen sowohl die in diesem Kapitel beschriebenen als auch weitere Kommandos, die in CMX V6.0 nur von Experten benötigt werden. Alle Man Pages liegen in Englisch vor.

Die Man Page zum gewünschten Kommando erhalten Sie mit dem Aufruf

`man` *kommando*

Die folgende Liste bietet eine nach Komponenten geordnete Übersicht der verfügbaren Kommandos.

10.1 Übersicht der Kommandos

Mit * gekennzeichnete Kommandos sind nur in den Man Pages beschrieben.

CMX-spezifische Kommandos

cmxconf
: Systemlokale Konfigurationsinfomation zu einem GLOBALEN NAMEN einer CMX-Anwendung ausgeben (sofern möglich für alle Ebenen des Transportsystems). Näheres siehe Abschnitt „Konfiguration einer CMX-Anwendung prüfen (cmxconf)“ auf Seite 231.

cmxdec
: CMX-Meldungen decodieren, siehe Abschnitt „CMX-Meldungen decodieren (cmxdec)“ auf Seite 233.

cmxdiag
: Diagnoseunterlagen bereitstellen, siehe Abschnitt „Diagnoseinformationen sammeln und bereitstellen (cmxdiag)“ auf Seite 237.

cmxinfo
: Informationen zur CMX-Konfiguration, siehe Abschnitt „Informationen zur CMX-Konfiguration (cmxinfo)“ auf Seite 239.

cmxm
: CMX-Monitor steuern, siehe Abschnitt „CMX-Monitor (cmxm)“ auf Seite 266.

cmxmd
: CMX-Monitordämon steuern, siehe Abschnitt „CMX-Monitordämon (cmxmd)“ auf Seite 279.

cmxprod
: Installierte Kommunikationsprodukte abfragen, siehe Abschnitt „Installierte Kommunikationsprodukte abfragen (cmxprod)“ auf Seite 281.

cmxstat
: Aktuell belegte TSP-spezifische Ressourcen ausgeben (Anwendungen, Verbindungen, Subnetzanschlüsse und interne Statistik), siehe Abschnitt „TSP-spezifische Statusinformation (cmxstat)“ auf Seite 283.

cmxtrc
: Traces in einem Transportsystem ein- und ausschalten, d.h. in den Protokollschichten 1 bis 4. Durch Angabe eines GLOBALEN NAMENS können die Traces auf die bestimmte CMX-Anwendungen eingeschränkt werden. Näheres siehe Abschnitt „Traces für Transportsystem (cmxtrc)“ auf Seite 287.

cmxtune
: Grenzwerte für den CMX-Automaten ändern, siehe Abschnitt „Grenzwerte für den CMX-Automaten ändern (cmxtune)“ auf Seite 291.

StartStop
: Starten und Stoppen von CMX und CCPs, siehe Abschnitt „Starten und Stoppen von CMX und TSPs (StartStop)“ auf Seite 303.

TSP-spezifische Kommandos

rfc1006
: TSP RFC1006 starten und stoppen, siehe Abschnitt „Starten und Stoppen von CMX und TSPs (StartStop)“ auf Seite 303.

rfc1006stat
: Status und Statistiken von RFC1006-TSP abfragen, siehe Abschnitt „Status/Statistiken des RFC1006-TSP abfragen (rfc1006stat)“ auf Seite 216.

rfc1006tune
: Betriebsparameter für RFC1006-TSP setzen, siehe Abschnitt „Betriebsparameter für RFC1006-TSP setzen (rfc1006tune)“ auf Seite 221.

ntp
: TSP NTP starten und stoppen, siehe Abschnitt „Starten und Stoppen von CMX und TSPs (StartStop)“ auf Seite 303.

tp02
: TSP TP0/2 starten und stoppen, siehe Abschnitt „Starten und Stoppen von CMX und TSPs (StartStop)“ auf Seite 303.

nea
: TSP NEA starten und stoppen, siehe Abschnitt „Starten und Stoppen von CMX und TSPs (StartStop)“ auf Seite 303.

Die zwei letztgenannten Kommandos sind nur verwendbar, wenn auf Ihrem System das entsprechende Produkt installiert ist.

TNS-spezifische Kommandos

tnsxchk
: Zustand eines TS-Directories prüfen, siehe Abschnitt „TS-Directory prüfen (tnsxchk)“ auf Seite 306.

tnsxcom
: TS-Directory-Einträge hinzufügen, ändern und löschen, siehe Abschnitt „TS-Directory erstellen, aktualisieren, lesen (tnsxcom)“ auf Seite 308.

tnsxd*
: TNS-Dämon starten.

tnsxdel
: Einträge aus einem TS-Directory löschen, siehe Abschnitt „TNS-Einträge löschen (tnsxdel)“ auf Seite 312.

tnsxinfo
: Informationen über ein TS-Directory anzeigen, siehe Abschnitt „Informationen zum TS-Directory anzeigen (tnsxinfo)“ auf Seite 315.

tnsxlock
: Zugriff auf TNS-Dämonen sperren und freigeben, siehe Abschnitt „Sperren der Zugriffe zum TNS-Dämon (tnsxlock)“ auf Seite 322.

tnsxprop
: Einträge eines TS-Directories anzeigen, siehe Abschnitt „Eigenschaften von TS-Anwendungen ausgeben (tnsxprop)“ auf Seite 323.

Trace-Kommandos

cmxl

CMX-Bibliotheks-Trace steuern und aufbereiten, siehe Abschnitt „CMX-Bibliotheks-Trace steuern und aufbereiten (cmxl)“ auf Seite 255.

comtr

Traces für CMX-abhängige Treiber überwachen, siehe Abschnitt „Traces für CMX-Treiber (comtr)“ auf Seite 292.

neal

NEA-Bibliotheks-Trace steuern und aufbereiten, siehe Abschnitt „NEABX-Bibliotheks-Trace steuern und aufbereiten (neal)“ auf Seite 299.

tnsxt

TNS-Trace-Information sicherstellen und aufbereiten, siehe Abschnitt „Sicherstellen und Aufbereiten der Trace-Information (tnsxt)“ auf Seite 326.

Die einzelnen Kommandos werden nachfolgend in alphabetischer Reihenfolge beschrieben.

10.2 Konfiguration einer CMX-Anwendung prüfen (cmxconf)

Das Kommando *cmxconf* überprüft zu einem GLOBALEN NAMEN die systemlokale Konfiguration auf Konsistenz und auf die Betriebsbereitschaft sämtlicher involvierter Komponenten. Die Konfiguration wird dabei soweit wie möglich ganzheitlich im Transportsystem betrachtet, d.h. von der Transportschicht bis zum Leitungsanschluss.

cmxconf kann in zwei Formaten angegeben werden: für eine lokale Anwendung (Option *-o)* oder für eine Partneranwendung (Option *-p*).

Syntax

cmxconf ␣**-o**␣own-appl [␣**-t**␣ts-prov]

cmxconf ␣**-p**␣part-appl

-o␣own-appl

(own) Es werden die lokalen Namensteile der Anwendung *own-appl* überprüft, d.h. es wird ausgegeben, ob die lokalen TSAPs bei den betreffenden Transport-Providern angemeldet sind. *own-appl* ist der GLOBALE NAME der lokalen Anwendung. Die Überprüfung kann mit *-t ts-prov* auf einen bestimmten Transport-Provider eingeschränkt werden.

-t␣ts-prov

Wenn eine Anwendung mehrere LOKALE NAMEN besitzt, wird die Überprüfung auf einen Transportsytem-Provider (*ts-prov*) eingeschränkt.

Mögliche Angaben für den Transportprovider:

```
nea
ntp
tp02
rfc1006
```

-p␣part-appl

(peer) Es wird zur vorgegebenen Partneradresse mit dem GLOBALEN NAMEN *part-appl* die gesamte lokale Konfiguration ausgegeben (TNS und FSS). Bei WAN-Transport-Providern wird zusätzlich überprüft, ob die Komponenten des Transportsystems aktiv sind, über die eine Verbindung aufgebaut werden soll. Diese Überprüfung umfasst sowohl die Transportadresse als auch die Routeninformation. Bei Partneradressen vom Typ RFC1006 wird nur überprüft, ob das Transportsystem aktiv ist.

Beispiel

Das folgende Kommando listet die Konfigurationsinformation zu der Partneradresse *nearem* auf:

```
cmxconf -p nearem

Configuration information for the remote partner: nearem
================================================================================
NEA-NSAP:       71/255  Remote T-Sel:   A'nearemot'

Configured routes for this remote NSAP:
SNPAROUTES  name=nearoute short-id=1 subnet=X25-1 type=X25 dte-addr=123 SNPAROUTES
name=nearoute1 short-id=2 subnet=X25-2 type=X25 dte-addr=123

Active local SNPA's for this partner:
CC  IF#  STATE TYPE  Bit/s LINK    LINKS NETW. SUBNET SUBNET-ID SUBNET-ADDR
W3    1  NETC  X.21    64k -         0/1 -     LEASED X25_1     1
W3    2  NETC  X.21    64k -         0/1 -     LEASED X25_2     2
```

Das Kommando *cmxconf* liefert somit eine Aussage über die prinzipielle Erreichbarkeit einer lokalen oder fernen Anwendung aus lokaler Sicht. Es kann eingesetzt werden, um die lokale Konfiguration aller möglichen Kommunikationspartner eines TNS-Directories auf Konsistenz zu überprüfen.

– Dieses Kommando ist TNS-abhängig und funktioniert nicht für andere Directories. Wenn der Kunde kein Directory nutzt, kann man sich bei Kommunikationsstörungen durch die Nachbildung des Eintrags im TNS behelfen.

– Zu einer Partneradresse muss eine Route nicht unbedingt definiert sein. Wenn keine Route für eine Partneradresse im SNPA-Format definiert ist, wählen die Transport-Provider (NTP und TP02) eigenständig einen Subnetzanschluss aus, der das Adressformat unterstützt.

10.3 CMX-Meldungen decodieren (cmxdec)

Mit dem Kommando *cmxdec* können Sie ICMX- und XTI-Meldungen decodieren. Dabei handelt es sich um folgende Meldungsarten:

– Fehlermeldungen, die in den Include-Dateien <*cmx.h*>, <*neabx.h*> und <*tnsx.h*> der ICMX bzw. in der Include-Datei <*xti.h*> von XTI definiert sind. Diese Meldungen werden an den Programmschnittstellen ICMX(L), ICMX(NEA) und XTI erzeugt.

– Meldungen, die aus fehlerhaft abgelaufenen Systemaufrufen von ICMX oder XTI resultieren, d. h. Fehlermeldungen, die in der Datei <*errno.h*> definiert sind.

– vom CMX-Automaten bzw. CCP übergebene Gründe für einen Verbindungsabbau durch den CMX-Automaten bzw. das CCP.

ICMX- und XTI-Fehlermeldungen, Systemmeldungen und Verbindungsabbaugründe werden i.a. in Form eines numerischen Codes, Dezimalzahl oder Hexadezimalzahl mit führendem „0x“, ausgegeben. Eine Fehlermeldung, die an der Programmschnittstelle zum TNS erzeugt wurde, ist nur eindeutig interpretierbar, wenn je ein Wert für den Fehlertyp, die Fehlerklasse und den Fehlerwert angegeben wird. Aus diesem Grund wird ein entsprechender Fehlercode in Form von drei Dezimalzahlen ausgegeben. Die Werte dieser Codes können negativ sein.

Ein Fehlercode bzw. der Code für einen Verbindungsabbau (reason) wird von *cmxdec* entschlüsselt, wenn Sie den Typ der Meldung (XTI-Fehlermeldung, ICMX(L)-Fehlermeldung usw.) und den angegebenen Code des Fehlers an *cmxdec* übergeben. *cmxdec* gibt dann den symbolischen Wert, der in der entsprechenden Include-Datei definiert ist, auf der Standardfehlerausgabe aus.

Bei entsprechender Umgebung des Aufrufers (Variable LANG) liefert *cmxdec* erläuternde Texte zu den Meldungen. Die Texte sind sprachabhängig und optional. Die Messagekataloge in deutscher und englischer Sprache werden immer zusammen mit CMX ausgeliefert.

Syntax

cmxdec [␣**-c**] [␣**-d**] [␣**-n**] [␣**-s**] [␣**-t**] [␣**-x**]␣code ...

Die Optionen geben den Typ der in *code* angegebenen Meldung an. Voreinstellung ist *-c*. *cmxdec* versteht die im Folgenden beschriebenen Optionen. Sie schließen sich gegenseitig aus.

-c

Der in *code* angegebene Wert ist eine ICMX-Fehlermeldung, wie sie von *t_error()* an der Schnittstelle ICMX(L) geliefert wird.

Ausgabeformat:

```
ICMX(L) Fehlerdecodierung (6.0)
CODE 0x%x = %d (TYP %d KLASSE %d WERT %d)
      symbolischer Wert und Erklärung für TYP
      symbolischer Wert und Erklärung für KLASSE
      symbolischer Wert und Erklärung für WERT
```

Bezeichnet WERT dabei eine Systemfehlermeldung, so wird in der letzten Zeile in der Regel statt eines symbolischen Wertes der Zahlenwert eingetragen. Die Erklärung erfolgt dann in Englisch.

-d

Der in *code* angegebene Wert ist ein Verbindungsabbaugrund, wie er von *t_disin()* (ICMX(L)), *x_disin()* (ICMX(NEA)) oder *t_rcvdis()* (XTI) geliefert wird. Beachten Sie, dass für XTI/Internet (XTI über TCP/IP) der Abbaugrund als Systemfehler, definiert in <*errno.h*>, decodiert wird.

Ausgabeformat:

```
CMX Reasondecodierung (6.0)
REASON 0x%x = %d:
ICMX(L):
      symbolischer Wert und Erklärung
XTI/Internet:
      symbolischer Wert und Erklärung
XTI/ISO:
      symbolischer Wert und Erklärung
```

-n

Der in *code* angegebene Wert ist eine NEABX-Fehlermeldung, wie sie von *x_error()* an der Schnittstelle ICMX(NEA) geliefert wird.

Ausgabeformat:

```
ICMX(NEA) Fehlerdecodierung (6.0)
CODE 0x%x = %d (TYP %d KLASSE %d WERT %d)
      symbolischer Wert und Erklärung für TYP
      symbolischer Wert und Erklärung für KLASSE
      symbolischer Wert und Erklärung für WERT
```

-s

Der in *code* angegebene Wert ist eine Systemmeldung, wie sie von Systemaufrufen geliefert wird.

Ausgabeformat:

```
CMX Systemfehlerdecodierung (6.0)
CODE 0x%x = %d
      Erklärung in Englisch
```

-t

Die drei in *code* angegebenen numerischen Werte sind eine TNS-Fehlermeldung, wie sie von den TNS-Aufrufen im Standardkopf geliefert wird.

Ausgabeformat:

```
ICMX(TNS) Fehlerdecodierung
TYP %x=%d KLASSE =x%x=%d WERT 0x%x=%d)
      symbolischer Wert und Erklärung für TYP
      symbolischer Wert und Erklärung für KLASSE
      symbolischer Wert und Erklärung für WERT
```

-x

Der in *code* angegebene Wert ist eine XTI-Fehlermeldung, wie sie von *t_error()* geliefert wird.

Ausgabeformat:

```
XTI Fehlerdecodierung
CODE 0x%x = %d
      symbolischer Wert und Erklärung
```

code

Für *code* ist der von ICMX, NEABX, TNS bzw. XTI gelieferte numerische Fehlercode, der Code einer Systemmeldung oder der Code eines Verbindungsabbaugrunds anzugeben, den *cmxdec* decodieren soll. Bei den Optionen *c, d, n, s, x* müssen Sie für *code* eine Dezimalzahl oder eine Hexadezimalzahl mit führendem „0x“ oder „0X“ angeben. Bei Option *-t* müssen Sie für *code* drei vorzeichenbehaftete Dezimalzahlen oder Hexadezimalzahlen mit führendem „0x“ oder „0X“ angeben.

Ein vorgegebener numerischer Wert hat für ICMX(L) und XTI unterschiedliche Bedeutung.

TYP, KLASSE und WERT entsprechen den bei ICMX, NEABX und TNS verwendeten Klassifizierungen der Fehlercodes, REASON ist der Grund für einen Verbindungsabbau. Der numerischen Angabe folgt die symbolische Definition gemäß den Include-Dateien <*cmx.h*>, <*neabx.h*>, <*xti.h*> und <*tnsx.h*>. Daran schließt sich der erklärende Text zu dieser Meldung an.

Wird für *code* ein ungültiger oder nicht definierter Wert angegeben, so entschlüsselt *cmxdec* diesen Wert möglichst weitgehend in Bezug auf die symbolische Darstellung des Verbindungsabbaugrundes oder auf Typ, Klasse und Wert der Fehlermeldung. Als erklärenden Text zum Wert von *code* wird von *cmxdec* die Meldung *Nicht decodierbar* ausgegeben.

10.4 Diagnoseinformationen sammeln und bereitstellen (cmxdiag)

Das Kommando *cmxdiag* sammelt Diagnoseinformationen zur späteren Fehleranalyse und stellt diese in der Datei */opt/SMAW/SMAWcmx/lib/ccp/diagfiles/collect/diag.tar* bereit. Die Diagnoseunterlagen umfassen Konfigurationsdaten, Statusinformationen, Logging-Dateien (System, CMX) und Traces. Der Umfang der erzeugten Diagnoseunterlagen wird über Aufrufargumente gesteuert.

Syntax

cmxdiag ␣ [**all** | **cctraces** | **konfig** | **ktraces** | **log** | **status** | **traces** | **cmxsnap**]

Bei Aufruf ohne Argument wird die Kommandosyntax angezeigt.

all
: Auslesen aller für eine Fehlerdiagnose relevanten Informationen und Kopieren dieser Daten in das Diagnosepaket */opt/SMAW/SMAWcmx/lib/ccp/diagfiles/collect/diag.tar*.

cctraces
: Kopieren aller vorhanden Tracelisten der Communication Controller (siehe Handbücher „CMX/CCP, ISDN-Kommunikation“ [3] und „CMX/CCP, WAN-Kommunikation“ [4]) in das Diagnosepaket */opt/SMAW/SMAWcmx/lib/ccp/diagfiles/collect/diag.tar*.

cmxsnap
: Auslesen der Prozesstabelle und Kopieren dieser Daten in das Diagnosepaket */opt/SMAW/SMAWcmx/lib/ccp/diagfiles/collect/diag.tar*.

konfig
: Auslesen der für eine Fehlerdiagnose relevanten Konfigurationsdaten und Kopieren dieser Daten in das Diagnosepaket */opt/SMAW/SMAWcmx/lib/ccp/diagfiles/collect/diag.tar*.

ktraces
: Aufbereiten der Traces für CMX-Treiber und Kopieren in das Diagnosepaket */opt/SMAW/SMAWcmx/lib/ccp/diagfiles/collect/diag.tar*.

log
: Funktionalität wie *konfig*. Zusätzlich werden das Loggingfile des cronjobs und die Loggingfiles der CMX-Komponenten in das Diagnosepaket */opt/SMAW/SMAWcmx/lib/ccp/diagfiles/collect/diag.tar* kopiert.

status

Funktionalität wie *log*. Zusätzlich werden der aktuelle Status der Communcation Controller, der Status der konfigurierten Schnittstellen sowie Informationen zu den TSPs und zur CMX-Konfiguration ermittelt und in das Diagnosepaket */opt/SMAW/SMAWcmx/lib/ccp/diagfiles/collect/diag.tar* kopiert.

traces

Funktionalität wie *ktraces* und *cctraces*. Zusätzlich werden das Loggingfile des cronjobs und die Loggingfiles der CMX-Komponenten in das Diagnosepaket */opt/SMAW/SMAWcmx/lib/ccp/diagfiles/collect/diag.tar* kopiert.

10.5 Informationen zur CMX-Konfiguration (cmxinfo)

Mit dem Kommando *cmxinfo* können Sie Informationen über die CMX-Konfiguration und über Art und Anzahl der bedienten CCs und der TSP Access Points abfragen. Das Kommando informiert über die mögliche und die aktuelle Auslastung von CMX und CCs/TSP Access Points. *cmxinfo* gibt die Informationen auf *stdout* aus.

In der folgenden Abbildung erhalten Sie ein Beispiel zur Implementierung von TSPs und Subnetz-Profilen. Beachten Sie auch die Informationen zur Architektur von CMX in Kapitel „Architektur der Solaris-Kommunikation“ auf Seite 11.

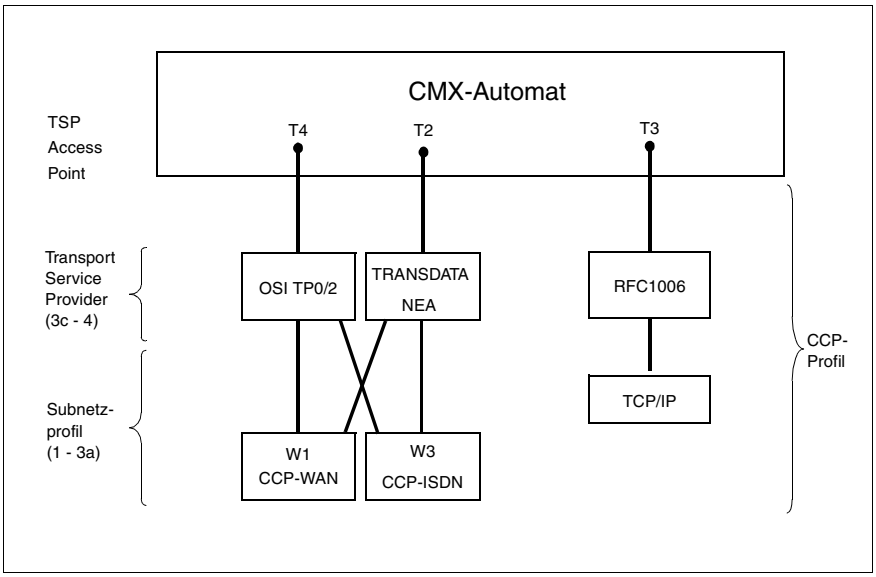


Bild 35: TSPs, Subnetz-Profile und CCP-Profile

Folgende Informationen werden von *cmxinfo* ausgegeben:

- die Grenzwerte des CMX-Automaten, die angeben, wieviele TS-Anwendungen (ICMX- und XTI-Anwendungen), Transportverbindungen und CCs gleichzeitig bedient werden können
- die Anzahl der derzeit bedienten CCs/TSP Access Points
- die Art der bedienten CCs/TSP Access Points
- Informationen zu den bedienten CCs/TSP Access Points, z. B. Speichergröße des CC, aktueller Zustand (betriebsbereit oder nicht)
- Eigenschaften der betriebsbereiten TSPs wie z. B. die Länge der Dateneinheit (TIDU), wieviele Transportverbindungen gleichzeitig über den TSP Access Point gehalten werden können
- Informationen zu allen aktuell bestehenden TSAPs oder weitere Informationen zu einem bestimmten TSAP

Mit Hilfe der Optionen kann die Ausgabe auf bestimmte Werte beschränkt werden.

Syntax

cmxinfo [␣**-t**]
[␣**-a**]
[␣**-b**␣{ id | **all** } [␣**-l**]]
[␣**-c**␣id]
[␣**-s**]
[␣-**C**␣{id | **all**}]
[␣**-S**␣{id | **all**}]
[␣**-i**] [␣**-M**] [␣**-v**]

Wenn Sie *cmxinfo* ohne Argumente aufrufen, werden die Grenzwerte des CMX-Automaten und die Art und Anzahl der bedienten CCs und TSP Access Points ausgegeben.

-t

Ausgabe von ablaufbegleitender Trace-Information. Es ist nur in Fehlerfällen sinnvoll, *-t* anzugeben.

-a

Es werden nur die Grenzwerte von CMX ausgegeben.

-b␣id | **all**

Es werden nur Hardware-Informationen zu dem in *id* angegebenen CC/TSP Access Point ausgegeben, sofern an Ihrem Rechner vorhanden. Die Angabe der einzelnen CCs/TSP Access Points erfolgt wie bei Option *-c* beschrieben. Wenn Sie *all* angeben, dann werden Hardware-Informationen zu allen verfügbaren CCs/TSP Access Points ausgegeben.

-l

Ausgabe von Information zu WAN-Interfaces des Controllers (nur zusammen mit Option *-b id*).

-c␣id

Es werden nur Informationen zu dem in *id* angegebenen CC/TSP Access Point ausgegeben, sofern an Ihrem Rechner vorhanden. Die einzelnen CCs/TSP Access Points werden in *id* wie folgt angegeben:

W[1-32]

für CC-WAN (X.21, V.24, V.35, ISDN), siehe Abschnitt „Architektur der CCP-Profile“ auf Seite 23.

T[1-6]

für TSP Access Point (siehe Abschnitt „Architektur der CCP-Profile“ auf Seite 23).

-s

Ausgabe einer Liste aller momentan verfügbaren TSAPs zusammen mit den jeweiligen PIDs, der Anzahl TCEPs (und dem TEP-Zustand, wenn die Anwendung die Schnittstelle XTI verwendet).

-S␣id | **all**

Ausgabe von Information für einen momentan verfügbaren TSAP mit der Identifikation *id*. Verfügbare TSAPs können Sie sich mit dem Kommando *cmxinfo -s* ausgeben lassen. Das Eingabeformat für *id* ist x.y (z. B. 12.0) oder hexadezimal mit „0x“ oder „0X“ zu Beginn. Wenn Sie *all* angeben, wird die Information für alle momentan verfügbaren TSAPs ausgegeben.

-**C**␣id | **all**

Ausgabe von Information für einen momentan verfügbaren TCEP mit der Identifikation *id*. Verfügbare TCEPs können Sie sich mit dem Kommando *cmxinfo -S* ausgeben lassen. Das Eingabeformat für *id* ist x.y (z. B. 12.0) oder hexadezimal mit „0x“ oder „0X“ zu Beginn. Wenn Sie *all* angeben, wird die Information für alle momentan verfügbaren TCEPs ausgegeben.

-i

Ausgabe von Information zu LAN-Interfaces, über die CMX-Anwendungen kommunizieren können.

-M

Zusammen mit *-b* oder *-C* erfolgt die Ausgabe in einem kürzeren Format: Kopf- und Fußzeilen werden nicht angezeigt.

-v

Ausgabe der Thread-Id (nur in Verbindung mit den Optionen *-s* oder *-S*).

Ausgabeformat von cmxinfo

Die Ausgabe von *cmxinfo* beginnt stets mit der Kopfzeile. Sie enthält den Eintrag:

```
CMX INFORMATION (6.0)
```

Der Umfang der folgenden Informationen ist abhängig von den Komponenten, zu denen Sie Informationen angefordert haben, d. h. von den Optionen, die Sie bei der Eingabe des Kommandos *cmxinfo* angegeben haben.

Im Folgenden wird beschrieben, wie die Ausgabe von *cmxinfo* zu den einzelnen Komponenten (CMX-Automat, CCs/TSP Access Points) aussieht.

Informationen zum CMX-Automaten (Option -a)

Sie werden über die Grenzwerte des CMX-Automaten und über die aktuelle Auslastung des CMX-Automaten informiert. Die Ausgabe erfolgt im folgenden Format:

```
CMX AUTOMAT VERSION G-6.0 x CCs/TSP Access Points
TEP a (g) ATT a (g) TSAP a (g) TCEP a (g) TSP a (g)
```

Im Folgenden werden die angegebenen Werte erläutert. In Klammern stehen jeweils die Grenzwerte (g), außerhalb der Klammern die aktuellen Werte (a).

CMX AUTOMAT VERSION
: Hier wird die Version des CMX-Automaten eingetragen. Der CMX-Automat ist die zentrale Komponente von CMX im Betriebssystemkern. Er ist das Bindeglied zwischen der Benutzerebene (TS-Anwendungen, ICMX- und XTI-Bibliothek) und den TSPs (siehe hierzu Abschnitt „Leistungsumfang von CMX und CCPs“ auf Seite 11).

x CCs/TSP Access Points
: x ist die Anzahl der vom CMX-Automaten bedienten CCs/TSP Access Points.

TEP

Der Wert a gibt die Anzahl der derzeit unterstützten Transportendpunkte (TEPs, siehe User Guide „X/Open Transport Interface" [2]) an. Die Zahl der TEPs ist die Summe aller XTI-Transportendpunkte plus der Anzahl aller über ICMX angemeldeten Prozesse bzw. Threads. Es können sich mehrere Prozesse bzw. Threads für eine TS-Anwendung bei CMX anmelden.

Der Grenzwert g gibt die maximal mögliche Zahl von TEPs an.

ATT

Der Wert a gibt an, wieviele Anmeldungen (Attachments) derzeit beim CMX-Automaten vorliegen.

a ist die Summe der Anmeldungen von Prozessen über alle aktiven TS-Anwendungen. Der Wert kann sich von TEP unterscheiden, da sich ein Prozess für mehrere TS-Anwendungen anmelden kann. Eine Anmeldung kann über die Schnittstelle ICMX oder XTI erfolgen.

Der Wert g gibt an, wieviele Anmeldungen gleichzeitig bestehen können.

TSAP

Der Wert a gibt an, wieviele TS-Anwendungen derzeit beim CMX-Automaten angemeldet sind. Über den LOKALEN NAMEN wird der TS-Anwendung ein Dienstzugriffspunkt (Transport Service Access Point = TSAP) zugeordnet. An diesen TSAP binden sich alle Prozesse, die sich für diese TS-Anwendung beim CMX-Automaten anmelden. a ist also die Anzahl der derzeit existierenden TSAPs.

Der Wert g gibt an, wieviele TS-Anwendungen (ICMX- und XTI-Anwendungen) maximal gleichzeitig bei CMX angemeldet sein können.

TCEP

Der Wert a gibt an, wieviele Transportverbindungen existieren (TCEP = Transport Connection End Point).

Der Wert g gibt an, wieviele Transportverbindungen maximal gleichzeitig über den CMX-Automaten gehalten werden können.

TSP

Der Wert a gibt die Anzahl der vom CMX-Automaten bedienten Transport Service Provider (TSPs) an. Der Wert g gibt an, wieviel TSPs CMX maximal bedienen kann.

Informationen zu den CCs/TSP Access Points (Option -b all)

Die Informationen zu den CCs/TSP Access Points sind in Form einer Tabelle aufbereitet. Die Bedeutung der ausgegebenen Werte wird anhand des folgenden Beispiels erläutert.

```
ID   TYPE         VERS  STATE    I1 I2  IO_ADDR    MEM  HW/FW_VERS CAB/BUS/SLOT
=============================================================================
T2 TPI-NEA       G-6.0 READY    - -      -         -      -        -/-/-
T3 TPI-RFC1006   G-6.0 READY    - -      -         -      -        -/-/-
T4 TPI-TPO/2     G-6.0 READY    - -      -         -       -       -/-/-
T5 TPI-NULL-TP   G-6.0 READY    - -      -          -      -       -/-/-
T6 TPI-LOOP      G-6.0 READY    - -      -          -      -       -/-/-
W1 PWS2          G-6.0 READY    1 -  0x0164a000 8192K  02/01.14   -/PCI/3
W2PWS0           G-6.0 READY    1 -  0x01694000 4096K  03/01.17   -/PCI/4
-----------------------------------------------------------------------------
```

In der Darstellung wurden stellenweise Zwischenräume und führende Nullen weggelassen. ’-’ bedeutet, dass kein passender Wert vorliegt.

Die Spalten der Tabelle haben folgende Bedeutung:

ID

Symbolische Bezeichnung und Typ der bedienten CCs/TSP Access Points. Folgende Werte sind möglich:

W[1-32]

für CC-WAN (X.21, V.24, V.35, ISDN), siehe Abschnitt „Architektur der CCP-Profile“ auf Seite 23.

T[1-6]

für TSP Access Point, siehe Abschnitt „Architektur der CCP-Profile“ auf Seite 23.

TYPE

Typ des TSP Access Points (z. B. TPI-NEA, siehe Beispiel oben) oder des CCs. In der folgenden Tabelle sind die bekannten CC-Typen mit angegeben. Falls der CC-Typ unbekannt ist, wird nur der hexadezimale Wert ausgegeben.

Folgende Werte für CC sind möglich:

PCI-Bus-CCs
PWS0
PWS2
PWXV-2
PWXV-4

Tabelle 27: Mögliche Werte für CC-Typ

VERS
: Version des CC-Adapters/TSPs. Der CC-Adapter ist der Gerätetreiber für den CC im Betriebssystemkern.

STATE
: Zustand des CCs/TSP Access Points. Folgende Werte sind möglich:

  NONEX
  : Der CC existiert, kann aber über CMX nicht administriert werden.

  EXIST
  : Der CC existiert, ist aber nicht bei CMX angemeldet bzw. der TSP ist installiert, aber nicht aktiv.

  ATTACH
  : Der CC ist bei CMX angemeldet, aber nicht betriebsbereit.

  READY
  : Der CC ist geladen und aus der Sicht des CC-Adapters betriebsbereit / der TSP Access Point ist aktiv.

  0xabcd
  : Hexadezimaldarstellung des Zustands.

Die folgenden Werte sind nur für CCs, nicht für TSPs relevant:

I1
: Interrupt des CC. Dargestellt durch:

  dd
  : Dezimalzahl (dd) aus dem Bereich 0-99.

  xxxx
  : 4-ziffrige Hexadezimaldarstellung.

I2
: Zweiter Interrupt des CC. Darstellung siehe I1.

MEM
: Speicherausbau des CC in Kbyte.

HW/FW_VERS
: Hardware- und Firmware-Version des CC.

CAB/BUS/SLOT
: Wenn möglich, wird die Position des CC mit der Nummer von Cabinet, Bus und Slot angezeigt.

Informationen zu den TSP Access Points

Die Informationen zu den Transportdiensten, die von den TSP Access Points angeboten werden, sind in zwei Tabellen aufgeteilt. Die Bedeutung der ausgegebenen Werte wird anhand des folgenden Beispiels für die Ausgabe erläutert.

```
ID    CCP      VERS  TSAP  TCEP  TIDU  TSP  SUBRV
================================================
T2 0x00a0    G-6.00  2000  2000  2183    1      0
T3 0x00a0    G-6.00  2048  2048  4089    1      0
T4 0x00a0    G-6.00  2048  2048  2043    1      0
T5 0x00a0    G-6.00  2048  2048  1409    1      0
T6 0x00a0    G-6.00  4096  4096 65321    1      0
W5 no info
------------------------------------------------
ID    TSPSEL  ETSDU UDCRQ UDCRS UDDRQ ADDRFORM
================================================
T2      -       12    92    92     1  WANNEA
T3      -       16    32    32    64  RFC1006
T3      -       16    32    32    64  LANINET
T4      -       16    32    32    64  WANSBKA
T5      -       32   256   256   129  WAN3SBKA
T5      -       32   256   256   129  SDLCSBKA
T6      -       16    32    32    64  LOOPSBKA
T6      -       16    32    32    64  TRSNASBKA
------------------------------------------------
```

In der Darstellung wurden stellenweise Zwischenräume und führende Nullen weggelassen. '-' bedeutet, dass kein passender Wert vorliegt.

Die Spalten haben folgende Bedeutung:

ID
: Symbolische Bezeichnung und Typ der bedienten CCs/TSP Access Points (siehe Abschnitt „Informationen zu den CCs/TSP Access Points (Option -b␣all)“ auf Seite 244).

CCP
: Ist der TSP Access Point (ID=T[1-6]) nicht aktiv oder wird vom CCP kein Transportdienst erbracht (andere IDs), so wird hier „no info“ ausgegeben, und alle anderen Felder sind leer.

VERS
: Version des TSP Access Points in hexadezimaler Darstellung.

TSAP
: Anzahl der TS-Anwendungen, die der TSP gleichzeitig unterstützen kann.

TCEP
: Anzahl der Transportverbindungen (TCEPs), die der TSP gleichzeitig aufrecht erhalten kann.

TIDU
: Vom TSP unterstützte maximale Länge einer TIDU (Transport Interface Data Unit). Eine TIDU ist die Dateneinheit, die die Anwendung bei einem einzelnen Aufruf zum Daten senden an CMX übergibt oder zum Daten empfangen von CMX erhält.

TSP
: Maximale Anzahl der Transport Service Provider (TSPs), die das CCP gleichzeitig bedienen kann. Bei TSP Access Points (ID=T[1-9]) ist dieser Wert stets 1.

SUBRV
: SUBRV enthält den Korrekturstand des CCPs. Bei TSP Access Points ist dieser Wert bedeutungslos.

TSPSEL
: Für TSPSEL wird die Kennung des TSP binär ausgegeben.

ETSDU
: Länge der vom CCP/TSP unterstützten ETSDU (Expedited Transport Service Data Unit). ETSDU gibt die Größe der Vorrangdateneinheit an, die das CCP/der TSP mit einem Sendeauftrag transferieren kann. Vor-

rangdaten sind Daten, die das CCP/der TSP mit Vorrang zu Normaldaten transferiert. Der Wert 0 bedeutet, dass Vorrangdaten nicht unterstützt werden.

UDCRQ, UDCRS, UDDRQ
: Beim Verbindungsaufbau und Verbindungsabbau können TS-Anwendungen Informationen in Form von Benutzerdaten an den Kommunikationspartner übergeben. Die erlaubte Länge dieser Benutzerdaten ist abhängig vom CCP/TSP und wird folgendermaßen definiert:

  UDCRQ
  : Maximale Länge der Benutzerdaten bei der Verbindungsanforderung (ICMX(L)-Aufruf *t_conrq*, XTI-Aufruf *t_connect*) einer lokalen an eine ferne TS-Anwendung.

  UDCRS
  : Maximale Länge der Benutzerdaten, wenn eine lokale TS-Anwendung die Verbindungsanforderung einer fernen TS-Anwendung beantwortet (ICMX(L)-Aufruf *t_conrs*, XTI-Aufruf *t_accept*).

  UDDRQ
  : Maximale Länge der Benutzerdaten beim Verbindungsabbau durch eine lokale TS-Anwendung (ICMX(L)-Aufruf *t_disrq*, XTI-Aufruf *t_snddis*).

ADDRFORM
: das vom TSP Access Point unterstützte Adress-Format bzw. die unterstützten Adress-Formate. Die Bedeutung der Adress-Formate ist im Abschnitt „Adress-Formate“ auf Seite 84 beschrieben. Das Adress-Format wird in hexadezimaler Darstellung ausgegeben, wenn kein Klartextstring für das unterstützte Adress-Format vorhanden ist.

Informationen zu allen aktiven TSAPs (Option -s -v)

```
    TSAP       PID     THREAD-ID    TSTAT   #TCEP
================================================
     0.0      9337             -     N/A       0
     1.0     10319             -     N/A       0
     2.0     10320             -     N/A       0
  5622.0      7942             1     N/A      75
  5622.0      7942             4     N/A      75
  5622.0      7942             5     N/A      75
  5622.0      7942             6     N/A      75
------------------------------------------------
```

Bei Aufruf ohne die Option *-v* entfällt die Spalte THREAD-ID.

Die Spalten der Tabelle haben folgende Bedeutung:

TSAP
: Identifikation des TSAP.

PID
: Prozess-Identifikation.

THREAD-ID
: Thread-Identifikation.

TSTAT
: Zustand des TEPs im Fall einer XTI-Anwendung. Folgende Werte sind möglich:

  UNBND
  : ungebunden

  IDLE
  : keine Verbindung aufgebaut

  OUTCON
  : abgehender Verbindungsaufbauwunsch noch nicht beantwortet

  INCON
  : ankommender Verbindungsaufbauwunsch noch nicht beantwortet

  DATA
  : Datentransfer

  OUTREL
  : abgehender Verbindungsabbauwunsch noch nicht beantwortet

  INREL
  : ankommender Verbindungsabbauwunsch noch nicht beantwortet

  N/A
  : not available (bei ICMX(L)- und ICMX(NEA)-Anwendung)

#TCEP
: Anzahl der Verbindungen

Information zu einem bestimmten TSAP (Option -S *id* -v)

```
  TSAP       PID THREAD-ID     TCEP  L/R CSTAT  SDAT  RDAT ID  OWN_GLOBAL_NAME
=============================================================================
5624.0      8874         3  11593.0    R  DATA   17M   17M  T6  DEMO_PD01
    -       8874         4  11595.0    R  DATA   18M   18M  T6  DEMO_PD01
    -       8874         6  11609.0    R  DATA   15M   15M  T6  DEMO_PD01
    -          -         6  11611.0    R  DATA   16M   16M  T6  DEMO_PD01
    -          -         6  11613.0    R  DATA   17M   17M  T6  DEMO_PD01
    -          -         6  11615.0    R  DATA   16M   16M  T6  DEMO_PD01
    -          -         6  11617.0    R  DATA   15M   15M  T6  DEMO_PD01
    -       8874         7  11619.0    R  DATA   16M   16M  T6  DEMO_PD01
    -          -         7  11621.0    R  DATA   16M   16M  T6  DEMO_PD01
    -          -         7  11623.0    R  DATA   15M   15M  T6  DEMO_PD01
    -          -         7  11625.0    R  DATA   14M   14M  T6  DEMO_PD01
    -          -         7  11628.0    R  DATA   14M   14M  T6  DEMO_PD01
    -          -         7  11632.0    R  DATA   13M   13M  T6  DEMO_PD01
    -       8874         5  11601.0    R  DATA   15M   15M  T6  DEMO_PD01
    -          -         5  11603.0    R  DATA   16M   16M  T6  DEMO_PD01
    -          -         5  11605.0    R  DATA   16M   16M  T6  DEMO_PD01
    -          -         5  11607.0    R  DATA   17M   17M  T6  DEMO_PD01
-----------------------------------------------------------------------------
```

Bei Aufruf ohne die Option *-v* entfällt die Spalte THREAD-ID.

Information zu allen momentan verfügbaren TSAPs (Option -S all -v)

```
  TSAP       PID THREAD-ID     TCEP  L/R CSTAT  SDAT  RDAT ID OWN_GLOBAL_NAME
=============================================================================
   0.0      9337         -        -    -      -     -     - - <CMX-Daemon>
   1.0     10319         -        -    -      -     -     - - N/A
   2.0     10320         -        -    -      -     -     - - $FJAM
5622.0      7942         1  11551.0    R   DATA    0B    0B T6  DEMO_PD01
    -          -         1  11552.0    R   DATA    0B    0B T6  DEMO_PD01

    -       7942         8  11482.0    R   DATA   18M   18M T6  DEMO_PD01
5623.0      8001        12  11480.0    L   DATA   18M   18M T6  DEMO_AD01
    -       8001        11  11481.0    L   DATA   15M   15M T6  DEMO_AD01
    -       8001         6  11484.0    L   DATA   18M   18M T6  DEMO_AD01
    -       8001        13  11485.0    L   DATA   17M   17M T6  DEMO_AD01
-----------------------------------------------------------------------------
```

Bei Aufruf ohne die Option *-v* entfällt die Spalte THREAD-ID.

Information zum TCEP *id* (Option -C *id*)

```
TCEP    L/R  CSTAT  SDAT  RDAT ID  OWN_GLOBAL_NAME  REMOTE_GLOBAL_NAME
=============================================================================
4712.0   L   DATA  202B  33B  T1  AD04             AD04
-----------------------------------------------------------------------------
```

Information zu allen momentan vefügbaren TCEPs (Option -C all)

```
TCEP   L/R CSTAT  SDAT  RDAT ID   OWN_GLOBAL_NAME     REMOTE_GLOBAL_NAME
=========================================================================
4709.0   L  DATA  202B  110B  T1  Mueller.Egon.Mch-P  Meier.Egon.Pad
4710.0   L  DATA    OF    OF  T2  Mueller.Egon.Mch-P  Meier.Franz.Pad
4711.0   L  DATA   22K  121K  W1  Meier.Otto.Mch-P    Mueller.Egon.Pad
4712.0   L  DATA  202B   12B  T1  Meier.Otto.Mch-P    Mueller.Emil.Pad
4713.0   L  DATEX   OF    OF  T2  Meier.Otto.Mch-P    N/A
4714.0   L  DATA  102K   13K  T6  Meier.Otto.Mch-P    N/A
4715.0   R  DATST  44M   56M  W1  Meier.Otto.Mch-P    Mueller.Fritz.Pad
```

Ausgabe mit der Option -C id -M

Falls das TS-Directory nicht alle GLOBALEN NAMEN von TCEPs enthält (Anzeige “N/A”), können Sie die Option *-M* benutzen. Sie liefert eine alternative Ausgabe für einen bestimmten TCEP (oder für alle TCEPs mit der Option *all* statt *id*).

```
TCEP    LOCAL_NAME                    TRANSPORT_ADDRESS
=========================================================================
4712.0  TSEL LOOPSBKA A'Otto1'
     -  TSEL LANINET A'4711'
     -  TSEL RFC1006 A'D018V008'
     -   -                             TA LOOPSBKA A'Emil1'
```

Die Spalten haben folgende Bedeutung:

TSAP
: Identifikation des TSAP. Ist dieser TSAP nicht aktiv, so wird „no info" ausgegeben.

PID
: Prozess-Identifikation

THREAD-ID
: Thread-Identifikation.

TCEP
: Identifikation des TCEP

L/R
: Angabe, wie der Aufbau der Transportverbindung initiiert wurde.

  L
  : Durch die lokale Anwendung

  R
  : Durch die ferne Anwendung

CSTAT
: Zustand des TCEP. Folgende Werte sind möglich:

  EXIST
  : TCEP existiert

  DATA
  : Datentransfer

  DATEX
  : Stop für Normaldaten

  DATST
  : Stop für Normal- und Vorrangdaten

  REDIN
  : Verbindungsumlenkung angekündigt

  DISIN
  : Verbindungsabbauanzeige angekündigt

  CONIN
  : Verbindungsaufbauanzeige angekündigt

  CONRQ
  : Verbindungsaufbauwunsch abgeschickt

SDAT
: Anzahl gesendeter Daten

  (T = Terabyte, G = Gigabyte, M = Megabyte, K = KByte, B = Byte).

  Maximal 9999T, sonst Überlauf, der mit „OF" gekennzeichnet wird.

RDAT
: Anzahl empfangener Daten (vgl. SDAT)

ID
: CC/TSP Access Point, über den die Verbindung aufgebaut wurde

OWN_/REMOTE_GLOBAL_NAME
: GLOBALER NAME, der für den TSAP im TS-Directory vergeben wurde (OWN = lokal, REMOTE = fern), oder „N/A“, wenn dieser nicht im TS-Directory gefunden wurde.

LOCAL_NAME
: Ein oder mehrere Transport-Selektoren (TSEL) der lokalen Anwendung

TRANSPORT_ADDRESS
Transportadresse der fernen Anwendung

Information zu LAN-Interfaces (Option -i)

Mit der Option *-i* erhalten Sie Informationen zu allen LAN-Interfaces, die für TS-Anwendungen zur Verfügung stehen.

```
INTERFACE   TYPE      STATE ADDRESS                    MACADDR
=============================================================================
hme0        ETHN(PCI) UP    172.25.236.26              0:80:17:28:7b:8
qfe0        ETHN(PCI) UP    10.99.218.44               8:0:20:e4:c4:34
qfe0        ETHN(PCI) UP    fe80::a00:20ff:fee4:c434   8:0:20:e4:c4:34
-----------------------------------------------------------------------------
```

Die Spalten haben folgende Bedeutung:

INTERFACE
Name des Netzwerk-Interface.

TYPE
Typ des Subnetzes, welches über das Interface erreicht wird.

STATE
Status des LAN-Interface. Folgende Werte sind möglich:

UP
Das Interface ist betriebsbereit.

DOWN
Das Interface ist nicht betriebsbereit.

ADDRESS
IP-Adresse des LAN-Interface.

MACADDR
MAC-Adresse des LAN-Interface, falls verfügbar.

Information zu WAN-Interfaces (Option -b *id* -l)

Zusammen mit der Option *-b␣-id* liefert die Option *-l* Informationen zu allen WAN-Interfaces des Controllers, der mit *id* angegeben wurde.

```
ID   TYPE    INTERFACE_1  INTERFACE_2  INTERFACE_3  INTERFACE_4
=============================================================
W14  PWXV-4      V.24         UNUSED       X.21         X.21
-------------------------------------------------------------
```

Der Aufruf *cmxinfo␣-b all␣-l* liefert folgende Ausgabe:

```
ID   TYPE    INTERFACE_1  INTERFACE_2  INTERFACE_3  INTERFACE_4
=================================================================
W3
W12  PWS2
W14  PWXV-4      V.24         UNUSED       X.21         X.21
-----------------------------------------------------------------
```

Die Spalten haben folgende Bedeutung:

ID
: Symbolische Bezeichnung des WAN-CCs mit den Werten W[1-32].

TYPE
: Typ des Controllers.

INTERFACE_[1-4]
: Interface-Typ. Für die folgenden PCI-Controller ist der Interface-Typ variabel:

  PWXV-[2|4] mit 2 oder 4 Leitungen: UNUSED, V.24, X.21, V.35V, V.35P, V.10, V.36, LOOP.

10.6 CMX-Bibliotheks-Trace steuern und aufbereiten (cmxl)

Der Trace der CMX-Bibliothek wird über die Umgebungsvariable CMXTRACE aktiviert und gesteuert. Die Trace-Einträge eines Prozesses werden kompakt und binär in einem dynamisch angelegten Puffer gesammelt und periodisch in temporäre Dateien geschrieben. Die Aufbereitung dieser Dateien erfolgt entkoppelt durch *cmxl*.

Für Multi-Threading (MT) gelten einige Abweichungen, auf die im Text hingewiesen wird.

Steuerung des Trace - CMXTRACE

Jeder CMX-Aufruf *t_attach* eines Prozesses wertet die Umgebungsvariable CMXTRACE aus und aktiviert gegebenenfalls den Trace. CMXTRACE muss vor dem Starten der Anwendung, d. h. vor dem ersten *t_attach* des zu überwachenden Prozesses, gesetzt worden sein. Nach dem Aktivieren des Trace wird die temporäre Datei *CMXLapid* mit der Prozess-Identifikation *pid* eröffnet, falls sie nicht bereits eröffnet ist. Für die Dateien werden die Zugriffsrechte `rw-------` (0600) vergeben. Anschließend wird dynamisch Speicher für die Pufferung der Trace-Einträge belegt.

Beim Multi-Threading werden die Trace-Einträge aller Threads eines Prozesses in die temporäre Datei *CMXLapid* geschrieben.

Speicher und Dateien bleiben für die Lebensdauer des Prozesses belegt.

Die in CMXTRACE angegebenen Optionen steuern den Trace. Die Optionen *s*, *S*, *D* und *G* bestimmen den Umfang der Protokollierung. Die Optionen *p, r* steuern Pufferung und Rundschreiben der Datei.

Syntax der Umgebungsvariable

CMXTRACE="[␣**-s**] [␣**-S**] [␣**-D**] [␣**-p**␣fac] [␣**-r**␣wrap] [␣**-f**␣directory] [␣**-G**␣dirx]";

export␣CMXTRACE

Die Optionen *-s, -S* und *-D* bestimmen die Art des Trace. Um den Trace zu aktivieren, muss ein Wert angegeben werden.

-s

Es erfolgt eine gewöhnliche Protokollierung der ICMX(L)-Aufrufe, ihrer Argumente, der Optionen und Benutzerdaten.

-S

Es erfolgt eine ausführliche Protokollierung der Aufrufe, ihrer Argumente, des Inhalts eventueller Optionen, der Benutzerdaten in ihrer Gesamtlänge.

Die Optionen *s* und *S* schließen sich gegenseitig aus.

-D

Es erfolgt eine ausführliche Protokollierung der Aufrufe mit zusätzlichen Informationen über Systemaufrufe. Diese Angabe kann nur zusätzlich zu *s* oder *S* gemacht werden.

-p␣fac

Durch die Dezimalziffer *fac* wird der Faktor der Pufferung bestimmt. Die Pufferung erfolgt im Betrag *fac* * 1024. Wird *fac* = 0 angegeben, so wird jeder Trace-Eintrag sofort (ungepuffert) in die Datei geschrieben.

fac=0...8.
-p␣fac nicht angegeben: Es wird *fac=1* angenommen.

-r␣wrap

Durch die Dezimalzahl *wrap* wird angegeben, dass nach *wrap* * 1024 in die zweite temporäre Datei
directory/CMXMapid protokolliert werden soll (*pid* = Prozess-ID).

Die zweite Datei *CMXMapid* behandelt der Trace genauso wie *CMXLapid*.

Nach jeweils *wrap* * 1024 schaltet der Trace zwischen *CMXLapid* und *CMXMapid* um. Dabei geht der alte Inhalt der jeweiligen Datei verloren.

-r wrap nicht angegeben: Es wird *wrap* = 512 angenommen.

-f␣directory

Die Trace-Dateien werden in das angegebene Directory geschrieben. *-f directory* nicht angegeben: Für *directory* wird */var/opt/SMAWcmx/tmp* angenommen.

-G␣dirx

Es erfolgt eine ausführliche Protokollierung mit zusätzlichen Informationen über DIR.X-Aufrufe. Die Protokolldatei wird im laufenden Directory unter dem Namen *logfile.pid* abgelegt.

dirx ist folgendermaßen zu belegen:

0x0
Verfolgen TNS-Namen

0x02
Verfolgen DIR.X-Namen

0x04
Verfolgen Konfiguration

0x08
Verfolgen interne Aufrufe

0x10
Verfolgen „alarms“

Diese Werte können binär kombiniert werden.

Die Option wird in der MT-Bibliothek ignoriert.

Aufbereitung des Trace - cmxl

cmxl liest die vom Trace erzeugten Einträge aus der temporären Datei *file*, verarbeitet sie entsprechend den angegebenen Optionen und gibt das Ergebnis auf *stdout* aus.

Syntax

cmxl [␣**-c**] [␣**-d**] [␣**-e**] [␣**-t**] [␣**-v**] [␣**-x**] [␣**-D**]␣file ...

Die Optionen geben an, welche Trace-Einträge aus *file* aufbereitet werden sollen. Es ist möglich, mehrere der im Folgenden beschriebenen Werte pro Aufruf von *cmxl* anzugeben. Lediglich die Optionen *v* und *x* schließen einander aus.

Keine Option angegeben: Es wird *cdex* angenommen.

-c

Die Aufbereitung erfolgt für die ICMX(L)-Aufrufe:

- zur An- und Abmeldung der TS-Anwendung bei CMX
- zum Verbindungsaufbau und -abbau
- zur Verbindungsumlenkung

-d

Die Aufbereitung erfolgt für die ICMX(L)-Aufrufe:

- zum Datenaustausch
- zur Flussregelung

-e

Die Aufbereitung erfolgt für die ICMX(L)-Aufrufe zur Ereignisbehandlung.

-t

Es erfolgt zusätzlich zu der Protokollierung der Fehlermeldungen eine explizite Aufbereitung der *t_error()*-Aufrufe.

Fehlermeldungen werden immer protokolliert, auch wenn diese Option nicht angegeben wird.

-v

Es erfolgt eine ausführliche Aufbereitung der ICMX(L)-Aufrufe, ihrer Argumente, der Optionen, der Benutzerdaten. Der Umfang der Aufbereitung der Daten ist abhängig von den bei CMXTRACE angegebenen Optionen.

-x

Es erfolgt eine eingeschränkte Aufbereitung der Aufrufe und ihrer Argumente *ohne* Optionen und Benutzerdaten.

-D

Es erfolgt eine ausführliche Aufbereitung mit zusätzlichen Informationen über Systemaufrufe.

file ...

Name einer oder mehrerer Dateien mit Trace-Einträgen, die aufbereitet werden sollen.

Ausgabeformate (CMXTRACE, NEATRACE, cmxl, neal)

Die Beschreibung der Trace-Informationen in diesem Abschnitt erfordert Kenntnisse über die Programmschnittstellen ICMX(L) bzw. ICMX(NEA) von CMX. Die Programmschnittstellen sind im Handbuch „CMX, Anwendungen programmieren“ [1] beschrieben.

Das Format der durch *cmxl* und *neal* aufbereiteten Trace-Informationen ist gleich. Aus diesem Grund wird hier nur das Ausgabeformat von *cmxl* beschrieben. Die Ausgabe von *neal* ist damit ebenfalls interpretierbar. Die Angaben t_... sind lediglich durch x_... zu ersetzen mit Ausnahme der Aufrufe *t_vdatarq()* und *t_vdatain()*.

Die von *cmxl* aufgearbeiteten Trace-Informationen werden in folgendem Format ausgegeben:

```
Kopf-  ICMX(L) TRACE (G-V6.0) datum hh:mm:ss
zeilen OPTIONS ‚cdex‘ TRACE FILE ‚trace file‘ ICMX(L) V6.0E

1. Zeile        zeitstempel t_xxxxx(args in %d, 0x%x, %s)
2. Zeile        [Optionen und Benutzerdaten in %d, %x, %s]
3. Zeile                  [TRANSPORTADRESSE, LOKALER NAME]
4. Zeile        [Ergebnisse, Ereignisse in %d, %x, %s]
1. Zeile        zeitstempel ... nächster Aufruf ...
```

Die zwei Kopfzeilen werden einmal zu Beginn der Ausgabe der Trace-Informationen ausgegeben. Sie enthalten:

- Versionsbezeichnung und CMX-Version (hier V6.0)
- Startdatum (datum) und Startzeit (hh:mm:ss) des Trace
- die ausgewählten Optionen zur Aufbereitung (hier sind es die Standardoptionen *c, d, e, x*)
- Name der aufbereiteten Trace-Datei (trace file)
- die Version der Programmschnittstelle.

Die Ausgabe der Trace-Informationen zu den einzelnen Funktionsaufrufen erfolgt in mehreren Zeilen unterschiedlichen Formats. Im Folgenden sind die Formate dieser Zeilen beschrieben (Bezeichnungen der Zeilenformate: 1. Zeile bis 4. Zeile).

1. Zeile

Jeder protokollierte Funktionsaufruf erhält am Anfang der 1. Zeile einen Zeitstempel. Bei *cmxl* hat der Zeitstempel die Form hh:mm:ss:msc mit hh = Stundnen, mm = Minuten, ss = Sekunden und msc = Millisekunden. Bei *neal* hat der Zeitstempel die Form hh:mm:ss.

Es folgen der protokollierte Funktionsaufruf *(t_xxxxx)* und in runden Klammern die Werte der Argumente *(args)* in der von ICMX(L) verlangten Reihenfolge. Die Argumente werden in dezimaler (%d), hexadezimaler (0x%x) oder symbolischer (%s) Form dargestellt.

Zur Interpretation der protokollierten Werte ist Folgendes zu beachten:

- Bei den Argumenten *datap, fromaddr, name, opt, toaddr* wird die übergebene Adresse (0x%x) dargestellt.
- Bei den Argumenten *chain, flags, cmode, pid, reason* ist hier der entsprechende Wert (0x%x, %d, oder %s) dargestellt, auch wenn die Argumente eigentlich die Adressen dieser Werte sind.
- Beim Argument *tref* ist der entsprechende Wert (0x%x) dargestellt. Ausnahmen sind die Aufrufe: *t_conrq()* und *t_event()*. Dort wird für *tref* die übergebene Adresse dargestellt.
- Bei der Protokollierung von Datenlängen, z. B. *datal* ist der beim Aufruf gültige Wert (%d) dargestellt. Bei folgenden Aufrufen wird zusätzlich der eventuell modifizierte Wert (%d), der nach der Rückkehr gültig ist, ausgegeben:

  t_concf(), t_conin(), t_datain(), t_vdatain(), t_xdatin(), t_redin() und *t_disin()*

  Beide Werte sind durch >< (Größerzeichen Kleinerzeichen) voneinander getrennt.

Die 2. Zeile und die 3. Zeile werden nur ausgegeben, wenn die Option *v* bei *cmxl* angegeben wurde und der Trace entsprechende Informationen gesammelt hat (Option *-S*).

2. Zeile

In der 2. Zeile sind die Optionen mit Optionsnummern und Optionsfeldern protokolliert. Sie stehen in der Reihenfolge, wie sie in der Optionsstruktur gemäß der Headerfile *<cmx.h>* deklariert sind.

Zur Interpretation der protokollierten Werte ist Folgendes zu beachten:

- Bei den Optionsfeldern *t_maxl, t_optnr, t_timeout* und *t_xdata* wird der übergebene Wert (0x%x, %d oder %s) dargestellt. Bei *t_udatap* wird die übergebene Adresse dargestellt.
- Bei dem Argument *t_udatal* wird der beim Aufruf gültige Wert (%d) dargestellt. Bei den Aufrufen *t_conin()*, *t_concf()*, *t_disin()* und *t_redin()* wird zusätzlich der eventuell modifizierte Wert (%d), der bei der Rückkehr gültig ist, ausgegeben. Die beiden Werte werden durch >< (Größerzeichen Kleinerzeichen) voneinander getrennt.

3. Zeile

Die Zeilen mit dem Format „3. Zeile" protokollieren die TRANSPORTADRESSE, den LOKALEN NAMEN und Benutzerdaten, sofern diese vom Trace protokolliert und von *cmxl* aufbereitet wurden. Es folgen die Daten, dargestellt in hexadezimaler und in abdruckbarer Form.

Ein Beispiel für die Ausgabe in der 3. Zeile:

```
Distanz   hexadezimal dargestellte Daten    . abdruckbar
  0     4c4f4b41 4c455220 4e414d45 20242424 .LOKALER NAME $$$
```

4. Zeile

In der 4. Zeile eines Eintrags wird das Ergebnis des Aufrufs protokolliert. Im Falle eines Fehlers wird T_ERROR eingetragen. Wurde der Aufruf erfolgreich durchgeführt, so wird das Ergebnis nur protokolliert, wenn es von T_OK abweicht. Das Ergebnis wird dann zusammen mit den Informationen protokolliert, die durch den Aufruf zurückgeliefert wurden.

Dies ist bei folgenden Aufrufen das folgende Ergebnis:

- *t_attach*: das Ergebnis T_NOTFIRST
- *t_conrq*: die gelieferte Transportreferenz (tref)
- *t_event*: das gemeldete Ereignis und die zugehörige Transportreferenz (tref)
- *t_datain, t_vdatain, t_xdatin*: die Restlänge noch zu lesender Daten
- *t_datarq, t_vdatarq, t_xdatrq*: die Ergebnisse T_DATASTOP, T_XDATSTOP

Die Darstellung der protokollierten Ergebnisse der Aufrufe erfolgt in der Regel auf folgende Weise:

- dezimal (%d) bei Längen oder Wertangaben
- symbolisch (%s), wenn eine entsprechende Definition in der Datei *<cmx.h>* existiert
- hexadezimal (0x%x) in allen anderen Fällen

Sind für ein Argument oder Optionsfeld zwar Symbole definiert, aber der Wert entspricht keinem der zulässigen Symbole, so erfolgt die Ausgabe hexadezimal (0x%x) mit einem Fragezeichen (?). Zum leichteren Auffinden sind die Ergebnisse T_ERROR mit mehreren # gekennzeichnet.

Beispiel für die Aufbereitung durch cmxl

Für die Aufbereitung des Trace wurden die Optionen *c, d, e, x* ausgewählt. D. h. es werden alle Aufrufe für An- und Abmeldung bei CMX, Verbindungsaufbau, -abbau und -umlenkung, Datenaustausch und Flussregelung aufbereitet mit ihren Argumenten ohne Optionen, ohne Benutzerdaten.

ICMX(L) Trace Expansion *Expanded: Jan 1 13:12:41*

```
Trace collected with data length 32 starting Jan 1 13:10:52.
Application was running in 32-bit mode.
Trace expanded by CMX 6.0 from file „CMXLa01180“ with options
```

```
c      expand ICMX connection handling calls (set by default)
d      expand ICMX data and flow control calls (set by
default)
e      expand ICMX event handling calls (set by default)
v      verbose mode showing args, options, user data
using T_MSG_SIZE=256, FD_SETSIZE=1024.

Traced System and CMX: SunOS PGTR0046 5.7 106541-04 sun4us;
CMX 6.0E00 14

hh:mm:ss.msc ICMX Call t_*****or (intermediate) results
13:10:52.000 t_getloc(0xffbefe00, NULL)
     glob:
     0 52315f32 315f3030 30302e50 322e7266 |R1_21_0000.P2.rf|
    10 63313030 362e5049 54                |c1006.PIT       |
     loc 0xff3870cc: RFC1006
     0 01000018 000e0000 00000004 0008d7c9 |                |
     10 e3f2f0f0 f0f00000                  |                |
13:10:52.000 t_attach(0x3e034, 0x3c454)
     opt:
     t_optnr T_OPTA1 (1) t_apmode T_PASSIVE (2)
     t_conlim 8
     name: RFC1006
     0 01000018 000e0000 00000004 0008d7c9 |                |
    10 e3f2f0f0 f0f00000                   |                |
14:24:16.000 returns T_OK (0)
14:24:16.000 t_event (0x3e02c, T_WAIT, 0x4e4a0)
     opt:
     t_optnr T_OPTE1 (1) t_timeout T_NOLIMIT (-1)
14:24:16.000 returns T_CONIN (5) tref 0xa53
     t_attid 0x104 t_uattid 0x0
     t_ucepid 0xe t_evdat 0x0 (0)
14:24:16.000 t_conin (0xa53, 0xffbef9e8, 0xffbef8b0,
0xffbefb40)
     opt:
     t_optnr T_OPTC1 (1) t_udatap 0xffbef7a8 t_utada1 256><0
     t_xdata T_NO (0) t_timeout T_NO (0)
     toaddr:RFC1006
```

Dateien

CMXLapid, CMXMapid
: Dateien mit kompakten Trace-Einträgen in Binärformat.

logfile.pid
: Enthält DIR.X-Trace-Einträge.

Falls bei CMXTRACE nicht anders angegeben, werden die Dateien *CMXLapid* und *CMXMapid* für den CMX-Bibliothekstrace im Verzeichnis */var/opt/SMAW-cmx/tmp* angelegt. *pid* bezeichnet die Prozess-ID.

10.6.1 Hinweise für Multi-Threading

Ein in CMX V6.0 erzeugter CMX-Bibliothekstrace kann thread-spezifisch aufbereitet werden. Dafür wird das Kommando *cmxl_mt* bereitgestellt. Es wertet die zuvor mit *cmxl* aus dem Binärformat erzeugte ASCII-Datei *file* aus (*cmxl ... file*). Für Diagnosezwecke sollte jeweils die komplette ASCII-Datei zur Verfügung gestellt werden.

Die Prozedur erlaubt folgende Aufruf-Syntax:

cmxl_mt [␣**-h**] [␣**-t**␣{ tid | **all** }] [␣file ...]

Optionen und Argumente

cmxl_mt (Aufruf ohne Argumente)
Die ASCII-Tracedatei wird von *stdin* gelesen; es werden lediglich der Trace-Header und eine Liste der in der Tracedatei vorkommenden Thread IDs auf stdout ausgegeben.

-h
Anzeige der Syntax-Beschreibung.

-t␣tid
Die ASCII-Tracedatei wird von *stdin* gelesen; es werden alle Einträge für die Thread-ID *tid* auf *stdout* ausgegeben.

-t␣**all**
Die ASCII-Tracedatei wird von stdin gelesen; für jeden Thread werden dessen Einträge in die Datei *file.tid* ausgegeben. Der Trace-Header und die Liste der Thread-IDs werden in der Datei *file.hdr* abgelegt

file ...
Es wird die ASCII-Tracedatei *file* gelesen und lediglich der Trace-Header und die Thread-IDs auf stdout ausgegeben.

-t␣tid␣file
Es wird die ASCII-Tracedatei *file* gelesen und die Einträge für den Thread *tid* auf *stdout* ausgegeben.

-t␣all␣file

Es wird die ASCII-Tracedatei *file* gelesen; für jeden Thread werden dessen Einträge in die Datei *file.tid* ausgegeben. Der Trace-Header und die Liste der Thread-IDs werden in der Datei *file.hdr* abgelegt.

Ausgabeformate

Das Ausgabeformat bei MT ändert sich gegenüber der bisherigen Ausgabe nur geringfügig. In der durch *cmxl* aufbereiteten ASCII-Datei ist zusätzlich die Thread ID in der Form "[*tid*]" ausgewiesen (siehe unten). Alle nachfolgenden Trace-Einträge sind solange diesem Thread zugeordnet, bis eine andere Thread ID ausgewiesen wird. Die Bedeutung der ausgegebenen Zeilen bleibt wie bisher.

Beispiele

Ausgabeformat des durch cmcl aufbereiteten *tracefile*:

```
ICMX(L) Trace Expansion                          Expanded: Oct  1 14:34:48

Trace collected with data length 0 starting Oct  1 14:30:58.
Application was running in 32-bit multithreaded mode.
Trace expanded by CMX 6.0 from file "CMXLa27504" with options
c       expand ICMX connection handling calls (set by default)
d       expand ICMX data and flow control calls (set by default)
D       expand system calls (sockets etc.; implies v)
e       expand ICMX event handling calls (set by default)
v       verbose mode showing args, options, user data
X       expand XTI system calls in case of ICMX over XTI
using T_MSG_SIZE=256, FD_SETSIZE=1024, openmax=256,
      fac=0, wrap=1024000000.

Traced System and CMX: SunOS PGTR0046 5.9 Generic sun4us; CMX 6.0E50
09hh:mm:ss.msc ICMX Call t_***** or (intermediate) results
[0001]
14:30:58.000 t_getloc(0xffbffb2d, NULL)
               glob:
              0  44454d4f 5f414430  31                 |DEMO_AD01       |
               loc 0x28d68: LOOPSBKA
              0  01000018 000e0000  00000400 00094445  |              DE|
             10  4d4f5f41 44303100                     |MO_AD01         |
14:30:58.000 t_getaddr(0xffbffb3a, NULL)
               glob:
              0  434d585f 504153                       |CMX_PAS         |
               addr 0x28e30: LOOPSBKA
              0  02000013 04000010  00098007 434d585f  |            CMX_|
             10  504153                                |PAS             |
[0002]

14:30:58.000 t_attach(0x23df4, 0xfeffbf7c)
               opt:
               t_optnr T_OPTA6 (6)  t_apmode T_ACTIVE (1)
               t_conlim 1
```

```
                name: LOOPSBKA
               0  01000018 000e0000  00000400 00094445                DE|
               10  4d4f5f41 44303100                    |MO_AD01         |
```

Ausgabeformat der Headerdatei *tracefile*.hdr:

```
ICMX(L) Trace Expansion                          Expanded: Oct  1 14:34:48

Trace collected with data length 0 starting Oct  1 14:30:58.
Application was running in 32-bit multithreaded mode.
Trace expanded by CMX 6.0 from file "CMXLa27504" with options
c       expand ICMX connection handling calls (set by default)
d       expand ICMX data and flow control calls (set by default)
D       expand system calls (sockets etc.; implies v)
e       expand ICMX event handling calls (set by default)
v       verbose mode showing args, options, user data
X       expand XTI system calls in case of ICMX over XTI
using T_MSG_SIZE=256, FD_SETSIZE=1024, openmax=256,
      fac=0, wrap=1024000000.

Traced System and CMX: SunOS PGTR0046 5.9 Generic sun4us; CMX 6.0E50 09

hh:mm:ss.msc ICMX Call t_***** or (intermediate) results

Found Thread Id's
    4
    3
    5
    1
    2
```

Ausgabeformat einer threadspezifischen Aufbereitung der Datei *tracefile.tid*:

```
hh:mm:ss.msc ICMX Call t_***** or (intermediate) results
14:30:58.000 t_attach(0x23df4, 0xfeffbf7c)
                opt:
                t_optnr T_OPTA6 (6)  t_apmode T_ACTIVE (1)
                t_conlim 1
                name: LOOPSBKA
               0  01000018 000e0000  00000400 00094445  |          DE|
              10  4d4f5f41 44303100                     |MO_AD01     |
14:30:58.000    T_COM (0)
14:30:58.000    open("/dev/SMAWcmx/cxnet", 0) == 5
14:30:58.899    ioctl(5, STINIRQ, 0x330cc) = 0
14:30:58.899    T_OK (0)
                t_uattid 0x0 t_attid 0x2d06 t_sptypes 0xc00
                t_cclist (0x331c0): 1 504
14:30:58.899  returns T_OK (0)
```

10.7 CMX-Monitor (cmxm)

Der CMX-Monitor beobachtet die laufenden Aktivitäten des CMX-Automaten. Er gibt Zählerstände aus und ermittelt Statistiken über die Auslastung der einzelnen Komponenten von CMX und der TSP Access Points im Betriebssystemkern.

Der CMX-Monitor liefert Informationen über:

– die Anzahl der angemeldeten TS-Anwendungen und die Anzahl der angemeldeten Prozesse, die diese TS-Anwendungen steuern

– die Anzahl der bestehenden Transportverbindungen zu fernen und lokalen Kommunikationspartnern

– die Aktivitäten der TS-Anwendungen, z. B. die Anzahl der Aufrufe von ICMX- und XTI-Funktionen pro Sekunde

– die Menge der gesendeten und empfangenen Daten pro Sekunde

– Informationen über die betriebenen TSP Access Points, z. B. ob ein TSP Access Point betriebsbereit ist oder nicht, wieviele Transportverbindungen auf den einzelnen TSP Access Points aktiv sind usw.

Die von dem CMX-Monitor ermittelten Statistikwerte sind bei der Beschreibung der Ausgabeformate im Detail dargestellt.

Den CMX-Monitor können Sie mit dem Kommando *cmxm* aufrufen. Beim Aufruf können Sie zwischen 3 Betriebsarten des CMX-Monitors wählen. Sie unterscheiden sich in Art und Umfang der gelieferten Werte und in der Form der Ausgabe:

– Tabellarische Ausgabe von Statistiken
– Semigraphische Ausgabe von Statistiken
– Summarische Ausgabe der Zählerstände

Tabellarische Ausgabe von Statistiken

Der CMX-Monitor berechnet zyklisch die Pegelstände, Änderungsraten (Einheiten pro Sekunde) und die Verhältnisse verschiedener Zähler. Die Zyklen, in denen der CMX-Monitor die Werte berechnet und ausgibt, können Sie beim Aufruf von *cmxm* bestimmen. Die Ausgabe erfolgt in Form einer Tabelle auf die Standardausgabe.

Semigraphische Ausgabe von Statistiken

Der CMX-Monitor ermittelt Statistikwerte wie bei der tabellarischen Ausgabe. Die Anzahl der ausgegebenen Größen ist eingeschränkt auf die Statistik für den Datenaustausch zwischen den verschiedenen Komponenten. Die Statistiken werden in Form von Diagrammen auf der Standardausgabe ausgegeben. Die Bedeutung der ausgegebenen Werte können Sie der Beschreibung der tabellarischen Ausgabe im nächsten Abschnitt entnehmen.

Summarische Ausgabe der Zählerstände

Hierbei gibt der CMX-Monitor einmal auf *stdout* die Daten für die einzelnen Komponenten aus, die entweder seit dem Start des Betriebssystems aufgelaufen sind oder nach dem letzten Reset der Statistiken mit der Option *-z*. Die ausgegebenen Werte können anhand der folgenden Beschreibung der tabellarischen Ausgabe interpretiert werden (siehe Abschnitt „Format der tabellarischen Ausgabe des CMX-Monitors“ auf Seite 270). Es ist zu beachten, dass die Werte absolut sind. Z. B. wird nicht wie bei der tabellarischen Ausgabe die mittlere Anzahl der gesendeten Daten pro Sekunde ausgegeben, sondern die Anzahl der seit Systemstart gesendeten Daten. Dabei ist zu berücksichtigen, dass die Zähler beim Überschreiten eines Schwellwertes rundgezählt werden.

Beispiel für eine summarische Ausgabe (Kommando cmxm -s):

```
CMX MONITOR (6.0): AUTOMAT STATISTICS Jan 14 14:38:58
CMX AUTOMAT G=6.0  8 CC BOOT Jan 13 19:47 TIME 14:38:58
       5 TSP
   15878 APPLICATIONS etablished
   16009 ATTACH invokes
   29194 TCEP set ups(4% refusals, 3% aborts)
39057077 ICMX(S3) commands (0% nomem 0% bad 0% busy)
 2277821 ICMX(S3) events (0% sync)
12357613 ICMX(CC) commands sent (0% blocked)
 9275589 ICMX(CC) commands got (0% blocked)
 9190014 blocks sent (33% with flow control)
 9218321 blocks got (33% with flow control)
       8.003240 Gbyte sent
       8.048487 Gbyte got
```

Bei allen drei Betriebsarten können Sie wählen, ob die vom CMX-Monitor ermittelten Werte alle Aktivitäten des CMX-Automaten erfassen sollen, oder nur die Aktivitäten, die sich auf einen bestimmten TSP Access Point beziehen.

Der CMX-Monitor verwendet zur Ermittlung der Statistikwerte das laufende System als Quelle. D. h. er liest die aktuellen Zählerstände ab, bereitet sie entsprechend der gewählten Betriebsart auf und gibt sie aus.

Alternativ zur Dialogausgabe können Sie auch mit Hilfe eines Hintergrundprozesses (Dämon) eine Statistikdatei erzeugen, die der CMX-Monitor dann aufbereitet. Der Dämon wird mit *cmxmd* gestartet und beendet. Er sammelt periodisch Statistikwerte in einer Datei. Diese Datei können Sie dann zu einem beliebigen Zeitpunkt nach der Beendigung des Dämonen durch den CMX-Monitor aufbereiten lassen. Die Ausgabe kann dann tabellarisch oder semigraphisch auf die Standardausgabe erfolgen.

Die Beendigung von *cmxm* erfolgt von der Tastatur durch DEL oder ENTER (und q bei semigraphischer Ausgabe) oder durch das Signal SIGINT. Bei Abbruch durch andere Mittel befindet sich das Terminal nach semigraphischer Ausgabe in einem undefinierten Zustand.

Das Kommando hat folgende Syntax:

cmxm [␣**-a**] [␣**-c**␣id] [␣**-f** [␣file]]
[␣**-l**␣{ ln | **all** }]
[␣**-i**␣sec] [␣**-b**␣hms] [␣**-e**␣hms]
[␣**-n**␣cnt] [␣**-s**] [␣**-v**] [␣**-z**]

Durch die Optionen *-f, -s* und *-v* wird die Art der Ausgabe der Statistikwerte und die von *cmxm* verwendete Quelle für die Statistikwerte festgelegt.

Beim Start von *cmxm* ohne Angabe von Optionen erfolgt eine zyklische Ausgabe der Statistik des CMX-Automaten in tabellarischer Form gemäß den Standardwerten der Optionen *-c␣id, -i␣sec, -n␣cnt*.

-a

Es wird die Statistik für den CMX-Automaten aufbereitet (Standardwert).

-c␣id

Es wird die Statistik für den TSP Access Points *id* aufbereitet, sofern an Ihrem Rechner vorhanden. Die einzelnen TSP Access Points werden in *id* wie folgt angegeben:

W[1-32]
für CC-WAN (X.21, V.24, V.35, ISDN).

T[1-6]
für TSP Access Point (siehe Abschnitt „Architektur der CCP-Profile“ auf Seite 23).

-f [␣file]

cmxm wertet die Statistik aus der Statistikdatei *file* aus, in der der *cmxm*-Dämon *cmxmd* Statistiken gesammelt hat.

Wenn *file* nicht angegeben ist, wird die Datei *cmxm*[*id*].*DD* angenommen. Dabei bezeichnet *id* entweder einen CC/TSP Access Point wie bei der Option *c* oder ist leer, und *DD* ist der Tag des Monats (beginnend bei 1).

-l␣ln | **all**

gibt Statistiken (Durchsatzraten) für alle (max. 4) Leitungen aus (*all*) oder für eine Leitung, die mit *ln* angegeben wurde. Der Controller wird mit *-c␣id* angegeben.

-i␣sec

sec bestimmt das Sekundenintervall für zyklische Ausgabe der Statistik. Für *sec* ist eine positive Dezimalzahl anzugeben. Die Ausgabe der Statistik erfolgt dann alle *sec* Sekunden auf *stdout*.

-i␣sec nicht angegeben: Es wird *sec=1* angenommen.

Wählen Sie im Fall von Leitungsstatistiken (Option *-l*) einen Wert ≥5.

-b␣hms

Für *hms* ist die Beginnzeit der Auswertung anzugeben. Die Angabe muss in der Form hh[:mm[:ss]] erfolgen (hh = Stunde, mm = Minute, ss = Sekunde). Es ist nur dann sinnvoll, *-b hms* anzugeben, wenn der Wert *f* angegeben wurde.

-b hms nicht angegeben:

Es wird *hms* = 00:00:00 angenommen.

-e␣hms

Für *hms* ist die Endezeit der Auswertung anzugeben. Die Angabe muss in der Form hh[:mm[:ss]] erfolgen (hh = Stunde, mm = Minute, ss = Sekunde). Es ist nur dann sinnvoll, *-e hms* anzugeben, wenn für *option* der Wert *f* angeben wurde.

-e hms nicht angegeben: Es wird *hms* = 24:00:00 angenommen.

-n␣cnt

cnt bestimmt die Anzahl Zyklen für zyklische Ausgabe. Für *cnt* ist eine positive Dezimalzahl anzugeben.

-n␣cnt nicht angegeben oder negativ: Es wird *cnt=unendlich* angenommen.

-s

Die Ausgabe der Statistik erfolgt summarisch.

-v

Die Option -*v* bewirkt eine visualisierte Darstellung der aufbereiteten Ergebnisse auf *stdout* in semigraphischer Form (Verbindung mit Terminal erforderlich). Zusammen mit -*c␣all* bewirkt -*v* eine semigraphische Darstellung von Statistiken des CMX-Automaten und der verfügbaren CCs.

-z

Die Option -*z* setzt alle Statistikdaten für den CMX-Automaten und die CCs/TSPs zurück auf null.

In den folgenden Abschnitten ist das Format der tabellarischen und summarischen Ausgabe des CMX-Monitors beschrieben.

Format der tabellarischen Ausgabe des CMX-Monitors

Die vom CMX-Monitor ermittelten Werte sind abhängig davon, ob Sie eine Statistik des CMX-Automaten oder eine Statistik für einen CC/TSP Access Point angefordert haben. Im Folgenden werden beide Statistiken getrennt beschrieben.

Statistik des CMX-Automaten

Die tabellarische Aufbereitung der Statistik erfolgt zeilenweise. Drei Kopfzeilen leiten die Ausgabe ein. Nach je 20 Zeilen erfolgt eine erneute Ausgabe der 3 Kopfzeilen.

```
 CMX MONITOR (6.0): AUTOMAT STATISTICS
 CMX AUTOMAT G-6.0 8CC BOOT Jan 1 08:09  TIME 13:29:40 IVAL 1
PG 1
TSAP   ATT     TEP      ICMX(S3)      ICMX(CC)     TCEP    DATA
SEND  DATA GET
 act   act   act bu  cmd n e  evt wt  snd  get   act rj ab bls
kbys blg  kbyg
  3    11    11  0   462 0 0  39  0   168  148    64  0  0 118
243.6 147 309.6
  3    11    11  0    9  0 0   3  0    1    3     64  0  0  0
 0.0   3   6.0
   ...
```

Die ersten Zeilen enthalten:

- die Version des CMX-Automaten
- die Anzahl der vom CMX-Automaten unterstützten CCs/TSP Access Points
- den Zeitpunkt des Systemstarts (BOOT)
- die aktuelle Zeit (TIME)
- das Zeitintervall (IVAL) der Statistikaufnahme
- eine Seitenangabe PG. PG entspricht der Anzahl der wiederholten Ausgabe der Kopfzeile.

Die vom CMX-Monitor ermittelten Werte sind zu Gruppen zusammengefasst. Die nächste Zeile enthält die Bezeichnungen der Gruppen, die Zeile darunter die zugehörigen Statistikgrößen. Die Einteilung in Gruppen erfolgt nach (internen) Schnittstellen. In der folgenden Beschreibung der Statistik sind jeweils die Gruppe, ihre Bedeutung und darunter die zugehörigen Statistikgrößen und deren Bedeutung aufgelistet.

TSAP
: TS-Anwendungen (TS-Applications).

  act
  : Derzeit aktive TS-Anwendungen (ICMX(L)- und XTI-Anwendungen).

ATT
: Anmeldungen (Attach-Aufrufe).

  act
  : Derzeit aktive Anmeldungen von Prozessen innerhalb dieser TS-Anwendungen.

TEP
: Transportendpunkte (Transport End Points).

  act
  : Derzeit aktive Transportendpunkte (TEPs). TEPs sind die XTI-Transportendpunkte und die in ICMX-Anwendungen angemeldeten Prozesse bzw. Threads.

  bu
  : Prozentsatz der Kollisionsfälle bei den Eröffnungen der Gerätedateien.

ICMX(S3)
: Die Statistik bezieht sich auf die Vorgänge an der Systemschnittstelle zwischen Benutzerprozess (TS-Anwendung, ICMX(L)- und XTI-Bibliothek) und dem CMX-Automaten im Betriebssystemkern.

cmd
: mittlere Anzahl der von den TS-Anwendungen abgesetzten Aufrufe an den CMX-Automaten pro Sekunde.

n
: Prozentsatz der Aufrufe in *cmd*, die wegen eines Betriebsmittelengpasses vorübergehend abgelehnt werden mussten.

e
: Prozentsatz der Aufrufe, die wegen eines Fehlers abgelehnt wurden.

evt
: mittlere Anzahl der eingetretenen Ereignisse (events) pro Sekunde.

wt
: Prozentsatz der Ereignisse, die von den TS-Anwendungen synchron erwartet wurden.

ICMX(CC)
: Diese Statistik bezieht sich auf die Vorgänge an der Schnittstelle zwischen CMX-Automat und den Treibern im Kern, die den Zugang zu den TSPs steuern (WAN-Adapter, TPI-Adapter). Die folgenden Werte beziehen sich auf alle bedienten TSPs.

  snd
  : Anforderungen vom CMX-Automaten an die TSPs, Daten zu senden. Es wird die mittlere Anzahl der Anforderungen pro Sekunde angegeben.

  get
  : Anforderungen der TSPs an den CMX-Automaten, empfangene Daten abzuholen. Es wird die mittlere Anzahl der Anforderungen pro Sekunde angegeben.

TCEP
: Diese Statistik bezieht sich auf die Aktivitäten der über den CMX-Automaten aufgebauten Transportverbindungen.

  act
  : derzeit aktive Transportverbindungen.

rj
: Prozentsatz der Verbindungen, die abgelehnt wurden. Der Wert enthält sowohl die von lokalen TS-Anwendungen abgelehnten Verbindungen als auch die abgelehnten Verbindungsanforderungen lokaler TS-Anwendungen durch deren Kommunikationspartner.

ab
: Prozentsatz der Transportverbindungen, die vom System zwangsweise abgebaut wurden.

DATA SEND
: Diese Statistik bezieht sich auf die gesendeten Daten, summiert über alle TSP Access Points.

bls
: mittlere Anzahl der gesendeten Blöcke pro Sekunde. Ein Block entspricht einer TIDU.

fcs
: Zahl der gesendeten Datenblöcke dividiert durch Zahl der erhaltenen Sendekredite in Prozent.

kbys
: mittlere Anzahl der pro Sekunde gesendeten Daten in Kbyte.

DATA GET
: Diese Statistik bezieht sich auf die empfangenen Daten, summiert über alle TSP Access Points.

blg
: mittlere Anzahl empfangener Blöcke pro Sekunde. Ein Datenblock entspricht einer TIDU.

fcg
: Zahl der empfangenen Datenblöcke dividiert durch Zahl der gegebenen Empfangskredite in Prozent.

kbyg
: mittlere Anzahl der pro Sekunde empfangenen Daten in Kbyte.

Statistik eines CC/TSP Access Points

Die tabellarische Aufbereitung der Statistik eines CC/TSP Access Points erfolgt zeilenweise. Drei Kopfzeilen leiten die Ausgabe ein. Nach je 20 Zeilen erfolgt eine erneute Ausgabe der 3 Kopfzeilen.

```
CMX MONITOR (6.0): CC STATISTICS                        Jan 1
13:42:43
CC ADAPTER  V2510  CC T6 RDY  BOOT Jan  1 08:09    TIME 13:42:43
CCP VERSION 0x00a0 ADDRFORMS:  LOOPSBKA TRSNASBKA
     0 TSAP set ups
     0 TCEP set ups (0% refusals, 0% aborts)
      0 ICMX(S3) events (0% sync)
      0 ICMX(CC) commands sent (0% blocked)
      0 ICMX(CC) commands got (0% blocked)
      0 blocks sent (0% with flow control)
      0 blocks got (0% with flow control)
     0 TSP starts
     0 TSP interrupts
     0.000000 Gbyte sent
     0.000000 Gbyte got
     0 ADM&DIAG starts
     0 ADM&DIAG interrupts
     0 ADM&DIAG Kbyte sent (0 bytes per block)
     0 ADM&DIAG Kbyte got (0 bytes per block)
```

Die ersten Zeilen enthalten:

– die Version des Treibers, der den Zugang zum CC/TSP Access Point steuert

– die symbolische Bezeichnung des CC/TSP Access Point und der aktuelle Status des CC/TSP Access Point

  Die symbolischen Bezeichnungen für die CCs/TSP Access Points haben die folgende Bedeutung:

  W[1-32]
  : für CC-WAN (X.21, V.24, V.35, ISDN).

  T[1-6]
  : für TSP Access Point (siehe Abschnitt „Architektur der CCP-Profile“ auf Seite 23).

  Für den Status des CC/TSP Access Point sind folgende Angaben möglich:

  ATT
  : der CC/TSP Access Point ist bei CMX angemeldet, aber nicht betriebsbereit.

RDY
: der CC/TSP Access Point ist bei CMX angemeldet und betriebsbereit.

– den Zeitpunkt, zu dem der CC/TSP Access Point zuletzt gestartet wurde (BOOT)
– die aktuelle Zeit (TIME) oder die Zeit, zu der der CC/TSP außer Betrieb genommen wurde (DOWN)
– das Zeitintervall der Statistikaufnahme (IVAL)
– eine Seitenangabe PG. PG entspricht der Anzahl der wiederholten Ausgabe der Kopfzeile.

Die vom CMX-Monitor ermittelten Werte sind zu Gruppen zusammengefasst. Die nächste Zeile enthält die Bezeichnungen der Gruppen, die Zeile darunter die zugehörigen Statistikgrößen. Die Einteilung in Gruppen erfolgt nach internen Schnittstellen. In der folgenden Beschreibung der Statistik sind jeweils die Gruppe, ihre Bedeutung und darunter die zugehörigen Statistikgrößen und deren Bedeutung aufgelistet.

ICCP
: Diese Statistik bezieht sich auf die Vorgänge an der Schnittstelle zwischen dem Gerätetreiber des CC im Betriebssystem (CC-Adapter) und dem CC. Nicht enthalten sind Administrationsvorgänge.

  cmd
  : mittlere Anzahl Aufrufe zur Kommunikation an den CC pro Sekunde.

  int
  : mittlere Anzahl Interrupts vom CC pro Sekunde.

ICMX(S3)
: Diese Statistik bezieht sich auf die Aktivitäten an der Schnittstelle zwischen Benutzerprozess und Betriebssystemkern bezogen auf diesen CC/TSP Access Point.

  evt
  : mittlere Anzahl der eingetreten Ereignisse (events) pro Sekunde.

  wt
  : Prozentsatz synchron erwarteter Ereignisse.

ICMX(CC)

Diese Statistik bezieht sich auf die Vorgänge an der Schnittstelle zwischen CMX-Automat und den Treibern im Kern, die den Zugang zu TSPs steuern (WAN-Adapter, TPI-Adapter).

snd

Anforderungen von CMX an den TSP Access Point, Daten zu senden. Es wird die mittlere Anzahl der Anforderungen pro Sekunde angegeben.

get

Anforderungen des TSP Access Point an CMX, empfangene Daten abzuholen. Es wird die mittlere Anzahl der Anforderungen pro Sekunde angegeben.

cw

Anzahl der Anforderungen in *snd* und *get*, die sich in einem Wartezustand befinden.

pw

Prozentsatz aller Sende- und Abholanforderungen, die einen Wartezustand durchliefen.

TSAP

Aktivitäten der TSAPs. TSAPs sind die Dienstzugriffspunkte, an die sich die TS-Anwendungen binden, und über die sie auf die Dienste der Transportsysteme (TSPs) zugreifen.

act

derzeit am TSP Access Point aktive TSAPs.

TCEP, DATA SEND, DATA GET

Die zu diesen Gruppen gehörenden Statistikgrößen haben die gleiche Bedeutung wie die entsprechenden Größen bei der Statistik des CMX-Automaten. Der ausgegebene Wert bezieht sich jedoch nur auf den einen TSP Access Point.

Leitungsstatistik

Die Option *-l␣ln|all* liefert Leitungsstatistiken für einen angegebenen Controller. Diese Statistiken werden zyklisch vom Controller zum Hostrechner transferiert. Da die anschließende Formatierung durch *cmxm* etwas Zeit erfordert, ist die Einstellung eines Zeitintervalls von 15 Sekunden für zeilenweises Editieren (*cmxm -i␣sec␣*15) sinnvoll.

Das folgende Beispiel zeigt die Ausgabe eines Kommandos *cmxm -cW2 -l␣all -i␣20*. Es liefert Leitungsstatistiken für alle Leitungen des Controllers W2 im Intervall von 20 Sekunden.

```
CMX MONITOR (6.0): LINE STATISTICS                          Oct 15 13:50:57
CC ADAPTER  V2510 CC W1/RDY  BOOT Oct 15 13:19 TIME 13:50:57  IVAL 20 PG1
   INTERFACE_1   |    INTERFACE_2   |   INTERFACE_3    |   INTERFACE_4
 SEND  GET  FCS- |  SEND  GET  FCS- |  SEND  GET  FCS- |  SEND  GET  FCS-
kBy/s kBy/s errs | kBy/s kBy/s errs | kBy/s kBy/s errs | kBy/s kBy/s errs
  0.0   0.0    0 |   -     -      - |     -     -    - |  0.0   0.0   0.0
  6.8   6.9    0 |   -     -      - |     -     -    - |  0.0   0.0   0.0
  6.6   6.7    0 |   -     -      - |     -     -    - |  0.0   0.0   0.0
```

Das Kommando cmxm -cW2 -l1 liefert Statistiken für Leitung 1:

```
CMX MONITOR (6.0): LINE STATISTICS                          Oct 15 13:54:44
INTERFACE (1) of CC W1/RDY  BOOT Oct 15 13:19 TIME13:54:44 IVAL 15 PG1
   DATA-SEND      DATA_GET       HDLC       HDLC       Parity-/
blk/s kBy/s     blk/s kBy/s    abrt/ovr   lng-errs     FCS-errs
    0   0.0         0   0.0           0          0            0
    9   9.4         8   9.5           0          0            0
    6   6.2         5   6.2           0          0            0
```

Die Bedeutung der zwei Kopfzeilen ist beim vorigen Ausgabe-Beispiel beschrieben. Die nächsten beiden Zeilen sind Überschriften für Komponenten, nach denen die Statistikwerte gruppiert sind.

SEND/GET kby/s
: Durchschnittlich über diese Leitung übertragene/empfangene Daten in Kilobyte pro Sekunde.

Parity-/FCS-errs
: Anzahl der FCS- oder Parity-Fehler pro Ausgabe-Intervall über diese Leitung.

DATA-SEND/DATA-GET
: Datentransfer-Statistik für diese Leitung.

  blk/s
  : Pro Sekunde durchschnittlich gesendete/empfangene Anzahl von Datenblöcken.

  kBy/s
  : Durchschnittlich gesendete/empfangene Daten in Kilobyte pro Sekunde.

HDLC

Zyklische Statistik des HDLC-Protokolls.

abrt/ovr

Anzahl der erhaltenen HDLC-Aborts/Overruns.

lng-errs

Anzahl der HDLC-Längen-Fehler (zu lang/zu kurz).

Beachten Sie, dass die Ausgabe in späteren Versionen in leicht verändertem Format erfolgen kann.

Beispiele

cmxm -v -c W1 -i 10 -n 20

Dieses Kommando erzeugt als Ausgabe die Aktivitäten des Controllers W1 in semigrafischem Format, alle 10 Sekunden, insgesamt zwanzig Mal.

cmxm -f -b 8:00 -e 16:30 -i 300 /var/opt/SMAWcmx/tmp/cmxm.21

Dieses Kommando erzeugt folgende Ausgabe: aus der Datei, die tägliche Statistiken für den 21. des Monats enthält (erzeugt mit *cmxmd*), werden in Tabellenform die Aktivitäten zwischen 8 und 16.30 Uhr (5-Minuten-Intervalle) ausgegeben.

cmxm -s -c T5

Dieses Kommando erzeugt als Ausgabe alle Aktivitäten des TSPs T5 bis zum aktuellen Zeitpunkt in summarischer Form.

Dateien

cmxm[*id*].*DD*

Statistik des Monatstages *DD* mit *id* gemäß *c*-Option oder leer. Wo sich die Datei im Dateisystem Ihres Rechners befindet, entnehmen Sie bitte der Freigabemitteilung.

Siehe auch

cmxinfo, cmxmd.

10.8 CMX-Monitordämon (cmxmd)

Der CMX-Monitordämon *cmxmd* sammelt im Hintergrund zyklisch Statistikdaten über die in CMX laufenden Aktivitäten und protokolliert diese zur späteren Tagesauswertung mit *cmxm* in einer Datei. Die Statistik umfasst die Aktivität an der Schnittstelle sowie die Statistik von ICMX- und XTI-Anwendungen, deren Verbindungen und deren Datendurchsatz. Sie kann global oder spezifisch für einen CC/TSP Access Point abgerufen werden.

Bei Aufruf ohne Argumente sammelt *cmxmd* periodisch (alle 10 Sekunden) bis zum Ende des Tages (24:00:00 h) die Statistik des CMX-Automaten aus dem laufenden System in die Datei *cmxm*[*id*].*DD* (*DD* ist der Tag des Monats, beginnend bei 1). Durch Angabe von Optionen und Argumenten kann der Monitordämon *cmxmd* detaillierter gesteuert werden. Die Beendigung von *cmxmd* erfolgt entweder von selbst am Tagesende, oder durch ein Signal (vorzugsweise SIGINT), oder durch Aufruf mit der Option *h*.

cmxmd schreibt seine Prozess-Identifikation in die Datei *cmxmd*[*id*].*pid* und selbsterklärende Angaben zum Verlauf in die Tracedatei *cmxmd*[*id*].*trc*.

Den Namen des Dateiverzeichnisses, in dem Sie den CMX-Monitordämon an Ihrem System starten können, entnehmen Sie bitte der Freigabemitteilung.

Das Kommando hat folgende Syntax:

cmxmd [␣**-h**] [␣**-c**␣id] [␣**-i**␣sec] [␣**-n**␣cnt] [␣**-o**␣file]

-h

Ein zuvor mit *cmxmd* gestarteter *cmxmd*-Dämon wird bei Angabe der Option *-h* beendet. Wurde beim Start des *cmxmd*-Dämonen die Option *-c* angegeben, so muss die Option *-c* in der gleichen Form auch beim Beenden angegeben werden.

-c␣id

id gibt den CC/TSP Access Point an, für den *cmxmd* Statistikwerte sammeln soll. Ist der angegebene CC/TSP Access Point an Ihrem Rechner nicht vorhanden, so weist der CMX-Automat das Kommando zurück. Eine entsprechende Fehlermeldung wird auf *stderr* ausgegeben.

Die einzelnen CCs/TSP Access Points werden in *id* wie folgt angegeben:

W[1-32]
: für CC-WAN (X.21, V.24, V.35, ISDN).

T[1-6]
: für TSP Access Point (siehe Abschnitt „Architektur der CCP-Profile“ auf Seite 23).

Wird die Option *c* beim Start von *cmxmd* angegeben, so muss sie in der gleichen Form auch beim Beenden dieses *cmxmd*-Dämonen (Option *h*) angegeben werden.

-i␣sec
: *sec* gibt die Länge des Sekundenintervalls für das periodische Sammeln der Statistik an. Für *sec* ist eine positive Dezimalzahl anzugeben.

-i␣sec nicht angegeben: Es wird *sec=10* angenommen.

-n␣cnt
: Für *cnt* ist anzugeben, wie oft *cmxmd* Statistiken aufnehmen soll. Es ist eine positive Dezimalzahl anzugeben.

-n cnt nicht angegeben: *cmxmd* sammelt die Statistiken bis zum Ende des laufenden Tages (24:00:00h).

-o␣file
: Die Statistikwerte sollen in die Datei *file* geschrieben werden. Für das Dateiverzeichnis, in dem *file* angelegt werden soll, muss Schreibberechtigung für alle Benutzer existieren.

Wenn *-o file* nicht angegeben ist, schreibt *cmxmd* die Statistik in die Datei *cmxm*[id].DD. Dabei ist *id* der bei Option *-c* angegebene Wert des CC/TSP Access Point oder leer, *DD* ist der Tag des Monats (beginnend bei 1). Bei jedem Aufruf wird diese Datei neu geschrieben.

Dateien

cmxmd[*id*].*pid*
: Prozess-ID des laufenden CMX-Dämonen.

/opt/SMAW/SMAWcmx/lib/cmx/cmxmd[*id*].*trc*
: Trace-Datei für den CMX-Dämonen.

/var/opt/SMAWcmx/tmp/cmxm[*id*].*DD*
: Statistik des Monatstages *DD* mit *id* gemäß *c*-Option oder leer.

10.9 Installierte Kommunikationsprodukte abfragen (cmxprod)

Mit Hilfe des Kommandos *cmxprod* können Sie abfragen, welche Kommunikationsprodukte auf Ihrem System installiert sind, und ob diese vollständig installiert sind. Das Kommando hat folgende Syntax:

cmxprod [␣**-a**] [␣**-n**␣ boot-env |␣**-r**␣rootdir] [␣product ...]

-a

Ausgabe ohne Kopfzeile und mit einem Buchstaben P, p oder c in der ersten Spalte. Nach P folgt die Produktbezeichnung, nach p steht die Package-Bezeichnung. Nach c stehen zusätzliche Parameter eines Package.

-n␣boot-env

Ausgabe der in der Boot-Umgebung *boot-env* installierten Produkte.

-r␣rootdir

Ausgabe der im Root-Verzeichnis *rootdir* installierten Produkte.

product

Gibt den Produktnamen an. Folgende Angaben sind möglich (das Präfix *CCP-* ist optional):

```
CMX
CCP-OSI/NEA
CCP-ISDN-LINK
CCP-WAN-LINK
CS-GATE
```

Es können ein oder mehrere Produktnamen angegeben werden. Wird kein Produktname angegeben, zeigt *cmxprod* die Informationen zu allen im lokalen System installierten Produkten an.

Für das Kommando *cmxprod* gilt ein Produkt dann als installiert, wenn das Package, das dem Produkt seinen Namen gibt, installiert ist.

Beispiel

```
[PGTR0046:root] cmxprod
CMX/CCP 6.0 Products and Packages:
CMX Communications Manager for UNIX systems
        SMAWcmx 6.0A0004 Apr 24 2003 06:00
        SMAWcxagt 6.0A0004 Apr 24 2003 06:04
        SMAWxti 6.0A0004 Apr 24 2003 06:03
        SMAWntp 6.0A0004 Apr 24 2003 06:04
        SMAWr6 6.0A0004 Apr 24 2003 06:04
        SMAWcsr 6.0A0004 Apr 24 2003 06:03
        SMAWwca is not installed.
        SMAWPbase 1.004 Apr 04 2003 16:31
        SMAWPglib 1.2.1002 Feb 04 2003 10:17
        SMAWPgtk+ 1.2.1002 Feb 04 2003 10:19
        SMAWPethe 0.9.1103 Apr 04 2003 16:33
CCP-OSI/NEA STREAMS-based NEA (NEATE/NEAN) and ISO (TP02) Protocols
        SMAWnea 6.0A0004 Apr 24 2003 06:04
        SMAWtp02 6.0A0004 Apr 24 2003 06:04
CCP-WAN-LINK Network Access to X.21, V.24, X.25 WANs
        SMAWwan 6.0A0004 Apr 17 2003 06:03
CCP-ISDN-LINK Network Access to ISDN/S2m and ISDN/S0
        SMAWisdn 6.0A0004 Apr 17 2003 06:03
CS-GATE STREAMS-based Transport Gateway TGW
        SMAWtgw 6.0A0004 Apr 24 2003 06:05
[PGTR0046:root]
```

Fehler

Auftretende Fehler werden in die Standard-Fehlerausgabe protokolliert.

Siehe auch

cmxinfo.

10.10 TSP-spezifische Statusinformation (cmxstat)

Das Kommando *cmxstat* liefert Zustandsinformation der verschiedenen Transport-Service-Provider in einem einheitlichen Format. Neben den angemeldeten TSAPs können die bei den TSPs angemeldeten Subnetzanschlüsse und die bestehenden Verbindungen aufgelistet werden.

Für das Kommando gibt es fünf verschiedene Varianten.

Syntax

cmxstat **-t** ts-prov **-a**

cmxstat **-t** ts-prov **-s**

cmxstat **-t** ts-prov **-c**{ **n** | **t**}

cmxstat **-t** ts-prov **-e**

cmxstat [**-t** ts-prov] [**-o** own-appl [**-d**]] [**-p** part-appl]

-t ts-prov
: Name des TSPs, mögliche Angaben ts-prov:
```
nea
ntp
tp02
rfc1006
```

-a
: (attached applications) Informationen über lokal gebundene TSAPs ausgeben.

-s
: (subnet) Informationen über Subnetz-Anschlüsse ausgeben.

-c{ **n** | **t**}
: (connection) Informationen über Verbindungen ausgeben. Mit dem Argument *n* erhält man die aktuellen Subnetzverbindungen und mit dem Argument *t* die aktuellen Transportverbindungen

-e
: (error) Fehlerstatistik ausgeben.

-o␣own-appl[␣**-d**]

(own) Informationen zur lokalen Anwendung *own-appl* ausgeben. Falls die lokale Anwendung im TNS konfiguriert ist, wird ihr GLOBALER NAME angegeben, andernfalls muss für *own-appl* der T-Selektor zusammen mit der Zusatzoption *-d* angegeben werden.

-p␣part-appl

(partner) Informationen zur Partnernwendung *part-appl* ausgeben. Für *part-appl* ist der im TNS konfigurierte GLOBALE NAME anzugeben.

Angemeldete TSAPs ausgeben (Option -a)

Mit dieser Variante werden optionsabhängig die aktuell an einem TSP gebundenen TSAPs aufgelistet. Zu jeder bei einem TSP lokal angemeldeten Anwendung (*t_attach* mit T_PASSIVE) werden neben der lokalen TSAP-Bezeichnung die Summe der entgegengenommenen Verbindungsaufbauwünsche sowie die Anzahl der davon angenommen und der davon abgelehnten ausgegeben. Diese Statistik erfasst nur die Verbindungswünsche, die von einer Partneranwendung zu dieser Anwendung eingehen. Die von der Anwendung aktiv aufgebauten Verbindungen werden hierunter nicht erfasst.

Das folgende Beispiel listet die am NEA-Transportprovider eingerichteteten TSAPs auf:

```
#cmxstat -t nea -a

NEA TP: Locally attached TSAPs
T-Selector      NSAP        ConnInd    ConAcc    ConRej
A:nealokal      70/255       20          19        1
```

Aktive Subnetzanschlüsse ausgeben (Option -s)

Diese Kommandovariante listet für WAN-TSPs die aktiven Subnetzanschlüsse auf. Beim RFC1006-TSP werden die Socket-Adressen aufgelistet, über die ankommende Rufe entgegengenommen werden.

Die WAN-TSPs geben neben der lokalen Adresse des Subnetzanschlusses die aus dem Subnetz kommenden passiven Verbindungsanforderungen, die Verbindungsannahmen und die -ablehnungen aus.

Aus Sicht des Null-Transport Providers sind z. B. die folgenden Subnetzanschlüsse gebunden:

```
#cmxstat -t ntp -s
Null TP:        Locally attached SNPAs
Subnet-Id      Addr               CC  Ln    CurrCon ConAcc    ConRej
X25-1 (0x11)   123                W2  L1          0        0        0
```

```
X25-2 (0x12)   321                W2  L33         0        0        0
X25-1 (0x11)   1                  W3  L1          0        0        0
X25-2 (0x12)   2                  W3  L2          0        0        0
```

Das Transportsystem OSI über TCP/IP führt die TCP-Sockets auf, an denen der RFC1006-TSP ankommende Verbindungswünsche entgegennimmt:

```
#cmxstat -t rfc1006 -s
OSI TP over TCP/IP:     Listening TCP sockets
Port IP address
102 ::
```

Standardmäßig belegt der RFC1006-TSP den TCP-Port 102 für alle IP-Interfaces unabhängig davon, ob eine Anwendung angemeldet ist oder nicht.

Die Anzeige der Subnetzanschlüsse sagt nichts über den aktuellen Zustand der Leitungen aus. Präzise Informationen zu den Leitungen erhalten Sie mit dem Kommando *bstv linkstat -bWx*.

Aktuelle Verbindungen ausgeben (Option -c)

Diese Kommandovariante listet die aktuellen Transportverbindungen (*-c t*) eines Transportsystems auf. Bei multiplexenden Transportsystemen (OSI-TP Cl.2 und NEA) werden mehrere Transportverbindungen über eine Netzverbindung betrieben; für sie können die bestehenden Subnetz-Netzverbindungen (*-c n*) separat aufgelistet werden.

Das folgende Beispiel gibt die Transportverbindungen des NEA-TSPs aus. In der NEA-Architektur setzt die Transportschicht auf eine verbindungslose Netzschicht auf. Der Bezug zur Subnetzverbindung, über die die Pakete geroutet werden, kann nicht geschaffen werden.

```
#cmxstat -t nea -c t
NEA TP:         Current connections
Tref  Ini TPI-State Lref  Rref  Ncon  Local TSel   LocAddr     RemoteTSel     RemAddr
16     1  DATA      1     2     -1      A:nearemot   70/255     A:nealokal     71/255
19     1  DATA      2     3     -1      A:nearemot   70/255     A:nealokal     71/255
```

Fehlerstatistik ausgeben (Option -e)

Mit der Option *-e* wird die TSP-spezifische Fehlerliste ausgegeben. Diese Liste enthält die von dem Transport Service Provider registrierten Fehlersituationen. Hierzu zählen gesendete/ empfangene Schnittstellensignale, Protokollfehler und Engpass-Situationen bei der Anforderung von Systemleistungen wie z.B. Kernelspeicher.

Die Fehlerstatistik für einen NEA-TSP kann z.B. so aussehen:

```
cmxstat -t nea -e
Collecting statistic data starts at: Thu Apr 15 06:07:27
2004
              M_ERROR sent:    0
          M_ERROR received:    2
  Interface error received:    0
      Interface error sent:    0
           Protocol errors:    0
 Memory allocation failure:    0
              Mblk failure:    0
           No free streams:    0
```

TSP-Ressourcen einer CMX-Anwendung ausgeben (Optionen -o, -p)

Mit dieser Variante werden die Ressourcen des oder der Transportsysteme ausgegeben, die zu einem LOKALEN NAMEN oder zu einer Partneradessse gehören.

Die Option *-o* gibt für die angegebene lokale CMX-Anwendung sämtliche in den Transportsysemen gebundenen Ressourcen aus. Dazu gehören die der Anwendung zugeordneten TSAPs und die in diesen TSAPs endenden Verbindungen. Falls die lokale Anwendung nicht im TNS konfiguriert ist, müssen Sie den T-Selektor zusammen mit der Option *-d* eingeben. Der T-Selektor ist in folgendem Format anzugeben:
typ:abdruckbare Darstellung, z.B. *A:nonx2902*.

Die Option *-p* gibt sämtliche Transportverbindungen aus, die zu der angegebenen Partneranwendung gehören. Wenn die Optionen *-o* und *-p* gleichzeitig angegeben werden, dann werden alle Verbindungen zwischen der lokalen Anwendung und der Partneranwendung aufgelistet.

10.11 Traces für Transportsystem (cmxtrc)

Das Kommando *cmxtrc* schaltet Traces in einem Transportsystem ein und aus. Die Protokollschichten, auf denen Traces gezogen werden können, und die aufzuzeichnenden Ereignisse sind abhängig von dem Transportsystem und den Kommandoeingabe-Optionen, insbesondere den GLOBALEN NAMEN der Anwendungen. Wenn z.B. die durch den GLOBALEN NAMEN einer Partneranwendung bestimmte Route genau einem WAN-Interface (Controller und Leitung) zugeordnet ist, so wird neben dem Trace im Transportprovider (NEA, NTP oder TP02) zusätzlich der Leitungstrace auf dem Controller eingeschaltet. Dieser Leitungstrace kann anschließend mit dem Protokollanalyser *ethereal* aufbereitet werden.

Das Kommando *cmxtrc* schaltet die Traces in den Transportsystemen unter Zuhilfenahme der Basismechanismen des Kommandos *comtr* ein; bei WAN-Transportsystemen zusätzlich mithilfe des leitungsspezifischen Kontrollertraces. Die Trace-Dateien werden im aktuellen Verzeichnis unter den Standardnamen von *comtr* abgelegt, siehe Seite 295f, Option *-f*.

Bitte beachten Sie Folgendes:

– Zu jedem Zeitpunkt kann pro Transportsystem maximal ein Transportsystemfilter aktiv sein.

– Es können gleichzeitig Tracefilter in verschiedenen TSPs eingerichtet sein. Zu jedem Zeitpunkt darf nur maximal ein Leitungstrace aktiv sein.

– *cmxtrc* schaltet einen zuvor mit dem Kommando *comtr* eingeschalteten Trace aus und sichert den aktuellen Inhalt des Tracepuffers. Entsprechendes gilt für leitungsspezifische Traces.

Das Kommando besitzt 3 Formate: Einschalten, Informieren, Ausschalten.

Syntax

cmxtrc [␣**-c** |␣**-d**] [␣**act** |␣**pas**] -o own-appl [␣**-h**␣hostname] [␣**-t**␣ts-prov] [␣**-n**␣num]

cmxtrc [␣**-c** |␣**-d**] [␣**act** |␣**pas**] -p part-appl [␣**-o**␣own-appl] [␣**-h**␣hostname] [␣**-n**␣num]

cmxtrc [␣**-c** |␣**-d**] [␣**act** |␣**pas**] ␣**-h**␣hostname [␣**-t** rfc1006] [␣**-n**␣num]

cmxtrc [-␣**-c** |␣**-d**] [␣**act** |␣**pas**] ␣**-t**␣ts-prov ␣**-n**␣num

cmxtrc␣**-i** [␣**-t**␣ts-prov]

cmxtrc␣-r [␣**-t**␣ts-prov]

Im Folgenden werden die einzelnen Varianten näher erläutert.

Trace einschalten (Optionen -c, -d)

-c [␣**act** |␣**pas**]
(connection) Verbindungsaufbau protokollieren, die Datentransferphase wird nicht aufgezeichnet, Standardeinstellung. Mit den optionale Argumenten *act* oder *pas* können Sie nach aktivem und passivem Verbindungsaufbau differenzieren, standardmäßig wird beides aufgezeichnet.

-d [␣**act** |␣**pas**]
(data) Verbindungsaufbau und Datentransferphase protokollieren. Es werden die Datenprimitive und deren Länge, nicht aber deren Inhalt aufgezeichnet.

-o␣own-appl
(own) Protokolliert die Aktionen der lokalen CMX-Anwendung mit dem GLOBALEN NAMEN *own-appl*. Betrifft der zugeordnete LOKALE NAME mehrere TSPs, dann müssen Sie über die Option *-t* das Transportsystem angeben. Existiert der TSAP bei Kommandoeingabe, dann erfüllen sämtliche neu eingerichteten Verbindungsendpunkte dieses TSAPs die Bedingungen des Tracefilters.

-h␣hostname
(host) Diese Option ist nur für RFC1006 relevant. Es werden nur die Ereignisse protokolliert, die dem mit *hostname* angegebenen lokalen IP-Interface zugeordnet sind.

-t ␣ts-prov
(TSP) Bestimmt den TSP. Diese Angabe ist nur dann erforderlich, wenn sich der TSP nicht bereits aus den anderen Angaben ergibt. Mögliche Angaben: nea, ntp, tp02, rfc1006

-p␣part-appl
(partner) Protokolliert die Aktionen zum Partner mit dem GLOBALEN NAMEN *part-appl*. Wenn in einem WAN-TSP die Verbindung über einen eindeutig konfigurierten Subnetzanschluss aufgebaut wird, dann wird gleichzeitig der entsprechende leitungsspezifische Trace eingeschaltet.

-n␣num
(number) Gibt die Anzahl der Verbindungen an, die aufgezeichnet werden sollen. Nachdem eine Anzahl von *num* Verbindungen protokolliert worden ist, wird der Tracefilter ausgeschaltet.

-v

(verbose) Steuert die Form der Traceaufbereitung. Mit der Option *-v* werden die Traceeinträge zusätzlich direkt nach stdout geschrieben. Standardmäßig werden die Trace-Ereignisse in einem Tracepuffer gespeichert und nach dem Ausschalten des Traces aufbereitet.

Einsatzhinweise

– Ein Tracefilter gilt nur für Verbindungen, die **nach** dem Einrichten des Filters aufgebaut werden. Der Filter gilt deshalb nicht für Verbindungen zwischen Kommunikationspartnern, die zwar die Tracefilterbedingungen erfüllen, jedoch schon vor dem Absetzen des Kommandos *cmxtrc* bestanden.

– Die Optionen *-o* und *-p* definieren TSAP-spezifische Tracefilter. Sie eignen sich daher zur Analyse von Kommunikationsproblemen einzelner CMX-Anwendungen.

– Traceunterlagen für reproduzierbare Probleme können auch durch die logische Spezifikation der zu tracenden Verbindungen angegeben werden, z.B. in Form der nächsten Verbindungen, die aufgebaut werden.

Informieren über eingeschaltete Traces (Option -i)

Die Option *-i* informiert darüber, ob ein Tracefilter in einem Transportsystem eingeschaltet ist. Zusätzlich werden die Filterattribute angezeigt.

Beispiel

Durch *cmxtrc -t ntp -n 2* werden die beiden als nächstes aufgebauten Verbindungen protokolliert:

```
# cmxtrc -n 2 -t ntp
clearing ntp trace buffer
comtr daemon writes into the following 2 files alternatively:
/comtr.bin_ntp.01 and /comtr.bin_ntp.02
```

Die Eingabe *cmxtrc -i -t ntp* gibt anschließend folgende Informationen aus:

```
# cmxtrc -i -t ntp
        Status des NTP TSP:     READY.
TSP ntp: Trace-Filter active
         Number of connection setups:    2
         Filter attribute: TRC_EV_CONN (connection setups without data
transfer)
```

Trace ausschalten (Option -r)

Diese Option setzt die Trace-Selektionskriterien für die angegebenen Transportsysteme zurück. Die Traceeinträge werden vom Tracepuffer ausgelesen und im aktuellen Verzeichnis unter den Standardnamen von *comtr* abgelegt, siehe Seite 295f, Option *-f*.

10.12 Grenzwerte für den CMX-Automaten ändern (cmxtune)

Mit dem Kommando *cmxtune* ändern Sie die Grenzwerte des CMX-Automaten für die maximale Anzahl von Transport End Points (TEPs), Transport Service Access Points (TSAPs), Transport Connection End Points (TCEPs) sowie Anmeldungen von Prozessen. *cmxtune* überschreibt die voreingestellten Werte in der Datei *CMXlimits* mit den neuen Grenzwerten. Die neuen Werte werden erst beim Reboot des Systems wirksam.

Das Kommando hat folgende Syntax:

cmxtune [**att** num] [**tep** num] [**tsap** num] [**tcep** num]

num
: Grenzwert für *att*, *tep*, *tsap* oder *tcep*.

  Wertebereich: 1024 bis 65535 (theoretischer Grenzwert).

att
: Maximale Anzahl der Anmeldungen über ICMX oder XTI festlegen.

tep
: Maximale Anzahl der Transportendpunkte (TEPs) festlegen.

tsap
: Maximale Anzahl der Transport Service Access Points (TSAPs) festlegen.

tcep
: Maximale Anzahl der Transport Connection End Points (TCEPs) festlegen.

Dateien

/opt/SMAW/SMAWcmx/lib/cmx/CMXinit
: Init-Skript für CMX.

/opt/SMAW/SMAWcmx/lib/cmx/CMXlimits
: Datei mit Grenzwerten für den CMX-Automaten.

10.13 Traces für CMX-Treiber (comtr)

Das *comtr*-Kommando bietet für alle CMX-Komponenten eine einheitliche Trace-Schnittstelle. Sie können Traces für folgende Komponenten erstellen und aufbereiten:

– für den CMX-Automaten,
– für den TPI-Adapter,
– für die Transport Service Provider NEA, RFC1006, TP0/2 und NTP,
– für den Forwarding Support Service,
– für die Communication Services CS-ROUTE,
– für den verbindungslosen WAN-Zugang,
– für den Routing-Scheduler,
– für das PPP-Interface,
– für die STREAMS- und HSI-basierten Teile des WAN-Adapters,
– für das Transport Gateway.

comtr ermöglicht eine lückenlose Aufzeichnung auch bei hohem Trace-Aufkommen. Das Ausgabeformat ist für alle CMX-Komponenten einheitlich, die Kommandosyntax ist identisch.

Traces können sowohl aus einer globalen als auch aus einer komponenten-spezifischen Liste gelesen und ausgewertet werden. Ein Trace kann jederzeit gestartet oder gestoppt werden, sein Inhalt jederzeit ausgegeben werden.

Ein globaler Trace ist nach dem Systemstart immer aktiv und schreibt in die Error-Liste. Mit dem Kommando *comtr␣-m␣glob␣-t* werten Sie diese Error-Liste aus.

Syntax

comtr kann nur vom Systemverwalter oder CMX-Administrator aufgerufen werden. Die Kommandosyntax ist abhängig von der gewünschten Funktionalität:

– Informationen abfragen
– Trace starten/überwachen
– Trace beenden und aufbereiten
– Traces aufbereiten

Für einige Funktionen wird die Angabe einer CMX-Komponente erwartet. Geben Sie dazu den Parameter *-m␣module-id* an.

-m␣module-id
Modul-Identifikation. Für *module-id* sind folgende Werte möglich:

glob Globale Error-Liste

cxauto
CMX-Automat

tpia TPI-Adapter

nea TSP NEA

rfc1006
TSP RFC1006

tp02 TSP TP0/2

ntp TSP NTP

fss Forwarding Support Service

ads Access Data Service

clw Connectionless WAN access

ppx Schnittstelle des Point-to-Point-Protokolls

rs Routing Scheduler

cdsx FSS-Interface (Anfragen vom CC)

cws STREAMS-basierter Teil des CC-WAN-Adapters

cwp PCI-basierter Teil des CC-WAN-Adapters

tgw Transport Gateway

Informationen abfragen

comtr ␣{ **-a** | **-h** }

-a

Sie erhalten für alle Komponenten folgende Informationen:

- Modulnamen
- Kurzbezeichnung
- Gesetztes Trace-Level
- Füllgrad des Trace-Puffers oder Überschreibkennzeichen *w*, falls überschrieben
- Größe der Dateien, die wechselweise von einem Dämon-Prozess beschrieben werden, und deren voller Pfadname

Am Ende der Ausgabe werden maximal zwei Transportreferenzen (TREF) und maximal zwei Prozess-IDs angezeigt, sofern mindestens eines dieser selektiven Trace-Kriterien gesetzt wurde.

Beispiel

```
COMTR -> INFORMATION ABOUT TRACE STATUS
Module          Id     Level  % full  File sizes (1/2)- file
                                                        names
------          --     -----  ------  ------------------------
Global          glob    unsp.     31
CMX-Automat     cxauto   1        00
CMX module      cdsx     1        26
CMX module      cws      1        24/w
CMX module      rs       1        07
FSS-Service     fss      1        01
TPI-Adapter     tpia     1        00
NEA TSP         nea      1        00
NTP TSP         ntp      1        00
RFC1006 TSP     rfc1006  1        00
TP02 TSP        tp02     1        01
CS-ROUTE module ads      1        00
CS-ROUTE module clw      1        03
CS-ROUTE module ppx      1        00
CC-WAN-Adapter cwp       1         00
```

-h

Die Hilfeoption kann CMX-global oder komponentenspezifisch aufgerufen werden. In der CMX-globalen Variante werden die Standardwerte des Trace-Levels und die Größe des kernelinternen Trace-Puffers ausgegeben, die beim Einschalten eines Traces der Komponente gesetzt werden, falls keine abweichenden Werte angegeben werden.

Beispiel für die globale Variante

```
COMTR -> INFORMATION ABOUT SELECTABLE TRACE LEVELS AND SIZE OF
TRACE BUFFER
Module           Id      default  buffer size
------           --      -------  ------------
CMX-Automat      cxauto    4       320 Kbytes
CMX module       cdsx      4        28 Kbytes
CMX module       cws       3        66 Kbytes
CMX module       rs        4        40 Kbytes
FSS-Service      fss       2        23 Kbytes
TPI-Adapter      tpia      3       144 Kbytes
NEA TSP          nea       3       304 Kbytes
NTP TSP          ntp       2       248 Kbytes
RFC1006 TSP      rfc1006   2       108 Kbytes
TP02 TSP         tp02      3       206 Kbytes
CS-ROUTE module  ads       2        72 Kbytes
CS-ROUTE module  clw       3       206 Kbytes
CS-ROUTE module  ppx       1       112 Kbytes
CC-WAN-Adapter   cwp       4        48 Kbytes
```

In der komponentenspezifischen Variante gibt die Hilfeoption die für die Komponente einstellbaren Trace-Level an. Einige Komponenten bieten DEBUG-Level an, die aber detaillierte Kenntnisse über die internen Abläufe in der Komponente voraussetzen.

Beispiel für die komponentespezifische Variante

```
# comtr -m ntp -h

level    meaning
-----    -------
  1      error level (always switched on)
  2      Connection primitives without any user data
  3      Connection primitives and user data in connection
         primitives, if present
  4 *    Connection primitives and data primitives
  5      Connection-, data primitives and user data
         (up to 100 bytes)
  6      Connection, data primitives and user data. The length and
         offset of the user data may differ from the default
         values(offs 0;len 100)
  8      Connection primitives for passive connection setup

* Default, if trace switched on and no level with the -sl flag
specified
```

Trace starten und überwachen

Die hier beschriebenen Optionen dienen dem Starten von Traces und der Überwachung des Trace-Mechanismus. Folgende Kommandos sind zulässig:

comtr␣**-m**␣module-id **-c**

comtr␣**-m**␣module-id␣**-sl**[␣level] [␣**-v**] [␣**-K**␣wrap] [␣**-f**␣file] [␣**-u**␣size]
[␣**-p**␣pid1[␣pid2]] [␣**-T**␣tref1[␣tref2]] [␣**-x**␣type]

comtr␣**-u**␣size

comtr␣**-p**␣pid1[␣pid2]

comtr␣**-T**␣tref1[␣tref2]

comtr␣**-r**

comtr␣**-x**␣type

-c

löscht den Inhalt des Trace-Puffers einer Komponente.

-sl␣level␣[**-v**]

startet den Trace, setzt den Trace-Level für die spezifizierte Komponente und weist einen Trace-Puffer zu. Optional geben Sie mit *level* ein Trace-Level an, andernfalls wird das vorgegebene Default-Level eingestellt (siehe *comtr -h*). Bei Angabe der Zusatzoption *-v* werden die anfallenden Trace-Daten direkt nach *stdout* geschrieben.

-K␣wrap

Die Dezimalzahl *wrap* gibt die maximale Größe der binären Trace-Datei in KByte an. Wird *wrap* = 0 angegeben, so gibt es keine Beschränkung für die Größe der binären Datei.

-K␣wrap nicht angegeben: Es wird *wrap=1024* eingestellt.

-f␣file

Anstelle der Default-Namen werden wechselweise *file.01* und *file.02* beschrieben. Ohne Angabe werden die Dateinamen *comtr.bin_modid.01* und *comtr.bin_modid.02* vergeben.

-u␣size

setzt eine veränderte Größe des Trace-Puffers für eine Komponente fest. *size* gibt die Größe des Trace-Puffers in KByte an.

Maximale Größe: 8096 KByte.

-p␣pid1␣pid2

beschränkt die Menge der Trace-Einträge auf 2 Prozess-IDs.

-T␣tref1␣tref2

gibt eine Transportreferenz an, für die der Trace erzeugt werden soll.

-r

löscht gesetzte, selektive Trace-Kriterien, d.h. Prozess-IDs und Transportreferenzen.

-x␣type

Beendet das Füllen eines modul-spezifischen Trace-Puffers, sobald ein bestimmter Eintrag *type* in den Puffer geschreiben wird.

Trace beenden und aufbereiten

Mit den folgenden Optionen schalten Sie einen Trace-Mechanismus aus, der unter der Regie eines Hintergrundprozesses gestartet wurde. Der Hintergrundprozess wird beendet.

comtr␣**-m**␣module-id␣**-t**[␣**-f**␣file]

comtr ␣ **-m** ␣ **glob** ␣ **-t** [␣ **-f** ␣ file][␣ **-b** ␣ binary-file]

-t

stoppt den Trace und setzt das Trace-Level für die spezifizierte Komponente auf das Error-Level zurück. Die Trace-Auswertung wird in die Datei *comtr.module-id.ascii* geschrieben.

-f ␣ file

schreibt die Trace-Auswertung in die Datei *file*.

-b ␣ binary-file

Nur bei Angabe von *glob*: Der globale Trace wird zusätzlich in eine Binär-Datei ausgegeben.

Traces aufbereiten

comtr ␣ **-e** ␣ **-f** ␣ *file*

bereitet Trace-Information zur Ausgabe im ASCII-Format auf, die in einer Binär-Datei *file* enthalten ist. Das Ergebnis wird auf *stdout* ausgegeben.

Beispiele

comtr -e -f comtr.bin_tpia.01 > tpia.read

Bereitet die Trace-Datei auf, die in *comtr.bin_tpia.01* enthalten ist, und schreibt das Ergebnis in die Datei *tpia.read*.

comtr -m nea -sl 5

startet einen Trace des NEA-TSP und setzt das Level auf 5. Die Trace-Information wird wechselweise in die Dateien *comtr.bin_nea.01* und *comtr.bin_nea.02* geschrieben.

comtr -p 1812 1814

beschränkt das Füllen des Trace-Puffers auf Einträge für die Prozess-IDs 1812 und 1814.

10.14 Protokoll-Traces mit ethereal

Mit CMX V6.0 wird der als Freeware bereitgestellte Software-Protokoll-Analyser *ethereal* ausgeliefert, der den Nachrichtenverkehr an einem LAN-Interface mitprotokollieren und die Protokollelemente symbolisch aufbereiten kann. Die Aufbereitung schließt auch das Konvergenzprotokoll RFC1006 mit ein, so dass der Nachrichtenverkehr von CMX-Anwendungen, die über diesen TSP mit einer Partneranwendung kommunizieren, aufgezeichnet und dargestellt werden kann. *ethereal* bietet eine Vielzahl von Filterungs- und Darstellungsmöglichkeiten für den Nachrichtenverkehr. Diese sind detailliert in der Dokumentation dieses Protokollanalysers beschrieben, auf die Sie über die Homepage von *ethereal http://www.ethereal.com* zugreifen können.

10.15 NEABX-Bibliotheks-Trace steuern und aufbereiten (neal)

Der Trace der NEABX-Bibliothek wird über die Umgebungsvariable NEATRACE aktiviert und gesteuert. Die Trace-Einträge eines Prozesses werden kompakt und binär in einem dynamisch angelegten Puffer gesammelt und periodisch in temporäre Dateien geschrieben. Die Aufbereitung dieser Dateien erfolgt entkoppelt durch *neal*.

Steuerung des Trace - NEATRACE

Jeder CMX-Aufruf *t_attach* eines Prozesses wertet die Umgebungsvariable NEATRACE aus und aktiviert gegebenenfalls den Trace. NEATRACE muss vor dem Starten der Anwendung, d. h. vor dem ersten *t_attach* des zu überwachenden Prozesses, gesetzt worden sein. Nach dem Aktivieren des Trace wird die temporäre Datei *NEALpid* mit der Prozess-Identifikation *pid* eröffnet, falls sie nicht bereits eröffnet ist. Für die Dateien werden die Zugriffsrechte 0600 vergeben. Anschließend wird dynamisch Speicher für die Pufferung der Trace-Einträge belegt.

Speicher und Dateien bleiben für die Lebensdauer des Prozesses belegt.

Die in NEATRACE angegebenen Optionen steuern den Trace. Die Optionen *s*, *S* bestimmen den Umfang der Protokollierung. Die Optionen *p,d, r* steuern Pufferung, Datenlänge und Rundschreiben der Dateien.

Die Variable NEATRACE wird in folgender Syntax angegeben:

NEATRACE= ␣{ **-s** | **S** } [␣**-p**␣fac] [␣**-d**␣length␣**-r**␣wrap] [␣**-f**␣file]

export␣NEATRACE

-s und *-S* bestimmen die Art des Trace. Sie dürfen nur einen der beiden Werte angeben. Um den Trace zu aktivieren, müssen Sie einen Wert angeben.

-s

Gewöhnliche Protokollierung: Es werden alle Aufrufe und deren Argumente protokolliert. Optionen und Benutzerdaten werden nicht protokolliert.

-S

Ausführliche Protokollierung: Es werden alle Aufrufe, deren Argumente, der Inhalt der Optionen und die Benutzerdaten protokolliert.

-p␣fac

Durch die Dezimalziffer *fac* wird der Faktor der Pufferung bestimmt. Die Pufferung erfolgt im Betrag *fac* * 1024.

Wird *fac* = 0 angegeben, so wird jeder Trace-Eintrag sofort (ungepuffert) in die Datei geschrieben.

Wertebereich für *fac*=0...8.
-p␣*fac* nicht angegeben: Es wird *fac=0* angenommen.

-d␣length

Die Dezimalzahl *length* gibt in Byte an, bis zu welcher Länge TIDUs bei Angabe der Steueroption *-S* protokolliert werden. Der Standardwert ist 16.

length=0...16 ...65535.

Bei Angabe 0 erfolgt keine Protokollierung, sonst bis zu der angegebenen Maximallänge.

-r␣wrap

Durch die Dezimalzahl *wrap* wird angegeben, dass nach *wrap* * 1024 in die zweite temporäre Datei *NEAMapid* protokolliert werden soll.

Die zweite Datei *NEAMapid* behandelt der Trace genauso wie *NEALapid*.

Nach jeweils *wrap* * 1024 Byte schaltet der Trace zwischen *NEALapid* und *NEAMapid* um. Dabei geht der alte Inhalt der jeweiligen Datei verloren.

-r wrap nicht angegeben: Es wird *wrap=256* angenommen.

-f␣file

Diese Option dient zur Angabe eines Dateiverzeichnisses *file*, in dem die Trace-Dateien *NEA[LM]apid* hinterlegt werden sollen. Dabei kann das Argument sowohl ein relativer als auch ein absoluter Pfadname sein.

Aufbereitung des Trace - neal

neal liest die vom Trace erzeugten Einträge aus den Dateien, die Sie für *file* angegeben haben. Die Einträge verarbeitet *neal* entsprechend der angegebenen Optionen und gibt das Ergebnis auf *stdout* aus. Das Kommando hat folgende Syntax:

neal [␣**-c**] [␣**-d**] [␣**-e**] [␣**-v**] [␣**-x**] [␣**-D**] [␣**-p**]␣file ...

Die gewählten Optionen geben an, welche Trace-Einträge aufbereitet werden sollen. Es ist möglich, mehrere der im Folgenden beschriebenen Werte pro Aufruf von *neal* anzugeben. Lediglich die Optionen *-v* und *-x* schließen einander aus. Ist keine Option angegeben, so wird *-cdex* angenommen.

-c

Die Aufbereitung erfolgt für die Funktionsaufrufe:

- zur An- und Abmeldung der TS-Anwendung bei NEABX
- zum Verbindungsaufbau und -abbau

-d

Die Aufbereitung erfolgt für die Funktionsaufrufe:

- zum Datenaustausch
- zur Fluss-Regelung

-e

Die Aufbereitung erfolgt für die Funktionsaufrufe zur Ereignisbehandlung.

-v

Es erfolgt eine ausführliche Aufbereitung der Funktionsaufrufe, ihrer Argumente, der Optionen, der Benutzerdaten. Der Umfang der Aufbereitung der Daten ist abhängig davon, ob bei NEATRACE die Option *-s* oder die Option *-S* angegeben wurde.

-x

Es erfolgt eine eingeschränkte Aufbereitung der Funktionsaufrufe und ihrer Argumente *ohne* Optionen und Benutzerdaten.

-D

Wurde bei der Trace-Steuerung die Option *D* angegeben, dann kann bei *neal* die interne Debugging-Information ausgewertet werden.

-p

Wird die Option *p* gesetzt, so müssen Sie für den Parameter *file* die Prozess-Identifikation (pid) des Prozesses einer TS-Anwendung angeben, für den die Trace-Einträge aufbereitet werden sollen.

file ...

Anzugeben ist der Name einer oder mehrerer Dateien, die die aufzubereitende Trace-Informationen enthalten. Falls u. a. die Option *p* angegeben wurde, ist für file nur die Prozess-Identifikation des Prozesses anzugeben, für den Trace-Einträge aufbereitet werden sollen. *neal* sucht dann im Dateiverzeichnis */var/opt/SMAWcmx/tmp* nach allen zu diesem Prozess gehörenden Trace-Dateien.

Ausgabeformate

Das Format der durch *neal* aufbereiteten Trace-Information ist im Abschnitt „CMX-Bibliotheks-Trace steuern und aufbereiten (cmxl)“ auf Seite 255 beschrieben.

Dateien

NEALapid, *NEAMapid*

Dateien mit kompakten Trace-Einträgen in Binärformat.

Falls bei NEATRACE nicht anders angegeben werden die Dateien für den NEABX-Bibliothekstrace *NEALapid* und *NEAMapid* im Directory */var/opt/SMAWcmx/tmp* abgelegt.

Siehe auch

cmxl

10.16 Starten und Stoppen von CMX und TSPs (StartStop)

Unter dem Namen StartStop sind mehrere Kommandos zusammengefasst, die Sie verwenden können, um die Komponenten von CMX (Transport Service Provider NTP und RFC1006, FSS) sowie die installierten TSPs (NEA, TP0/2) zu starten, zu stoppen bzw. ihren Start bei der System-Inbetriebnahme zu steuern.

Nach der Installation der Kommunikations-Software können die (neu) installierten Komponenten mit diesen Kommandos manuell in Betrieb genommen werden.

Die Kommandos schließen alle Aufgaben ein, die nacheinander beim Starten bzw. Stoppen der Komponenten anstehen. Außerdem überprüfen sie Voraussetzungen, aktuelle Status und Abhängigkeiten der anstehenden Operationen und sorgen damit auf effektive Weise für die Konsistenz des Systems. Sie protokollieren Ablauf und Ergebnis der Operationen für spätere Diagnose in Protokolldateien.

Die Ausgabe der Kommandos erfolgt auf Standardausgabe bzw. Standardfehlerausgabe.

Diese StartStop-Skripte sind sehr hilfreich für die Handhabung der CMX-Komponenten und der TSPs. Sie brauchen für ihre Verwendung Systemverwalterberechtigung. Sie können sie von der Kommandoebene mit den folgenden Kommandos oder über das CMX-Menü aufrufen:

Syntax

cmx [␣{ **autostart** | **start** | **restart** | **stop** | **boot** | **shutdown** | **autostop** | **diag** }]

cmxsnmp [␣{ **autostart** | **start** | **restart** | **stop** | **autostop** | **diag** }]

ntp [␣{ **autostart** | **start** | **restart** | **stop** | **boot** | **shutdown** | **autostop** | **diag** }]

nea [␣{ **autostart** | **start** | **restart** | **stop** | **boot** | **shutdown** | **autostop** | **diag** }]

tp02 [␣{ **autostart** | **start** | **restart** | **stop** | **boot** | **shutdown** | **autostop** | **diag** }]

rfc1006 [␣{ **autostart** | **start** | **restart** | **stop** | **boot** | **shutdown** | **autostop** | **diag** }]

csr [␣{ **autostart** | **start** | **stop** | **boot** | **shutdown** | **autostop** }]

autostart

fügt Start-Routinen in die System-Start-Dateien ein. Diese Routinen werden dann bei jedem Systemstart durchlaufen und starten die entsprechende Komponente.

start

startet die Komponente nach verschiedenen Überprüfungen. Es wird eine Meldung auf Standardausgabe ausgegeben, wenn dieses Modul bereits gestartet ist. Das Ergebnis dieser Operation entspricht dem von *autostart* beim Hochfahren des Systems.

start veranlasst außerdem durch Einträge in die *crontab*-Datei, dass einige Aktionen regelmäßig ausgeführt werden. Diese überprüfen, ob die Komponente noch arbeitet, und starten sie erneut, wenn dies nicht der Fall ist. Diese Aktionen werden solange ausgeführt, bis die Komponente explizit durch *stop* beendet wird oder bis der Neustart dreimal nacheinander nicht gelingt. In beiden Fällen wird die Ausführung der Aktionen eingestellt und eine entsprechende Meldung in die Protokolldatei geschrieben.

restart

startet die Komponente erneut. Diese Operation hat dieselbe Wirkung wie die Ausführung von *stop* und *start* direkt nacheinander.

stop

stoppt die Komponente. Wird eine Komponente deaktiviert, so gibt sie ihre reservierten System-Ressourcen frei. Regelmäßige Aktionen, die aus der *crontab*-Datei des Systems aufgerufen werden, werden ebenfalls beendet. Dieser Status bleibt bis zum nächsten Start oder Autostart erhalten.

boot

nimmt nach der Software-Installation die neu installierten bzw. aktualisierten Komponenten wieder in Betrieb.

Die Kommandoeingabe *cmx boot* startet CMX und alle auf CMX aufsetzenden Produktkomponenten. Nach der Installation/Aktualisierung einer auf CMX aufsetzenden Produktkomponente, z.B. dem TP0/2-TSP, wird diese Komponente durch die Eingabe von *tp02 boot* in Betrieb genommen, ohne die bereits in Betrieb befindlichen anderen Transportsysteme zu beeinträchtigen.

shutdown

nimmt die angegebene Komponente bzw. im Fall von *cmx* die gesamte CMX/CCP-Software außer Betrieb und schafft die Voraussetzungen für eine Aktualisierung der installierten Software.

cmx shutdown wird nur dann ausgeführt, wenn bei CMX keine Anwendungen mehr angemeldet sind.

autostop

entfernt die Komponenten-spezifischen Routinen, die durch *autostart* in die System-Start-Dateien eingefügt wurden, aus diesen Dateien. Damit wird die Komponente nicht mehr automatisch beim Systemstart in Betrieb genommen.

diag

gibt Protokolldateien für die entsprechende Komponente aus.

Dateien

Bei den folgenden Dateinamen ist *$Name* jeweils durch einen der folgenden Komponentennamen zu ersetzen: CMX, FSS, NEA, NTP, TP02, RFC1006, cmxsnmp.

/var/opt/SMAWcmx/adm/log/$Name.log

Protokolldatei der entsprechenden Komponente.

/var/opt/SMAWcmx/adm/log/CCP.log

Protokolldatei für Restart eines CCP.

/etc/rc0.d/K[0-9] [0-9] $Name

Stop-Skript für entsprechende Komponente.

/etc/rc2.d/S[0-9] [0-9]-$Name

Start-Skript für entsprechende Komponente.

10.17 TS-Directory prüfen (tnsxchk)

Das Kommando *tnsxchk* sucht Fehler in dem angegebenen TS-Directory. *tnsxchk* untersucht zunächst die Zeiger und Indizes innerhalb der Dateien des zu überprüfenden TS-Directory *DIRnum* (*num* = 1...9) auf Plausibilität. Erkennt es hierbei keinen Fehler, so prüft *tnsxchk* das Format der Einträge innerhalb der Dateien. *tnsxchk* gibt an, ob eine der Dateien fehlerhaft ist oder nicht. Zur Diagnose liefert *tnsxchk* auf Anforderung detaillierte Informationen zu den gefundenen Fehlern. *tnsxchk* erkennt auch, wenn das TS-Directory nicht die vom TNS-Dämon erwartete Struktur hat. Das Kommando hat folgende Syntax:

tnsxchk [␣**-d**␣num] [␣**-v**] [␣**-f**]

-d␣num

num gibt die Nummer des zu überprüfenden TS-Directory *DIRnum* an.

Ohne Angabe wird *num* = 1 eingestellt.

-v

tnsxchk prüft, ob das TS-Directory *DIRnum* die Version hat, die der TNS-Dämon verwenden kann.

Ohne Angabe von *-v* überprüft *tnsxchk* alle Dateien des TS-Directory.

-f

tnsxchk liefert detaillierte Informationen zu den gefundenen Fehlern. Diese Informationen sind sehr komplex und werden nur für die Diagnose durch den Kundendienst benötigt.

Ausgabeformat

Die Ausgabe von *tnsxchk* wird durch eine Kopfzeile eingeleitet, die den Programmnamen, den Namen des überprüften TS-Directory, das Datum und den Zeitpunkt des Programmstarts enthält.

Danach wird der Inhalt der Datei VERSION des TS-Directory ausgegeben. Die in der Datei VERSION enthaltenen Werte sind betriebssystemabhängig. Ein Beispiel für die Ausgabe des Kommandos *tnsxchk -v* ist:

```
VERSION:
========
VERSION=V6.0 BYTEORDER=MSB LONG=32BIT INTEGER=32BIT SHORT=16BIT
```

Für jede Datei des TS-Directory, die *tnsxchk* überprüft, gibt *tnsxchk* folgende zwei Zeilen aus:

```
tnsxchk: datei wird überprüft:
tnsxchk: ergebnis
```

Für *datei* wird der absolute Pfadname der gerade geprüften Datei angegeben. Für *ergebnis* wird „OK" oder „Fehlerhaft" ausgegeben.

Dateien

DIR1 ... DIR9

TS-Directory, aufsteigend nummeriert, maximal 9.

Die TS-Directories *DIR1* bis *DIR9* befinden sich unter dem Verzeichnis */opt/SMAW/SMAWcmx/lib/cmx*. Sie enthalten folgende Dateien (i = 1,...,5):

NAMEP*i*
NPVALUES
PROPERTIES
PRVALUE
ROOT
FREENPV
FREENPRV
VERSION

LOG

Die Datei *LOG* für das pauschale Ändern von TNS-Einträgen wird im Directory */var/opt/SMAWcmx/tmp* abgelegt.

tnsxd.trc

Die Datei *tnsxd.trc* zur Protokollierung der TNS-Zugriffe befindet sich im Directory */opt/SMAW/SMAWcmx/lib/cmx*. Dort gibt es auch die Datei *tnsxd.pid*, in der die aktuelle Prozessnummer des aktiven *tnsxd*-Prozesses abgelegt ist.

Siehe auch

tnsxd

10.18 TS-Directory erstellen, aktualisieren, lesen (tnsxcom)

tnsxcom bearbeitet Einträge des Formates *tnsxfrm* (standard TNS entry format). Über den Parameter *modus* kann die Betriebsweise des TNSXCOM eingestellt werden. Folgende Betriebsweisen können angegeben werden:

UPDATE
: Aktualisieren des TS-Directory entsprechend der Einträge in der Datei *file* (den Dateien *file* ...).

INTERAKTIV
: Aktualisieren des TS-Directory entsprechend der Angaben von *stdin*.

LOAD
: Erstellen eines zuvor leeren TS-Directory aus den Einträgen in der Datei *file* (den Dateien *file* ...).

DUMP
: (Sortierte) Ausgabe des Inhalts eines TS-Directory im Format *tnsxfrm* in die Datei *file*.

CHECK
: Überprüfen der Syntax der Datei *file*.

CHECK_UPD
: Überprüfen der Syntax der Datei *file* und bei korrekter Syntax Aktualisieren des TS-Directory entsprechend der Einträge in *file*.

Bei Abwesenheit von Optionen compiliert *tnsxcom* die Einträge aus *file* im UPDATE-Modus (zeilenweise) in das TS-Directory.

tnsxcom mit der Betriebsweise UPDATE, INTERAKTIV, LOAD und CHECK_UPD (*modus* = *-u, -i, -l, -S*) kann nur vom Systemverwalter aufgerufen werden.

Sobald *tnsxcom* eine Datei *file* abgearbeitet hat, gibt der TNSXCOM folgende Werte auf *stderr* aus:

- die Anzahl der aufgetretenen Fehler und Warnungen,
- die während der Bearbeitung der Datei *file* abgelaufene Zeit (real),
- den Anteil von real, den der *tnsxcom*-Prozess verbraucht hat, in %
- die verbrauchte Zeit aufgeteilt in Benutzermodus (user) und Systemmodus (sys).

Wurden mehrere Dateien von *tnsxcom* bearbeitet (*file* ...), so gibt *tnsxcom* am Ende der Bearbeitung diese Werte für den gesamten Compilierlauf aus.

Syntax

tnsxcom [␣**-d**␣num] [␣modus] [␣**-p**␣prmt] [␣**-t**] [␣**-o**␣orig] [␣file ...]

-d␣num

Nummer des TS-Directory, das bearbeitet werden soll. Mögliche Angabe für *num*: 1,...,9

Ohne Angabe wird *num* = 1 eingestellt.

modus

Betriebsweise von *tnsxcom* festlegen.

Mögliche Angaben für *modus* sind *-D*, *-i*, *-l*, *-s*, *-S*, *-u*; die einzelnen Angaben schließen sich gegenseitig aus. Ohne Angabe wird *-u* eingestellt. Die Angaben für *modus* haben folgende Bedeutung:.

-D

DUMP-Modus

tnsxcom bereitet den Inhalt des TS-Directory in Einträgen im Format *tnsxfrm* in die Datei *file* auf, die wiederum als Eingabe verwendbar ist. Die Ausgabe erfolgt sortiert in aufsteigender ASCII-Sortierreihenfolge nach den GLOBALEN NAMEN gemäß ihrer Namenshierarchie und für jeden Namensteil. Es wird also zuerst nach Namensteil[1] sortiert. Existieren im TS-Directory mehrere TS-Anwendungen mit demselben Namensteil[1], so werden diese nach Namensteil[2] sortiert usw.

-i

INTERAKTIV-Modus

tnsxcom liest Einträge im Format *tnsxfrm* von *stdin*, nachdem er durch Ausgabe einer Promptzeichenfolge (siehe *-p prmt*) seine Eingabebereitschaft angezeigt hat, und mischt sie in das TS-Directory, indem er bisher im TS-Directory noch nicht vorhandene Einträge erfasst oder existierende Einträge aktualisiert.

-l

LOAD-Modus

tnsxcom nimmt die Einträge einzeln aus der Datei *file* und füllt das (bisher leere) TS-Directory mit den syntaktisch korrekten Einträgen.

-s

CHECK-Modus

tnsxcom wendet nur die Syntaxprüfung auf die Datei *file* an und protokolliert mögliche Syntaxfehler. Das TS-Directory wird nicht verändert.

-S

CHECK_UPD-Modus

Wie bei Option *-s* erfolgt in einem ersten Lauf zuerst die Syntaxprüfung auf die gesamte Datei *file*. Treten in *file* keine Syntaxfehler auf, so aktualisiert *tnsxcom* dann das TS-Directory in einem zweiten Lauf.

-u

UPDATE-Modus

tnsxcom nimmt die Einträge einzeln aus der editierbaren Datei *file* und mischt die syntaktisch richtigen Einträge in das TS-Directory durch Erfassung bisher nicht vorhandener oder Aktualisierung existierender Einträge.

Option *-u* ist der Standardwert von *modus*.

-p␣prmt

Durch *-p prmt* wird die Zeichenfolge *prmt* im interaktiven Modus *-i* als Prompt verwendet, bei anderen Modi wird *-p* ignoriert.

Keine Angabe von *-p prmt*: es wird *prmt* = * angenommen.

-t

Trace von *tnsxcom* einschalten.

Der Trace wird eingeschaltet und dessen Ausgabe in der Datei *tnsxcom.trc* im laufenden Dateiverzeichnis protokolliert. Bei Angabe von *modus* = *-D* wird die Option *-t* ignoriert.

-o␣orig

orig gibt den Ursprung vor. Der Ursprung ist eine Folge von Namensteilen. Der Inhalt des *name*-Feldes eines *tnsxfrm*-Eintrags wird um . (Punkt) und dem Wert von *orig* ergänzt, falls der Inhalt nicht mit . (Punkt) endet. Das Resultat muss ein syntaktisch korrekter GLOBALER NAME mit maximal fünf Namensteilen sein.

file ...

Name der Datei mit Einträgen im Format *tnsxfrm*, die im Fall *modus* = *-l*, *-s*, *-S* oder *-u* von *tnsxcom* ausgewertet werden soll. Es können mehrere Dateien angegeben werden. Im Fall *modus* = *-D* ist der Name der Datei anzugeben, in die *tnsxcom* den Inhalt des TS-Directory aufbereiten soll.

Fehler

Fehlermeldungen, die sich auf Syntax oder Semantik der Einträge in *file* beziehen, gibt *tnsxcom* zusammen mit dem Namen der Datei und der Zeilennummer auf *stderr* aus.

Dateien

tnsxcom.trc

Name der Tracedatei im Dateiverzeichnis, in dem *tnsxcom* aufgerufen wurde.

LOG

Die Datei *LOG* für das pauschale Ändern von TNS-Einträgen wird im Directory */var/opt/SMAWcmx/tmp* abgelegt.

tnsxd.trc

Die Datei *tnsxd.trc* zur Protokollierung der TNS-Zugriffe befindet sich im Directory */opt/SMAW/SMAWcmx/lib/cmx*. Dort gibt es auch die Datei *tnsxd.pid*, in der die aktuelle Prozessnummer des aktiven *tnsxd*-Prozesses abgelegt ist.

DIR1 ... DIR9

TS-Directory, aufsteigend nummeriert, maximal 9.

Die TS-Directories *DIR1* bis *DIR9* befinden sich unter dem Verzeichnis */opt/SMAW/SMAWcmx/lib/cmx*.

Siehe auch

tnsxd, *tnsxprop*, sowie Beschreibung des Eingabeformats für den TNSXCOM im Abschnitt „Syntax der TNS-Konfigurationsdatei“ auf Seite 77.

Beispiel

Der Aufruf *tnsxcom -d 2 -l -o np4..np2.np1 input1 input2* überführt die Einträge aus den Dateien *input1* und *input2* in das leere TS-Directory 2; ein *inhalt* jeden *name*-Feldes, der nicht mit ’.’ endet, soll dabei zu "*inhalt.np4..np2.np1*" (mit leerem Namensteil 3) erweitert werden.

10.19 TNS-Einträge löschen (tnsxdel)

Mit *tnsxdel* löschen Sie einzelne Einträge aus dem TS-Directory. Neben bestimmten GLOBALEN NAMEN können Sie auch ganze Hierarchien von Namen löschen, indem Sie die Namensteile, die unterschiedlich sind, bei der Eingabe des Kommandos leer lassen und nur die identischen Namensteile angeben.

tnsxdel [␣**-d**␣num] [␣**-a**] [␣**-v**] [␣**-f**␣file] [␣name...]

Die Optionen des Kommandos haben folgende Bedeutung.

-d␣num
: Definiert mit *num* die Nummer des gewünschten TS-Directories.

  Wertebereich: 1-9. Voreinstellung: 1

-a
: Mit dieser Option werden alle Einträge und das Directory selbst gelöscht.

-v
: Stellt den Kommandomodus auf „verbose“, d. h. es wird für jedes entfernte Objekt eine Zeile mit dem Objektnamen nach *stdout* geschrieben. In Kombination mit der Option *-a* wird lediglich eine Zeile mit dem Namen des entfernten TS-Directory erzeugt.

-f␣file
: Hier geben Sie eine Input-Datei *file* an. Sie enthält alle GLOBALEN NAMEN oder Namensteile, die gelöscht werden sollen.

name
: Hier geben Sie den zu löschenden GLOBALEN NAMEN ein.

Beispiele

Ein TS-Directory DIR5 enthält folgende Einträge (sie erhalten die Ausgabe mit tnsxpprop -d 5):

```
Gilbert.sales.dep1  \
     TA  LOOPSBKA A´Gilbert´
Meier.sales.dep1 \
     TA  LOOPSBKA A´Meier´
Ruth.sales.dep1 \
     TA  LOOPSBKA A´Ruth´
Schulz.purchase.dep2 \
     TA RFC1006 139.22.96.29 A´HUGO´
Huber.warehouse.dep2 \
```

```
     TA RFC1006 139.22.96.32 A´EGON´
Kruse.warehouse.dep2 \
     TA RFC1006 139.22.96.38 A´EMIL´
```

Ihre Eingabedatei *input*, die die zu löschenden Elemente enthält, könnte folgende Einträge enthalten.

Löschen von Gilbert (nicht in dep1):

```
Gilbert.sales.dep2
```

Löschen von Meier in der Datei:

```
Meier.sales.dep1
```

Löschen der gesamten Hierarchie incl. Huber und Krause:

```
warehouse.dep2
```

Der anschließende Aufruf von *tnsxcom -d 5 -v -f input one.more name* würde folgende Ausgabe erzeugen:

```
entry Gilbert.sales.dep2 not existing
entry Meier.sales.dep1 removed
entry Kruse.warehouse.dep2 removed
entry Huber.warehouse.dep2 removed
entry one.more name not existing
```

Ein folgender Aufruf von *tnsxcom -d 5* ergibt:

```
Gilbert.sales.dep1  \
   TA  LOOPSBKA A´Gilbert´
Ruth.sales.dep1 \
   TA  LOOPSBKA A´Ruth´
Schulz.purchase.dep2 \
   TA  RFC1006 139.22.96.29 A´HUGO´
```

Sie erreichen erheblich kürzere Laufzeiten beim Löschen von TNS-Einträgen, wenn Sie in umgekehrter Einfügereihenfolge löschen. Dazu sollten Sie beim Anlegen großer TS-Directories per *tnsxcom* die Einträge zunächst in sortierter Reihenfolge vornehmen. Vor dem Löschen per *tnsxdel* können Sie dann mit dem Kommando *sort -r* die Eingabedatei in umgekehrter Reihenfolge sortieren.

Beispiel

Die Datei *tnsxdel.input* enthält die zu löschenden GLOBALEN NAMEN. Die Einträge werden in umgekehrter Reihenfolge sortiert und anschließend aus dem Directory 1 gelöscht:

```
sort -r tnsxdel.input > tnsxdel.input.sort
tnsxdel -f tnsxdel.input.sort
```

Fehler

Fehlermeldungen sind selbsterklärend; Fehlercodes (Dezimalzahl) können mit dem Kommando *cmxdec* entschlüsselt werden. Wenn angegebene Objektnamen nicht im TS-Directory enthalten sind, so wird dies nicht als Fehler interpretiert. Dasselbe gilt bei der Angabe eines nicht vorhandenen TS-Directories.

Siehe auch

tnsxcom, *tnsxprop*, *tnsxt*, *cmxdec*.

10.20 Informationen zum TS-Directory anzeigen (tnsxinfo)

tnsxinfo gibt den Inhalt, die Grenzwerte und die aktuelle Belegung eines TS-Directory in lesbarer Form auf der Standardausgabe *stdout* aus. Die ausgegebenen Informationen erhält *tnsxinfo* durch direkten Zugriff auf das jeweilige TS-Directory, d. h. ohne Anforderung an den TNS-Dämon.

tnsxinfo liefert statistische Informationen über die Belegung der TS-Directories, bereitet die GLOBALEN NAMEN der TS-Anwendungen des TS-Directory in einer Baumstruktur auf und gibt die den TS-Anwendungen zugeordneten Eigenschaften aus.

Syntax

tnsxinfo [␣**-d**␣num] [␣**-g**] [␣**-p**] [␣**-v**]

-d␣num
: definiert *num* als die Nummer des zu verwendenden TS-Directory. Ohne Angabe wird *num* = 1 eingestellt.

-g
: *tnsxinfo* gibt den Inhalt der Datei VERSION im TS-Directory, die Grenzwerte für die Belegung und Informationen über die aktuelle Belegung des TS-Directory aus. Alle im TS-Directory vorhandenen GLOBALEN NAMEN werden in einer Baumstruktur aufbereitet ausgegeben.

-p
: *tnsxinfo* gibt den Inhalt der Datei VERSION im TS-Directory, die Grenzwerte für die Belegung und Informationen über die aktuelle Belegung des TS-Directory aus. Zusätzlich werden alle im TS-Directory enthaltenen Eigenschaften aufbereitet ausgegeben.

-v
: *tnsxinfo* gibt nur den Inhalt der Datei VERSION des TS-Directory aus.

Wird keine der Optionen *-g, -p, -v* angegeben, so gibt *tnsxinfo* den Inhalt der Datei VERSION und Tabellen mit den Grenzwerten des TS-Directory und Informationen über seine aktuelle Belegung auf *stdout* aus.

Ausgabeformat

Die Ausgabe von *tnsxinfo* beginnt immer mit zwei Kopfzeilen und dem Inhalt der Datei VERSION im TS-Directory.

Die Kopfzeilen enthalten:

– Name und Version des Programms
– aktuelles Datum und Zeit des Programmstarts
– Name des TS-Directory, auf das sich die Ausgabe bezieht.

Ausgabe der Grenzwerte und der aktuellen Belegung

Grenzwerte und aktuelle Belegung des TS-Directory werden in Form von drei Tabellen ausgegeben. Die in den Tabellen enthaltenen Informationen können unvollständig sein, wenn zur Zeit der Informationsabfrage der TNS-Dämon läuft (z. B. wenn Daten zur Zeit der Abfrage über den TNS-Dämon verändert werden oder der TNS-Dämon Daten zum schnelleren Zugriff in einem cache hält). Die erste Tabelle enthält die Angaben zu den einzelnen Namensteilen der GLOBALEN NAMEN. Die Tabelle ist wie folgt aufgebaut:

```
GRENZWERTE FUER NAMENSTEILE UND EIGENSCHAFTEN:
Namensteil TS_COUNTRY TS_ADMD TS_PRMD  TS_OU  TS_PN
Index             1       2       3      4      5
Laenge            2      16      16     10     30
Maximum         113     223    1093   4093  16381
Grenze          100     200    1000   4000  16000
belegt            1       2       2      2      5
```

Die erste Zeile der Tabelle enthält die symbolischen Bezeichnungen der Namensteile des GLOBALEN NAMENS (siehe Abschnitt „Syntax der TNS-Konfigurationsdatei“ auf Seite 77). Darunter steht der Index des entsprechenden Namensteils. Index 1 gehört zu Namensteil1 usw. Die Einträge in den weiteren Zeilen haben folgende Bedeutung:

Laenge
: Erlaubte Maximallänge der einzelnen Namensteilwerte in Byte

Maximum
: Größe der Hash-Tabellen in den einzelnen Dateien NAMEP*i* (i = 1,...,5).

Grenze
: Anzahl der Namensteile mit Index *i*, die Sie maximal in das TS-Directory eintragen können.

belegt
: Anzahl der Namensteile mit Index *i*, die derzeit im TS-Directory existieren.

Die zweite Tabelle enthält Grenzwerte und aktuelle Belegung der Eigenschaften, der Namensteil- und Eigenschaftswerte (dies entspricht der maximalen Länge und der aktuellen Länge der Dateien NPVALUES und PRVALUES).

```
              Eigenschaften    Namensteilwerte    Eigenschaftswerte
Maximum              320000             560700             21406500
belegt                   14                 54                  134
Luecken (max.)       -/-              0 (100)               0 (50)
```

Die Einträge haben folgende Bedeutung:

Maximum
: maximale Anzahl der Eigenschaften, die Sie im TS-Directory ablegen können bzw. maximale Länge der Dateien NPVALUES, die alle Namensteilwerte der GLOBALEN NAMEN enthält, und PRVALUES, die alle Werte der Eigenschaften enthält (siehe Abschnitt „Syntax der TNS-Konfigurationsdatei“ auf Seite 77). Die Längen werden in Byte angegeben.

belegt
: Anzahl der Eigenschaften, die im TS-Directory abgelegt sind bzw. aktuelle Länge der Dateien NPVALUES und PRVALUES in Byte.

Luecken (max.)
: Anzahl der Lücken in den Dateien PROPERTIES (sie enthält die Namen der Eigenschaften), NPVALUES und PRVALUES, die durch Löschen entstanden sind. Wenn die Anzahl der Lücken den in Klammern angegebenen Maximalwert übersteigt, wird das TS-Directory reorganisiert und die Lücken geschlossen.

Die dritte Tabelle enthält die erlaubte Maximallänge der verschiedenen Eigenschaftswerte in Byte.

```
properties:
type            Laenge(Byte) type             Laenge(Byte)
TS_EMPTYPROP               0   TS_TRANS               200
TS_NEABX                   1   TS_TRSYS                 1
TS_LNAME                 200   TS_USER1PR             200
TS_USER2PR               200   TS_USER3PR             200
```

Die Bezeichnungen für die Eigenschaften haben folgende Bedeutung:

```
TS_EMPTYPROP   Leereintrag,
TS_LNAME       LOKALER NAME,
TS_USER2PR     USER2,
TS_TRANS       TRANSPORTADRESSE,
TS_TRSYS       TRANSPORTSYSTEM,
TS_USER1PR     USER1,
TS_USER3PR     USER3.
```

Die Eigenschaften sind in Abschnitt „Konfigurieren mit tnsxcom“ auf Seite 75 beschrieben.

Ausgabe der GLOBALEN NAMEN

Die im TS-Directory abgelegten GLOBALEN NAMEN werden in Form eines Namensbaums ausgegeben (Beispiel für einen Namensbaum siehe Abschnitt „GLOBALER NAME“ auf Seite 77). Hier wird nur ein Teil der im Beispiel von *tnsxinfo* gelieferten Informationen abgedruckt.

```
GLOBALE NAMEN:
     ----i-[Npi] [x] NonLeaf, Name part i, Value Npi, Index x
    ====i=[Npi] [x] [p] Leaf, Name part i, Value Npi, Index x,
Propindex p

     ROOT
     |----1-[co] [53]
     |       |----2-[admd1] [82]
     |       |       |----3-[prmd1] [461]
     |       |       |       |----4-[ou1] [873]
     |       |       |       |       |====5=[pn4] [875] [10]

       :
       :
```

Jede einfach gestrichelte Linie `---i-` führt zu einem Knoten im Namensbaum. Jede doppelt gestrichelte Linie `===i=` führt zu einem Blatt im Namensbaum. Jedem Blatt ist eindeutig eine TS-Anwendung zugeordnet. Der Weg im Namensbaum von der Wurzel ROOT bis zu dem Blatt entspricht dem GLOBALEN NAMEN der TS-Anwendung. Nur den Blättern sind Eigenschaften zugeordnet.

Die angegebenen Werte haben folgende Bedeutung:

i
: Index des dargestellten Namensteils.

[Npi]
: Wert des dargestellten Namensteils.

[x]
: Hash-Tabellenindex des dargestellten Namensteils in der Datei NAMEP*i*. Der Wert liegt zwischen 1 und dem in der 1. Tabelle angegebenen Maximum für die Hashtabelle.

[p]
: Index der ersten Eigenschaft der TS-Anwendung in der Tabelle in der Datei PROPERTIES. Dieser Wert wird nur ausgegeben, wenn Sie von *tnsxinfo* auch die Ausgabe der Eigenschaften angefordert haben.

Ausgabe der Eigenschaften

Die Eigenschaften aller TS-Anwendungen in dem TS-Directory werden wie folgt aufbereitet: (Es ist nur ein Teil der im Beispiel von *tnsxinfo* gelieferten Informationen abgedruckt.)

```
PROPERTIES: 14 Eintraege
  #  xfnl next [t property    lng   off] [np# ind]
  0: XF..    1 [I TS_...        1     0] [ 5  864]
           0  00
  1: X.N.    2 [I TS_TRANS     20     1] [ 5  864]
           0  02001400 01004000  0a005bc4 c9c1d3d6
          10  c7401208
  2: X..L   -1 [I TS_TRSYS      1    15] [ 5  864]
           0  00
  3: XF..    4 [I TS_...        1    16] [ 5  868]
           0  00
  4: X.N.    5 [I TS_TRANS     37    17] [ 5  868]
           0  02002500 02001000  1b00300f 03490000
          10  00090001 08001410  1996fe80 08d3c1d5
          20  c1d5e640 40
  5: X.N.    6 [I TS_TRSYS      1    3c] [ 5  868]
           0  02
```

In der ersten Zeile der Tabelle gibt *tnsxinfo* an, wieviele Eigenschaften in dem TS-Directory eingetragen sind. Für jede dieser Eigenschaften werden mehrere Zeilen ausgegeben. Die erste Zeile enthält die Werte zu den in der Kopfzeile der Tabelle angegebenen Größen, die folgenden Zeilen den Wert der Eigenschaft in hexadezimaler Form.

Die Größen der Kopfzeile haben die folgende Bedeutung:

#
: Index der Eigenschaft in der Datei PROPERTIES des TS-Directory.

xfnl
: gibt an, ob es sich um die erste oder die letzte Eigenschaft der TS-Anwendung in der Liste handelt. Mögliche Werte für xfnl sind:

  X
  : Eintrag ist belegt.

  F
  : erster Eintrag einer Eigenschaft der TS-Anwendung.

  L
  : letzter Eigenschaftseintrag für diese TS-Anwendung.

  N
  : Folgeeintrag einer Eigenschaft für die TS-Anwendung.

  Die Werte F, L, N schließen einander aus.

next
: Index # der nächsten Eigenschaft, die zu derselben TS-Anwendung gehört.

t
: Typ der Eigenschaft. Es wird derzeit nur der Wert *I* für TS_ITEM ausgegeben. Der Typ TS_GROUP (G) wird zur Zeit noch nicht unterstützt.

property
: Bezeichnung der Eigenschaft. Es werden die Bezeichnungen ausgegeben, die auch bei der Ausgabe der erlaubten Längen der Eigenschaftswerte verwendet wurden (siehe oben).

lng
: Länge des Wertes der Eigenschaft in Byte.

off
: Offset in der Datei PRVALUES des TS-Directory, in der die Eigenschaftswerte stehen. Offset gibt den Abstand vom Dateianfang bis zum Beginn dieses Eigenschaftswertes in Byte an.

np#
: Index Namensteil, dem die Eigenschaft zugeordnet ist (np# = 1,...,5).

ind
: Hash-Tabellenindex dieses Namensteils in der Datei NAMEPnp#

Fehler

Besitzt das in *-d num* angegebene TS-Directory eine falsche Datei VERSION, so gibt *tnsxinfo* den Inhalt der Datei VERSION und die in VERSION erwarteten Werte zusammen mit einer Fehlermeldung aus.

tnsxinfo gibt ebenso eine Fehlermeldung aus, wenn er die Datei VERSION eines TS-Directory nicht öffnen kann und die Datei ROOT eine falsche Länge hat. Beides deutet auf ein TS-Directory hin, das mit einer alten CMX-Version erzeugt und nicht gemäß Abschnitt „Migration“ auf Seite 98 umgesetzt wurde.

Dateien

DIR1 ... DIR9
: TS-Directory, aufsteigend nummeriert.

10.21 Sperren der Zugriffe zum TNS-Dämon (tnsxlock)

tnsxlock sperrt und entsperrt Zugriffe zum TNS-Dämon *tnsxd*. Solange die Sperre gesetzt ist, weist *tnsxd* sämtliche Zugriffe mit der Fehlermeldung (typ, class, value) = (TS_TEMPERR, TS_INTERR, TS_NORQ) ab. Das bedeutet, dass temporär von *tnsxd* keine Anforderungen bearbeitet werden (siehe <*tnsx.h*>).

tnsxlock kann nur vom Systemverwalter (root-Berechtigung erforderlich) aufgerufen werden.

Syntax

tnsxlock␣{ **on** | **off** }

on
Sperre setzen

off
Sperre lösen

Eine gesetzte Sperre bleibt wirksam, bis sie gelöst oder *tnsxd* neu gestartet wird.

Siehe auch

tnsxd, *tnsxt*

10.22 Eigenschaften von TS-Anwendungen ausgeben (tnsxprop)

tnsxprop gibt die Werte aller Eigenschaften in einem abdruckbaren Format auf *stdout* aus, die in einem TS-Directory für die angegebenen TS-Anwendungen enthalten sind.

Mit Hilfe des Parameters *option* kann man festlegen, in welchem Format die Eigenschaften ausgegeben werden sollen.

Die TS-Anwendungen werden durch die Parameterwerte von *name* bestimmt. Die Parameterwerte für *name* können auch aus der Datei *file* an *tnsxprop* übergeben werden. Wird weder für *name* noch für *file* eine Angabe gemacht, so bereitet *tnsxprop* die Eigenschaften aller TS-Anwendungen des TS-Directory im angegebenen Format auf.

Syntax

tnsxprop [␣**-d**␣num] [␣{ **-h** | **-S** }] [␣**-f**␣file] [␣name ...]

-d␣num

Nummer des TS-Directory, das verwendet werden soll.

Mögliche Angabe für *num*: 1,...,9

Ohne Angabe wird *num* = 1 eingestellt.

-h | **-S**

Mit Hilfe dieses Parameters wird das Format festgelegt, in dem die Eigenschaften ausgegeben werden sollen.

Ohne Angabe wird *-S* eingestellt.

Die Optionen *-h* und *-S* schließen sich gegenseitig aus. Sie haben folgende Bedeutung:

-h

Aufbereitung der Eigenschaften in hexadezimaler Darstellung. Die Ausgabe der Eigenschaftswerte erfolgt als Hexadezimalziffern-String zusammen mit der entsprechenden Bitdarstellung, wobei das niederwertigste Bit ganz rechts steht.

-S

Aufbereitung der Eigenschaften in symbolischer Darstellung im Format *tnsxfrm*. Siehe hierzu Abschnitt „Syntax der TNS-Konfigurationsdatei“ auf Seite 77.

-f␣file

Für *file* ist der Name einer Datei anzugeben, die die GLOBALEN NAMEN der TS-Anwendungen enthält, deren Eigenschaften abgefragt werden sollen. Die GLOBALEN NAMEN sind wie unter *name* beschrieben anzugeben.

name

Für *name* ist der GLOBALE NAME der TS-Anwendung im TS-Directory wie folgt anzugeben:

NP5.NP4.NP3.NP2.NP1. Die NPi sind die Namensteile des GLOBALEN NAMENS. Dabei ist NP5 der Namensteil[5], also der Namensteil der unteren Hierarchiestufe. NP1 ist der Namensteil[1], also der in der Hierarchie höchste Namensteil. Die Namensteile sind in von links nach rechts aufsteigender hierarchischer Reihenfolge anzugeben.

Ist bei einem GLOBALEN NAMEN einer der Namensteile nicht belegt (z. B. NP4) und folgt diesem Namensteil noch ein Namensteil höherer Hierarchie (z. B. NP3), so ist von dem nicht belegten Namensteil das Trennzeichen (.) anzugeben. Eine Folge von Trennzeichen am Ende des Wertes von *name* kann weggelassen werden.

Enthalten die Namensteile Sonderzeichen, deren Sonderbedeutung eine Mehrdeutigkeit der Syntax verursachen würde, so müssen diese Sonderzeichen mit \ (Gegenschrägstrich) entwertet werden. Im Zweifelsfall sollten Sie jedes Sonderzeichen entwerten. Ist die Entwertung überflüssig, so wird sie von *tnsxprop* ignoriert.

Gibt man für einen der Namensteile den Wert * (Stern) an, so liefert *tnsxprop* die Eigenschaften von allen TS-Anwendungen, die für diesen Namensteil einen beliebigen Wert haben und in den anderen Namensteilen mit der Angabe in *name* übereinstimmen (Filtermodus TS_RESTRICTED).

Dateien

DIR1 ... DIR9

TS-Directory, aufsteigend nummeriert, maximal 9.

Beispiel 1

Der TS-Anwendung im TS-Directory 1, die nur Namensteil[5] mit dem Wert *Beispiel_1* hat, sind alle möglichen Eigenschaften zugeordnet. Diese Eigenschaften sollen in hexadezimaler Darstellung auf *stdout* ausgegeben werden. Geben Sie folgendes Kommando ein:

tnsxprop -d 1 -h Beispiel_1

Beispiel 2

Für die TS-Anwendung "Beispiel_1" im TS-Directory 1 sollen die Eigenschaften in symbolischer Darstellung auf *stdout* ausgegeben werden. Geben Sie folgendes Kommando ein:

tnsxprop -d 1 Beispiel_1

Die Aufrufe *tnsxprop -S > DAT1* und *tnsxcom -D DAT1* sind äquivalent, beide schreiben den Inhalt des TS-Directory 1 in symbolischer Darstellung in die Datei *DAT1*.

10.23 Sicherstellen und Aufbereiten der Trace-Information (tnsxt)

tnsxt kann von jedem Benutzer aufgerufen werden. Mit Hilfe von *tnsxt* kann der Trace der Zugriffe auf den TNS jederzeit gestartet und angehalten werden. *tnsxt* sammelt die letzten Zugriffe zum TNS in abdruckbarer Darstellung in einem Ringpuffer, solange der Trace eingeschaltet ist. Der Ringpuffer wird in der Datei *tnsxd.trc* angelegt und bleibt während der gesamten Lebensdauer des Systems bestehen. Der Trace kann jederzeit gestartet werden, der Start muss jedoch vor den zu überwachenden Zugriffen erfolgen. Hält man den Trace an, so kann der Ringpuffer ausgegeben werden. Bei Fortstart des Trace wird der Ringpuffer fortgeschrieben.

Syntax

tnsxt ␣ { **on** | **off** }

on
: der Trace wird (fort)gestartet,

off
: der Trace wird angehalten.

Dateien

/opt/SMAW/SMAWcmx/lib/cmx/tnsxd.trc
: Datei mit Trace-Einträgen

Siehe auch

tnsxd

11 SNMP Subagent für CMX

Das vorliegende Kapitel enthält Informationen zu Funktion und Betrieb des CMX-Agenten.

Im Abschnitt „Funktion des CMX-Agenten“ auf Seite 328 werden insbesondere die Einbettung des CMX-Agenten in das SNMP-Netzmanagement-Konzept sowie die Gruppen und Objektklassen der CMX-MIB beschrieben.

Der Abschnitt „Betrieb des CMX-Agenten“ auf Seite 352 wendet sich an den Systemverwalter des lokalen UNIX-Systems und beschreibt die erforderlichen Aufgaben für die Installation des CMX-Agenten auf dem lokalen UNIX-System und die Möglichkeiten der lokalen Administration.

11.1 Übersicht zum CMX-Agenten

Der EMANATE Subagent des Produkts CMX ist ein SNMP-Agent für die Kommunikationsprodukte auf UNIX-Systemen. In diesem Handbuch wird für diesen Agenten die Kurzbezeichnung CMX-Agent verwendet. Unter Kommunikationsprodukten versteht man die Produkte CMX, CCP (Communication Control Program) und XTI (X/OPEN Transport Interface).

Der CMX-Agent unterstützt das Management von CMX und Transportsystemen, die durch die CCP-Produkte realisiert werden, über SNMP. Der CMX-Agent wird auf Ihrem lokalen UNIX-Rechner installiert. Diese Management-Anwendung ermöglicht dem Netzverwalter der Management-Station, Informationen vom CMX-Agenten direkt abzurufen. Ein Großteil der CMX- und CCP-spezifischen Konfigurations- und Administrationsdaten, die CMX lokal verwaltet, können über den CMX-Agenten von der Management-Station aus abgefragt und zum Teil auch verändert werden. Administrierbare Objekte sind u.a. die Communication Controller (CC), die Transport Service Provider (TSP), der zentrale CMX-Automat und die Subnetz-Profile.

Der CMX-Agent ermöglicht die ferne Administration über SNMP, dem Simple Network Management Protocol. Mit SNMP steht ein Hilfsmittel zur Verfügung, um die Komponenten eines Netzes von einer Management-Station aus zu verwalten und zu überwachen.

Die zentrale Schnittstelle zwischen Agent und Management-Station ist die Management Information Base (MIB). Die MIB beschreibt die Typen von administrierbaren Objekten und die darauf erlaubten Operationen.

Der CMX-Agent realisiert eine CMX-spezifische Management Information Base, im folgenden CMX-MIB genannt.

11.2 Funktion des CMX-Agenten

Der CMX-Agent ist ein SNMP-Agent für CMX ab der Version 5.0 und unterstützt das Management von CMX und den durch die CCP-Produkte realisierten Transportsystemen über SNMP.

Dieser Abschnitt beschreibt:

- die Kommunikation zwischen dem CMX-Agenten und einer SNMP-Management-Station
- die EMANATE-basierte Architektur des Agenten (EMANATE Master Agent und UNIX SNMP Agent Adapter)
- Struktur und Operationen der Management Information Base (MIB)
- die für den CMX-Agenten relevanten Gruppen der Internet MIB-II
- die Objektklassen und Gruppen der CMX-MIB
- die Trap-Nachrichten der CMX-MIB

11.2.1 CMX-Agent und SNMP-Management-Stationen

Der Ausfall eines Netzes kann hohe Kosten und große Probleme verursachen. Deshalb ist es wichtig, das Netz und seine Komponenten zu überwachen, Probleme zu erkennen und entsprechende Maßnahmen rechtzeitig einzuleiten.

In einem Netz gibt es eine oder mehrere Netzmanagement-Stationen, von denen aus das Netz überwacht und verwaltet wird. Mit SNMP (Simple Network Management Protocol - RFC1157) steht ein Hilfsmittel zur Verfügung, um auf Basis von TCP/IP die Komponenten eines Rechnernetzes zu verwalten und zu überwachen. SNMP ist ein Protokoll, über das in einem Netz Lese- und Schreibanforderungen für Objekte übertragen werden können.

SNMP wird - in den heute verfügbaren Implementierungen - über ein TCP/IP basiertes Netz übertragen. In den so erreichbaren Netzelementen können dann beliebige Objekte verwaltet werden, auch wenn sie zu anderen Transportsystemen wie z.B. denen von CMX gehören.

SNMP muss sowohl in der Management-Station als auch in den zu überwachenden Netzkomponenten implementiert sein. Netzkomponenten sind z.B. UNIX-Endsysteme, Bridges, Router und Gateways. In unserem Kontext sind die Netzkomponenten UNIX-Systeme.

Der CMX-Agent, der auf einem lokalen UNIX-System abläuft, kommuniziert mit einer Management-Station über SNMP. Die Management-Station fordert vom CMX-Agenten CMX- und CCP-spezifische Informationen mit den SNMP-Operationen *GET* und *GETNEXT* (siehe „GET-Request (bzw. -Response)“ auf Seite 333 und „GETNEXT-Request (bzw. -Response)“ auf Seite 333) an. Schreibende Zugriffe führt sie mit der SNMP-Operation *SET* durch. Der CMX-Agent bildet die Anforderungen der Management-Station auf die CMX-spezifischen Systemfunktionen zur lokalen Administration, z.B. *cmxinfo* (siehe Abschnitt „Informationen zur CMX-Konfiguration (cmxinfo)“ auf Seite 239) und *bstv*, ab und liefert die von der Management-Station angeforderte Information zurück. Alle administrierbaren Objekte und die darauf erlaubten Operationen sind in der CMX-MIB definiert (siehe Abschnitt „Die CMX-MIB“ auf Seite 336).

Der CMX-Agent wird immer erst dann aktiv, wenn er von der Management-Station eine Nachricht empfängt. Eine Ausnahme bildet die SNMP-Operation *TRAP*. Mit einer Trap-Nachricht kann der CMX-Agent der Management-Station unaufgefordert Meldungen senden, wenn besondere Ereignisse (Zustandsübergänge) im Agent-System festgestellt werden (siehe Abschnitt „Trap-Nachrichten der CMX-MIB“ auf Seite 350).

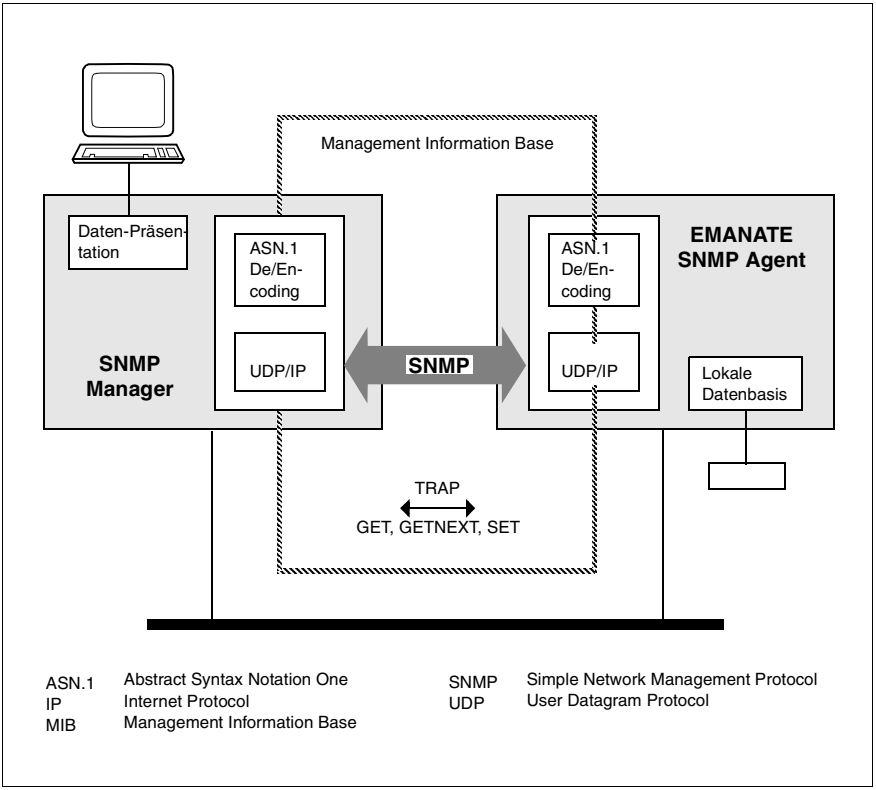


Bild 36: Das SNMP-Netzmanagement-Konzept

11.2.2 EMANATE-basierte Architektur des Agenten

Mit der Einführung von EMANATE wird der UNIX SNMP Agent (oder Core Agent) durch den UNIX SNMP Agent Adapter V3.0 und den EMANATE Master Agenten abgelöst. Die bisherige TCP-Extension *SItcpext* wird durch den MIB-II-Subagenten *SImib2* ersetzt.

Der CMX-Subagent ist als eigener Dämon-Prozess realisiert. Dieser kann unabhängig vom EMANATE Master Agenten gestartet und gestoppt werden.

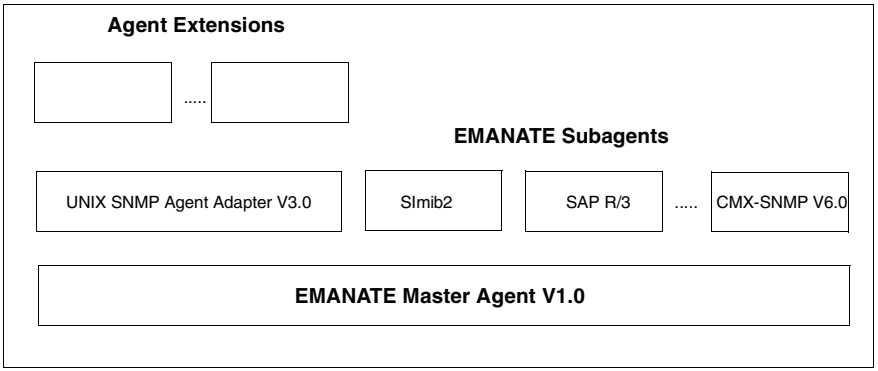


Bild 37: EMANATE-basierte Architektur des Agenten

11.2.3 Die Management Information Base (MIB)

Neben dem Protokoll, über das die Netzmanagement-Station und der SNMP-Agent miteinander kommunizieren, muss auch festgelegt werden, auf welche Objekte der Netzkomponenten die Management-Station über den SNMP-Agenten lesend oder schreibend zugreifen kann.

Diese Objekte werden in den so genannten Management Information Bases (MIBs) festgelegt. Die Standard-MIB für das Netzmanagement TCP/IP-basierter Netze ist die MIB-II (Management Information Base for Network Management of TCP/IP-based Internets - RFC1213). Der in Solaris vorhandene „Solstice Enterprise Agent“ unterstützt zusammen mit dem TCP/IP-Subagenten SMAWmibii die MIB-II in vollem Umfang.

Die MIB beschreibt die administrierbaren Objekte und die erlaubten Operationen. In der MIB werden die von der Management-Station zu verwaltenden Informationen über die Agenten gespeichert. Jeder Agent verwaltet seine statischen und dynamischen Informationen, z.B. Messwerte und Auslastungen, selbst.

Die MIB beschreibt den Maximalumfang an Objekten und Operationen, die einer Management-Station zur Verfügung stehen. Sie beschreibt nicht, wie die Objekte im Agent-System technisch realisiert sind und wie die Informationen dem Benutzer präsentiert werden.

SNMP legt im Request for Comment RFC1155 (Structure and Identification of Management Information for TCP/IP-based Internets) fest, welche Strukturmittel zur Definition einer SNMP-MIB verwendet werden dürfen. RFC1157 (Simple Network Management Protocol) beschreibt in ASN.1-Notation das SNMP-Protokoll für den Datentransfer zwischen einer Management-Station und dem SNMP-Agenten.

Allen Objekten der MIB wird zur formalen Identifikation in einem weltweit eindeutigen Registrierungsbaum ein *Object Identifier* zugeordnet. Die Registrierung ist dabei baumartig strukturiert. Die mit SNMP administrierbaren Objekte erscheinen als „Blätter“ im Registrierungsbaum. Zur Strukturierung können beliebige Zwischenknoten definiert werden, die dann eigene Object Identifier haben. Teilbäume in der tiefsten Hierarchiestufe werden manchmal auch als *Objektklassen* bezeichnet.

Auf der tiefsten Strukturebene beschreibt die MIB alle administrierbaren Objekte des Agent-Systems. Dazu werden OBJECT-TYPE Makros definiert, die jeweils den zugehörigen Object Identifier, die Syntax des Objektwertes, die erlaubten Operationen und einen Beschreibungstext enthalten.

SNMP-Objekte haben keine einzelnen Attribute - wie z.B. beim Netzmanagement nach OSI-Normen - sondern nur einen Attributwert. Soll ein Objekt mit mehreren Attributen gemäß SNMP-Konventionen dargestellt werden, so wird das Objekt zu einer SNMP-Objektklasse, die einzelnen Attribute werden zu einzelnen SNMP-Objekten gemacht.

Enthält ein SNMP-Agent mehrere gleichartige Objekte derselben Objektklasse, so werden diese in einer SNMP-Tabelle dargestellt.

Ein Object Identifier im weltweit eindeutigen Registrierungsbaum beschreibt zunächst den Typ des Objekts. Das Objekt selbst wird innerhalb eines Agent-Systems durch Anfügen eines geeigneten Suffix identifiziert. Bei SNMP-Tabellen ist das der Wert des Tabellenindex, sonst der Wert null.

Unter *System* versteht SNMP aus Sicht des Managers immer die Einheit, die durch eine IP-Adresse ausgezeichnet ist.

Auf den in der Management Information Base definierten Objekten bietet SNMP die folgenden Operationen an.

GET-Request (bzw. -Response)

Mit dieser Operation kann eine Objektinstanz (oder auch eine Liste von Objektinstanzen) vom Agent-System abgefragt werden.

Wichtig ist, dass der SNMP-Manager beim GET-Request den Object Identifier der Instanz (bzw. eine Liste davon) genau angibt, d.h.

<object type>.0>
bei einfachen Instanzen (z.B. *cmxTsapMax.0*)

<object type>.<Indexwert>
bei SNMP-Tabellen (z.B. *cmxCcType.3*)

GETNEXT-Request (bzw. -Response)

Mit dieser Operation kann ebenfalls eine Objektinstanz (oder auch eine Liste von Objektinstanzen) vom Agent-System abgefragt werden. Bei einem GETNEXT-Request ermittelt der SNMP-Agent den lexikographischen Nachfolger des im GETNEXT-Request angegebenen Object Identifier und stellt diesen zusammen mit dem Attributwert in der GETNEXT-Response bereit. Die folgenden Beispiele aus der CMX-MIB verdeutlichen dies:

Beispiel 1

Ein GETNEXT auf *cmxCcType* liefert – sofern mindestens ein Communication Controller installiert ist – *cmxCcType.1* und den Wert des Attributs (also den Controller-Typ mit Index 1).

Ein GETNEXT auf *cmxCcType.1* liefert – sofern mehr als ein Communication Controller installiert ist – *cmxCcType.2* und den Wert des Attributs (also den Controller-Typ mit Index 2).

Ein GETNEXT auf *cmxCcType.2* liefert – sofern kein weiterer Communication Controller installiert ist – *cmxCcDescr.1* und den Wert des Attributs (also die Beschreibung des Controllers mit Index 1). In der CMX-MIB sind die Object Types *cmxCcType* und *cmxCcDescr* hintereinander spezifiziert und registriert.

Dieses Beispiel veranschaulicht, wie mit GETNEXT eine ganze SNMP-Tabelle ausgelesen werden kann. Der SNMP-Manager hat vorher keine Kenntnis über die vorhandenen Tabellenzeilen und deren aktuelle Indexwerte.

Beispiel 2

Ein GETNEXT auf *cmxTsapMax* liefert *cmxTsapMax.0* und den Wert des Attributs.

Ein GETNEXT auf *cmxTsapMax.0* liefert *cmxTcepMax.0* und den Wert des Attributs. In der CMX-MIB sind diese beiden Attribute hintereinander spezifiziert und registriert.

SET-Request (bzw. -Response)

Mit dieser Operation kann der Wert einer Objektinstanz (oder auch einer Liste von Objektinstanzen) im Agent-System verändert werden.

Wichtig ist, dass der SNMP-Manager beim SET-Request den Object Identifier der Instanz (bzw. eine Liste davon) genau angibt, d.h.

<object type>.0
bei einfachen Instanzen (z.B. cmxNeateMaxConn.O)

<object type>.<Indexwert>
bei SNMP-Tabellen (z.B. cmxCcAdminState.3)

Trap-Nachrichten

Die Operationen GET-, GETNEXT- und SET-Request sind Aufträge von der Management-Station an das Agent-System, die der SNMP-Agent mit einem entsprechenden Response beantwortet.

Trap-Nachrichten hingegen sind Meldungen, die von einem SNMP-Agenten unaufgefordert an die Management-Station gesendet werden. Meist meldet der Agent damit gravierende Fehlersituationen im System. Eine Trap-Nachricht kann neben dem Trap-Typ auch noch weitere Informationen enthalten (also Attributwerte zu Object Types). Trap-Nachrichten, die der CMX-Agent senden kann, sind in Abschnitt „Trap-Nachrichten der CMX-MIB“ auf Seite 350 beschrieben.

11.2.4 Die Internet MIB-II

Die im RFC1213 spezifizierte MIB-II ist der zentrale MIB-Teil, der von jedem SNMP-Agenten unterstützt werden muss. Für die CMX-MIB sind die beiden folgenden MIB-II-Gruppen relevant.

– Die MIB-II-Gruppe *System*

– Die MIB-II-Gruppe *Interface*

Die MIB-II-Gruppe *System*

Diese Gruppe enthält Object Types zur Identifikation des Gesamtsystems, die der zentrale SNMP-Agent für UNIX-Systeme verwaltet; z.B. der Object Type *sysDescr* enthält Informationen zum Solaris-System (z. B. Sun SNMP-Agent).

Die MIB-II-Gruppe *Interface*

Der Begriff Interface spielt eine zentrale Rolle im SNMP-Datenmodell. Ein Interface beschreibt insbesondere einen Zugangspunkt zu einem Subnetz. Alle Subnetz-Anschlüsse eines Systems werden in einer SNMP-Tabelle präsentiert. Die einheitliche Darstellung der Subnetz-Anschlüsse ist vom LAN- oder WAN-Anschluss-Typ sowie von der Implementierungsart und Produktzugehörigkeit im UNIX-System unabhängig. Die verschiedenen SNMP-Subagenten eines UNIX-Systems tragen hier die Subnetz-Anschlüsse ein, die in ihrem Kontext unterstützt werden. Insbesondere werden hier auch die von CMX und den CCP-Produkten verwalteten WAN-Subnetz-Anschlüsse angezeigt.

Die SNMP-Tabelle *ifTable* enthält Object Types zu Typ, Subnetz-Adresse und Zustand des Subnetz-Anschlusses. Monitoringinformationen werden abhängig vom Typ des Subnetz-Anschlusses angeboten. Der CMX-Agent präsentiert in dieser Version keine Statistiken für die WAN-Subnetz-Anschlüsse. Die Zähler stehen in diesen Fällen immer auf Null.

Die anderen MIB-II-Gruppen wie *IP Group*, *UDP Group* und *TCP Group* beschreiben die Protokoll-Entities für die Internet-Anbindung. Für die CMX-MIB sind diese Gruppen nicht relevant.

11.2.5 Die CMX-MIB

Die CMX-MIB enthält alle Objekte der Kommunikation, auf die die Management-Station über den CMX-Agenten lesend und teilweise auch schreibend zugreifen kann. Nach der Installation des CMX-Agenten sind Schreibzugriffe auf der CMX-MIB standardmäßig gesperrt und müssen explizit freigeschaltet werden (siehe Abschnitt „Lokale Administration“ auf Seite 353).

Die Menge der Objekte, die durch die Management-Station über einen CMX-Agenten administriert werden kann, umfasst alle in der CMX-MIB definierten Objekte sowie eine Teilmenge der in der RFC1213 MIB-II definierten Objekte (die CMX-spezifischen Subnetz-Anschlüsse in der *ifTable* der MIB-II, siehe Abschnitt „Die Internet MIB-II“ auf Seite 335).

Die Objekte der CMX-MIB sind in der Abstract Syntax Notation One (ASN.1) spezifiziert. Der CMX-Agent unterstützt die folgenden, in den Requests for Comments (RFCs) festgeschriebenen Standards:

– RFC1155 SMI: Structure and Identification of Management Information for TCP/IP-based Internets.

– RFC1157 SNMP: Simple Network Management Protocol.

– RFC1212: Concise MIB Definitions

Das ASN.1-Modul *cmx.asn1* beschreibt die CMX-MIB und liegt auf dem lokalen Agent-System im Dateiverzeichnis */opt/SMAW/SMAWsnmpm/asn1/snmpv1*. Diese Spezifikation ist die formale Schnittstellenbeschreibung zwischen einer Management-Station und einem CMX-Agenten.

Für die Fujitsu Siemens Computers GmbH ist im weltweit eindeutigen Registrierungsbaum ein Unterbaum mit der Wurzel 1.3.6.1.4.1.231 reserviert (Object Identifier *sni(231)* im Bild „Die CMX-MIB im SNI-Registrierungsbaum“ auf Seite 337).

Die CMX-MIB ist in diesem Unterbaum als Zwischenknoten mit dem Object Identifier *sniCMX(2)* realisiert.

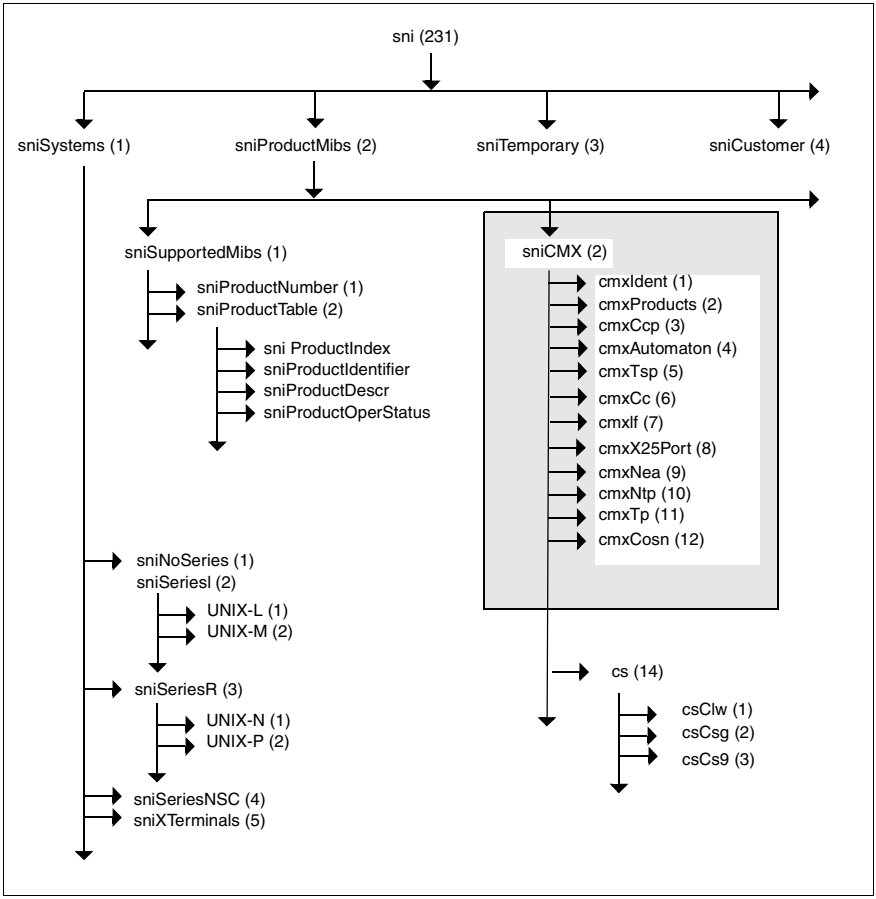


Bild 38: Die CMX-MIB im SNI-Registrierungsbaum

Die CMX-MIB ist in verschiedene CMX-MIB-Gruppen organisiert, denen die einzelnen Objektklassen zugeordnet sind. Die Gruppen und die Objektklassen der CMX-MIB modellieren die logische Struktur der Kommunikation, die in Bild „OSI-Protokollstacks“ auf Seite 18 dargestellt ist.

Bevor im Folgenden die einzelnen MIB-Gruppen genauer beschrieben werden, erhalten Sie einen kurzen Überblick zu wesentlichen Zusammenhängen in der Kommunikation auf UNIX-Systemen.

Ein zentraler Begriff ist das CCP-Profil. Ein CCP-Profil definiert für jede der vier unteren Schichten im OSI-Referenzmodell ein bestimmtes Protokoll und legt damit bestimmte Netzeigenschaften fest.

Je nach Typ sind die Protocol Entities eines CCP-Profils unterschiedlich realisiert: teilweise im UNIX-Kernel und/oder als Komponenten der Loadware auf den Communication Controllern. Abhängig davon sind auch die Attribute und Operationen, die für das Netzmanagement verfügbar sind.

Um für das Management der unterschiedlichen CCP-Profiltypen eine einheitliche Sicht zu erreichen, sind in diesem Zusammenhang folgende Objektklassen definiert:

– Transport Service Provider (TSP)

  Hiermit werden alle Komponenten (Protocol Entities) von CCP-Profilen zusammengefasst, die im UNIX-Kernel als ‚managable object' ablaufen und die Subnetz-Profile auf den Communication Controllern steuern. Die einzelnen Protocol Entities innerhalb der TSPs werden der Management-Station über jeweils eigene MIB-Untergruppen zugänglich gemacht.

– Communication Controller (CC)

  Diese Objektklasse beschreibt die Kommunikations-Hardware zum Anschluss des UNIX-Systems an ein Subnetz in einer einheitlichen Sicht für das Management. Dem Communication Controller ist das darauf ablaufende Subnetz-Profil als Attribut zugewiesen.

– TSP Access Point

  Der zentrale CMX-Automat bietet an seinen Zugangspunkten zu den Kommunikationskomponenten eine einheitliche Sicht von Konfigurations- und Monitoringattributen. Für das Management dieser Attribute wurde die Objektklasse TSP Access Point definiert.

Das Zusammenwirken dieser drei Objektklassen veranschaulicht die folgende Grafik. Lesen Sie dazu auch den Abschnitt „Architektur der CCP-Profile“ auf Seite 23.

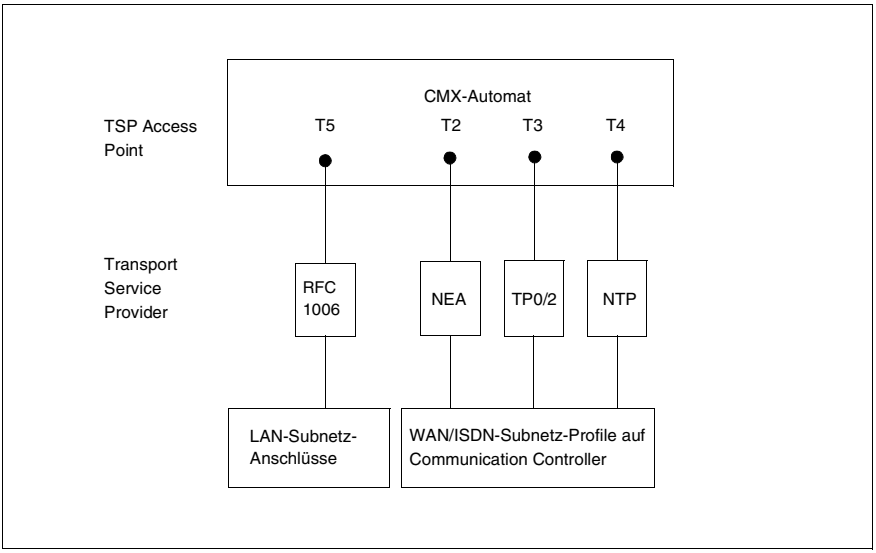


Bild 39: Logische Struktur der Kommunikation in UNIX-Systemen

Weitere Informationen zur Architektur und den einzelnen Kommunkationskomponenten befinden sich in den Handbüchern „CMX/CCP, ISDN-Kommunikation“ [3] und „CMX/CCP, WAN-Kommunikation“ [4]).

Das folgende Bild veranschaulicht die Zuordnung der Transport Service Provider und der Subnetz-Anschlüsse bzw. Subnetz-Profile zu den Gruppen der CMX-MIB.

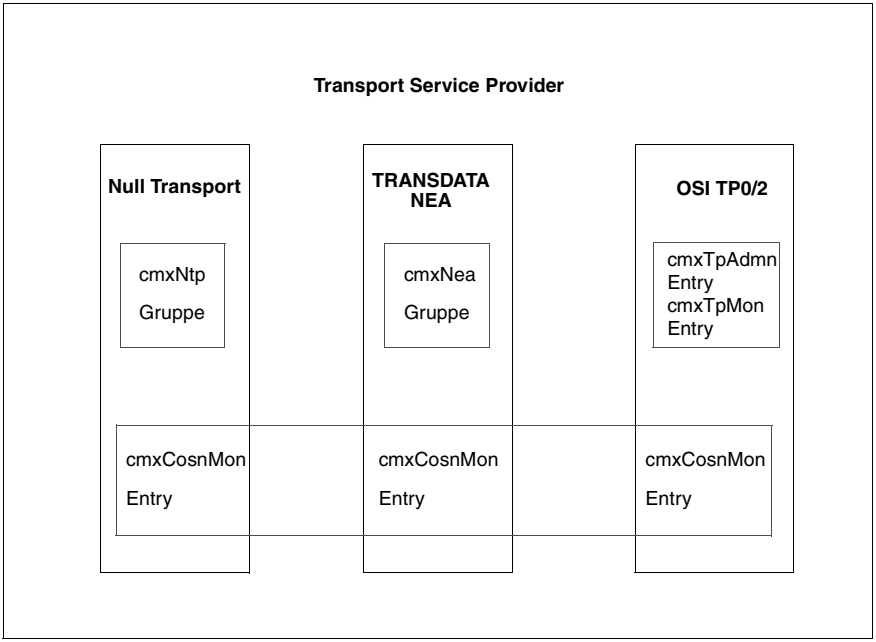


Bild 40: Struktur der Transport Service Provider aus der Sicht von SNMP

Die Transport Entities im Transport Service Provider OSI TP0/2 und OSI TP4/CLNP werden in einer einheitlichen Sicht in der Gruppe *cmxTp* der CMX-MIB dargestellt. Zur Beschreibung der Konfigurations- und Monitoringattribute gibt es zu jeder Entity jeweils einen Eintrag in der *cmxTpAdmnTable* bzw. *cmxTpMonTable* der Gruppe *cmxTp*.

Die verbindungsorientierte Subnetzschicht innerhalb der Transport Service Provider für WAN wird in der Gruppe *cmxCosn* der CMX-MIB beschrieben. Jeder Transport Service Provider im WAN hat einen Eintrag in der *cmxCosnMonTable* (Monitoring).

In den folgenden Abschnitten finden Sie in der Reihenfolge der CMX-MIB (siehe Bild „Die CMX-MIB im SNI-Registrierungsbaum“ auf Seite 337) zu jeder Gruppe der CMX-MIB eine Zusammenfassung der wesentlichen Inhalte. Dabei werden insbesondere Zweck und Zusammenhänge der einzelnen Objektklassen beschrieben, ohne jedoch auf Details einzugehen. Weitere Informationen zu den einzelnen Objektklassen finden Sie in den inline-Beschreibungen des ASN.1-Moduls *cmx.asn1*.

11.2.5.1 Die CMX-MIB-Gruppe cmxIdent

Diese MIB-Gruppe enthält drei Object Types mit allgemeinen Produkt- und Versionsinformationen zu CMX (*cmxProductDescr*) und zum CMX-Agent (*cmxSnmpDescr*).

Mit *cmxMibVer* zeigt der Agent die Version der MIB an, die aktuell unterstützt wird. CMX V6.0 trägt hier die Zahl 1 ein. Diese Version wird bei künftigen Änderungen/Erweiterungen in den MIB-Definitionen hochgezählt. Hierdurch können auf einer Management-Station ablaufende Anwendungen im administrierten Netzwerk Agenten mit unterschiedlichen Versionsständen der CMX-MIB unterscheiden.

11.2.5.2 Die CMX-MIB-Gruppe cmxProducts

Diese MIB-Gruppe bietet in zwei SNMP-Tabellen eine Übersicht aller installierten Kommunikations-Produkte und Packages:

– *cmxProdTable* – Tabelle der Produkte

– *cmxProdPkgTable* – Tabelle der Packages mit der jeweiligen Produktzuordnung

Die Unterscheidung in diese zwei Tabellen ist erforderlich, da einerseits ein Produkt aus mehreren Packages bestehen kann, und andererseits ein Package von mehreren Produkten verwendet werden kann.

Die Information wird jeweils in abdruckbarer Form geliefert. Zur Indizierung werden die Produkt- bzw. Package-Namen als Display Strings verwendet. Die *cmxProdPkgTable* ist dabei als Matrix mit einem zweistufigen Index organisiert.

11.2.5.3 Die CMX-MIB-Gruppe cmxCcp

Diese MIB-Gruppe bietet Informationen zu den Subnetz-Profilen. In der Tabelle *cmxCcpCfTable* werden alle zu den installierten Subnetz-Profilen vorhandenen Konfigurationsdateien (KDs) aufgelistet. Der Tabelle entnehmen Sie, welche Konfigurationsdateien Sie einem Communication Controller in Abhängigkeit vom Subnetz-Profil zuweisen können. Die in *cmxCcpCfName* aufgelisteten Dateien sind zulässige Attributwerte bei einer SNMP-Set-Operation bzgl. *cmxCcCfAss* in der *cmxCcTable* der CMX-MIB-Gruppe *cmxCc*.

Die *cmxCcpCfTable* hat zwei Indexattribute:

– *cmxCcpCfPIndex*

Dieser primäre Index zeigt als ASN.1 Named Number den Typ des Subnetz-Profils.

– *cmxCcpCfIndex*

Dieser zweite Index nummeriert für jedes Subnetz-Profil die Konfigurationsdateien fortlaufend durch.

An dieser Stelle ist ausnahmsweise die SNMP-Konvention verletzt, wonach ein Indexwert zur Laufzeit des SNMP-Agenten immer dieselbe Objektinstanz bezeichnen soll. Der CMX-Agent listet die Konfigurationsdateien – je Subnetz-Profil – alphabetisch auf. Dadurch kann es innerhalb der Indexwerte zu Verschiebungen kommen, wenn zur Laufzeit des Agenten Dateien hinzugefügt oder gelöscht werden.

11.2.5.4 Die CMX-MIB-Gruppe cmxAutomaton

Die Object Types zum zentralen CMX-Automat sind in der MIB-Gruppe *cmxAutomaton* zusammengefasst. Diese MIB-Gruppe umfasst vier Objektklassen:

– *cmxAutGlob* – globale CMX-Konfiguration und Statistik

– *cmxTsapTable* – Transport Service Access Points

– *cmxTcepTable* – Transport Connection End Points

– *cmxTspAccCTable* – TSP Access Points

Globale CMX-Konfiguration und Statistik

Da es den CMX-Automaten nur genau einmal im System gibt, sind seine Konfigurations- und Monitoringattribute als einfacher Teilbaum in der MIB registriert (*cmxAutGlob*).

Konfigurationsattribute

Hier wird die maximale Anzahl von Transport Service Access Points (*cmxTsapMax*) bzw. Transport Connection End Points (*cmxTcepMax*) angegeben, die von CMX unterstützt werden.

Einfache Monitoringattribute, Zähler

Die Zähler laufen ab dem Booten. Durchsatzwerte können durch eine entsprechende Manager-Anwendung durch zyklisches Polling auf die Zähler ermittelt werden.

Da in einer SNMP-MIB nur 32-Bit-Zähler definiert werden können, stellt die CMX-MIB die Low/High Werte als getrennte Object Types dar, die eine Manager-Anwendung bei Bedarf wieder zusammenführen kann.

Beachten Sie folgende Besonderheit bei den Zählern für die gesendeten und empfangenen Bytes: der CMX-Automat zählt hier intern jeweils in zwei „unsigned long“, wobei der Überlauf von Low zum High Wert jeweils bei 10^9 erfolgt. Auf diese Weise sind auch bei langer Laufzeit des Systems und hohem Durchsatz eindeutige Zählerstände gewährleistet.

Zur Unterstützung der einfachen Darstellungsformate an einer Management-Station fasst der CMX-Agent die Low/High Werte auch zu einer lesbaren Exponentialdarstellung zusammen und präsentiert dies in einem eigenen Object Type. Die Management-Station entscheidet, welche dieser Darstellungsformen sie nutzen will.

Transport Service Access Points (TSAP)

Die im System aktiven TSAPs werden in der *cmxTsapTable* aufgelistet.

Ein TSAP kann mehrere TSEL-Werte umfassen. Jeder TSEL ist wiederum einem oder mehreren Transport Service Providern zugeordnet, über die eine Verbindung mit diesem TSEL aufgebaut werden kann.

Jeder TSAP ist im CMX-Automaten durch eine Zahl (ID) im Bereich von 0 bis *cmxTsapMax* (siehe oben) identifiziert. Diese ID wird als primärer Tabellenindex *cmxTsapIndex* verwendet. Dieselbe ID kann freigegeben und später für einen neu zu öffnenden TSAP wieder vergeben werden, d.h. sie ist während einer längeren Laufzeit des Systems nicht eindeutig. Deshalb wird bei jeder neuen Belegung derselben ID ein Inkarnationszähler erhöht, der als sekundärer Tabellenindex dient (*cmxTsapInc*).

Der dritte Index *cmxTsapTselInd* nummeriert die TSEL-Werte innerhalb eines TSAP fortlaufend durch.

Tranport Connection End Point (TCEP)

Die im System aktiven TCEPs werden in der *cmxTcepTable* aufgelistet.

Ein TCEP ist genau einem Transport Service Provider zugeordnet. Deshalb werden ID und Inkarnationszähler des TSAP als primäre Tabellenindizes des TCEPs verwendet, wodurch die eindeutige Beziehung zum entsprechenden TSAP gewährleistet ist.

Jeder TCEP ist im CMX-Automat durch eine Zahl (ID) im Bereich von 0 bis *cmxTcepMax* (siehe „Tranport Connection End Point (TCEP)“ auf Seite 343) identifiziert. Diese ID wird als dritter Tabellenindex *cmxTcepIndex* verwendet. Dieselbe ID kann freigegeben und später für einen neu zu öffnenden TCEP wieder vergeben werden, d.h. sie ist während einer längeren Laufzeit des Systems nicht eindeutig. Deshalb wird bei jeder neuen Belegung derselben ID ein Inkarnationszähler erhöht, der als vierter Tabellenindex (*cmxTcepInc*) dient.

Neben allgemeinen Informationen wie Zustand, Zeitstempel des Verbindungsaufbaus und benutzter Transport Service Provider werden zu einem TCEP Adress-Informationen und Statistikwerte angeboten.

Anmerkungen zu den Adress-Informationen

Bei einer lokalen Anwendung wird der TSEL dargestellt. Bei einer fernen Anwendung wird neben dem TSEL noch eine Information zur Adresse des fernen Systems mitgeliefert. Falls bekannt, wird die Netzadresse (OSI-, NEA- oder IP-Adresse) dargestellt. Andernfalls wird die Subnetz-Adresse (MAC-, DTE-Adresse, ISDN-Rufnummer etc.) dargestellt, über die das System erreicht wird. Ist auch die Subnetz-Adresse nicht bekannt (z.B. bei Standleitungen), so wird der lokale Subnetz-Anschluss ausgegeben, der für die Kommunikation benutzt wird.

Anmerkungen zu den Statistikwerten

Analog zur Darstellung der Zähler des zentralen CMX-Automaten werden hier jeweils zwei Object Types als Low/High Value mit einem Überlauf von 10^9 verwendet. Es gelten die im vorigen Abschnitt aufgeführten Erläuterungen.

TSP Access Points

Die dem CMX-Automaten bekannten TSP Access Points werden in der *cmxTspAccCTable* aufgelistet.

CMX definiert für jeden Transport Service Provider (TSP) einen TSP Access Point. Ein TSP Access Point definiert den Zugriffspunkt zum Transport Service Provider. Für alle TSPs verwendet CMX eine einheitliche Darstellung für die Konfigurations- und Monitoringattribute.

Bei den im UNIX-Kernel implementierten TSPs sind die TSP Access Points jeweils genau einem TSP zugeordnet. Diese TSPs sind in der *cmxTspTable* aufgelistet.

Bei den in der CC-Loadware realisierten TSPs sind die TSP Access Points jeweils genau einem CC zugeordnet. Diese TSPs werden nicht in der *cmxTspTable* aufgeführt.

Die beiden Attribute *cmxTspAccCTsp* bzw.*cmxTspAccCCc* kennzeichnen die entsprechende Zuordnung zum TSP bzw. zum CC und enthalten jeweils einen Verweis auf *cmxTspIndex* bzw. *cmxCcIndex* der *cmxTspTable* (MIB-Gruppe *cmxTsp*) bzw. der *cmxCcTable* (MIB-Gruppe *cmxCc*). Das jeweils nicht relevante Attribut hat den Wert null.

Die Object Types zu den Konfigurationsattributen, den Monitoringattributen, Zählern und dem Messauftrag für Durchsatzwerte sind entsprechend organisiert wie in der Untergruppe *cmxAutGlob* des CMX-Automaten (siehe Abschnitt „Die CMX-MIB-Gruppe cmxAutomaton“ auf Seite 342). Während dort global für den CMX-Automaten gezählt wird, beziehen sich hier die Werte auf jeweils einen TSP Access Point. Die Erläuterungen insbesondere zum Messauftrag gelten hier analog.

11.2.5.5 Die CMX-MIB-Gruppe cmxTsp

Eine wichtige Rolle im Konzept der CMX-MIB spielen die Transport Service Provider. Ein TSP bezeichnet alle Komponenten (Protocol Entities) von CCP-Profilen, die zur Steuerung von Subnetz-Profilen erforderlich sind. Alle Typen von TSPs werden unabhängig von ihrer Realisierung in einer Objektklasse beschrieben.

Die SNMP-Tabelle *cmxTspTable* gibt einen Überblick zu den installierten TSPs und verwendet zur Indizierung einen fortlaufenden Integer-Index. Andere Objektklassen verweisen auf einen TSP durch Referenz auf genau diesen Indexwert.

Die *cmxTspTable* besitzt nur wenige einheitliche Attribute wie Typ und Zustand des TSPs. Außerdem sind hier die Operationen Start und Stop verankert. Durch eine Set-Operation auf den Object Type *cmxTspAdminState* können Sie den TSP starten oder stoppen.

Die verschiedenen Protocol Entities innerhalb der TSPs haben teilweise sehr unterschiedliche Konfigurations- und Monitoringattribute, die in der einheitlichen *cmxTspTable* nicht aufgenommen werden können. Zur Administration dieser Entity-typischen Object Types bietet die CMX-MIB daher eigene MIB-Gruppen (z.B. *cmxNea* und *cmxNtp*) an. Einen Überblick hierzu vermittelt Ihnen auch das Bild „Struktur der Transport Service Provider aus der Sicht von SNMP“ auf Seite 340).

11.2.5.6 Die CMX-MIB-Gruppe cmxCc

Die Objektklasse Communications Controller (CC) beschreibt die Kommunikations-Hardware zum Anschluss des UNIX-Systems an ein Subnetz in einer einheitlichen Sicht für das Management. Dem Communication Controller ist das darauf ablaufende Subnetz-Profil bzw. ein komplettes CCP-Profil als Attribut zugewiesen. Die CCs werden in der SNMP-Tabelle *cmxCcTable* aufgelistet.

Wichtige Attribute sind cmxCcOperState, cmxCcAdminState cmxCcCcpAss und cmxCcCfAss:

– *cmxCcOperState*. Hier steht der aktuelle Zustand eines installierten CC.

– *cmxCcAdminState*. Durch eine Set-Operation auf diesen Object Type können Sie den CC laden, stoppen oder dumpen. Eine Get-Operation liefert immer den Wert *none(0)* zurück. Eine erfolgreiche SNMP-Set-Operation bedeutet, dass der Auftrag im Agent-System angenommen wurde. Der Auftrag wird nach Quittierung im Hintergrund ausgeführt, da z.B. das Laden eines CC mehrere Sekunden dauern kann. Der tatsächliche Erfolg einer Lade-Operation muss durch anschließendes Polling auf *cmxCcOperState* festgestellt werden.

– *cmxCcCcpAss*, *cmxCcCfAss*. Hier stellen Sie mit einer SNMP-Operation ein, welches Subnetz-Profil und welche Konfigurationsdatei beim Laden des CC benutzt werden soll. Die in der MIB-Gruppe *cmxCcp* beschriebene *cmxCcpCfTable* listet auf, für welche Subnetz-Profile Konfigurationsdateien auf dem Agent-System zur Verfügung stehen. Änderungen an den Zuweisungen werden beim nächsten Laden des CC wirksam.

  Wird nur das Attribut *cmxCcCfAss* geändert, so bezieht sich dies auf das gerade zugewiesene Subnetz-Profil.

  Wird nur das Attribut *cmxCcCcpAss* geändert, so wird die zuletzt mit dem neuen Subnetz-Profil verknüpfte Konfigurationsdatei als zugewiesene Datei wirksam.

– *cmxCcCcpLoad*, *cmxCcCfLoad*. Bei geladenem CC wird hier das Subnetz-Profil und die benutzte Konfigurationsdatei angezeigt. Diese Werte können von den Werten in *cmxCcCcpAss* und *cmxCcCfAss* abweichen, wenn zur Laufzeit eines CC andere Zuweisungen erfolgten, die jedoch erst beim nächsten Laden wirksam werden.

11.2.5.7 Die CMX-MIB-Gruppe cmxIf

Diese MIB-Gruppe listet in der Tabelle *cmxIfTable* alle konfigurierten Subnetz-Anschlüsse auf, die über CMX administriert und bedient werden. Die LAN-Anschlüsse, die von der TCP/IP-Extension verwaltet werden, sind in dieser Tabelle nicht erfasst.

Die Objektklasse Subnetz-Anschluss beschreibt die Zugangspunkte zu den Subnetzen: ISDN, DATEX-L, DATEX-P etc. Ein Subnetz-Anschluss umfasst alle Attribute eines Subnetz-Zugangs und definiert eine am Communication Controller angeschlossene Leitung zum Subnetz. Die *cmxIfTable* verweist daher mit dem Attribut *cmxIfCc* auf den zugehörigen CC.

Durch eine Set-Operation auf das Attribut *cmxIfAdminState* können Sie den entsprechenden Subnetz-Anschluss aktivieren bzw. deaktivieren.

Die unterschiedlichen Subnetztypen (z.B. X.25) bieten jeweils verschiedene Typen von Konfigurationsattributen an, die in der *cmxIfTable* nicht dargestellt werden können. Dies erfolgt ggf. in zusätzlichen SNMP-Tabellen, die genau auf den jeweiligen Anschluss-Typ ausgerichtet sind (z.B. die Tabelle *cmxX25PortTable* der CMX-MIB-Gruppe *cmxX25Port*). Wenn zu einem Eintrag in der *cmxIfTable* eine solche Extension existiert, wird mit den Attributen *cmxIfSpecific* und *cmxIfSpecificIndex* darauf verwiesen.

11.2.5.8 Die CMX-MIB-Gruppe cmxX25Port

In der CMX-MIB ist zu X.25-Anschlüssen eine Erweiterung der Interface-Tabelle *cmxIfTable* (siehe MIB-Gruppe *cmxIf*) definiert. Die Tabelle *cmxX25PortTable* enthält Zusatzinformationen zu Subnetz-Anschlüssen an ein X.25-Paketnetz.

Das folgende Beispiel verdeutlicht die Verknüpfung der *cmxX25PortTable* mit der *cmxIfTable*.

SNMP-Tabelle *cmxIfTable*

cmxIfIndex	**...**	**3**	**4**	**5**
. . .		Dx-P4	Dx-P	ISDN
cmxIfSpecific		cmxX25PortTable	cmxX25PortTable	0.0
cmxIfSpecificIndex		1	2	0

Tabelle 28: cmxIfTable

SNMP-Tabelle *cmxX25PortTable*

cmxX25PortIndex	**1**	**2**	**3**	**4**	**5**
cmxX25IfIndex	3	4	0	0	0
.....					

Tabelle 29: cmxX25PortTable

CMX unterstützt die drei folgenden Anschlussarten an ein Paketnetz (PSDN).

- Direkter Zugang zum PSDN

  Beschreibt der Subnetz-Anschluss der *cmxIfTable* einen direkten Anschluss an ein Paketnetz, so wird dazu genau ein Eintrag in der *cmxX25PortTable* erzeugt, um die X.25-Konfigurationsparameter der Packet Level Entity anzuzeigen. Die beiden zugeordneten Einträge sind wechselseitig durch die Attribute *cmxIfSpecificIndex* und *cmxX25IfIndex* miteinander verknüpft. Im Beispiel sind das die Einträge 3 und 4 der *cmxIfTable* bzw. die Einträge 1 und 2 der *cmxX25PortTable*.

- Zugang zum PSDN über einen fest generierten ISDN-B-Kanal

  Hierbei wird der B-Kanal in der *cmxIfTable* als ISDN-Subnetz-Anschluss angezeigt, dem in der *cmxX25PortTable* genau ein X.25-Parametersatz zugeordnet ist. Die Verknüpfungen der beiden Tabellen erfolgen wie beim oben beschriebenen, direkten Zugang zum PSDN.

- Switched Access zum PSDN über das ISDN-Wählnetz

  Hierbei wird der ISDN-S0-Anschluss als Subnetz-Anschluss in der *cmxIfTable* angezeigt. Diesem Wählanschluss an das ISDN-Netz können keine X.25-spezifische Eigenschaften zugeordnet werden.

  Bei der Zweistufenwahl in das Paketnetz werden abhängig von der Partneradresse die X.25-Parameter und die zu verwendende, eigene DTE-Adresse bestimmt. Für die Konfigurierung sind bestimmte X.25-Parametersätze vorgesehen, die in der *cmxX25PortTable* mit aufgelistet werden. Diese Einträge haben jedoch keine Verknüpfung zur *cmxIfTable*. Im Beispiel sind dies die Einträge 3, 4 und 5 der *cmxX25PortTable*.

11.2.5.9 Die CMX-MIB-Gruppe cmxNea

Diese Gruppe fasst alle Konfigurations- und Monitoringattribute der NEATE- und NEAN-Protocol Entity für den TRANSDATA NEA-Transport Service Provider zusammen. Für jede dieser Protocol Entities ist eine eigene Untergruppe in der MIB vorgesehen: *cmxNeate* und *cmxNean*.

Beim Start des TRANSDATA NEA-TSPs werden alle Zähler zurückgesetzt. Alle Konfigurationsattribute beziehen sich auf die laufende Protocol Entity. Änderungen wirken sofort auf die Protocol Entity und sind nur bei aktivem TSP möglich. Abhängig von der lokalen Konfigurierung gelten wieder die im lokalen Agent-System eingestellten Attribute, wenn der Transport Service Provider neu gestartet wird, jedoch spätestens beim Rebooten des UNIX-Systems.

Monitoringattribute zur verbindungsorientierten Subnetzschicht innerhalb des TRANSDATA NEA-TSP finden Sie in der MIB-Gruppe *cmxCosn*.

11.2.5.10 Die CMX-MIB-Gruppe cmxNtp

Diese Gruppe fasst alle Konfigurations- und Monitoringattribute der NULLTP-Protocol Entity für den Transport Service Provider Null Transport zusammen.

Beim Start des Null Transport-TSPs werden alle Zähler zurückgesetzt. Alle Konfigurationsattribute beziehen sich auf die laufende Protocol Entity. Änderungen wirken sofort auf die laufende Protocol Entity und sind nur bei aktivem TSP möglich. Abhängig von der lokalen Konfigurierung gelten wieder die im lokalen Agent-System eingestellten Attribute, wenn der Transport Service Provider neu gestartet wird, jedoch spätestens beim Rebooten des UNIX-Systems.

Monitoringattribute zur verbindungsorientierten Subnetzschicht innerhalb des Null Transport-TSP finden Sie in der MIB-Gruppe *cmxCosn*.

11.2.5.11 Die CMX-MIB-Gruppe cmxTp

Diese Gruppe enthält alle Objektklassen, die zur Administration der ISO-Protocol Entities für den Transport Service Provider OSI TP0/2 definiert sind. Die SNMP-Tabellen enthalten jeweils alle möglichen Object Types zu den Transport-Protokollklassen 0/2. Abhängig von der jeweils konkreten Protokollklasse sind dann einige Object Types nicht relevant bzw. haben unterschiedliche Wertebereiche.

Es sind zwei SNMP-Tabellen definiert. Die Indexwerte nummerieren die Transport Entities fortlaufend durch; gleiche Indexwerte in den beiden Tabellen beziehen sich auf dieselbe Protocol Entity.

- *cmxTpAdmnTable* - SNMP-Tabelle zur Konfiguration der ISO-Protocol Entities

  Die meisten der hier definierten Object Types beschreiben initiale Attributwerte für neu aufzubauende Transportverbindungen. Abhängig von der lokalen Konfigurierung gelten wieder die im lokalen Agent-System eingestellten Attribute, wenn der Transport Service Provider neu gestartet wird, jedoch spätestens beim Rebooten des UNIX-Systems.

- *cmxTpMonTable* - SNMP-Tabelle zum Monitoring der ISO-Protocol Entities

  Hier befinden sich alle Attribute zum Monitoring der Protocol Entities.

  Monitoringattribute zur verbindungsorientierten Subnetzschicht innerhalb des OSI TP0/2-TSP finden Sie in der MIB-Gruppe *cmxCosn*.

11.2.5.12 Die CMX-MIB-Gruppe cmxCosn

Diese Gruppe enthält alle Objektklassen, die zur Administration der verbindungsorientierten Subnetzschicht innerhalb der Transport Service Provider im WAN definiert sind.

Das Monitoring innerhalb der Subnetzschicht erfolgt immer bezogen auf jeweils einen TSP. Dieser TSP wird im Index *cmxCosnMonTsp* durch Referenz auf den Indexwert in der *cmxTspTable* eingetragen.

Abhängig vom Typ des TSP werden Protokoll- (PDU) und/oder Interface Data Units (IDU) zur Netzschicht gezählt. Nicht relevante Zähler stehen auf null. Dies ermöglicht eine einheitliche SNMP-Tabelle für alle TSP-Typen.

11.2.6 Trap-Nachrichten der CMX-MIB

SNMP bietet die Möglichkeit, dass Agenten unaufgefordert Trap-Nachrichten an Management-Stationen senden können, wenn die Agenten besondere Ereignisse feststellen.

Trap-Nachrichten sind in SNMP vordefiniert oder sind herstellerspezifische Erweiterungen für bestimmte Systeme. Die Definition von Trap-Nachrichten ist in dem Dokument RFC 1215 festgelegt (RFC 1215: A Convention for Defining Traps for Use with the SNMP).

Trap-Nachrichten linkDown und linkUp

Der CMX-Agent unterstützt die zwei im Dokument RFC 1157 (Simple Network Management Protocol) definierten Trap-Nachrichten *linkDown* und *linkUp*. Sie kennzeichnen den Ausfall bzw. die Inbetriebnahme eines Subnetz-Anschlusses. Standardmäßig kann der CMX-Agent diese beiden Trap-Nachrichten senden. Sollen die beiden Trap-Nachrichten nicht gesendet werden, muss der Systemverwalter im lokalen Agent-System den Parameter *IFPOLLTIME* in der Datei *AgentParams.rc* auf den Wert null setzen (siehe Abschnitt „Datei AgentParams“ auf Seite 354).

Trap-Nachrichten cmxCcUp und cmxCcDown

Die beiden weiteren Trap-Nachrichten *cmxCcUp* und *cmxCcDown* sind herstellerspezifische Erweiterungen für die Kommunikation und teilen der Management-Station die Zustandsübergänge der Communication Controller mit. Die Trap-Nachricht *cmxCcDown* teilt den Ausfall eines Communication Controllers mit, die Trap-Nachricht *cmxCcUp* teilt die Inbetriebnahme eines Communication Controllers mit. Sollen die beiden Trap-Nachrichten nicht gesendet werden, muss der Systemverwalter im lokalen Agent-System den Parameter *CCPOLLTIME* in der Datei *AgentParams.rc* auf den Wert Null setzen (siehe Abschnitt „Datei AgentParams“ auf Seite 354).

11.3 Betrieb des CMX-Agenten

Die Informationen im gesamten Kapitel „Adressierungskonzept“ auf Seite 37 wenden sich ausschließlich an den Systemverwalter des lokalen UNIX-Systems. Dieser ist in diesem Kapitel auch unter der Anrede „Systemverwalter“ angegeben oder bei der direkten Anrede der Adressat.

Sie haben bei der Inbetriebnahme des CMX-Agenten im lokalen UNIX-System folgende Aufgaben durchzuführen, die in den Unterkapiteln näher beschrieben werden:

1. den CMX-Agenten installieren
2. den CMX-Agenten lokal administrieren, sofern erforderlich

11.3.1 Installation und Start des CMX-Agenten

Installation

Voraussetzung für die Installation des CMX-Agenten ist, dass CMX und der EMANATE Master Agent bereits installiert sind. Der EMANATE Master Agent SMAWsnmpm muss eventuell von einer Fujitsu-Siemens-Computers-Add-on-CD installiert werden.

Weitere Informationen insbesondere zu Versionsabhängigkeiten entnehmen Sie bitte der Freigabemitteilung.

Führen Sie zur Installation folgende Schritte durch:

1. Prüfen Sie, ob der EMANATE Master Agent (Package SMAWsnmpm) und das Paket SMAWadapt (Siemens Native Agent Adapter) bereits installiert ist.
2. Installieren Sie das Package SMAWcxagt.
3. Nach erfolgreicher Installation sind automatisch der EMANATE Master Agent und der CMX-Agent gestartet.

   Der CMX-Agent ist jetzt betriebsbereit. Auch wenn Sie Ihr UNIX-System neu hochfahren, wird der CMX-Agent automatisch mit aktiviert.

Nach der Installation des CMX-Agenten sind die Schreibzugriffe auf der CMX-MIB standardmäßig gesperrt und müssen explizit freigeschaltet werden (siehe Abschnitt „Lokale Administration“ auf Seite 353).

Starten und Stoppen, Diagnose

Durch den Start des Betriebssystems wird der CMX-Agent automatisch gestartet und mit dem Stopp des Systems automatisch gestoppt (Init-Skript */etc/init.d/cmxema*). Unabhängig vom Master Agenten kann der Subagent jederzeit mit dem Kommando *cmxsnmp [start|stop]* gestartet bzw. gestoppt werden (siehe Abschnitt „Starten und Stoppen von CMX und TSPs (StartStop)“ auf Seite 303).

Der automatische Start des CMX-Agenten beim Systemstart kann über *cmxsnmp [autostart|autostop]* gesteuert werden.

Mit *cmxsnmp diag* können Trace-Einträge des CMX-Agenten aus den Dateien */var/opt/SMAWcmx/tmp/cmxsnmp.[12].trc* aufgelistet werden.

11.3.2 Lokale Administration

Für die zur Inbetriebnahme und zum Start des CMX-Agenten benötigten Parameter werden zwei Dateien mit Standardwerten mit ausgeliefert, die Sie bei Bedarf ändern können.

– Datei *AgentParams.rc*, in der Sie Parameter z.B. für bestimmte Timer festlegen können.

– Datei *AgentTraces.rc*, in der Sie Tracepunkte festlegen können.

Für Konfigurationsänderungen können Sie die beiden Dateien entweder direkt mit einem Editor bearbeiten oder mit dem Kommando *cmxsnmpadm* (siehe Abschnitt „Rekonfiguration“ auf Seite 359). Bei direkter Bearbeitung mit einem Editor beachten Sie bitte die Hinweise am Ende von Abschnitt „Rekonfiguration“ auf Seite 359.

Die Voreinstellungen in den beiden Dateien sind so gewählt, dass der CMX-Agent sofort in Betrieb genommen werden kann, ohne dass Sie Parameter in den Dateien verändern müssen.

Beachten Sie bitte die folgenden Voreinstellungen in der Datei *AgentParams.rc*, die für Ihre Konfiguration von besonderer Bedeutung sind:

– Schreibzugriffe der Management-Station auf die Objekte der CMX-MIB sind nicht zugelassen (siehe Parameter *SETENABLE*).

– Der CMX-Agent erzeugt Trap-Nachrichten für Zustandsübergänge der Subnetz-Anschlüsse *linkUp* und *linkDown* (siehe Parameter *IFPOLLTIME*).

– Der CMX-Agent erzeugt Trap-Nachrichten für Zustandsübergänge der Communication Controller *cmxCcUp* und *cmxCcDown* (siehe Parameter *CCPOLLTIME*).

Für den EMANATE Master Agenten müssen Sie konfigurieren, welche Management-Stationen Trap-Nachrichten erhalten sollen.

11.3.2.1 Datei AgentParams

Die Datei *AgentParams.rc* wird im Dateiverzeichnis */opt/SMAW/SMAWcmx/lib/cmxsnmp* mit den unten aufgeführten Voreinstellungen ausgeliefert.

Datei AgentParams.rc mit Voreinstellungen

```
# COPYRIGHT (C) Fujitsu Siemens Computers GmbH 2000
#               All Rights Reserved
#
#        SNMP EMANATE Subagent for CMX
#
#        Parameterfile
#
# Comments must be marked by # in the first line position.
#
################################################################
#
 SETENABLE 0     # SNMP SET Operations allowed (1) or not (0)
 MAXTRACE 100    # maximum length of trace files (in kilo bytes)
#
# Timer values in seconds.
#
 MAXHOLDTIME 10  # holding timer until update of internal tables
#                # (proposed value for MAXHOLDTIME: 10)
 CFPOLLTIME 3600 #if not 0: poll for updates in CCP config files
#                # (proposed value for CFPOLLTIME: minimum 900)
 CCPOLLTIME 180  # if not 0: poll for CC state an send CC-TRAP
#                # (proposed value for CCPOLLTIME: 60...300)
 IFPOLLTIME 1800 # if not 0: poll for IF state an send IF-TRAP
#                # (proposed value for IFPOLLTIME: minimum 900)
#
# The following values are used to determine the
# size of internal tables created during startup.
# Normally they need not be changed.
#
 MAXCC 70        # maximum number of elements in cmxCcTable
 MAXTSP 10       # maximum number of elements in cmxTspTable
```

```
 MAXTSPACC 40    # maximum number of elements in cmxTspAccCTable
 MAXIF 160       # maximum number of elements in cmxIfTable
 MAXX25 130      # maximum number of elements in cmxX25PortTable
 MAXTSPSET 100   # maximum number of elements in cmxTspSetTable
 MAXTSEL 0       # maximum number of elements in cmxTsapTable
#                # (0: CMXSNMP uses the CMX maxTSAP
configuration)
 MAXTCEP 0       # maximum number of elements in cmxTcepTable
#                # (0: CMXSNMP uses the CMX maxTCEP
configuration)
```

Bedeutung der Parameter

SETENABLE

Mit dem Parameter *SETENABLE* können Sie einstellen, ob die Management-Station über den CMX-Agenten Schreiboperationen auf Objekte der CMX-MIB ausführen darf. Die Schreibberechtigung können Sie für den CMX-Agenten nur global vergeben und nicht für einzelne Objekte der CMX-MIB. Voreinstellung ist „0“, d.h. die Management-Station hat keine Schreibberechtigung für Objekte der CMX-MIB, auch wenn in der Konfiguration des EMANATE Master Agenten dieser Management-Station die Schreibberechtigung zugeteilt ist.

0:
keine Schreibberechtigung

1:
Schreibberechtigung zugelassen

Für den EMANATE Master Agenten müssen Sie konfigurieren, von welchen Management-Stationen Schreibzugriffe erlaubt sein sollen.

MAXTRACE

Hiermit bestimmen Sie die maximale Größe der beiden Trace-Dateien *cmxsnmp.1.trc* und *cmxsnmp.2.trc* in Kbyte (siehe Abschnitt „Datei Agent-Traces“ auf Seite 358).

Wertebereich in Kbyte: 1.. 100 ...

MAXHOLDTIME

Bei einer SNMP-GET-Anforderung ruft der CMX-Agent die erforderlichen Systemfunktionen, z.B. *bstv* oder *cmxinfo*, auf, um die Informationen der CMX-MIB zu ermitteln. Er hinterlegt diese Daten, z.B. die *cmxCcTable*, in einem internen Speicher. Ab diesem Zeitpunkt läuft der Timer *MAXHOLDTIME*. Bis zum Ablauf des Timers wird die Tabelle im internen Speicher gehalten. Informationen, die innerhalb dieses Zeitintervalls zu

derselben CMX-MIB-Tabelle angefordert werden, bedient der CMX-Agent aus dem internen Speicher, ohne eine Systemfunktion aufzurufen. Nach Ablauf des Timers wird bei einer weiteren GET-Anforderung wieder die entsprechende Systemfunktion aufgerufen, die CMX-MIB-Tabelle im internen Speicher neu aufgebaut und der Timer gestartet. Für einige Tabellen der CMX-MIB ist dieser Timer-Algorithmus realisiert. *MAXHOLDTIME* bestimmt die Laufzeit dieser Timer.

Durch diesen Mechanismus erzielt der CMX-Agent eine bessere Performance. Das Auslesen einer CMX-MIB-Tabelle mit einer Folge von GETNEXT-Anforderungen führt somit in der Regel nur zu einem einmaligen Aufruf der entsprechenden Systemfunktion.

Wertebereich in Sekunden: 1 .. 10 ...

Die Laufzeit des Timers beträgt in der Voreinstellung 10 Sekunden. Wählen Sie einen Wert kleiner als der Default-Wert, kann sich die Performance des CMX-Agenten verschlechtern. Wählen Sie einen Wert größer als der Default-Wert, berücksichtigen Sie, dass alle während der Laufzeit des Timers eintretenden Veränderungen in der entsprechenden Tabelle, z.B. Statusänderungen oder auch neue Objektinstanzen, in der CMX-MIB bei einer GET-Anforderung noch nicht sichtbar werden.

CFPOLLTIME

Beim Hochfahren liest der CMX-Agent die spezifischen Konfigurationsdateien der Subnetz-Profile auf den Communication Controllern aus, um die CMX-MIB-Tabellen *cmxIfTable* und *cmxX25PortTable* aufzubauen. Aus Performancegründen werden diese Konfigurationsdateien nur dann sofort in den internen Tabellen (Tabellen im internen Speicher) aktualisiert, wenn die geänderte Konfigurationsdatei auch ihren Namen ändert. Bei gleichbleibendem Namen werden Änderungen in einer Konfigurationsdatei während der Laufzeit des CMX-Agenten jeweils nur nach Ablauf des durch *CFPOLLTIME* spezifizierten Zeitintervalls berücksichtigt. Selbstverständlich wird der Zustand eines Subnetz-Anschlusses in der *cmxIfTable* ständig aktuell gehalten.

Erwarten Sie solch „kritische“ Änderungen in diesen Konfigurationsdateien der Subnetz-Profile, die dem CMX-Agenten sofort mitgeteilt werden müssten, so können Sie mit dem Timer *CFPOLLTIME* einen häufigeren zyklischen Update der internen Tabellen veranlassen. Aus Performancegründen ist eine „lange“ Laufzeit des Timers, z.B. 3600 Sekunden, empfehlenswert.

0:
kein zyklischer Update der internen Tabellen

60 ... 3600 ...:
zyklischer Update der internen Tabellen (Wert in Sekunden)

CCPOLLTIME
Mit dem Parameter *CCPOLLTIME* können Sie wählen, ob die CC-spezifischen Trap-Nachrichten *cmxCcUp* und *cmxCcDown* gesendet werden sollen. Sollen Zustandsübergänge eines Communication Controllers der Management-Station mitgeteilt werden, müssen Sie mit dem Timer *CCPOLLTIME* eine zyklische Abfrage des CC-Status veranlassen. Nach Ablauf des Timers wird jeweils die Systemfunktion *cmxinfo* aufgerufen. Aus Performancegründen empfiehlt sich daher ein Wert im Bereich von 60 .. 300 s.

0:
kein Senden von CC-Trap-Nachrichten

Wenn Sie mit dem Wert null das Senden von CC-Trap-Nachrichten ausschalten, so sollten Sie für die Überwachung der Subnetz-Anschlüsse einen kleineren Timer-Wert für *IFPOLLTIME* einstellen (siehe unten).

60 ...180 ...:
Senden von CC-Trap-Nachrichten (Wert in Sekunden)

Für den EMANATE Master Agenten müssen Sie konfigurieren, welche Management-Stationen Trap-Nachrichten erhalten sollen.

IFPOLLTIME
Mit dem Parameter *IFPOLLTIME* können Sie wählen, ob die SNMP-Trap-Nachricht *linkUp* bzw. *linkDown* bei Zustandsänderung eines Subnetz-Anschlusses gesendet werden soll.

Wenn Sie das Senden von CC-Trap-Nachrichten ausgeschaltet haben (Parameter *CCPOLLTIME=0*, siehe oben), so sollten Sie für den Parameter *IFPOLLTIME* einen kleineren Wert wählen (z.B. 180 Sekunden), da sonst der Ausfall eines Communication Controllers zu spät erkannt wird.

0:
kein Senden von Trap-Nachrichten *linkUp* und *linkDown*

1... 1800 ...:
Senden von Trap-Nachrichten *linkUp* und *linkDown* (Wert in Sekunden)

Im Normalfall wird der CMX-Agent durch News-Meldungen der Communication Controller über Zustandsänderungen von Subnetz-Anschlüssen direkt informiert und kann dann sofort eine entsprechende SNMP-Trap-Nachricht *linkUp* bzw. *linkDown* senden. Um den Verlust von News-Meldungen in speziellen Fehlersituationen zu vermeiden, wertet der CMX-Agent zyklisch auch die News-Dateien */var/opt/SMAWcmx/tmp/cc_NEWSFILE_0/1* aus. Dieses Zeitintervall wird mit dem Parameter *IFPOLLTIME* eingestellt. Aus Performancegründen - und da im Normalfall die News wie oben beschrieben sofort erkannt werden - ist eine „lange“ Laufzeit, z.B. 1800 Sekunden, empfehlenswert.

Für den EMANATE Master Agenten müssen Sie konfigurieren, welche Management-Stationen Trap-Nachrichten erhalten sollen.

Die restlichen acht Parameter *MAX** beziehen sich auf die Größe der internen Tabellen des CMX-Agenten und sind durch den jeweiligen Kommentar in der Datei beschrieben. Bei Auslieferung des Produkts sind diese Parameter ausreichend dimensioniert. Die Parameterwerte sollten nicht verändert werden.

11.3.2.2 Datei AgentTraces

Die Datei *AgentTraces.rc* wird im Verzeichnis */opt/SMAW/SMAWcmx/lib/cmxsnmp* mit den unten aufgeführten Voreinstellungen ausgeliefert.

```
# COPYRIGHT (C) Fujitsu Siemens Computers GmbH 2000
#               All Rights Reserved
#
#        SNMP EMANATE Subagent for CMX
#
#        Tracepointfile
#
# Comments must be marked by # in the first line position.
#
# Delete the # to activate the appropriate trace.
#
#################################################################
#
# TRAPS  # trace trap generation
# POLLS  # trace all cyclic polling algorithms
# CFUPD  # trace evaluation of CFs and update of interface table
# HOLDT  # trace start and expiration of holding timers
# GETARG # trace all SNMP GET request
# SYSCALL # trace all calls of system functions and scripts
# ADDR    # trace address evaluation in TSAPs and TCEPs
# NEWS    # trace news reception from bstvd
```

Die Bedeutung der einzelnen Tracepunkte ist in den Kommentarzeilen der Datei beschrieben.

Tracepunkte, die am Zeilenanfang kein „#“-Zeichen besitzen, sind eingeschaltet. Tracepunkte, die ausgeschaltet sind, werden am Zeilenanfang mit einem „#“-Zeichen gekennzeichnet. Defaultmäßig sind alle Tracepunkte ausgeschaltet.

Die Trace-Informationen werden zuerst in die Datei */var/opt/SMAWcmx/tmp/cmxsnmp.1.trc*, nach deren Überlauf dann in die Datei */var/opt/SMAWcmx/tmp/cmxsnmp.2.trc* geschrieben und umgekehrt. Die Größe der Trace-Dateien ist durch den Parameter *MAXTRACE* in der Datei *AgentParams.rc* vorgegeben.

Beim Hochfahren bzw. einem Restart des CMX-Agenten werden einige Informationen wie z.B. die aktuellen Parameterwerte und die eingeschalteten Tracepunkte in die erste Trace-Datei geschrieben.

Traceausgaben werden für alle eingeschalteten Tracepunkte geschrieben. Bei auftretenden Systemfehlern erfolgt immer eine entsprechende Traceausgabe unabhängig davon, ob Tracepunkte eingeschaltet sind.

Das Layout und der Umfang der Trace-Einträge können sich in späteren Versionen ändern. Es gibt keine Garantie für das Layout dieser Dateien.

11.3.2.3 Rekonfiguration

Wenn Sie den CMX-Agenten mit der oben genannten Standardkonfiguration betreiben wollen, können Sie den Rest dieses Unterkapitels überlesen.

Wenn Sie den CMX-Agenten mit einer anderen Konfiguration betreiben wollen, müssen Sie die Dateien *AgentParams.rc* und *AgentTraces.rc* Ihren Wünschen entsprechend anpassen.

Sie können Parameterwerte in den beiden Dateien *AgentParams.rc* und *AgentTraces.rc* entweder mit dem Kommando *cmxsnmpadm* oder mit einem beliebigen Editor ändern.

Ändern mit dem Kommando cmxsnmpadm

Mit dem Kommando *cmxsnmpadm* können Sie dialogorientiert die gewünschten Änderungen in einer oder in beiden Dateien durchführen.

Nach dem Aufruf von *cmxsnmpadm* werden Sie zu jedem Parameterwert gefragt, ob Sie diesen Wert ändern wollen und wenn ja, werden Sie in der nächsten Zeile um die Eingabe des neuen Wertes gebeten; z.B.:

```
MAXTRACE 100 # maximum length of trace files (in kilo bytes)
change parametervalue? y | n | q (default: n) y
new value for MAXTRACE : 60
```

Nacheinander werden die Parameter von der Datei *AgentParams.rc* und anschließend von der Datei *AgentTraces.rc* abgefragt. Die Reihenfolge der Parameterabfrage entspricht der Reihenfolge in der entsprechenden Datei.

Mit *q* (quit) können Sie die Parameterabfrage abbrechen. Die bis dahin eingegebenen Änderungen sind in jedem Fall wirksam.

Am Ende des Kommandos werden Sie gefragt, ob die Änderung noch im laufenden Betrieb aktiviert werden soll oder erst nach dem nächsten Hochfahren des Systems:

```
activate updates by restarting CMX Subagent? y | n (default: y)
```

Sie haben die folgenden drei Möglichkeiten, das Kommando aufzurufen.

Format 1**: cmxsnmpadm**

Format 2: **cmxsnmpadm␣-p**[␣parameter]

Format 3: **cmxsnmpadm␣-t**[␣tracepoint]

Format 1: Änderungen in beiden Dateien durchführen

Rufen Sie das Kommando *cmxsnmpadm* ohne Parameter auf, werden nacheinander die Parameter der Datei *AgentParams.rc* und danach der Datei *AgentTraces.rc* zur Änderung angeboten.

Format 2: Änderungen in Datei AgentParams.rc durchführen

-p

Wenn Sie nach dem Schalter *-p* keinen Parameter der Datei *AgentParams.rc* explizit angeben, so werden Sie zu jedem Parameter in der Datei nach Ihrem Änderungswunsch gefragt.

parameter

Sie geben den Namen des Parameters an, den Sie in der Datei *AgentParams.rc* ändern wollen. Sie erhalten dann nur die Änderungsabfrage für diesen Parameter.

parameter = {SETENABLE | MAXTRACE | MAXHOLDTIME | CFPOLLTIME| CCPOLLTIME | IFPOLLTIME | TIMEOUT | TIMEOUTLOG | MAXCC | MAXTSP | MAXTSPACC | MAXIF | MAXX25 | MAXTSPSET | MAXTSEL | MAXTCEP}

Eine Änderung der Parameter *MAX** im laufenden Betrieb bewirkt einen Restart des CMX-Agenten.

Beispiel

Sie wollen die Werte für die Parameter *MAXTRACE* und *CCPOLLTIME* ändern und die Änderungen im laufenden Betrieb aktivieren:

```
cmxsnmpadm -p

##### process parameters in /opt/SMAW/SMAWcmx/lib/cmxsnmp/AgentParams.rc ####

SETENABLE 0      # SNMP SET Operations allowed (1) or not (0)
change parametervalue? y | n | q (default: n) n

MAXTRACE 100     # maximum length of trace files (in kilo bytes)
change parametervalue? y | n | q (default: n) y
new value for MAXTRACE : 60

MAXHOLDTIME 10   # holding timer until update of internal tables
change parametervalue? y | n | q (default: n) n

CFPOLLTIME 3600  #if not 0: poll for updates in CCP config files
change parametervalue? y | n | q (default: n) n

CCPOLLTIME 180   # if not 0: poll for CC state an send CC-TRAP
change parametervalue? y | n | q (default: n) y
new value for CCPOLLTIME : 100

IFPOLLTIME 1800  # if not 0: poll for IF state an send IF-TRAP
change parametervalue? y | n | q (default: n) q

activate updates by restarting CMX Subagent? y| n (default: y) n
```

Format 3: Änderungen in Datei AgentTraces.rc durchführen

-t

Wenn Sie nach dem Schalter *-t* keinen Parameter der Datei *AgentTraces.rc* explizit angeben, so werden Sie zu jedem Parameter in der Datei nach Ihrem Änderungswunsch gefragt.

tracepoint

Sie geben den Namen des Tracepunkts an, den Sie in der Datei *AgentTraces.rc* aktivieren bzw. deaktivieren wollen. Sie erhalten dann nur die Änderungsabfrage für diesen Tracepunkt.

traces = { TRAPS | POLLS | CFUPD | HOLDT | GETARG | SYSCALL | ADDR | NEWS }

Beispiel

Sie wollen den Tracepunkt *TRAPS* aktivieren. Diese Änderung soll erst nach dem nächsten Hochfahren des CMX-Agenten wirksam werden:

cmxsnmpadm -t TRAPS

```
####
##### process parameters in /opt/lib/cmxsnmp/AgentTraces.rc ####
####

# TRAPS     # trace trap generation
trace is OFF, do you want to switch ON? y| n | q (default: n)  y

activate updates by restarting CMX Subagent? y| n (default: y) n
```

Ende-Status

0
: *cmxsnmpadm* wurde erfolgreich ausgeführt.

≠0
: Bei der Ausführung von *cmxsnmpadm* ist ein Fehler aufgetreten.

Dateien

/opt/SMAW/SMAWcmx/bin/cmxsnmpadm
: Kommando *cmxsnmpadm*

/opt/SMAW/SMAWcmx/lib/cmxsnmp/AgentParams.rc
: Datei, die die Parameter für die lokale Administration enthält.

/opt/SMAW/SMAWcmx/lib/cmxsnmp/AgentTraces.rc
: Datei, die die Tracepunkte enthält.

Ändern per Editor

1. Führen Sie die gewünschten Änderungen in der entsprechenden Datei (*AgentParams.rc* bzw. *AgentTraces.rc*) mit einem beliebigen Editor durch.
2. Sollen die Änderungen bereits im laufenden Betrieb gültig werden, rufen Sie *cmxsnmp restart* auf. Die Änderungen sind anschließend für den laufenden CMX-Subagenten gültig. Ansonsten werden die Änderungen automatisch beim nächsten Start des CMX-Agenten gültig.

Eine Änderung der Parameter *MAX** im laufenden Betrieb bewirkt einen Restart des CMX-Subagenten.

12 Ablauf von TLI-Anwendungen

CMX erlaubt es TLI-Anwendungen, die von den Produktgruppen CCP-WAN und CCP-ISDN bereitgestellten Transportsysteme zu nutzen.

TLI ist eine Programmschnittstelle im Solaris-Betriebssystem und ist mit XTI (X/Open Transport Interface) verwandt.

TLI-Anwendungen können über die Transportsysteme der CCPs ablaufen, wenn die folgenden Bedingungen erfüllt sind:

1. Die TLI-Anwendung muss so konzipiert sein, dass sie grundätzlich über ein ISO-Transportsystem ablaufen kann. Sie muss beispielsweise vor einem Verbindungsabbau sicherstellen, dass alle gesendeten Daten vom Partner empfangen wurden, da ISO-Transportsysteme keinen 'orderly release' kennen.
2. Die TLI-Anwendung muss den Netzwerkauswahl-Mechanismus von System V Release 4 (über */etc/netconfig*) und die NETDIR-Funktionen zur Namens-Adressabbildung verwenden. Sie darf keine Annahmen über den Namen des Transportdienstes oder das Adressformat des Transportsystems machen.

Die Konfigurationsdatei /etc/netconfig

Mit dem Solaris-Basissystem wird eine Netzkonfigurationsdatei ausgeliefert, die TLI-Anwendungen hilft, das richtige Transportsystem auszuwählen. Eine Beschreibung dieser Datei und des darauf aufbauenden Auswahlmechanismus finden Sie im Handbuch „XTI, X/Open Transport Interface“ [2]. Der Systemverwalter muss die Datei um die folgenden Einträge erweitern:

Für ISO-Transportdienste:

Netzwerk-ID: cx-osicots

Semantik: tpi_cots

Flag: –

Protokollfamilie: osi

Protokollname: –

Netzwerk-Gerätedatei: /dev/osicots3

Nachschlagebibliotheken: /usr/lib/tnsxaddr.so

Für NEA-Transportdienste (ohne NEABX):

Netzwerk-ID: cx-nea

Semantik: tpi_cots

Flag: –

Protokollfamilie: nea

Protokollname: –

Netzwerk-Gerätedatei: /dev/neat3

Nachschlagebibliotheken: /usr/lib/tnsxaddr.so

Für ISO- und NEA-Transportdienste (ohne NEABX) gemeinsam:

Netzwerk-ID: cx-msg

Semantik: tpi_cots

Flag: –

Protokollfamilie: msg

Protokollname: –

Netzwerk-Gerätedatei: /dev/msg3

Nachschlagebibliotheken: /usr/lib/tnsxaddr.so

Den Zusammenhang zwischen den genannten Transportdiensten und CCP-Profilen entnehmen Sie der folgenden Tabelle:

Transportdienst	**CCP-Profil**
ISO	CCP-ISDN-CONS CCP-WAN-CONS
NEA	CCP-ISDN-NEA CCP-ISDN-NX25 CCP-WAN-NEA CCP-WAN-NX25

Tabelle 30: Zuordnung Transportdienst und Profil

Die Namens-Adressabbildung

Wie ICMX-Anwendungen benötigen auch TLI-Anwendungen Zugang zu einem Name Service, der symbolische Namen auf Adressen abbildet und umgekehrt. TLI-Anwendungen, die die NETDIR-Programmschnittstelle benutzen, verwenden zweistufige symbolische Namen, die den Rechner („Host“) und die Anwendung im Rechner („Service“) kennzeichnen. Nach der oben beschriebenen Erweiterung der Datei */etc/netconfig* greifen die TLI-Anwendungen über die NETDIR-Programmschnittstelle auf den TNSX zu. Die TNSX-Einträge erfolgen in derselben Weise wie für ICMX-Anwendungen. Lediglich bei der Namensgebung für die GLOBALEN NAMEN muss der Systemverwalter folgende Regeln beachten:

1. GLOBALE NAMEN zur Eigenschaft LOKALER NAME („eigene Adresse“)

   Die Namensbestandteile NP1, NP2 und NP3 des GLOBALEN NAMENS müssen leer bleiben. Im Namensbestandteil NP4 müssen Sie den eigenen Rechnernamen angeben, wie ihn das Kommando *uname -n* zurückliefert. NP5 ist beliebig (jedoch nicht leer) und identifiziert die TLI-Anwendung innerhalb des lokalen Endsystems.

2. GLOBALE NAMEN zur Eigenschaft TRANSPORTADRESSE („ferne Adresse“)

   Die Namensbestandteile NP1, NP2, NP3 und NP4 des GLOBALEN NAMENS sind beliebig (jedoch mindestens einer der Namensteile ist nicht leer) und identifizieren aus Sicht der TLI-Anwendung das ferne Endsystem. NP5 ist beliebig (jedoch nicht leer) und identifiziert die TS-Anwendung innerhalb des fernen Systems.

Fachwörter

Anwendung

Eine Anwendung ist ein System von Programmen, das ein bestimmtes Dienstangebot eines EDV-Systems anwendet, um einem menschlichen oder maschinellen Nutzer eine höherwertige Dienstleistung anzubieten. Kommunikationsanwendungen sind Anwendungen, die die Kommunikationsfunktionen eines EDV-Systems nutzen, um unter Nutzung eines Netzes systemübergreifende Dienstleistungen zu erbringen.

Den meisten Anwendungen wird ein Präfix zur Kennzeichnung des untergelagerten Dienstangebots vorangestellt (CMX-Anwendung, UTM-Anwendung, DCAM-Anwendung, Motif-Anwendung und Windows-Anwendung, etc.). Beispiele für Kommunikationsanwendungen sind Filetransfer, Terminalemulation, Electronic Mail, World Wide Web Browser und Server, Transaktionssysteme wie UTM, allgemein alle Anwendungen nach dem Client/Server-Prinzip.

API (Application Program Interface)

APIs sind Programmschnittstellen, die die Funktionen eines Programmsystems zur Verfügung stellen. Als Programmierer nutzen Sie die APIs bei der Programmierung von Anwendungen. APIs bieten Funktionen zum Verbindungsmanagement, zum Datenaustausch und zur Abbildung von Namen in Adressen. APIs im CMX-Umfeld sind ICMX, XTI,TLI und NLI.

CC (Communications Controller)

Ein CC ist eine Baugruppe zum Anschluss eines Solaris-Rechners an ein Netz. Sie benötigen einen CC, um Ihren Rechner physisch an ein Subnetz anzuschließen, es sei denn, der Anschluss ist auf einer anderen Baugruppe, z. B. der Mutterplatine, mit integriert (onboard-Anschluss).

Um einen logischen Anschluss zum Netz zu erhalten, werden CCs mit dem zugehörigen Subnetzprofil geladen. Das Subnetzprofil ist Bestandteil des *CCPs*. Beispiele für ladbare CCs zum Anschluss an X.25- und Telefonnetze und ISDN sind PWXV, PWS0 und PWS2.

CCP (Communication Control Program)

Ein CCP ist ein Programmsystem, das zusammen mit einem oder mehreren *CCs* den logischen Zugang eines Solaris-Rechners an ein *Netz* leistet. Ein CCP implementiert die vier unteren Schichten (Transportsystem) des OSI-Referenzmodells zur Datenkommunikation.

Ein CCP besteht aus *Subnetz-Profilen* und *Transport Service Providern*.

CLI (Command Line Interface)

CLI ist die Summe der Kommandos für *OA&M* von *CMX* und den *CCPs*. Als Administrator können Sie Initialisierungs-, Überwachungs-, Steuer- und Wartungsfunktionen von *CMX*, den *CCPs* und den *Communication Services* über die Kommandozeile des UNIX-Systems vornehmen (die Kommandos cmxinfo, cmxm(onitor), tnsxcom, bstv, ccpgen, etc.).

CLIs bieten ein breites Spektrum an Optionen mit zum Teil komplexer Syntax. Die Benutzeroberfläche *CMXCUI* ermöglichen eine einfache, interaktive Handhabung der gewünschten Routinemaßnahmen.

CMX (Communications Manager for UNIX Systems)

CMX erbringt Kommunikationsdienste zur Nutzung von CMX-Anwendungen und *Communication Services* im Netz und ermöglicht die Programmierung von CMX-Anwendungen. CMX vereinheitlicht die Dienste unterschiedlicher Netze und ermöglicht damit die Nutzung derselben CMX-Anwendung unabhängig vom unterliegenden Netz. Als Laufzeitsystem vermittelt CMX zwischen aktuellem Netzangebot und CMX-Anwendungen und bietet dem Netzadministrator einheitliche Funktionen für *OA&M* (Operation, Administration, Maintenance) von *CCPs* und *CCs*. Als Entwicklungssystem bietet CMX Schnittstellen (APIs) und Verfahren zur Programmierung von netzunabhängigen CMX-Anwendungen.

CMX-Anwendungen

CMX-Anwendungen sind Anwendungen, die die Dienste von CMX nutzen. CMX-Anwendungen haben im Netz eine Adresse, die *TRANSPORT-ADRESSE*. Sie identifizieren sich untereinander durch symbolische Namen, dem *GLOBALEN NAMEN* einer Anwendung.

CMXCUI (Character User Interface)

Das CMXCUI ist eine zeichenorientierte Benutzeroberfläche zu den Funktionen des *OA&M* von *CMX* und den *CCPs*. Als Administrator können Sie damit die komfortable Bedienung des *OA&M* über Menüs und Formulare nutzen. Das CMXCUI nutzt FMLI und setzt auf dem *CLI* auf.

Communication Services

Communication Services dienen der Verknüpfung von heterogenen Netzen verschiedener Architektur bzw. unterschiedlicher Technologie. Durch Einsatz von Communication Services können beispielsweise unterschiedliche LAN-WAN-Kopplungen realisiert werden, wobei der entsprechende Communication Service eine Software-Komponente z. B. auf einem Server ist.

FSS (Forwarding Support Service)

Der FSS ist eine Komponente von *CMX*, die die korrekte Adressierung von Anwendungen im Netz und die Wahl einer Route durch das *Netz* und seine Subnetze unterstützt. Sie können als Administrator den FSS dazu mit den netzspezifischen Angaben konfigurieren, die Sie für Ihr Netz vorgesehen bzw. mit dem Netzbetreiber abgestimmt haben.

Eine wichtige Information im FSS ist die Abbildung einer Netzadresse, z. B. der NEA-Adresse "47/11", auf eine Subnetzadresse des fernen Rechners, z. B. die X.25-Adresse "8963647658". Eine weitere wichtige Information ist die Definition einer Route mit ihrem lokalen Ausgangspunkt und den verschiedenen Stationen durch die Subnetze zum fernen Rechner. Der lokale Ausgangspunkt einer Route ist eine *Subnetz-Id*, die einen bestimmten von mehreren vorhandenen Subnetz-Anschlüssen identifiziert.

GLOBALER NAME einer Anwendung

Jede CMX-Anwendung identifiziert sich selbst und ihre Kommunikationspartner im Netz durch symbolische, hierarchische GLOBALE NAMEN. Ein GLOBALER NAME besteht aus bis zu fünf Namensteilen (NP[1- 5]), die Sie zur Definition der Anwendung (NP5), des Rechners (NP4) und (bis zu drei) administrativer Domänen (NP[3-1]) verwenden können.

Beispiel: Der GLOBALE NAME "IhreAnwendung.D018S065.mch-p.sni.de" bedeutet: "IhreAnwendung" residiert im Host "D018S065" in der Domäne "mch-p.sni.de".

Bei der Wahl eines GLOBALEN NAMENS müssen Sie als Administrator die Vorgaben und Empfehlungen der speziellen Anwendung beachten.

Als Administrator können Sie dem GLOBALEN NAMEN einer Anwendung 1:1 eine *TRANSPORTADRESSE* oder einen *LOKALEN NAMEN* der Anwendung zuordnen. Als Programmierer können Sie die von CMX

erwartete TRANSPORTADRESSE oder den LOKALEN NAMEN mit Hilfe der Funktionsaufrufe des *Transport Name Service* (TNS) aus dem GLOBALEN NAMEN gewinnen.

KOGS (Konfigurationsorientierte Generatorsprache)

KOGS ist die konfigurationsorientierte Generatorsprache, mit der die physischen und logischen Eigenschaften der Subnetz-Anschlüsse eines Rechners in einer Textdatei beschrieben werden. Sprachelemente der KOGS sind Makros, Operanden und Operandenwerte. Im Normalfall definiert der System- bzw. Netzverwalter die spezifischen Eigenschaften seiner Subnetz-Anschlüsse mit dem *CMXCUI*. Nur in Ausnahmefällen verwendet er dazu die KOGS.

LOKALER NAME einer Anwendung

Eine CMX-Anwendung meldet sich in ihrem lokalen Rechner mit dem LOKALEN NAMEN bei CMX zur Kommunikation an. Der LOKALE NAME besteht aus einem oder mehreren *T-Selektoren*, die jeweils das Transportsystem bezeichnen, über das die CMX-Anwendung kommunizieren soll. Als Administrator können Sie mit dem LOKALEN NAMEN die Kommunikation einer CMX-Anwendung über bestimmte Transportsysteme ermöglichen oder ausschließen und etwaige Anforderungen der CMX-Anwendung nach bestimmten T-Selektor-Werten, z. B. beim Filetransfer, erfüllen.

Beispiel: Eine Anwendung soll den T-Selektor "cmxappl" (in Kleinbuchstaben!) für die Kommunikation über das TCP/IP- RFC1006 Transportsystem und den T-Selektor "$CMXAPPL" (in Großbuchstaben!) für die Kommunikation über das NEA-Transportsystem verwenden.

Den LOKALEN NAMEN einer Anwendung können Sie als Administrator in CMX mit dem User Interface *CMXCUI* dem *GLOBALEN NAMEN* der Anwendung zuordnen. Als Programmierer können Sie den von CMX erwarteten LOKALEN NAMEN mit Hilfe der Funktionsaufrufe des *Transport Name Service* (TNS) aus dem GLOBALEN NAMEN gewinnen.

Netz

Ein Netz ist ein Verbund zusammenwirkender Übertragungskomponenten (Leitungen, Vermittlungsknoten, Verfahren) mit einheitlich definierten Diensten, Protokollen und Zugangseinrichtungen für EDV-Systeme. Ein Netz verbindet Rechner zur Nutzung systemübergreifender Anwendungen miteinander. Das Netz eines Netzbetreibers kann sofort für Anwen-

dungen oder zur Definition darauf aufbauender, überlagerter, privater Netzstrukturen genutzt werden. Im UNIX-Umfeld sind folgende Netze relevant: das Internet, SNA-, TRANSDATA- und OSI-Netze.

Ein Netz kann aus einem oder mehreren *Subnetzen* bestehen, die über das homogene Ende- zu-Ende-Protokoll des Netzes verknüpft sind. Die oben genannten Beispielnetze können Überlagerungen aus öffentlichen oder privaten Subnetzen wie dem X.25-Netz, dem Telefon- oder Datennetz, dem ISDN oder ATM-Netz und verschiedenen privaten, lokalen Netzen basierend auf Ethernet, Token Ring und FDDI sein.

Netzadresse

Jeder Rechner in einem *Netz* ist durch seine Netzadresse eindeutig identifiziert. Ein Rechner kann in unterschiedliche Netze eingebunden sein und hat dann für jedes dieser Netze eine spezifische Netzadresse.

Im Internet heißen die Netzadressen IP-Adressen. Sie sind einem sogenannten IP-Interface eindeutig zugeordnet. Ein Rechner kann mehrere IP-Interfaces besitzen. An einem IP-Interface können nur die IP-Version 4, nur die IP-Version 6 oder beide IP-Versionen 4 und 6 gleichzeitig unterstützt werden. Für jede unterstützte IP-Version ist dem IP-Interface eine entsprechende IP-Adresse zugeordnet (Beispiel einer IPv4-Adresse: 129.144.89.171, Beispiel einer IPv6-Adresse: fe80::280:17ff:fe28:7b08).

Im NEA-Netz hat ein Rechner eine NEA-Netzadresse, die sich zusammensetzt aus Rechner-/Regions-Nummer (z. B. 124/213).

Die OSI-Netzadresse (NSAP-Adresse) setzt sich zusammen aus dem Initial Domain Part (IDP) und dem Domain Specific Part (DSP) und hat das Format: IDP+DSP (z. B. 470058+0144458100007391100308001411961301).

OA&M (Operation, Administration and Maintenance)

Das OA&M ist die Summe der Funktionen zur Inbetriebnahme, Betriebsüberwachung und -steuerung, Konfigurierung und Wartung der CMX- und CCP-Komponenten. Wesentliche OA&M-Tätigkeiten im CMX-Umfeld sind das Laden und Überwachen eines *CC*, das Konfigurieren von Laufzeitparametern des *CCPs* und das Schalten von Traces. Die einfache, interaktive Handhabung von Routinemaßnahmen im OA&M ermöglichen Ihnen das *CMXCUI*. Für spezielle, außergewöhnliche Administrationsaufgaben können Sie auch das *CLI* verwenden.

Route

Eine Route beschreibt den Weg vom lokalen Rechner zu einem fernen Rechner innerhalb eines *Subnetzes*. Liegt der ferne Rechner in einem anderen Subnetz als der lokale Rechner, dann beschreibt die Route den Weg vom lokalen Rechner bis zum Netzübergang ("Next Hop"), wo dann das weitere Routing zum fernen Rechner erfolgt. Eine Route ist durch ihre Endpunkte definiert: die *Subnetz-ID* im lokalen Rechner und die *Subnetz-Adresse* des fernen Rechners, wenn der ferne Rechner im selben Subnetz liegt, oder die Subnetz-Adresse des "Next Hop", wenn der ferne Rechner nicht im selben Subnetz liegt. Hat ein Rechner mehrere Subnetz-Adressen, so kann er über mehrere Routen erreicht werden.

Subnetz

Ein Subnetz ist ein technisch oder administrativ homogener Teil eines *Netzes*. Subnetze sind u.a. das X.25-Netz, das Telefon- oder Datennetz, das ISDN oder ATM-Netz und verschiedene private, lokale Netze basierend auf Ethernet, Token Ring und FDDI. Der Zugang zu einem Subnetz kann über einen oder mehrere Subnetz-Anschlüsse erfolgen. Ein Subnetz-Anschluss wird durch seine *Subnetz-Adresse* identifiziert.

Subnetz-Adresse

Die Subnetz-Adresse beschreibt eindeutig einen Subnetz-Anschluss, der den Zugang zum *Subnetz* ermöglicht. Die Subnetz-Adresse ist beispielsweise eine ISDN-Rufnummer, eine DTE-Adresse oder eine Ethernet-Adresse.

Subnetz-ID

Die Subnetz-ID, auch SNID genannt, benennt eine Gruppe gleichartiger Subnetz-Anschlüsse, die in dasselbe *Subnetz* führen. Die Subnetz-ID gibt die Art des Subnetzes an und identifiziert, um welche Gruppe von Zugängen zu diesem Subnetz es sich handelt. Eine Subnetz-ID steht beispielsweise für zwei ISDN-Anschlüsse oder für mehrere X.25-Anschlüsse in einem Subnetz.

Subnetz-Profil

Das Subnetz-Profil bezeichnet die Komponenten eines *CCPs*, die einen *Communication Controller* steuern.

SWK (Softwarekonfiguration)

Eine SWK ist eine definierte Kombination von Versionen von Software-Produkten, die zusammen ein abgegrenztes und verifiziertes Leistungsspektrum abdecken.

Eine SWK aus *CMX* und *CCP*-Produktversionen garantiert deren definiertes Zusammenwirken. Dies wird durch Qualitätssicherungsmaßnahmen gewährleistet. Bei einer Mischung von CMX- und CCP-Produktversionen, die nicht als SWK oder nicht ausdrücklich als verträglich definiert sind, können unerwartete Störungen und Ausfallsituationen mit nicht definierten Folgen auftreten.

TNS (Transport Name Service)

Der TNS ist eine Komponente von *CMX*, die die korrekte Abbildung der *GLOBALEN NAMEN* von CMX-Anwendungen im Netz in *TRANSPORTADRESSEN* und *LOKALE NAMEN* unterstützt. Als Administrator konfigurieren Sie die von Ihnen gewählte Zuordnung von GLOBALER NAME zu TRANSPORTADRESSE für ferne Anwendungen sowie die Zuordnung von GLOBALER NAME zu LOKALER NAME für lokale Anwendungen. Als Programmierer von Anwendungen können sie diese Abbildungen über ein *API* nutzen und damit allein mit den *GLOBALEN NAMEN* von Anwendungen ohne Bewertung der Abbilder arbeiten.

Der TNS bietet die netzweite Identifikation von Anwendungen durch logische GLOBALE NAMEN und deren Abbildung in eine entsprechende *Netzadresse*. Damit können Sie die Anwendungen vom Wissen um ihre Netzadressen entkoppeln. Zusammen mit dem *FSS* bietet der TNS die vollständige Abbildung des logischen Namens in eine konkrete *Subnetz-Adresse* und eine *Route* durch die verschiedenen Subnetze des Netzes.

TRANSPORTADRESSE einer Anwendung

Eine rufende CMX-Anwendung übergibt die TRANSPORTADRESSE eines gerufenen Kommunikationspartners beim Aufbau der Kommunikation an *CMX*. CMX verwendet die TRANSPORTADRESSE, um den Kommunikationspartner im Netz zu lokalisieren und eine *Route* durch das Netz zu bestimmen. Die TRANSPORTADRESSE hängt im allgemeinen von der logischen und physischen Struktur des Netzes (und seiner Subnetze) ab. Sie enthält die für Ihr Netz spezifischen Vorgaben Ihrer/Ihres Netzbetreiber(s).

Als Administrator können Sie unabhängig von der Anwendung die TRANSPORTADRESSE und damit die Kommunikationswege beeinflussen. Bestandteile einer TRANSPORTADRESSE sind: eine Netzadresse zur eindeutigen Bestimmung des fernen Rechners, auf dem die Anwendung residiert, der Typ des *Transportsystems*, über das die ferne Anwendung erreicht werden kann, und der *T-Selektor*, der die ferne Anwendung im fernen Rechner identifiziert.

Als Administrator können Sie dem *GLOBALEN NAMEN* einer Anwendung 1:1 eine TRANSPORTADRESSE der Anwendung zuordnen.

Als Programmierer können Sie die von CMX erwartete TRANSPORTADRESSE mit Hilfe der Funktionsaufrufe des *Transport Name Service* (TNS) aus dem GLOBALEN NAMEN gewinnen.

Transportsystem

Das Transportsystem bezeichnet die unteren vier Schichten des *OSI-Referenzmodells*. Ein *CCP* implementiert die vier Schichten des Transportsystems. Das Transportsystem sorgt für den gesicherten Datenaustausch zwischen Rechnern, deren *Anwendungen* miteinander kommunizieren, und zwar unabhängig von den darunterliegenden Netzstrukturen. Das Transportsystem verwendet dazu Protokolle.

T-Selektor

Der T-Selektor identifiziert eine Kommunikationsanwendung innerhalb des Rechners, auf dem die Anwendung abläuft. Der T-Selektor bildet zusammen mit der *Netzadresse* des Rechners die *TRANSPORTADRESSE* einer Anwendung, mit der diese Anwendung innerhalb eines Netzes eindeutig adressiert werden kann. Das Format und der Wertebereich des T-Selektors hängen vom Typ des *Netzes* ab. Im NEA-Netz entspricht der T-Selektor dem Stationsnamen (z. B. T'DSS01').

TSP (Transport Service Provider)

Ein TSP ist eine Komponente eines *CCP* oder von *CMX*, die mit Ausnahme des NTP (Null-Transport) mittels eines Transportprotokolls den OSI-Transportdienst im Netz anbietet. Sie können als Administrator die Nutzung eines bestimmten TSP für die Kommunikation von *Anwendungen* bestimmen. Der RFC1006 ist der TSP in CMX, der zusammen mit TCP/IP im Internet den OSI-Transportdienst bietet. Der NTP (Null-Transport) bietet CMX-Anwendungen den Direktzugriff auf die Netzdienste des X.25-Subnetzes. TP0/2, und NEA sind die TSPs für ein OSI-Umfeld und das TRANSDATA-Netz.

Ein TSP bildet zusammen mit einem *Subnetzprofil* ein *Transportsystem*. Er bietet einen Satz von konfigurierbaren Laufzeit- und Tuningparametern, bewertet die *TRANSPORTADRESSE* und findet eine geeignete Route durch das Netz. Der TSP nutzt dazu Ihre Angaben im *FSS*, soweit erforderlich.

Abkürzungen

ASCII
American Standard Code of Information Interchange

CC
Communication Controller

CCITT
Comité Consultatif International Télégraphique et Téléphonique

CCP
Communication Control Program

CMX
Communications Manager for UNIX Systems

CMXCUI
CMX Character User Interface

DCAM
Data Communication Access Method

EBCDIC
Extended Binary Coded Decimals Interchange Code

ETHN
ETHERNET

ETSDU
Expedited Transport Service Data Unit

FSS
Forwarding Support Service

FSB
Forwarding Support Base

FT
File Transfer

ICMX
Programming Interface CMX

ISDN
Integrated Services Digital Network

ISO
International Organization for Standardization

KD
Konfigurationsdatei

KOGS
Konfigurationsorientierte Generatorsprache

LAN
Local Area Network

MIB
Management Information Base

MT
Multi-Threading, multi-threaded

NEA
Netzwerk-Architektur bei TRANSDATA-Systemen

NLI
Network Layer Interface

NSAP
Network Service Access Point

OSI
Open Systems Interconnection

PDN
Programmsystem für Datenfernverarbeitung und Netzsteuerung

PID
Process Identifier

PSDN
Packet Switched Data Network

PSTN
Public Switched Telephone Network

PVC
Permanent Virtual Circuit

SNA
Systems Network Architecture

SNID
Subnetz-Identifikation

SNPA
Subnet Point of Access

SVC
Switched Virtual Circuit

TCEP
Transport Connection Endpoint

TCP/IP
Transmission Control Protocol/Internet Protocol

TEP
Transport Endpoint

TIDU
Transport Interface Data Unit

TLI
Transport Layer Interface

TNS
Transport Name Service

TPDU
Transport Protocol Data Unit

TPI
Transport Provider Interface

TREF
Transport Reference

TS
Transport Service

TSAP
Transport Service Access Point

TSDU
Transport Service Data Unit

TSP
Transport Service Provider

TSTAT
TEP-Status

WAN
Wide Area Network

XTI
X/OPEN Transport Interface

Literatur

Die Handbücher sind online unter *http://manuals.fujitsu-siemens.com* zu finden oder in gedruckter Form gegen gesondertes Entgelt unter *http://FSC-manual-shop.com* zu bestellen.

[1] **CMX V6.0**
Anwendungen programmieren

Zielgruppe
Programmierer

Inhalt
Das Handbuch beschreibt die Programmierschnittstellen von CMX, d.h. alle Werkzeuge, die Sie benötigen, um selbst TS-Anwendungen zu entwickeln.

[2] **XTI V6.0**
X/Open Transport Interface
User Guide

Zielgruppe
Programmierer von TS-Anwendungen.

Inhalt
Das Handbuch enthält implementierungsabhängige Ergänzungen zu den Funktionsaufrufen von XTI.

[3] **CMX/CCP V6.0** (Solaris)
ISDN-Kommunikation
Benutzerhandbuch

Zielgruppe
Netzverwalter.

Inhalt
Das Handbuch beschreibt die Rechnerkopplung über ISDN (Integrated Services Digital Network).

[4] **CMX/CCP V6.0** (Solaris)
WAN-Kommunikation
Benutzerhandbuch

Zielgruppe
Netzverwalter und Systemadministratoren

Inhalt
Das Handbuch beschreibt die Rechnerkopplung über WAN (Wide Area Network); damit wird Kommunikation im Fernbereich (Wide Area Network, WAN) ermöglicht.

[5] **CMX V6.0** (Solaris)
TCP/IP über WAN/ISDN
Benutzerhandbuch

Zielgruppe
Netzverwalter und Systemadministratoren.

Inhalt
Das Handbuch beschreibt, wie CMX den verbindungslosen IP-Verkehr über das verbindungsorientierte WAN ermöglicht.

[6] **Anschluss an SNA-Netze**
TRANSIT-BAS
Basismanual

Zielgruppe
Solaris-Anwender in SNA-Netzen

Inhalt
Basisbeschreibung der TRANSIT-Produkte

[7] **System Administration Guide, Volume 2**
Solaris 8/9
Systemverwalterhandbuch

Zielgruppe
Solaris-Systemverwalter

Inhalt
Einführung in die Solaris-Systemverwaltung

[8] **openNet Server V3.0** (BS2000/OSD)
IPSec V1.0
Benutzerhandbuch

Zielgruppe
Das Handbuch wendet sich an Netzadministratoren, Entwickler von Netzanwendungen im BS2000/OSD sowie an alle, die sich für Fragen der Internet-Sicherheit, insbesondere im BS2000/OSD-Umfeld, interessieren.

Inhalt
In einem allgemeinen Teil stellt das Handbuch nach einem kurzen Überblick über die Bedrohungen für die Internet-Sicherheit ausführlich das Konzept des IPSec-Protokolls dar. Anschließend beschreibt das Handbuch die IPSec-Implementierung im BS2000/OSD und liefert alle Informationen zu Installation, Konfiguration und Inbetriebnahme des IPSec-Subsystems im BS2000/OSD.

Sonstige Literatur

[9] **WebSysAdmin/DomainAdmin V2.1**
Rechnerverwaltung in einer Domäne

Zielgruppe
Solaris-Systemverwalter

Inhalt
Rechnerverwaltung in einer Domäne

[10] **[Sol_LU]**
Solaris Live Upgrade 2.0 Guide
SUN Microsystems October 2001

Zielgruppe
Solaris-Systemverwalter

Inhalt
Beschreibung der Solaris-Installation per Live Upgrade

Stichwörter

Fujitsu Siemens Computers GmbH
Handbuchredaktion
81730 München

Fax: 0 700 / 372 00000

email: manuals@fujitsu-siemens.com
http://manuals.fujitsu-siemens.com

Kritik
Anregungen
Korrekturen

Absender

Kommentar zu CMX V6.0B (Solaris)
Betrieb und Administration

Fujitsu Siemens Computers GmbH
Handbuchredaktion
81730 München

Fax: 0 700 / 372 00000

email: manuals@fujitsu-siemens.com
http://manuals.fujitsu-siemens.com

Kritik
Anregungen
Korrekturen

Absender

Kommentar zu CMX V6.0B (Solaris)
Betrieb und Administration

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/*...
and the user documentation at *http://manuals.ts.fujitsu.com*.

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/*..., und unter *http://manuals.ts.fujitsu.com* finden Sie die Benutzerdokumentation.