

Umgebungen und Views im Library-Workspace

Der Library-Workspace dient zur hierarchischen Verwaltung der Natural-Systemdateien in der aktuellen Umgebung als Tree-View. Bei Natural ist eine Systemdatei eine Sammlung von Natural-Libraries, und eine Natural-Library ist eine Sammlung von Natural-Objekten und Ressourcen.

Dieses Kapitel behandelt die folgenden Themen:

- Lokale Umgebung
- Remote-Umgebung
- Logical-View
- Flat-View
- File-View
- Symbole für die Knoten
- Eigenschaften der Knoten

Lokale Umgebung

Alle Knoten unterhalb des Knotens **Local Environment** (Lokale Umgebung) gehören zur lokalen Windows-Umgebung, die normalerweise Ihr eigener PC ist. Dieser Knoten steht in allen unten beschriebenen Views zur Verfügung. Beispiel für den Logical-View:



Wenn die lokale Umgebung aktiv ist, wird der Knoten-Name **Local Environment** (Lokale Umgebung) in Fettdruck angezeigt, und in der Kommandozeile enthält das Textfeld **Environment** die Zeichenkette "Local" (Lokal).

Mit der Configuration Utility können Alias-Namen für die Systemdateien definiert werden. Wenn ein Alias-Name definiert wurde, wird er für den entsprechenden Knoten im Library-Workspace angezeigt. Siehe *System Files* (globale Konfigurationsdatei) in der *Configuration Utility*-Dokumentation.

Remote-Umgebung

Ein Knoten für eine Remote-Umgebung kann nur angezeigt werden, wenn Sie Natural Single Point of Development (SPoD) für die Remote-Entwicklung auf einem Großrechner, UNIX-, OpenVMS- oder Windows-Rechner benutzen. Weitere Informationen finden Sie unter *Remote-Entwicklungsumgebung*

aufrufen in der Dokumentation *Remote-Entwicklung mit SPoD*.

Der Knoten-Name ist entweder eine benutzerdefinierte Beschreibung oder (wenn eine Beschreibung nicht angegeben wurde) der Name des Servers und die Port-Nummer. Dieser Knoten steht in allen im Folgenden beschriebenen Views zur Verfügung. Beispiel für den Logical-View:



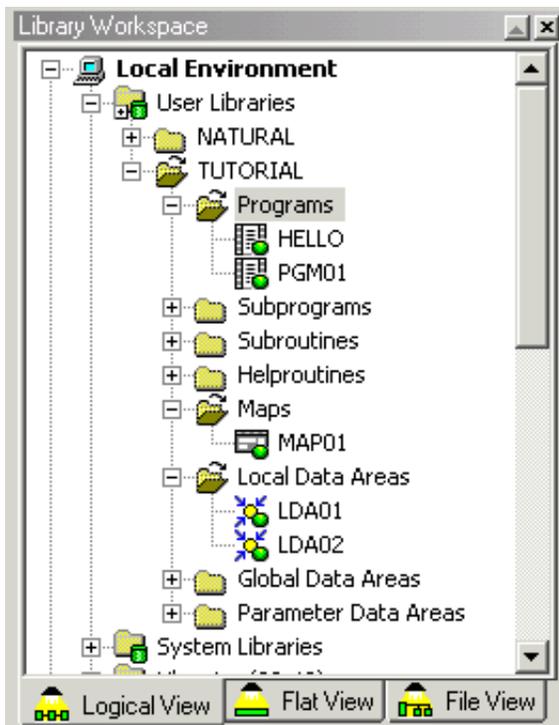
Wenn die Remote-Umgebung aktiv ist, wird der Knoten-Name in Fettdruck angezeigt, und in der Kommandozeile enthält das Textfeld **Environment** ebenfalls den Knoten-Namen.

Wenn gerade keine Verbindung zur Remote-Umgebung besteht, erscheint die Zeichenkette "Disconnected" (Nicht verbunden) rechts neben dem Knoten-Namen.

Alias-Namen können auch für die Systemdateien in einer Remote-Umgebung unter UNIX, OpenVMS oder Windows erscheinen. Die Alias-Namen werden mit der Configuration Utility definiert, die unter Natural für diese Plattform verfügbar ist.

Logical-View

Beim Logical-View stehen unterschiedliche Knoten für Benutzer-Libraries und System-Libraries zur Verfügung. Die Objekte in einer Library sind in verschiedene Knoten unterteilt, und zwar nach ihren Natural-Objektypen.



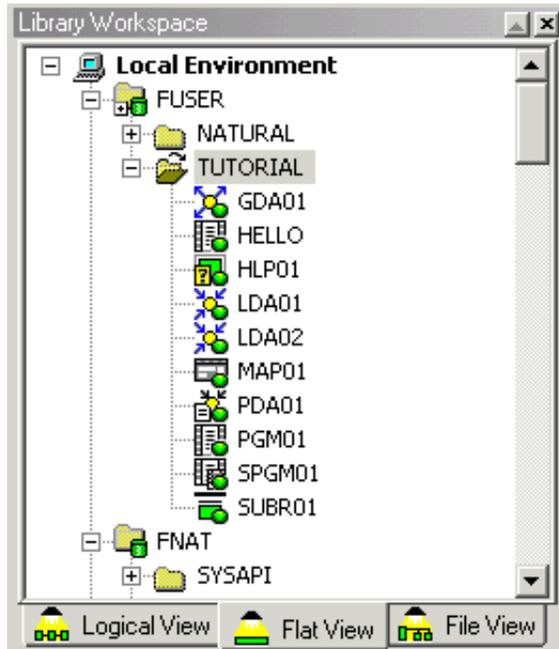
Beispielsweise erscheinen alle Programme in einem Gruppen-Knoten mit dem Namen **Programs**. Wenn Sie also die verfügbaren Programme in einer Library ansehen möchten, müssen Sie den **Programs**-Knoten aufklappen. Dies ist beim Flat-View nicht erforderlich (siehe unten).

Bei Subroutinen, Funktionen, Klassen und DDMs können die Langnamen im Logical-View gezeigt werden. Bei den anderen Views erscheinen die Kurznamen (max. 8 Zeichen; siehe auch *Namenskonventionen für Objekte*). Beispielsweise erscheint bei einer Subroutine der im Programm definierte Name (der länger als 8 Zeichen sein darf). Im Flat-View und File-View erscheint der beim Speichern der Subroutine definierte Name.

Flat-View

Beim Flat-View stehen unterschiedliche Gruppen-Knoten für Systemdateien zur Verfügung. Diese enthalten vom Benutzer geschriebene Programme (FUSER) und Systemprogramme (FNAT).

Die Objekte in einer Library werden ohne Gruppierungen angezeigt. Der Objekttyp wird stets durch das Symbol neben einem Objekt angezeigt. Siehe *Objekt-Knoten*.



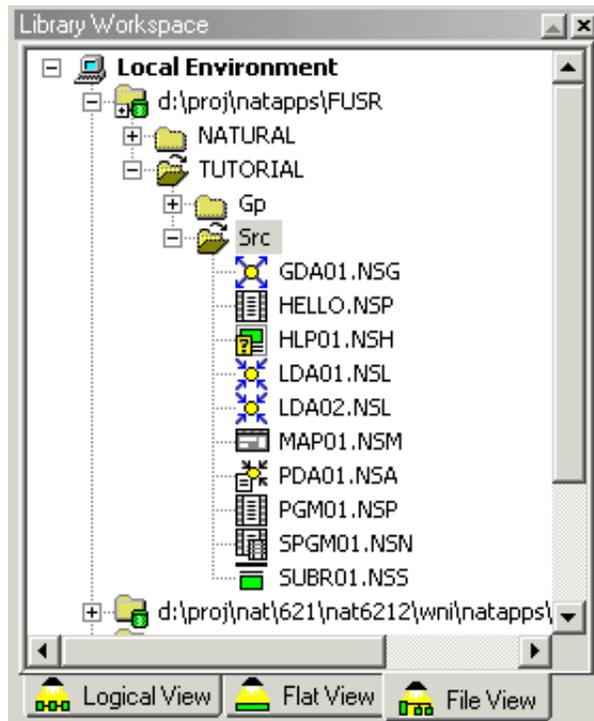
File-View

Beim File-View wird die Struktur der Natural-Umgebung so wie im Dateisystem dargestellt. Eine Natural-Systemdatei wird als Pfad zu den *Src*- und *Gp*-Unterverzeichnissen angezeigt.

Der Gruppen-Knoten **Src** enthält alle Natural-Sourcen, und der Gruppen-Knoten **Gp** enthält alle generierten Programme (ausführbare Natural-Programme). Gruppen-Knoten für andere Unterverzeichnisse (beispielsweise **Res** für Ressourcen oder **Err** für Fehlercodes) können auch verfügbar sein.

Anmerkung:

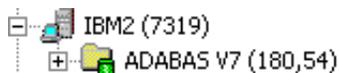
Generierte Programme werden auch als katalogisierte Objekte bezeichnet. Sie werden erstellt, wenn Objekte in Source- und Objekt-Form gespeichert oder katalogisiert werden.



Der Objekttyp wird stets durch das Symbol neben einem Objekt angezeigt. Siehe *Objekt-Knoten*.

Die Natural-Objekte werden mit den entsprechenden Datei-Extensionen angezeigt (zum Beispiel ".NGP" für "Natural Generated Program" (von Natural generiertes Programm) oder ".NSP" für "Natural Source-Programm"). Weitere Informationen zu den möglichen Datei-Extensionen finden Sie unter *System File Structure* in der *Operations*-Dokumentation.

In einer Remote-Umgebung erscheint der Name jeder per Mapping zugeordneten Remote-Datenbank mit der Datenbank-ID und der Dateinummer. Beispiel:



Symbole für die Knoten

Dieser Abschnitt beschreibt die Symbole, die in den Tree-View-Fenstern und List-View-Fenstern für die verschiedenen Knoten-Arten benutzt werden.

- Gruppen-Knoten
- Objekt-Knoten

Gruppen-Knoten

Die folgenden Symbole werden für die verschiedenen Gruppen-Knoten benutzt:



Lokale Umgebung.



Systemdatei mit Benutzer-Libraries oder System-Libraries. Wenn sie mit SPoD in einer Remote-Umgebung arbeiten, enthält die Systemdatei auch DDMs. Siehe auch *Library-Typen*.



Inaktive Systemdatei. Siehe auch *Library-Typen*.



Systemdatei mit einem aktiven Filter. Weitere Informationen finden Sie unter *Libraries und Objekte filtern*.



Knoten für Libraries oder Gruppen von Natural-Objekten. Der Inhalt einer Gruppe ist abhängig von dem View (Logical-View, Flat-View oder File-View).



Library mit einem aktiven Filter. Weitere Informationen finden Sie unter *Libraries und Objekte filtern*.



Library, in der die Zeilennummern im Source-Code unterdrückt sind. Weitere Informationen finden Sie unter *Zeilennummern im Source-Code unterdrücken*.



Library mit einem aktiven Filter, in der die Zeilennummern im Source-Code unterdrückt sind.



Remote-Umgebung.

Objekt-Knoten

Eine detaillierte Beschreibung jedes mit Natural verfügbaren Objekttyps entnehmen Sie *Objekttypen* im *Leitfaden zur Programmierung*.

Die folgenden Symbole werden für die unterschiedlichen Objekttypen benutzt:

	Programm
	Klasse (siehe auch <i>Class Builder Nodes</i> in der <i>Editors</i> -Dokumentation)
	Subprogramm
	Funktion
	Subroutine
	Copycode
	Helproutine
	Text
	Dialog
	Map
	Adapter
	Local-Data-Area (LDA)
	Global-Data-Area (GDA)
	Parameter-Data-Area (PDA)
	Datendefinitionsmodul (DDM)

Natural-Objekte können aus der Natural-Source, dem von Natural generierten Objekt (zum Beispiel ein generiertes Programm oder eine generierte Local-Data-Area) oder aus beidem bestehen. Der Unterschied wird im Symbol reflektiert. Die Symbole in der obigen Tabelle können auch wie folgt erscheinen (Beispiel für ein Programm):

	Ohne einen grünen Ball und nicht grau: nur die Source des Objekts ist verfügbar.
	Mit einem grünen Ball und nicht grau: Source und generiertes Objekt sind verfügbar.
	Mit einem grünen Ball und grau: nur ein generiertes Objekt ist verfügbar und keine Source.

Informationen zum Erstellen oder Öffnen eines solchen Objekts entnehmen Sie *Editoren aufrufen*. Oder entnehmen Sie detaillierte Informationen zur Benutzung der verschiedenen Editoren aus der *Editors*-Dokumentation.

Symbole erscheinen auch für Ressourcen. Weitere Informationen finden Sie unter *Ressourcen verwalten*.

Eigenschaften der Knoten

Sie können die Eigenschaften für jeden im Library-Workspace oder in einem List-View-Fenster gezeigten Knoten anzeigen. In Abhängigkeit von dem ausgewählten Element werden unterschiedliche Informationen im daraufhin erscheinenden Dialogfeld **Properties** (Eigenschaften) angezeigt (siehe die Beispiele unten).

Eigenschaften für einen Knoten anzeigen

1. Markieren Sie den Knoten, für den sie die Eigenschaften anzeigen möchten.
2. Wählen Sie aus dem Menü **View** den Befehl **Properties**.

Oder:

Rufen Sie das Kontextmenü auf, und wählen Sie den Befehl **Properties**.

Oder:

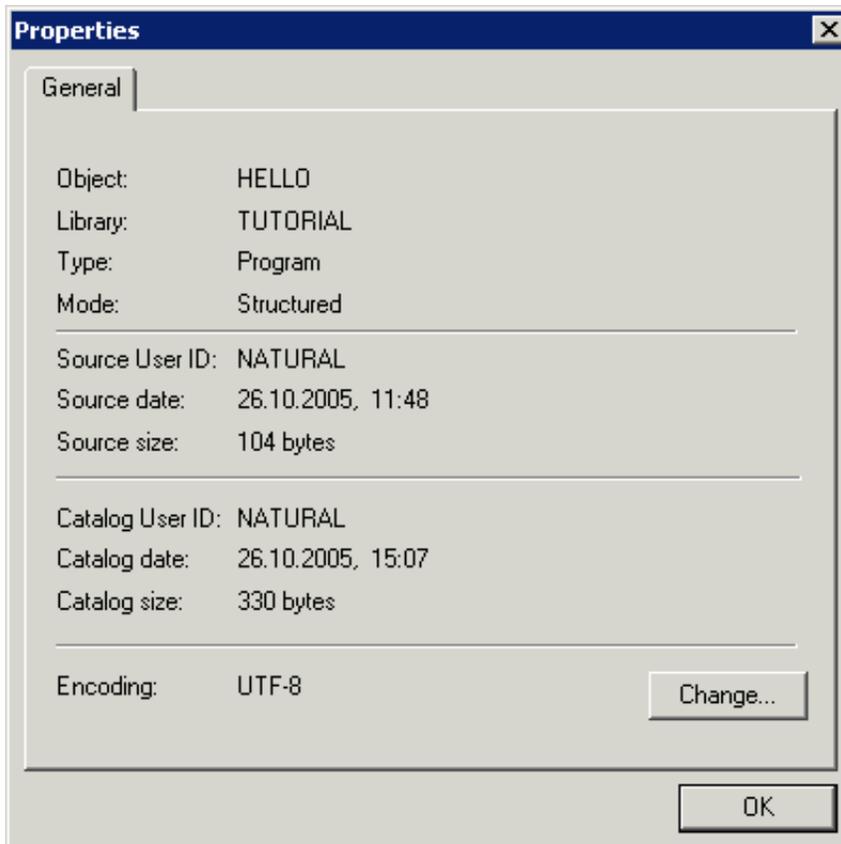
Drücken Sie ALT+EINGABE.

Es erscheint das Dialogfeld **Properties**.

Anmerkung:

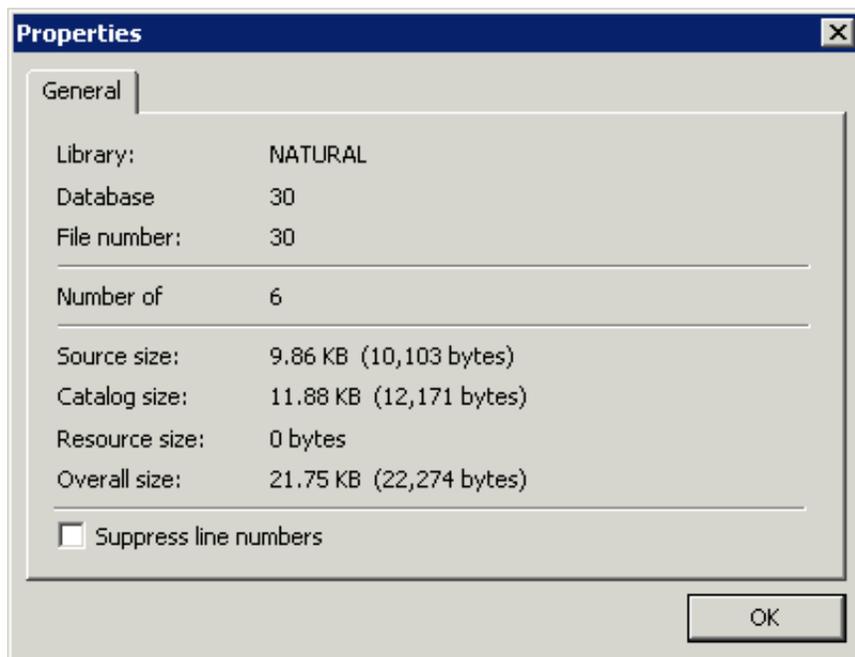
Sie können auch die Eigenschaften für die Knoten im Application-Workspace anzeigen. Im Gegensatz zu den für die Knoten im Library-Workspace erscheinenden **Properties**-Dialogfeldern bieten die **Properties**-Dialogfelder, die für den Application-Server und für eine Anwendung aufgerufen werden können, mehr als eine Seite. Es ist möglich, auf den zusätzlichen Seiten Informationen einzugeben. Weitere Informationen finden Sie unter *Anwendungen verwalten* in der Dokumentation *Remote-Entwicklung mit SPoD*.

Beispiel für ein Programm



Falls verfügbar für den ausgewählten Objekttyp, wird die Befehlsschaltfläche **Change** (Ändern) angezeigt. Wenn Sie diese Befehlsschaltfläche wählen, wird das Dialogfeld **Encoding** angezeigt, in dem Sie eine andere Codepage definieren können. Weitere Informationen finden Sie unter *Codepage für ein Natural-Objekt ändern*.

Beispiel für eine Library



Wenn Sie die Eigenschaften für eine Library anzeigen, steht das Kontrollkästchen **Suppress line numbers** zur Verfügung. Wenn dieses Kontrollkästchen markiert ist, werden neue Sourcen in dieser Library ohne Zeilennummern im Dateisystem abgespeichert. Die Zeilennummern in bereits bestehenden Sourcen werden unterdrückt sobald diese Sourcen erneut abgespeichert werden. Weitere Informationen finden Sie unter *Zeilennummern im Source-Code unterdrücken*.