

# Allgemeine Informationen

Dieses Kapitel behandelt die folgenden Themen:

- Über den Debugger
  - Remote-Debugging
- 

## Über den Debugger

Ab Natural für Windows Version 6.2 ist der Debugger in Natural Studio integriert. Die vollständige Natural Studio-Funktionalität kann somit parallel zum Debugger benutzt werden. Wenn beispielsweise der Debugger aktiv ist, können Sie zu einem anderen Objekt im Library-Workspace navigieren, oder aber können Sie mit dem Befehl **Find Object** nach einem bestimmten Objekt suchen.

Der Debugger wird benutzt, um Natural-Anwendungen in den folgenden Umgebungen auszutesten:

- in der lokalen Umgebung und
- in einer Remote-Umgebung, d.h.: auf einem per Mapping zugeordneten Entwicklungs-Server (SPoD). Die Voraussetzung dazu ist, dass eine der folgenden Versionen auf dem Entwicklungs-Server installiert ist:
  - Natural for Mainframes Version 4.2 oder darüber.
  - Natural for UNIX Version 6.2 oder darüber.

Für den Debugger sind keine zusätzlichen Einstellungen erforderlich. Natural Studio wickelt alle Schritte intern ab (wie z.B. das Herstellen oder Beenden der Kommunikation mit dem entsprechenden Server).

Siehe auch *Umgebungen und Views im Library Workspace* in der Dokumentation *Natural Studio benutzen* und *Remote Entwicklungsumgebung aufrufen* in der Dokumentation *Remote-Entwicklung mit SPoD*.

### Anmerkung:

Es gibt mehrere Unterschiede, wenn Sie Anwendungen in einer Remote-Großrechnerumgebung austesten. Diese Unterschiede sind in der plattform-spezifischen Natural Development Server-Dokumentation (NDV) aufgeführt, die für diese Natural-Release gilt. Die NDV-Dokumentation steht separat zur Verfügung; sie ist nicht Bestandteil der Natural für Windows-Dokumentation.

### Wann wird der alte Debugger noch mit SPoD benutzt?

Wenn Sie mit Natural Studio arbeiten und den Debugger für ein Objekt auf einem per Mapping zugeordneten Entwicklungs-Server aufrufen, ist es möglich, dass der alte Debugger anstatt des neuen integrierten Debuggers aufgerufen wird. Dies ist der Fall, wenn eine alte Version von Natural auf dem Entwicklungs-Server installiert ist. Es gibt die folgenden alten Versionen:

- Natural for UNIX Version 6.1.1 oder darunter.

Siehe *Alten Debugger benutzen*.

## Remote-Debugging

Mit einer der nächsten Versionen wird Remote-Debugging nicht mehr unterstützt. Statt dessen müssen Sie dann den Debugger benutzen, der in Natural Studio integriert ist.

Remote-Debugging erfolgt, wenn Sie eine ursprüngliche Natural für UNIX-Anwendung von einem Windows-Computer aus austesten, oder wenn Sie eine Natural-Dialoganwendung von einem anderen PC aus im Remote-Betrieb austesten. Dies erfolgt außerhalb des Rahmens von SPoD.

Um Remote-Debugging einzuschalten, müssen Sie wie folgt vorgehen:

- Installieren Sie das Debug-Frontend auf einem Windows-Computer. Dadurch wird auch der Remote-Debugging-Service `natdbgsv` installiert, der für Remote-Debugging aktiv sein muss. Siehe *Remote-Debugger installieren*.
- Definieren Sie die Parameter `RDNODE`, `RDPOR` und `RDACTIVE` in der Umgebung, die die Anwendung enthält, die ausgetestet werden soll. Weitere Informationen finden Sie unter *Einrichten Ihrer Umgebung für Remote-Debugging*.
- Rufen Sie den Debugger auf, indem Sie das Systemkommando `DEBUG objektname` in der Umgebung eingeben, die die Anwendung enthält, welche ausgetestet werden soll.

Die folgenden Themen werden nachfolgend behandelt:

- Remote-Debugger installieren
- Einrichten Ihrer Umgebung für Remote-Debugging
- Szenarien für Remote-Debugging

### Wichtig:

Um den Remote-Debugger unter Microsoft Windows zu starten, muss die Personal-Firewall deaktiviert sein. Siehe *Configuring the Microsoft Windows Personal Firewall to Run Natural* in der *Operations*-Dokumentation für Natural für Windows.

## Remote-Debugger installieren

Wenn bei Ihnen Natural für Windows installiert ist, müssen Sie den mit Natural für Windows ausgelieferten Remote-Debugger benutzen. Wenn der Remote-Debugger noch nicht installiert wurde, benutzen Sie die Option **Modify** (Ändern) des Natural-Installationspakets, um den Remote-Debugger zu Ihrer Natural für Windows-Installation hinzuzufügen. Siehe *Maintaining Your Natural or Natural Runtime Environment* in der *Installation*-Dokumentation für Natural für Windows.

Sie müssen den Remote-Debugger nur dann alleinstehend installieren, wenn Natural für Windows bei Ihnen nicht installiert haben. Wenn Sie eine Natural-Anwendung austesten möchten, die auf einer UNIX-Plattform gespeichert wird, kopieren Sie `$NATDIR/$NATVERS/dbrmt/1386/nrd.exe` vom UNIX-Installationsmedium auf Ihren Windows-Computer (zum Beispiel in ein temporäres Verzeichnis) und entpacken Sie die Datei. Starten Sie `setup.exe`, um die Installation des Remote-Debugger zu starten.

## Einrichten Ihrer Umgebung für Remote-Debugging

Die folgenden Themen werden nachfolgend behandelt:

- Windows-seitig ohne Terminal-Services
- Windows-seitig mit Terminal-Services
- Natural-seitig

### Windows-seitig ohne Terminal-Services

Installieren Sie entweder den Remote-Debugger (die entsprechenden Dateien finden Sie auf dem UNIX-Installationsmedium), oder installieren Sie Natural für Windows (der Remote-Debugger kann optional mit dem Setup-Typ "Custom" von Natural für Windows installiert werden; siehe die *Installation*-Dokumentation für Natural für Windows). Dadurch wird auch der Natural-Remote-Debugging-Service `natdbgsv` installiert.

Um den Remote-Debugging-Service zu deinstallieren, geben Sie `natdbgsv -u` in der Kommandozeile ein. Um den Portnamen und die Version des aktuellen Service zu erfahren, geben Sie `natdbgsv -s` ein. Um den Service auf einem anderen Port neu zu installieren, deinstallieren Sie ihn zuerst, und geben Sie dann `natdbgsv -i portnummer` ein, wobei *portnummer* der Wert des Profilparameters `RDPOR`T ist. Wenn die Portnummer bereits in Benutzung ist, erscheint ein Dialog, in dem Sie eine neue Portnummer eingeben können.

#### Anmerkung:

Bevor Sie den Remote-Debugging-Service auf einem anderen Port als dem 2600er (Standardwert) installieren, müssen Sie den Wert des Profilparameters `RDPOR`T ändern, so dass er der Portnummer des Client-Computers entspricht, auf dem die Natural-Anwendung ausgetestet wird.

### Windows-seitig mit Terminal-Services

Installieren Sie den Remote-Debugger (die entsprechenden Dateien finden Sie auf dem UNIX-Installationsmedium), oder installieren Sie Natural für Windows (der Remote-Debugger kann optional mit dem Setup-Typ **Custom** von Natural für Windows installiert werden; siehe die *Installation*-Dokumentation für Natural für Windows). Dadurch wird auch die Debugger-Verknüpfung für den Listener-Prozess `natdbgsv` im **Start**-Menü erstellt (in demselben Programme-Verzeichnis, in dem Sie die Verknüpfungen für Natural finden können). Um Remote-Debugging zu verwenden, muss `natdbgsv` gestartet werden. Wenn der Listener-Prozess zum ersten Mal in einer bestimmten Benutzer-Session aufgerufen wird, wird eine freie Portnummer angezeigt, die in dem entsprechenden Feld des Profilparameters `RDPOR`T eingegeben werden muss.

Jede nachfolgende Aktivierung von `natdbgsv` bewirkt, dass der Listener mit derselben Portnummer gestartet wird. Wenn diese Nummer bereits von einer anderen Anwendung benutzt wird, dann muss der Benutzer im Port-Dialog von `natdbgsv` eine neue Portnummer angeben, und `RDPOR`T muss entsprechend angepasst werden.

### Natural-seitig

Starten Sie Natural mit den folgenden Profilparameter-Einstellungen:

- `RDACTIVE` ist auf "ON" gesetzt.

- RDNODE mit dem Knotennamen des Windows-Servers.
- RDPORNT ist auf "2600" oder eine andere Portnummer gesetzt: entweder die Nummer des Ports, auf dem Sie den Remote-Debugging-Service installiert haben (siehe *Windows-seitig ohne Terminal-Services*), oder des Ports auf dem der Listener-Prozess gestartet wurde (siehe *Windows-seitig mit Terminal-Services*).

## Szenarien für Remote-Debugging

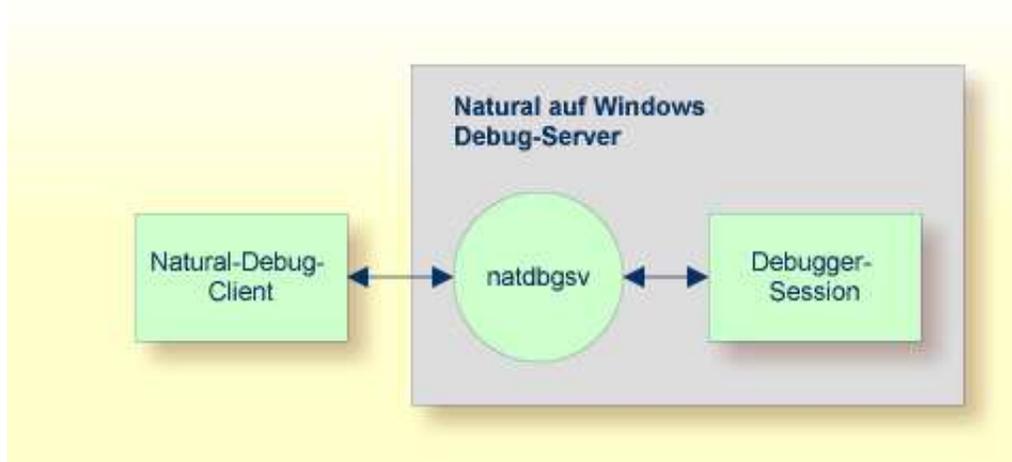
Es gibt unterschiedliche Szenarien, wie Sie Remote-Debugging verwenden können: Ein einzelner Natural-Client läuft unter der Kontrolle einer Remote-Debugging-Session, oder eine verteilte Natural-Anwendung läuft unter der Kontrolle mehrerer Remote-Debugging-Sessions. Eine solche verteilte Anwendung kann sowohl Natural-RPC- und DCOM-Server enthalten als auch nicht in Natural geschriebene Komponenten, wie z.B. Visual Basic-Clients.

Die folgenden Themen werden nachfolgend behandelt:

- Szenario 1: Einzelne Natural-Anwendung austesten
- Szenario 2: Verteilte Natural-Anwendung austesten
- Szenario 3: Natural-Teil einer heterogenen Anwendung austesten

### Szenario 1: Einzelne Natural-Anwendung austesten

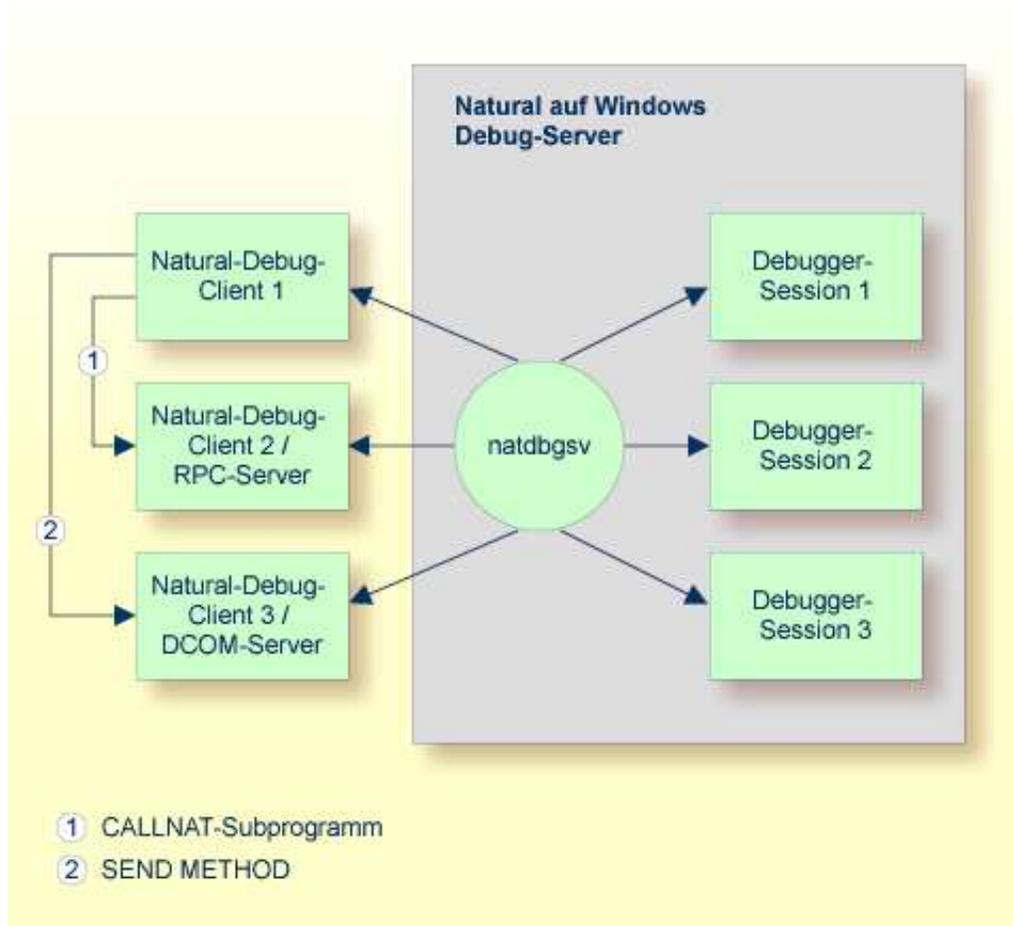
Die folgende Abbildung veranschaulicht das Austesten einer einzelnen Natural-Anwendung.



### Szenario 2: Verteilte Natural-Anwendung austesten

Um jede Komponente der folgenden verteilten Natural-Anwendung auszutesten, geben Sie in der Kommandozeile des Natural-Debug-Clients 1 das folgende Kommando ein: `DEBUG objektname`. Wenn der Natural-Debug-Client zum ersten Mal ein Subprogramm auf einem Natural-RPC-Server aufruft, wird eine neue Debugging-Session für den RPC-Server geöffnet. Dann wird die Verarbeitung des RPC-Servers ausgetestet. Die Debugging-Session wird geschlossen, sobald der RPC-Server beendet wird.

Dasselbe gilt für einen Natural-DCOM-Server.



### Szenario 3: Natural-Teil einer heterogenen Anwendung austesten

Wie auch im vorherigen Szenario: Wenn eine Methode auf dem DCOM-Server zum ersten Mal aufgerufen wird, dann wird eine neue Debugging-Session für den DCOM-Server geöffnet, die Verarbeitung des DCOM-Servers wird ausgetestet, und die Debugger-Session wird geschlossen, sobald der DCOM-Server beendet wird:

