

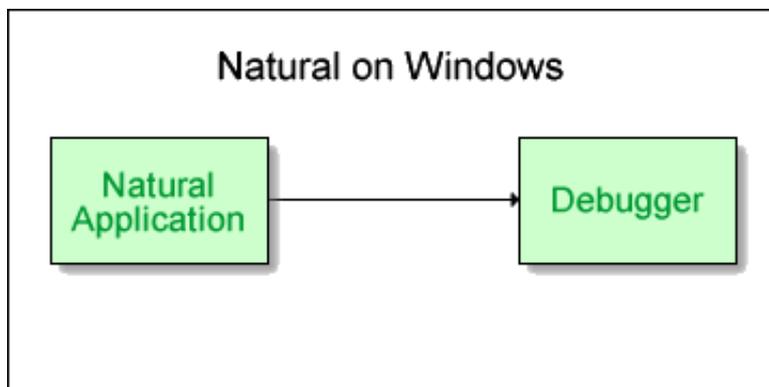
Local and Remote Debugging

The following topics are covered below:

- Local Debugging
 - Remote Debugging
 - Setting Up Your Environment
 - Running a Natural Remote Debugging Session
-

Local Debugging

Local debugging, as opposed to remote debugging, is done with a debugger integrated in Natural for Windows. This debugger can be used to debug Natural applications running on the same computer.

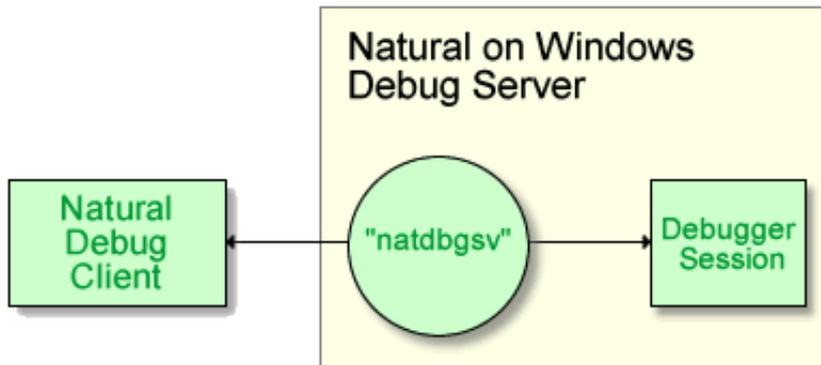


Remote Debugging

There are several scenarios of how you can use remote debugging: a single Natural client runs under the control of one remote debugging session or a distributed Natural application runs under the control of several remote debugging sessions. Such a distributed application may include both Natural RPC and DCOM servers or even components not written in Natural, like Visual Basic clients.

- Scenario 1: Debugging a Single Natural Application
- Scenario 2: Debugging a Distributed Natural Application
- Scenario 3: Debugging The Natural Part of a Heterogeneous Application

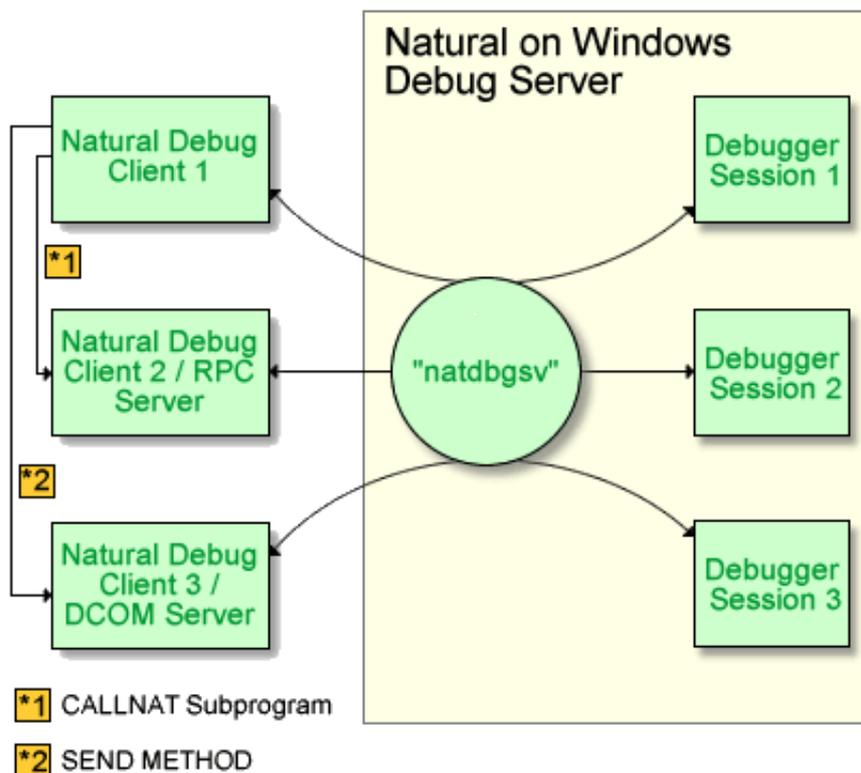
Scenario 1: Debugging a Single Natural Application



Scenario 2: Debugging a Distributed Natural Application

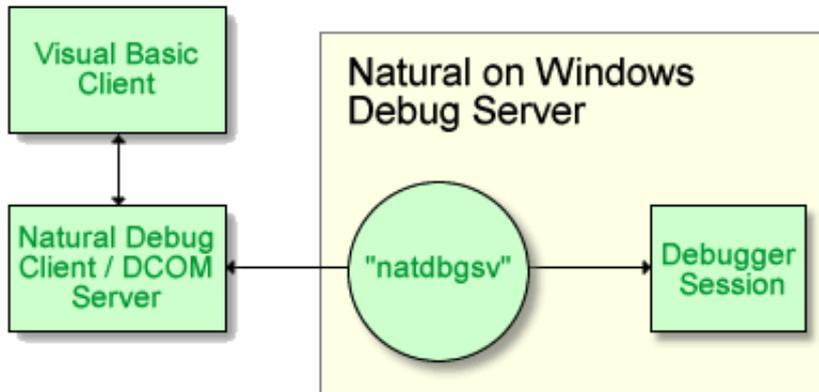
To debug each component of the following distributed Natural application, you enter "DEBUG *objectname*" in the command line of Natural debug client 1. The first time the Natural Debug Client calls a subprogram on a Natural RPC server, for example, a new debug session is opened for the RPC server. Then, the RPC server's processing is debugged. The debug session is closed as soon as the RPC server is terminated.

The same applies to a Natural DCOM server.



Scenario 3: Debugging The Natural Part of a Heterogeneous Application

Like in scenario 2, the first time a method on the DCOM server is called, a new debug session is opened for the DCOM server, the DCOM server's processing is debugged, and the debugger session is closed as soon as the DCOM server is terminated:



Setting Up Your Environment

Windows Side

Install the Natural remote debugger or install Natural for Windows Version 5.1 PL 1 or above. This also installs the Natural Remote Debugging Service.

To uninstall the Natural Remote Debugging Service, use "natdbgsv -u" in the command line. To view the current service's port name and version, enter "natdbgsv -s". To re-install the service on a different port, uninstall it first and then enter "natdbgsv -i *portnumber*", where *portnumber* is the value of the RDPORT profile parameter. If the portnumber is already used, a Dialog appears where the user can enter a new port number.

Note:

If you install the Natural Remote Debugging Service on another port than 2500, you have to change the value of the RDPORT profile parameter according to the client computer where the Natural application is being debugged.

Windows with Terminal Services

Install the Natural remote debugger or install Natural for Windows Version 5.1 PL 1 or above. This also creates the Debugger Icon in the Program Group representing the listener process natdbgsv. In order to use remote debugging natdbgsv must be started. The first time the listener process is launched in a specific user session, a free portnumber is exposed, which must be taken as the value of the RDPORT profile parameter.

Any subsequent activation of natdbgsv causes the listener to be started with the same portnumber except this number is already used by a different application. In this case the user must provide natdbgsv's port dialog with a new portnumber and RDPORT must be adapted accordingly.

Natural Side

Start Natural with the following three profile parameter settings:

- RDACTIVE=ON;
- RDNODE=*nodename*, where *nodename* is the name of the Windows server;
- RDPORT=2500 (or another port number, depending on which port you have installed the Natural Remote Debugging Service (Windows)), or with which port the listener process was started (Windows with Terminal Services).

Running a Natural Remote Debugging Session

To run a Natural Remote Debugging Session:

1. Ensure that the Windows server has been started and that the service "natdbgsv" is active.
2. Enter "DEBUG *objectname*" in the Natural command line, where *objectname* is the name of the Natural object you wish to debug.

The Natural Debugger becomes active on the Windows computer, with the following window caption: "Debugging remote Natural client (*nodename*::*username*::*process-id*)", where *nodename* is the name of the computer where Natural is running, *username* is the name of the Natural user and *process-id* is the Natural process ID.