NetWeave Interface Class for JAVA

Installation, Configuration and User's Guide

Revised 5 January 2010

www.netweave.com

Copyright © NetWeave Integrated Solutions, Inc. 2003-2010. All rights reserved.

NetWeave is a registered trademark of NetWeave Integrated Solutions, Inc.
Windows is a registered trademark of Microsoft Corporation.
CICS, MVS, and MQSeries are registered trademarks of the IBM Corporation.
UNIX is a registered trademark of X/OPEN Ltd.
Tandem, Guardian, VMS, and OpenVMS are registered trademarks of HP.
All other trademarks are noted in the text and are the property of their respective owners.

Table of Contents

Introduction

This document describes the enhancements of Netweave support for the java environment. This provides a stateless, context free java class library, NWDSInterface Class (N2IC), for communicating with other Netweave servers. N2IC also provides Netweave transaction support, so a java client can start a transaction against a Netweave supported transaction service, such as Tandem TM/MT (TMF). The local server daemon initiates and maintains connections with other Netweave servers and also maintains transaction handles.

The audiences for this document are java developers and administrators. It is assumed the reader is familiar with java concepts, object oriented programming (OOPS) and architecture. It is not the intent of this document to explain these.

In the previous implementation a developer had to incorporate the transport classes into his/her component. This required the developer to learn the evolve architecture, master several interrelated classes and create and maintain several configuration files. With the phase2 architecture the developer just instantiates the N2IC like any other component. The configuration is provided in the N2IC.properties file and contains only one entry, the port of the local server daemon. If one is not using transactions, there is only one method to call in the object, namely writeRead. It is a much more elegant approach that uses concepts that should be completely familiar to a java developer.

The N2IC classes are distributed as N2IC.jar and shipped together with the daemon's executable appropriate to the computer that runs the java client. The distribution package also contains a simple control program for managing the daemon.

General features of NetWeave are described in its standard documentation. NetWeave documentation may be downloaded from our ftp site in PDF format or as Microsoft Word documents.

- //ftp.netweave.com/pub/doc/200/pdf/NWDoc200_PDF.zip
- //ftp.netweave.com/pub/doc/200/msword/NWDoc200_MSWord.zip

NWDS Interface Class (N2IC)

Configuration

N2IC.jar and the sample N2IC properties file exist in the distribution file. The properties file is a set of name/value key pairs used for storing and retrieving configuration settings. Here is a sample file:

#properties file for N2IC localSvrPort

= 19344

There is one parameter to configure for the local server daemon, localSvrPort.

Programming – Exchanging Data

To send a reqeust and get a reply from a Netweave server, instantiate the N2ICTransport class and call the writeRead method of N2IC.

```
      public com.netweave.N2IC.IOData writeRead(
java.lang.String connectionName,
com.netweave.N2IC.IOData wData,                          long
timeout) throws NWDSException
```

sends and receives data from remote Netweave server

Parameters:

`connectionName` - name of server as defined in local server's ini file

`wData` - io data written from java client to Netweave server

`timeout` - time in milliseconds to wait for read to complete.

Returns:

IOData data returned from Netweave server

Throws:

`NwdsException`

The writeRead method requires a connectionName string that needs to be defined in the local server daemon ini file, and an IOData object that contains the data to be written to the server. The timeout parameter is optional. If not provided your program will block until the Netweave server responds. If successful writeRead returns an IOData object with the response, otherwise an NWDSException is thrown. Here is a programming snippet that initializes the bean and exchanges data 1000 times with a server called ECHO_SERVER, which just returns the same data that was sent:

```
N2ICTransport transport = new N2ICTransport();
IOData i = new IOData(3000); byte b[];
b = new String("hello world").getBytes(); i.data
= b;
IOData o;
String server = "ECHO_SERVER";
for (int x=0; x < 1000; x++) {      o =
transport.writeRead(server, i);
    System.out.println("Response = " + new String(o.data);
}
```

Programming – Transactions

N2IC also supports transactions to Netweave supported transaction servers such as Tandem TM/MT (TMF) and IBM CICS. Transactions are a series of writes that are sometimes referred to as a "unit of work". Either all the writes of the unit of work are accepted or rolled back by the transaction server.

There are three additional methods in N2IC for transaction support: startTransaction, commitTransaction, and abortTransaction. The startTransaction returns a TxHandle object. The TxHandle object is passed to the writeRead method for all writes that are a part of the unit of work. The transaction must then be completed with a call to commitTransaction or all writes will be rolled back with a call to abortTransaction. Here is a programming snippet of a transaction:

```
N2ICTransport = new N2ICTransport();
IOData i = new IOData(3000); byte b[];
b = new String("hello world").getBytes();
```

```
i.data = b;
String server = “ECHO_SERVER”;
String node = “TANDEM::”;


TxHandle tx = transport.startTransaction(node);
transport.writeRead(server, tx, i); transport.writeRead(server,
tx, i); transport.writeRead(server, tx, i);
transport.commitTransaction(tx);
```

Since the local daemon server holds the transaction handle, an inactivity timer exists on the daemon to ensure that all transactions are eventually resolved. If the transaction is not used for the timer period, the daemon will abort the transaction. The timer can be set in the ini file. Its default value is 30 minutes.

The Daemon

The key to the NetWeave Transport is the daemon, a local process that provides the services for the N2IC instrumented java code. The daemon uses the NetWeave IPC library to make TCP/IP connections to remote servers. It pools these connections for repeated use by the N2ICinstrumented components.

The daemon accepts local connections from the N2IC java clients. A request message names the remote server to which the daemon forwards the request. When the reply returns from the remote server, the daemon relays it back to the specific object that made the request. The local socket connection between a java object and the daemon exists for the lifetime of the object.

As the name 'daemon' suggests, this process must be running when the application server is running. The daemon is a NetWeave application that uses information in its configuration file ("INI file") to make TCP/IP connections to one or more server applications on a remote computer. General information about NetWeave INI files is in the "NetWeave Configuration Guide" and not reproduced here.

The daemon implements a minimal management interface described in section "Control the Daemon". The control interface is shown in figure 1 as a remote application, but it may be a local application.

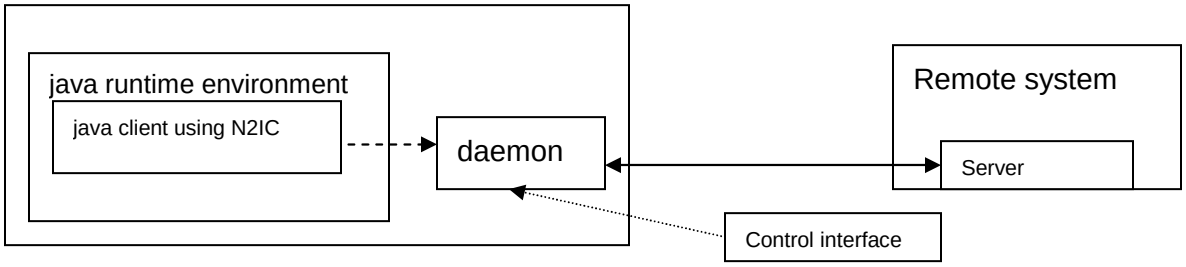

Figure 1

Figure 1 illustrates relationships among the java runtime environment, java client using N2IC, daemon, and the remote server application. The java programmer instantiates the N2ICTransport class, connects to the daemon and sends the request. The daemon extracts the name of the remote server from the request and searches its pool for an available connection to that process. It creates a new one if the pool is empty. The daemon sends the request to the server who processes it and replies. The daemon relays the reply to the N2IC object.

Create a Daemon Service on Windows

The Windows version provides two service executables, localsrv.exe and vicelocl.exe. The localsrv executable runs from the command line interface exactly as the version described above. The vicelocl runs as a traditional Windows service. Both versions of the daemon share the same code for the core functionality. The vicelocl.exe includes an additional layer of functionality to interoperate with the Windows Service Control Manager. This "service shell" supports 5 commands to manage the daemon service.

- INSTALL – installs the service as NWDS_LCL
- UNINSTALL – uninstalls the service

- START – starts the service
- STOP – stops the service
- QUERY – tells whether the service is running or stopped.

A sixth command, CONSOLE, runs the service directly from the command line. It is the same as running localsrv.

The service shell has its own runtime syntax:

<service daemon> <service command> <INI file> <start group> <control group> where

- <service daemon > is the full file name of the daemon executable. The full file name must include the path. NetWeave names the daemon "vicelocl.exe", but you may rename to suit your installation.
-
- <service command> is one of the commands above.
- <INI file> is the full file name (including path) of the NetWeave INI file.
- <start group> is a combination of a typical NetWeave start group and the NetWeave IPC connection group to which the client stub object in the transport classes connect.
- <control group> is a connection group for the daemon's command interface. This is described below.

Daemon's INI File

License Key Information

At the top of the INI file, there is a key that tells the Daemon process that it is authorized to operate on the current hardware platform.

For instance:

```
***    SAMPLE INI CONFIGURATION FILE      ***
[LICENSE_GROUP]
* sun2 -- 20091231
LICENSE_KEY = GLSFDKPSGCFDPKCSBNFTNB
```

In order to use the daemon NetWeave will need to send you a license key that matches your hostname (or IP address), and an expiration date. Note that the daemon will not function without the hostname/IP address, but will function past the expiration date, albeit with some complaints in the log file.

When you receive a key from the NetWeave license server, simply replace key value highlighted above with the contents of the key value for the associated platform.

Configuration

The Daemon is configured by use of a text-based INI file that consists of a few key parameters that instruct the Daemon how to operate. Change the start group to include connection information for the local connection between each client stub object and the daemon. For example, assume the current start group, `[START]`, contains only the `@TRACE_FILE@` macro.

```
[START]
@TRACE_FILE@ = c:\usr\nwds\daemon.err
```

Add connection parameters for a TCP/IP connection on localhost, i.e., IP address 127.0.0.1. This is the address at which the client expects to find the daemon. The port is selectable; in the following example it is arbitrarily set to 19344. As described below, this parameter must match the localSvrPort parameter in the java client N2IC.properties configuration file.

Caution: unlike typical TCP/IP connections to a process on another computer, this localhost connection must specify `LOCAL_PROCESS = 1`. For example,

```
[START]
@TRACE_FILE@ = c:\usr\nwds\daemon.err
PROTOCOL = TCPIP
LOCAL_PROCESS = 1
TCPIP_ADDRESS = 127.0.0.1
TCPIP_PORT = 19344

The daemon has two inactivity timers for removing old connections to the
Netweave servers and stale transaction handles.  The default values for both are
30 minutes and can be modified in the USER_NAME_GROUP.  For example,
[USER_NAME_GROUP]
# maximum time to maintain forward handles in seconds
FORWARD_EXPIRE_TIME=900
# maximum time to maintain tx handles in seconds
TX_EXPIRE_TIME=600
```

Daemon's Control Group

This connection group contains parameters similar to the start group. The TCPIP_PORT must be unique, and the `LOCAL_PROCESS` must be 0, the default. The `LOCAL_PROCESS = 0` instructs NetWeave to use its standard library. This example for a control group named `[ADMIN]` sets the port to 5002.

```
[ADMIN]
PROTOCOL = TCPIP
LOCAL_PROCESS = 0
TCPIP_ADDRESS = 127.0.0.1
TCPIP_PORT = 5002
```

Audit the Daemon

The daemon implements an audit logging feature that writes a message to the NetWeave trace file to identify each request and reply. Audit logging is enabled by the `AUDIT_ON` parameter in the `[USER_NAME_GROUP]`. Refer to the "Configuration Guide" for details of this special group.

```
[USER_NAME_GROUP]
AUDIT_ON = 1
```

Control the Daemon

The daemon control program (dcp) is an administrative tool that sends commands to the daemon.

Syntax of the control program:

<dcp> <INI file> <start group> <command> <daemon's control group>

The dcp implements three commands:

- S – "graceful shutdown" tells the daemon to close all connections and stop.
- X – "toggle logging" turns logging on or off. The `AUDIT_ON` parameter sets the initial state for logging.
- I – "info" reports whether logging is on or off. The report includes TCP/IP addresses of the current remote server connections.

The dcp may be run from a remote computer or on the same computer as the daemon. When the daemon and dcp run on the same computer, they should share a common configuration file. Create a unique start group for the dcp in order to specify a unique FILE_NAME for the traces. By sharing a common configuration file, there can be no mismatch on the parameters for the control group. When the dcp is run on a computer different from the daemon, the daemon control group must be identical in both INI files.