

Distributing NaturalX Applications

An application consisting of NaturalX classes can be distributed across several processes and machines using DCOM.

This chapter covers the following topics:

- General
- Globally Unique Identifiers - GUIDs

See also *Using Statements and Commands in a NaturalX Server Environment* in the *Operations* documentation.

General

Using NaturalX, you can make Natural classes and their services available to local and remote clients, thus creating distributed applications. Local clients are processes that run on the same machine as a given NaturalX server, and remote clients are processes that run on a different machine.

In order to distribute applications, a widely used distributed object technology is used - the Microsoft Distributed Component Object Model (DCOM). When you register a Natural class to DCOM, its interfaces are presented to clients in a quasi-standardized fashion as dynamic COM interfaces, which are also known as dispatch interfaces. These interfaces can be easily addressed by many programming languages including Visual Basic, Java, C++ and, of course, Natural.

There are several points that must be taken into consideration when organizing the distribution of a NaturalX application. Each of these points is discussed in more detail in this section and in the *Operations* documentation.

- Determine whether each class should be internal, external or local (see the section *Internal, External and Local Classes*).
- Globally unique IDs (GUIDs) must be assigned to the internal and external classes and their interfaces in order to be able to address them uniquely in the network (see the section *Globally Unique Identifiers (GUIDs)*).
- You can define the activation policy for each class in order to control the conditions under which DCOM activates it (see section *Activation Policies* in the *Operations* documentation).
- In order to organize classes to applications, you can define NaturalX servers and assign the classes to them (see the section *NaturalX Servers* in the *Operations* documentation).
- Classes must be registered to make them known to DCOM (see section *Registration* in the *Operations* documentation).
- You can configure an application in order to further control its behavior (see the sections *Configuration Overview* and *DCOM Configuration on Windows* in the *Operations* documentation).

Internal, External and Local Classes

It is important to distinguish between classes for internal use, classes for external use and those for local use only.

Internal Classes

Objects (instances) of internal classes can only be created in the client process.

Internal classes have the following features:

- Access to client session-dependent resources such as files and system variables.
- Can run within the client transaction.
- Can be debugged using the Natural debugger (local debugging).

External Classes

Objects (instances) of external classes can be created in a different process or on a different machine. If the client process is simultaneously a server for the class, they can also be created in the client process.

External classes have the following features:

- No access to client session-dependent resources such as stacks, files and system variables.
- Do not run within the client transaction.
- Can be used by remote nodes.
- Can be used by various clients using a variety of languages such as Natural, Java, Visual Basic, C/C++, etc.
- Can be debugged with the Natural debugger (remote debugging).

Local Classes

Local classes are classes, which are executed in local execution mode. Natural executes a class locally (within the Natural session) if it is either not registered or if DCOM is not available.

Local classes have the following features:

- Can be used even if DCOM is not available.
- Need not be registered with DCOM.
- Cannot be used from outside the client process.

Globally Unique Identifiers - GUIDs

DCOM uses global unique identifiers (GUIDs) - 128-bit integers that are virtually guaranteed to be unique throughout the world - to identify every interface and every class. This helps to ensure that server components can be located and to prevent clients connecting to an object accidentally.

If a class is to be registered to DCOM, every interface defined in a Natural class and the class itself must be associated with such a globally unique ID.

Once a globally unique ID has been assigned to an interface or a class, the ID must never be changed.

Using the Class Builder

Natural provides the Class Builder as the tool to develop Natural classes. The Class Builder automatically assigns a GUID to every class and interface.