# DLLCOM – Multithreaded COM Support

## Introduction

DLLCOM introduces the ability to perform non-blocking COM calls in VisualWorks. This improves responsiveness of COM servers implemented in VisualWorks using COM Connect

## Requirements

DLLCOM will run on VisualWorks Version from 7.3 to 7. 6 on Window XP platforms with 32Bit Intel Architecture.

## Installation instructions

DLLCOM consists of two parts, a DLL file and three parcel files.

The installation consists of two steps:

1. The DLL file needs to be copied to the default com directory specified inside the image.

2. Depending on whether threaded Automation functionality is needed or not, please load one of the following parcels: (we suggest loading the latter)
   - COM – Ole Multithreaded Extension
   - COM – Automation Multithreaded Extensions.

The parcels will automatically load COM-Connect and other prerequisites into the image. It may take some time to finish the loading process as the parcels extend some of the system classes and requires additional processing by VisualWorks. You will be notified when VisualWorks has switched to free-threaded mode during parcel load.

## Changes

DLLCOM changes the threading model of an existing VisualWorks to free-threaded but synchronization is still performed via VisualWorks DLLCC mechanisms. Threaded COM call-outs and all in-bound calls will be routed over DLLCOM. Non-threaded callouts will still go through the original COM primitives provided by the VisualWorks virtual machine.

## Usage

COM Callouts will work as before, except that it is now possible to declare COM functions to be performed threaded, that is they will not block the virtual machine. So other Smalltalk processes may continue to run and even may accept call-ins from other processes. For doing so, place the "__threaded" into the COM pragma in the respective method in the COMInterfacePoiner subclass. The compiler will recognise this flag and perform all calls to this function in an additional thread

## *OLE Mode*

Some OLE Calls like accessing clipboard require COM Connect to switch to OLE mode using the OLEDLL>inOleModeDo:. The OLE mode is a special mode for which VisualWorks has to be reinitialized to single-threading including OLE support. It is only possible to use this mode as long as there are no pending callouts or call-ins. If mode switching fails the method will not execute the block. To handle mode switch failures, there is also a method named #doInOleMode:ifImpossibleDo:.