

# **Using Integrity simulator with IBM® Rational® Rhapsody® Developer for Ada**

## Note

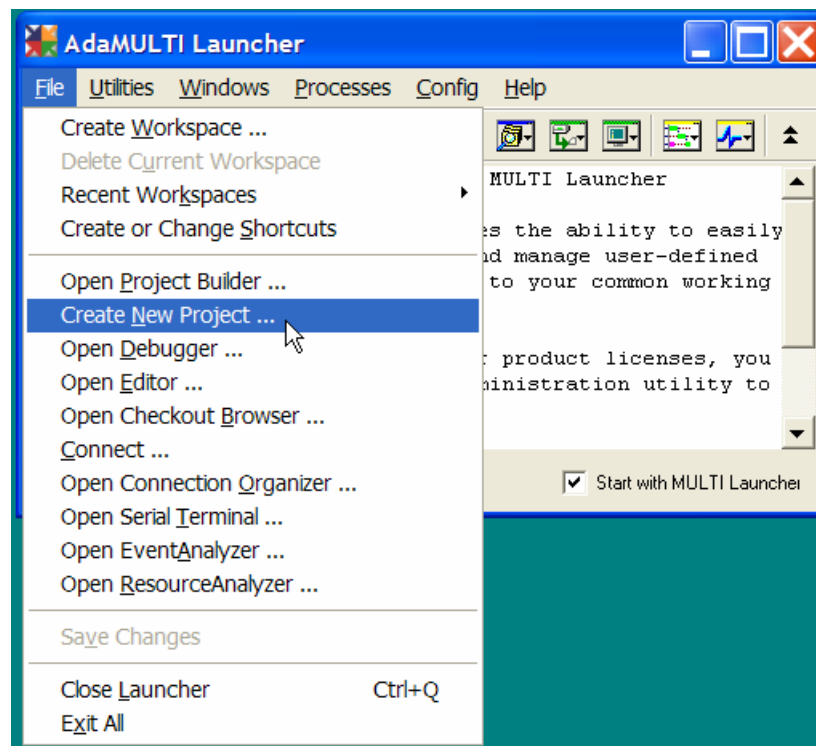
- The instructions in this document have been designed for AdaMulti v4.0.7 and Integrity 5.0.4

## Prerequisites

- Make sure you have followed the steps in section 1.1 « Installation notes » of the « IBM® Rational® Rhapsody® Developper for Ada Code Generator User's Guide »
  - Recompile the code generated from the behavioral framework model for Integrity
  - Recompile the animation libraries for Integrity
  - Note that for the simulator, the target CPU shall be set to « sim800 »
- This is the [RiA\\_CG\\_User\\_Guide.pdf](#) file available under <Rhapsody>\Sodius\RiA\_CG\help directory

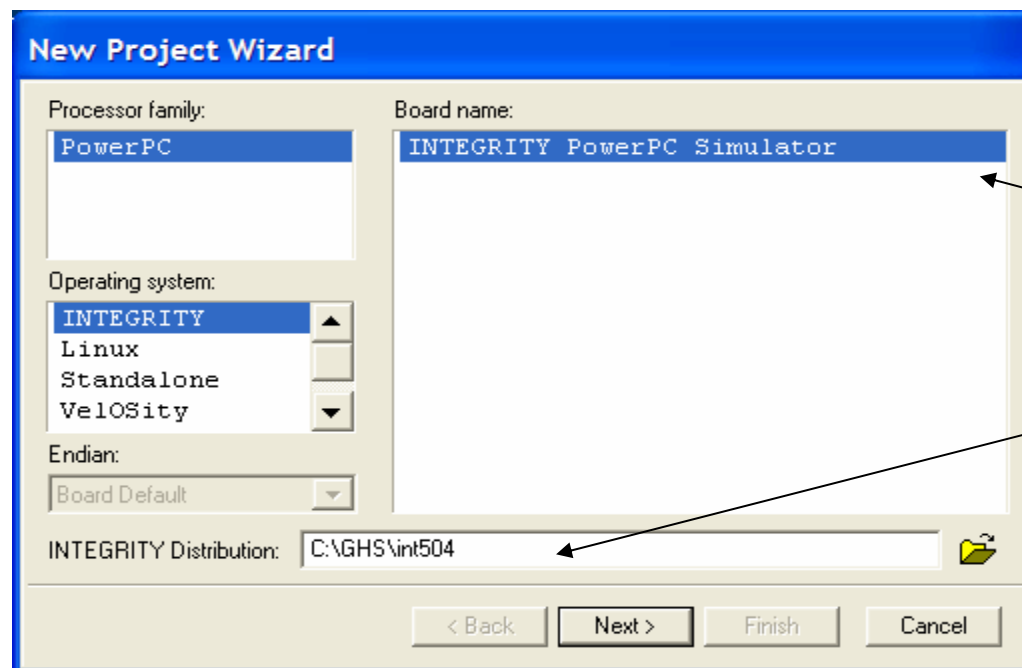
# Building a Kernel

- Before we can load a Rhapsody built application onto the simulator, we must build a Kernel
- Create a new Integrity Project



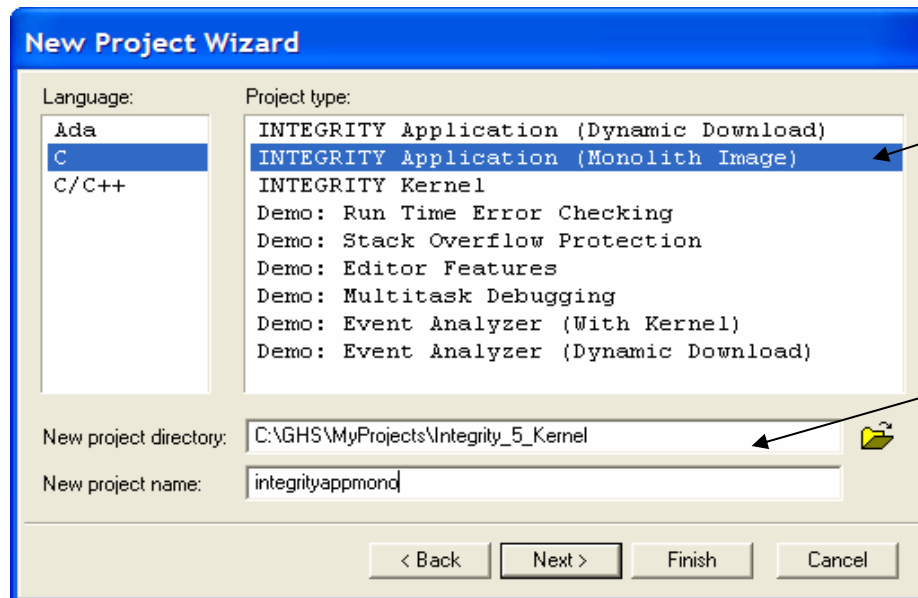
# New Project Wizard

- The New Project Wizard appears.
  - 1. Choose PowerPC/Integrity/Integrity Power PC Simulator BSP
  - 2. Make sure to have the correct path to the Integrity Distribution
- Click Next



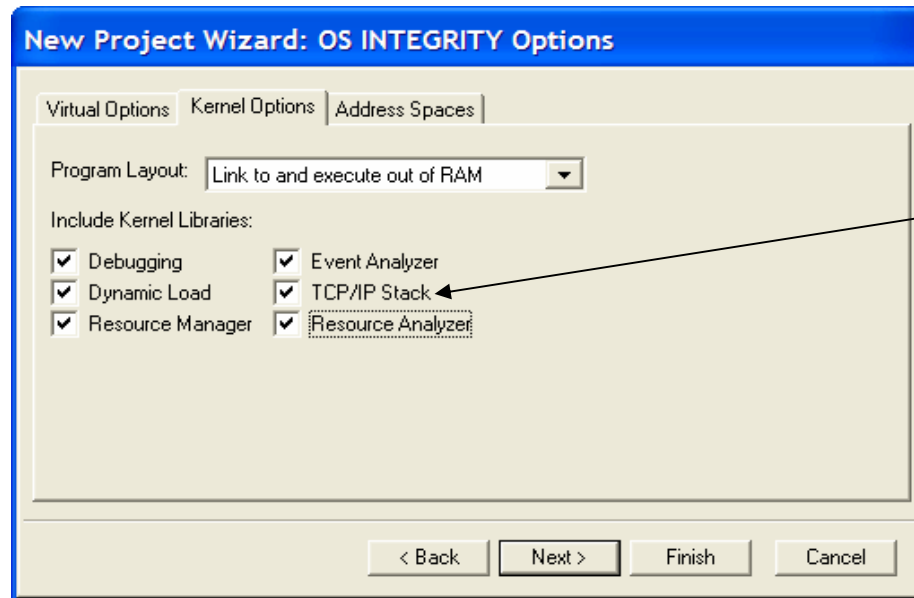
# Create an Image

- Choose Language C (does not matter that your Rhapsody project maybe Ada or C++)
- Choose Monolith Image
- You may want to place the project in a convenient location so you can find it easily before a simulation



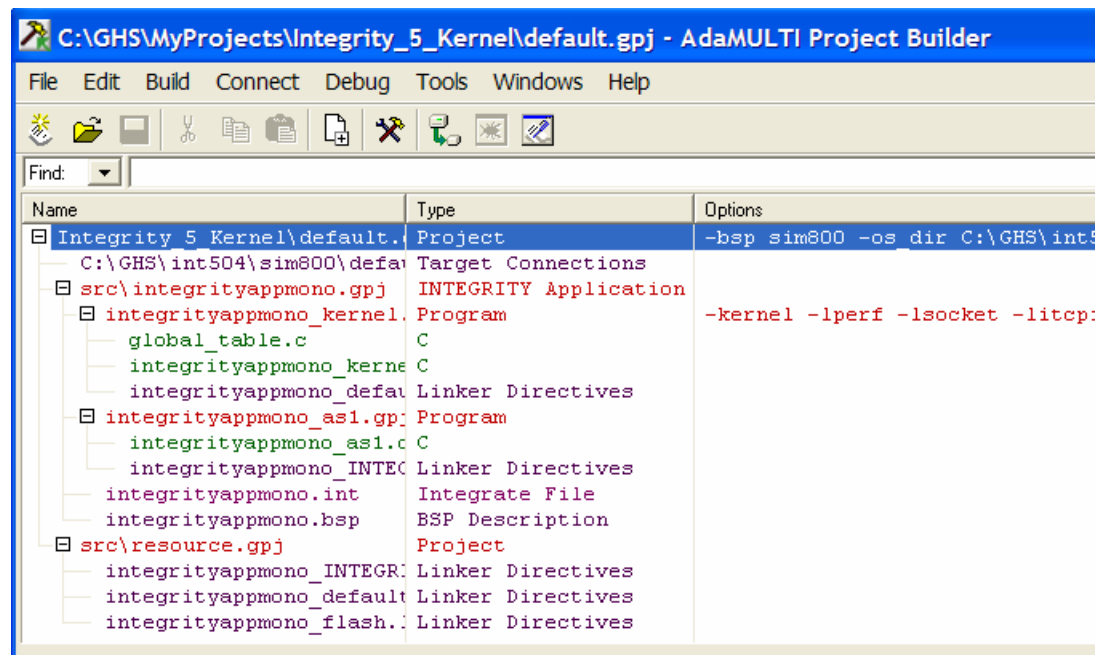
# Kernel Options

- Click Next
- Goto Kernel Options Tab and tick everything
- Click Finish



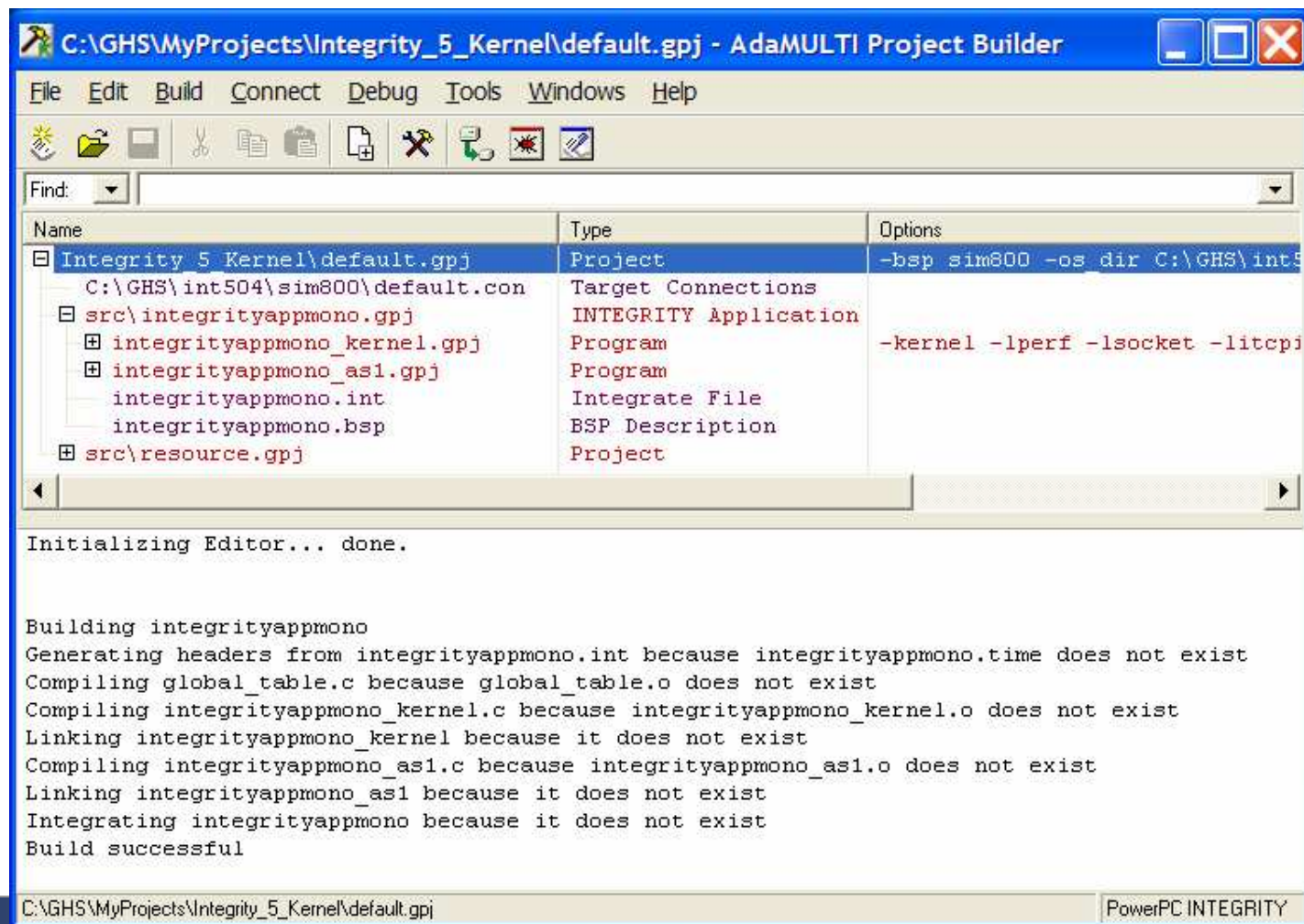
# Kernel Project

- The Builder Window now opens up with this default project containing one address space and a simple hello world app.



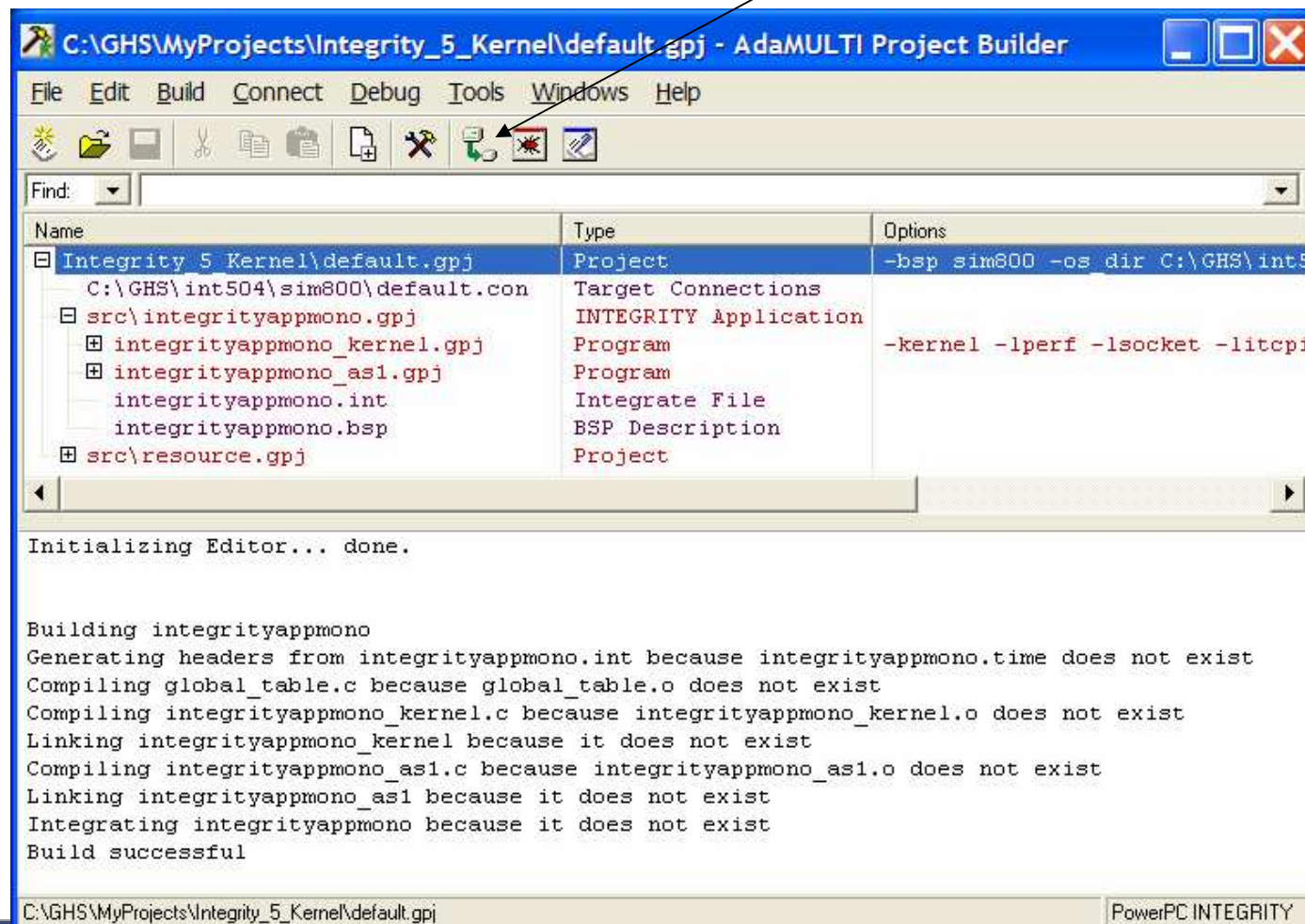
# Build the Kernel

- Build the Kernel from the Build menu (or F7)



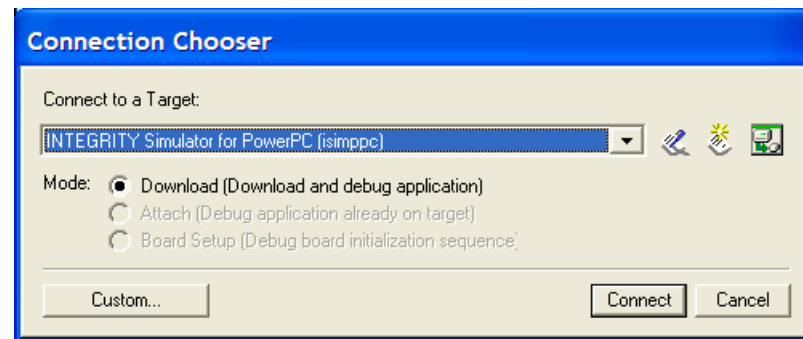
# Connect to Target

- Click Connect on the Toolbar

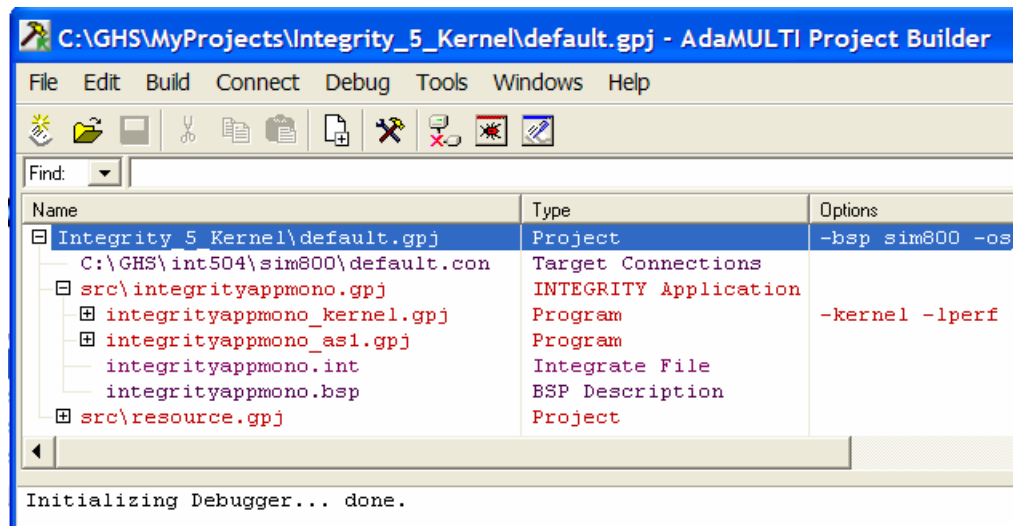


# Connect ..

- When the Connection Chooser appears, click Connect

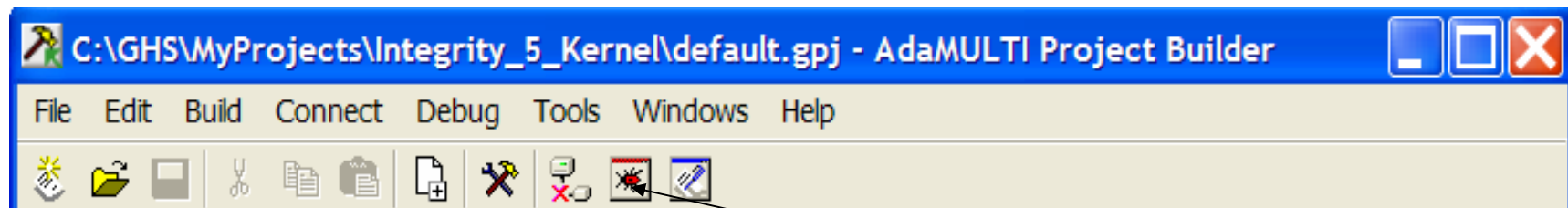


- The Builder window indicates a connection

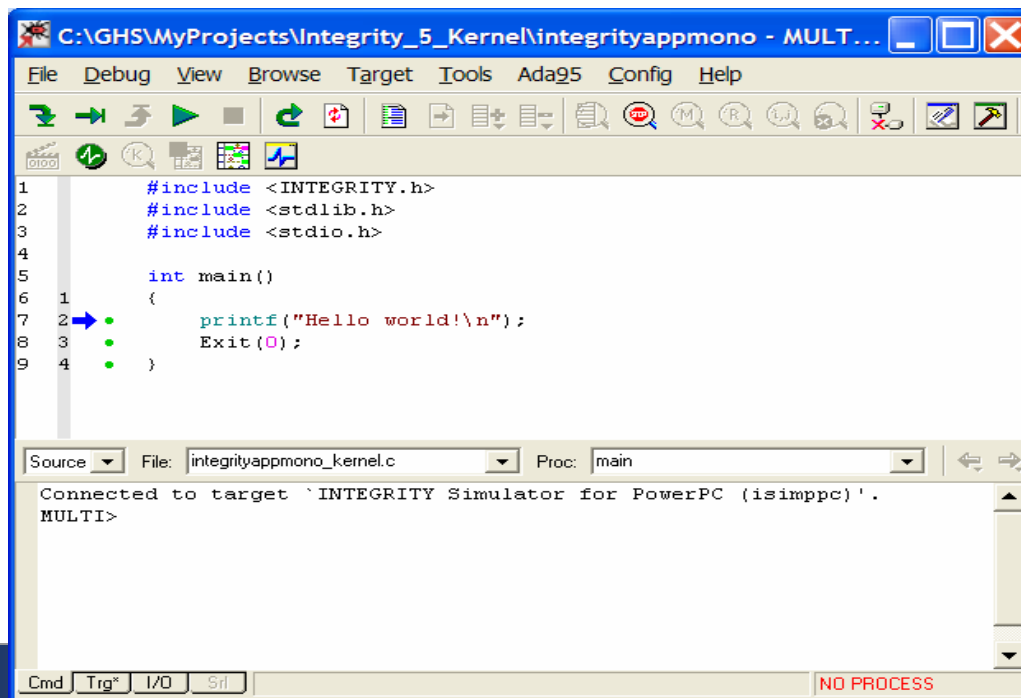


# Loading the Kernel

- In the Builder Window, hit Debug

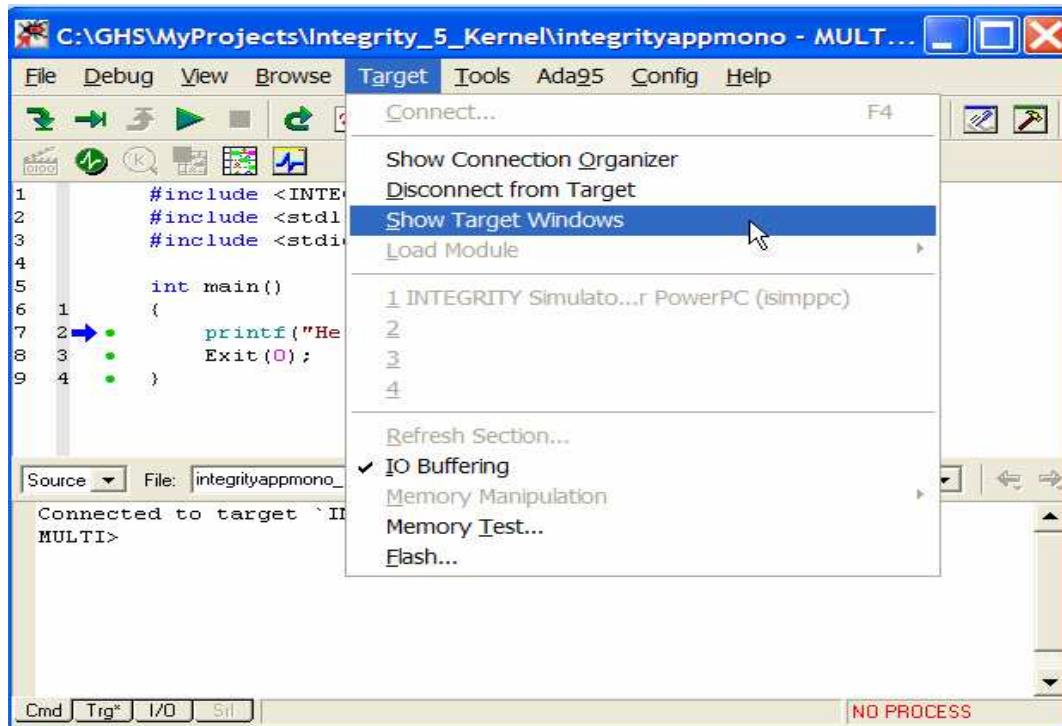


- This opens the default application:



# Open Target Windows

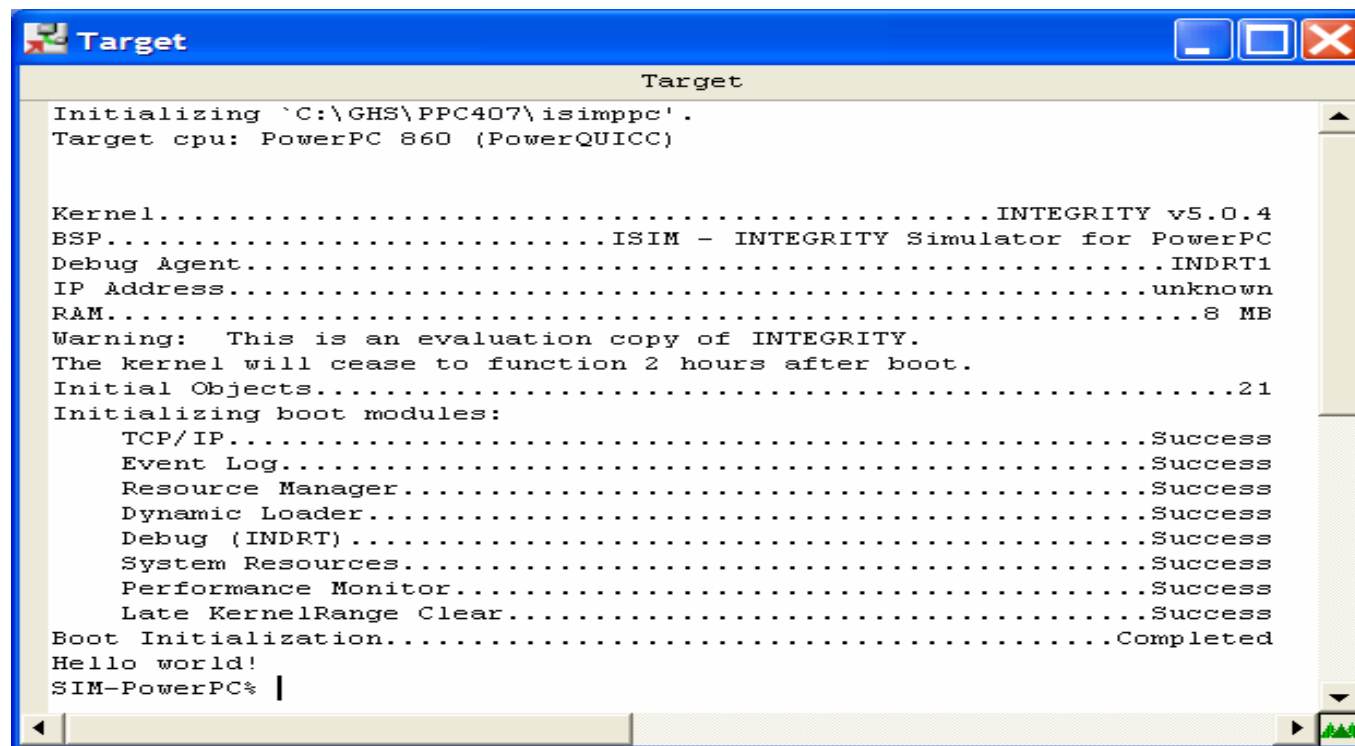
- From the Target Menu, choose Show Target Windows



- A Target Window and an IO window have opened
- The IO is not required so close that window

## Run the Default App

- Back on the Debug window, click Run
- The Target window shows the Kernel loading and the simple Hello World runs



The screenshot shows a window titled "Target" with a standard Windows-style title bar (blue with minimize, maximize, and close buttons). The window contains a text area with the following output:

```
Target
Initializing `C:\GHS\PPC407\isimppc'.
Target cpu: PowerPC 860 (PowerQUICC)

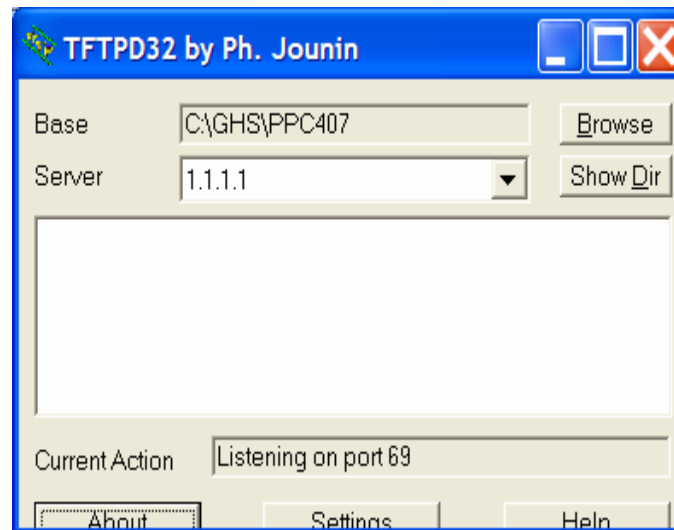
Kernel.....INTEGRITY v5.0.4
BSP.....ISIM - INTEGRITY Simulator for PowerPC
Debug Agent.....INDRT1
IP Address.....unknown
RAM.....8 MB
Warning: This is an evaluation copy of INTEGRITY.
The kernel will cease to function 2 hours after boot.
Initial Objects.....21
Initializing boot modules:
  TCP/IP.....Success
  Event Log.....Success
  Resource Manager.....Success
  Dynamic Loader.....Success
  Debug (INDRT).....Success
  System Resources.....Success
  Performance Monitor.....Success
  Late KernelRange Clear.....Success
Boot Initialization.....Completed
Hello world!
SIM-PowerPC% |
```

## Before a Simulation

- We have used a simple Hello World project we can use to Boot up a Kernel that we can place our Rhapsody projects onto.
- So – before a simulation:
  - Load Multi
  - Open this kernel project.
  - Click Debug and click Run.
  - The Kernel will load
- Thus we have a Kernel, (with one address space containing the hello world). Our Rhapsody project will run on this Kernel in it's own address space.

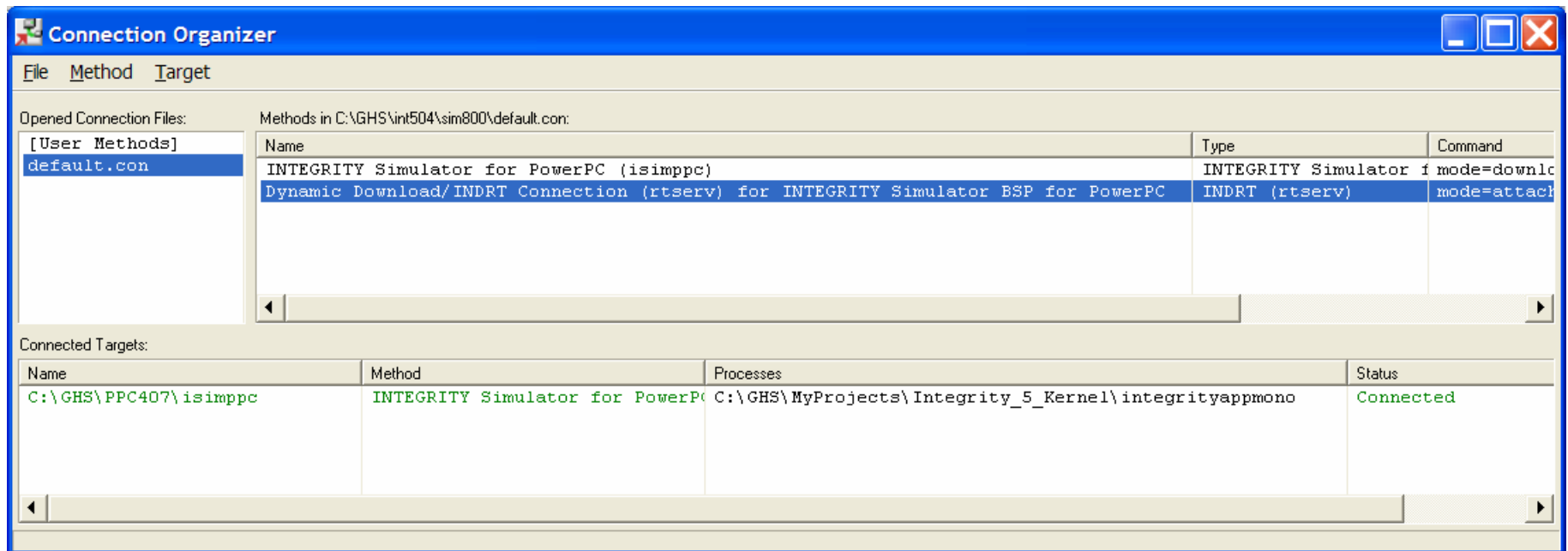
# Starting TFTP

- Start the tftpd32.exe program located in the GHS installation directory.
- This program is used to load executables into the simulator



# Creating a Runtime Debug Server

- First minimise all windows but the Debug window
- From the Target menu, click Show Connection Organiser:



- Click default.con
- Double-Click the INDRT connection (rtserv)

# Host or Target?

- In the Target Name box, enter either your hostname or the IP address of your target.
- Hit Apply
- Hit Connect

**INDRT (rtserv) Connection Editor**

Name:

Type:

☐ Log Connection to file:

Target Setup script:

☒ MULTI ☐ Legacy

Connect for: ☐ Download (Download and debug application)  
☒ Attach (Debug application already on target)  
☐ Board Setup (Debug board initialization sequence)

Connection | Advanced | Debug

☒ Ethernet / IP Connection

Target Name(s) or Address(es) in the form: string1, string2, ...

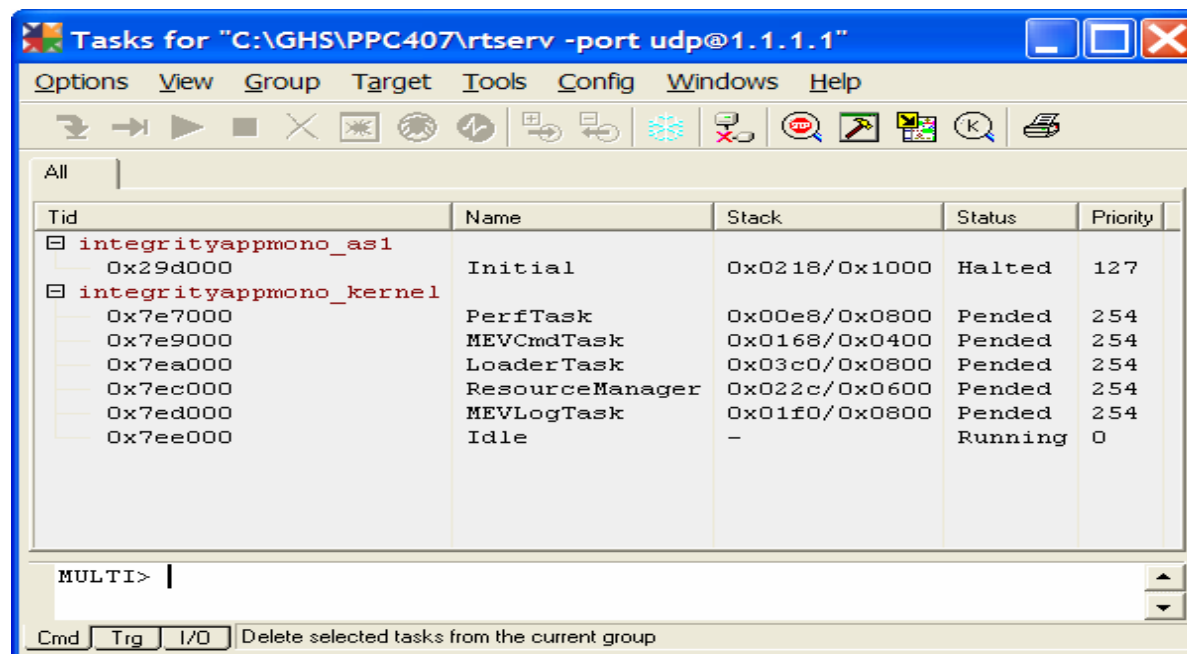
TFTP Load Directory:

☐ Serial Connection

Serial Port:  Baud Rate:

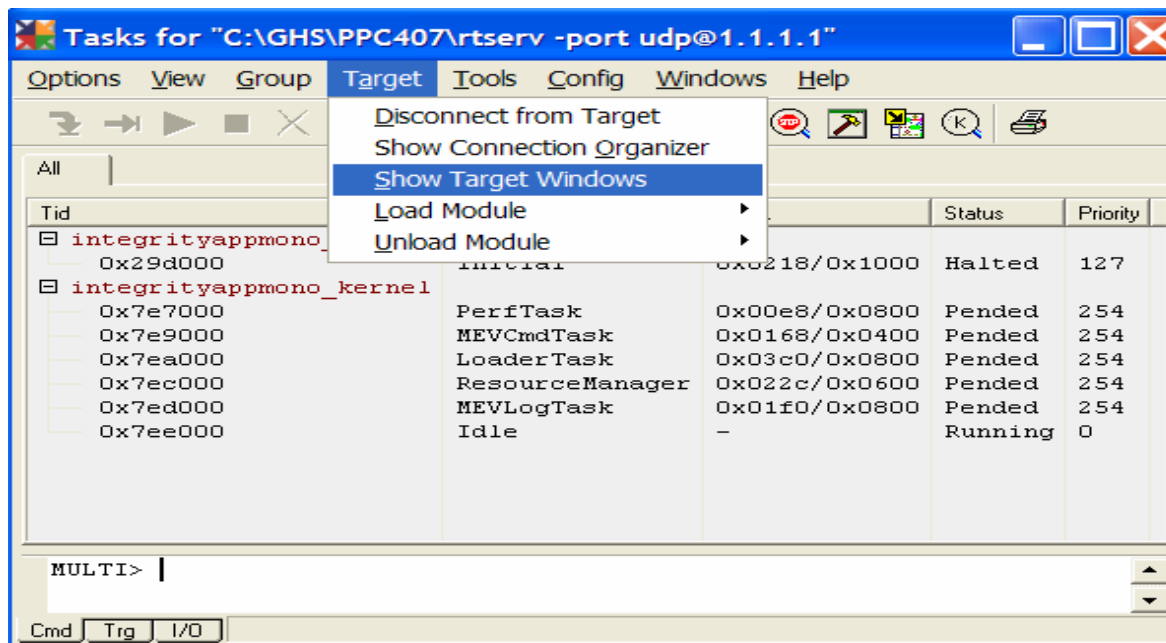
# The Tasks Window

- The Task window should now appear. This is a dynamic view of the tasks on the Kernel at the time. It also serves as a 'front end' to the Runtime Debug Server.



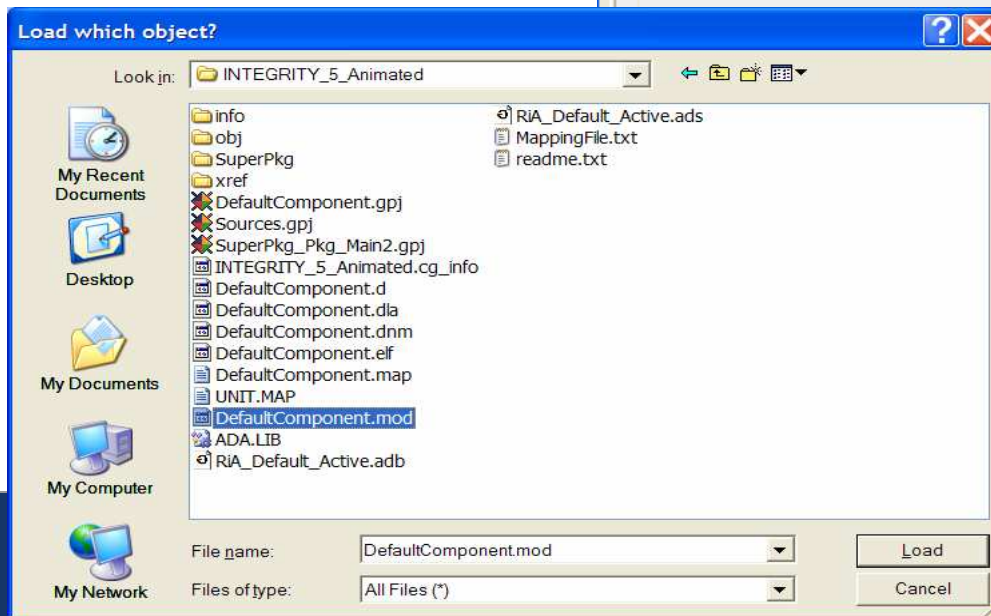
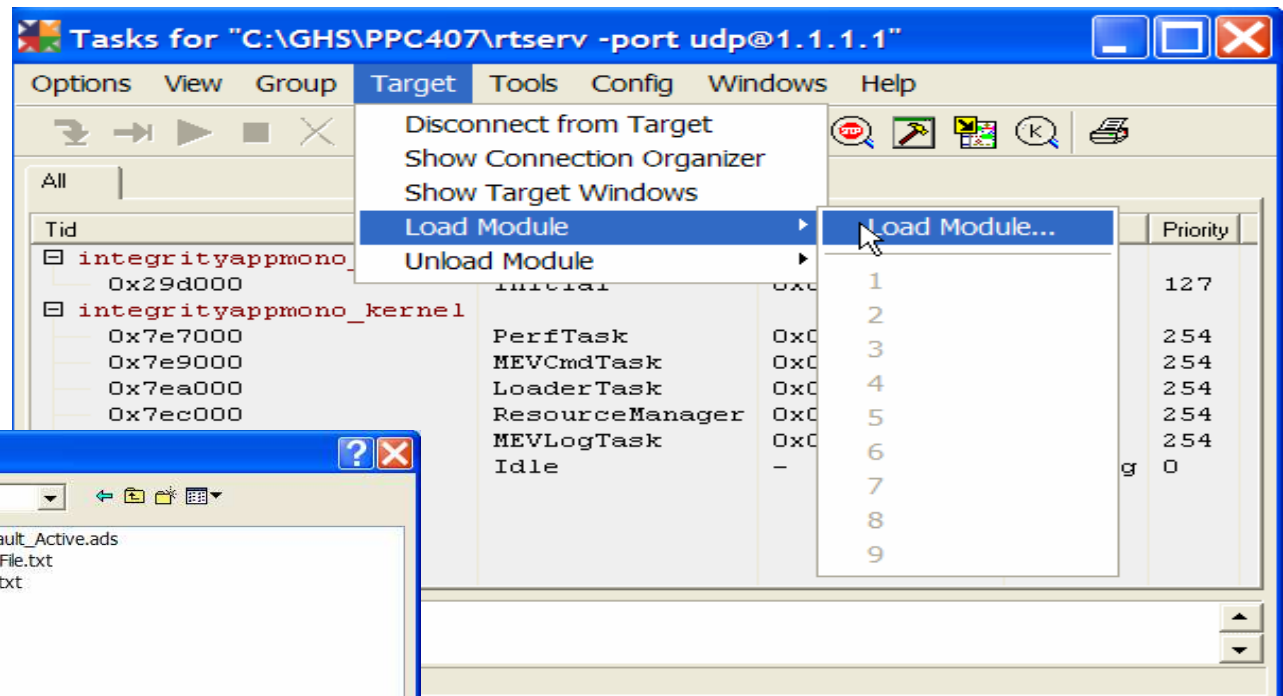
# Target Windows

- From the Target Menu on the Tasks Window, Choose 'Show Target Windows'
- Note that these new windows are for the application – so we will keep the IO window open this time



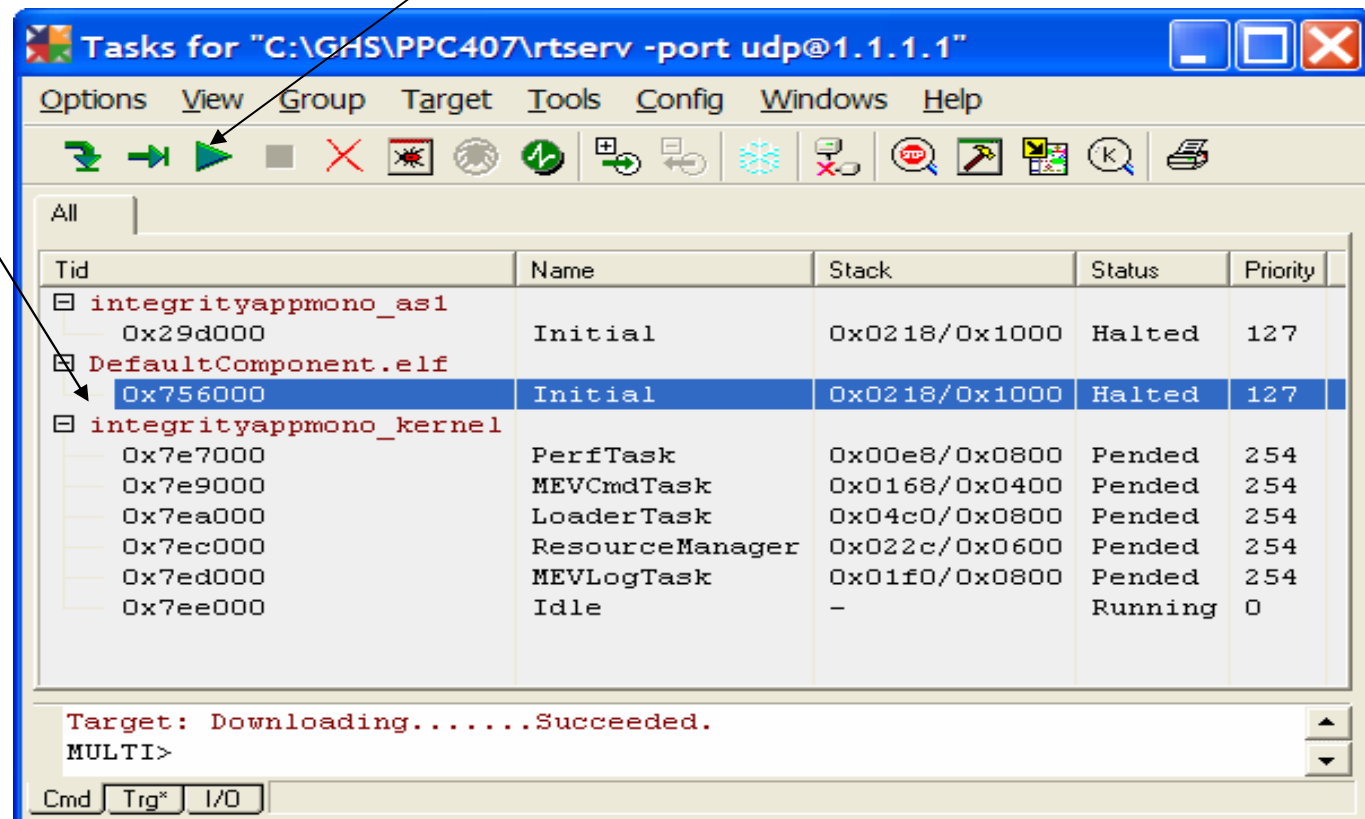
# Load a Module

- From the Target menu – choose Load Module
- Load the .mod file generated by Rhapsody



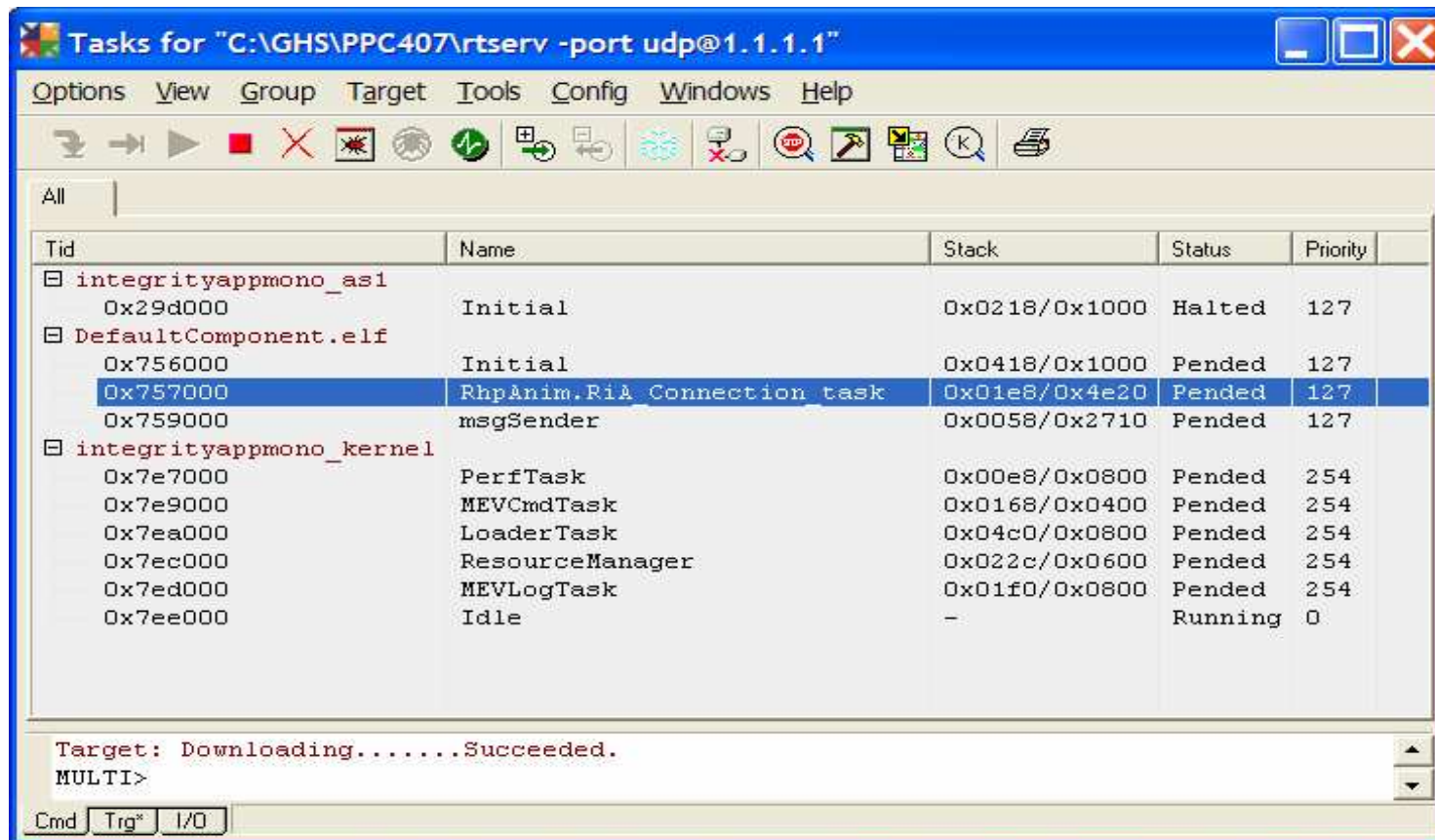
# Starting the module

- A new task shall appear, named after the executable
- Select it and click on 'Continue selected tasks'



# Debug

- If the model is animated, a new task shall appear shortly
- And the animation bar in Rhapsody shall be enabled



# Debug

- Now you can animate the model
- Beware – the Simulator is quite slow

# Resetting the animation toolbar

- Once you're done animating, to reset the animation toolbar, you have to disconnect the kernel from the simulator

