

# **Quick Start Guide: Configuring the BASIS IDE**

April 2009

By

Bruce Gardner

The purpose of this guide is to demonstrate the initial setup and configuration of the BASIS Integrated Development Environment.

## ***Prerequisites***

### ***Downloading and Installing the BASIS IDE***

### ***Overview of Creating Projects in the BASIS IDE***

#### ***Creating a BBJ Project***

- A. Create the BBJ project
- B. Configure the BBJ project

#### ***Creating a (V)PRO/5 Project***

- A. Create the (V)PRO/5 project
- B. Configure the (V)PRO/5 project

# Quick Start Guide: Configuring the BASIS IDE

Based on the NetBeans OpenSource Project, the BASIS Integrated Development Environment (IDE) is a comprehensive set of development and project management tools. The BASIS IDE is bundled with BBJ® and requires BBJ to run. However, the IDE does not restrict the developer to developing only in BBJ; it also supports Visual PRO/5®, PRO/5®, and Java.

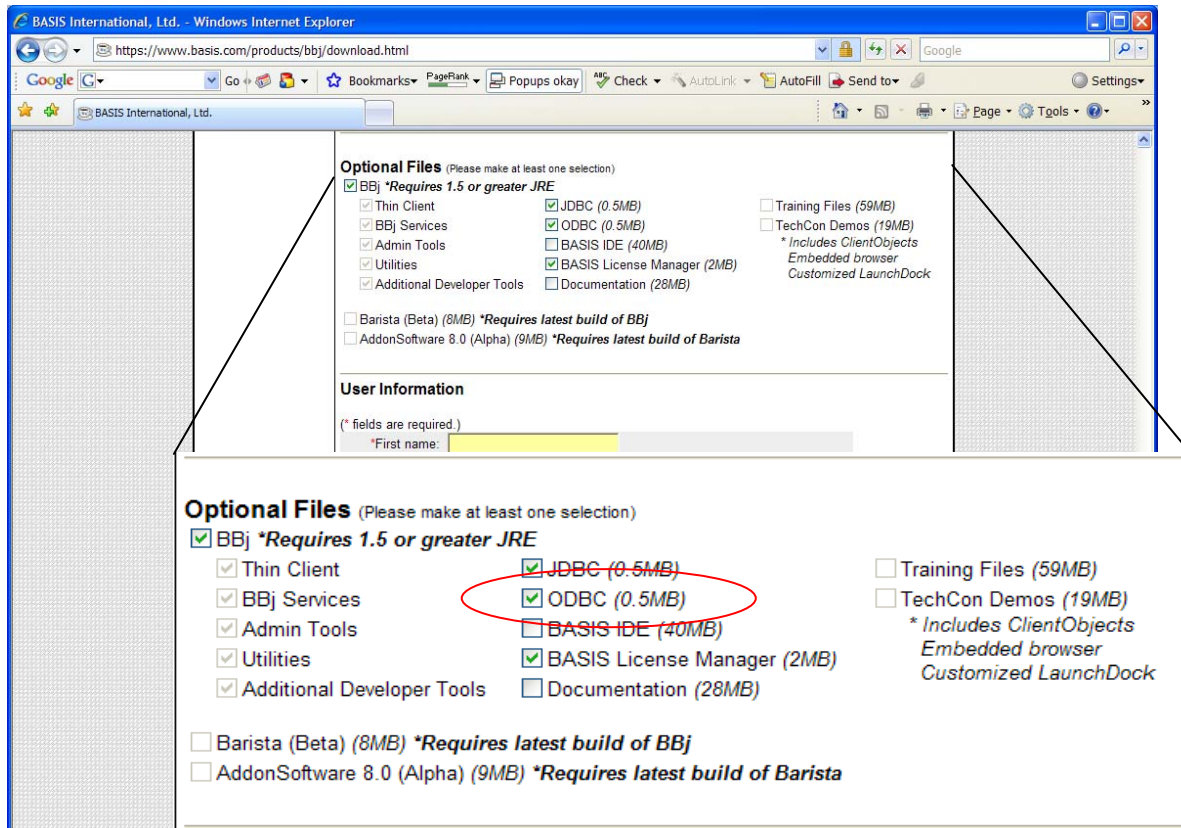
This Quick Start Guide assumes that the BASIS IDE is installed on Windows for the first time for evaluation purposes. To demonstrate the utility and flexibility of the BASIS IDE, this guide creates two example projects – a BBJ project and a Visual PRO/5 or PRO/5 project, later referred to as (V)PRO/5.

## Prerequisites

- BBJ 6.10 or higher with the BASIS IDE (required)
- Java Development Kit 1.5.x (required)
- Visual PRO/5 6.10 or higher (recommended)

## Downloading and Installing the BASIS IDE

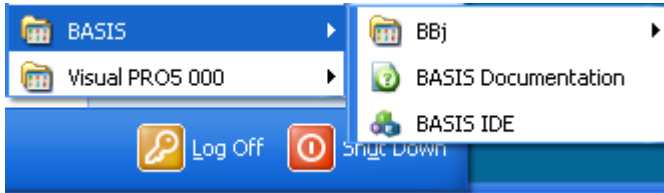
The BASIS IDE is bundled with BBJ and can be downloaded from the BBJ Download page at <https://www.basis.com/products/bbj/download.html>. After selecting the BBJ platform and correct version, check BASIS IDE box in the Optional Files section. Installing the BASIS IDE requires an additional 40MB of disk space (see in **Figure 1**).



**Figure 1.** The default Optional Files section of the BBJ Download page

After downloading the installation package, proceed with the installation. This process installs the BASIS IDE into the <bbjhome>/ide directory.

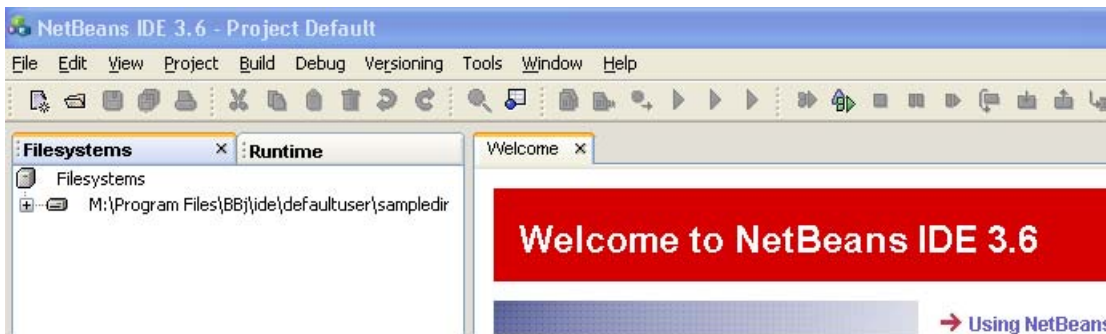
Next, run the BASIS IDE by clicking the BASIS IDE shortcut in the BASIS program group shown below in **Figure 2**.



**Figure 2.** BASIS program group

## Overview of Creating Projects in the BASIS IDE

Everything in the BASIS IDE is organized by projects. Notice that a default sample project already exists when the BASIS IDE is launched for the first time. See **Figure 3**.



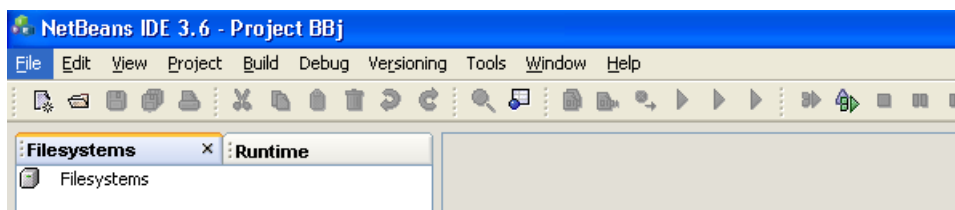
**Figure 3.** The BASIS IDE Filesystem tree and default project directory

Developers can maintain multiple projects within the IDE using separate files systems and separate options settings. This makes it possible to compile (V)PRO/5 code in one project with a (V)PRO/5 compiler and compile BBJ code in a separate project with a BBJ compiler.

## Creating a BBJ Project

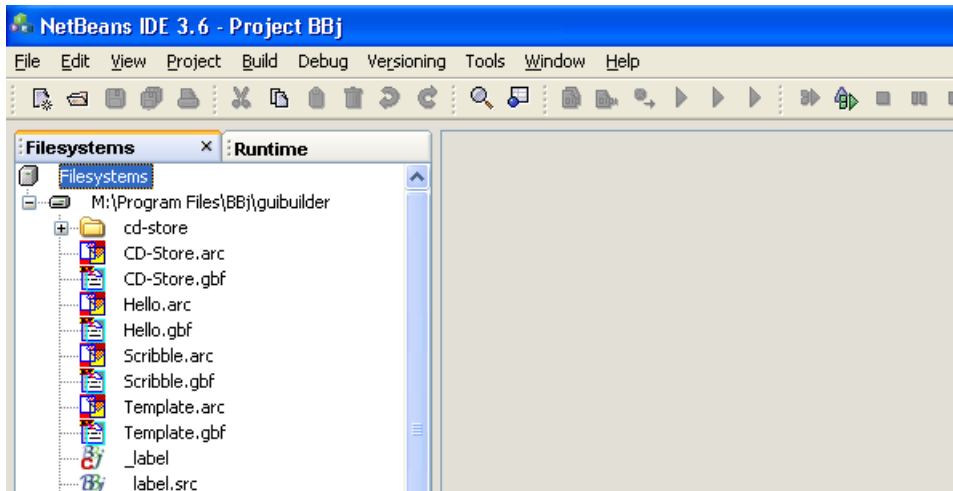
### A. Create the BBJ project

1. From the Project Menu in the BASIS IDE, select Project Manager.
2. In the Project Manager window, click 'New'.
3. In the Create New Project window, enter BBJ as the project name, then click [OK] to display the new project (see Figure 4).



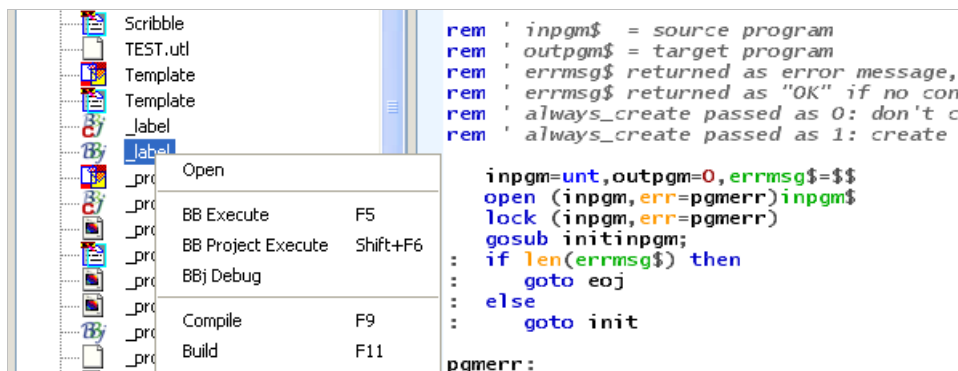
**Figure 4.** New BBJ project with empty filesystem tree

4. To associate a project directory with the BBJ project right-click 'Filesystems,' select 'Mount' from the drop-down menu, then select 'Local Directory' from the submenu. This displays the local directory window.
5. Browse to and select a project directory. For evaluation purposes, mount the <bbjhome>/guibuilder directory, installed with BBJ. The project directory is now associated with the newly created BBJ project shown in **Figure 5**.



**Figure 5.** *Filesystem tree with associated project directory*

Each recognized file type has a specific icon and action associated with it, making it easy to distinguish files at a glance. Right-clicking on a file in the project directory displays a listing all of the possible actions that apply to that particular file type, shown in **Figure 6**.



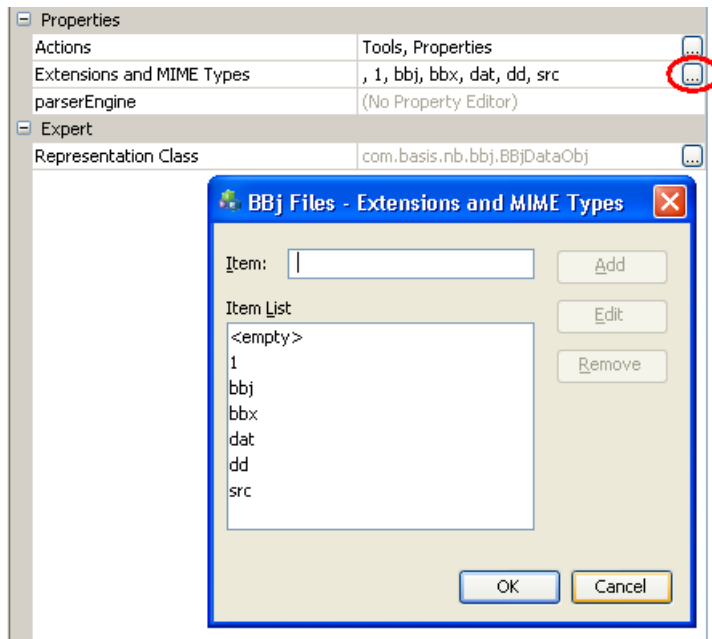
**Figure 6.** *Submenu listing actions associated with file*

Notice in **Figure 6** that the file TEST.utl does not have an associated icon. By default, the BASIS IDE recognizes the extensions .1, .bbj, .bbx, .dat, .dd, .src, and files with no extensions. The headers of recognized files are scanned by the IDE to determine their type. The developer can edit or add to the list of recognized extensions as discussed in the next section.

## B. Configure the BBJ project

### 1. Associate New File Extensions

- a. From the Tools menu, select **Options**.
- b. In the Options window, expand the IDE Configuration tree.
- c. Within the IDE Configuration tree, expand System.
- d. Within the System tree, expand Object Types and double-click the **BBJ Files** module.
- e. In the **Properties** of BBJ files window that now appears, click the ellipse [...] button to the right of **Extensions and MIME Types**, circled in red in **Figure 7**.



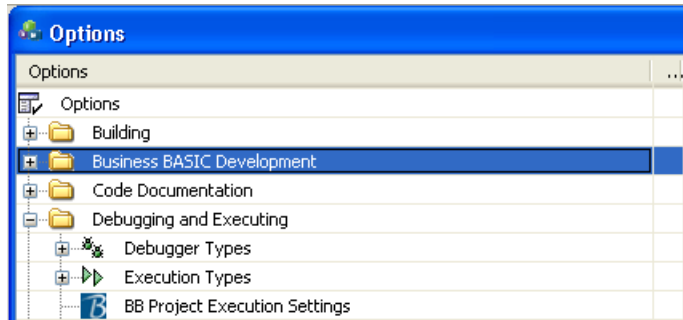
**Figure 7.** The Extensions and MIME Types selected from the Properties of BBJ files window

- f. In the BBJ Files – Extensions and MIME Types window, type the new extension `.utl` in the Item field and click [OK].
- g. Close the Properties of BBJ Files window and the Options window.

**Note:** the BASIS IDE scanned the files with extensions `.utl` in the Filesystem tree and displayed the appropriate icon next to each file.

### 2. Pre-configured BBJ Settings

- a. From the Tools/Options menu, expand **Debugging and Executing**.
- b. Notice in **Figure 8** that this node in the Options tree contains options settings for both (V)PRO/5 and BBJ projects.



**Figure 8.** The Options window showing BBJ and (V)PRO/5 settings

Since the BASIS IDE is bundled with BBJ, all BBJ-specific settings within the Tools/Options menu are pre-configured. For (V)PRO/5 options settings, refer to the next section, **Creating a (V)PRO/5 Project**.

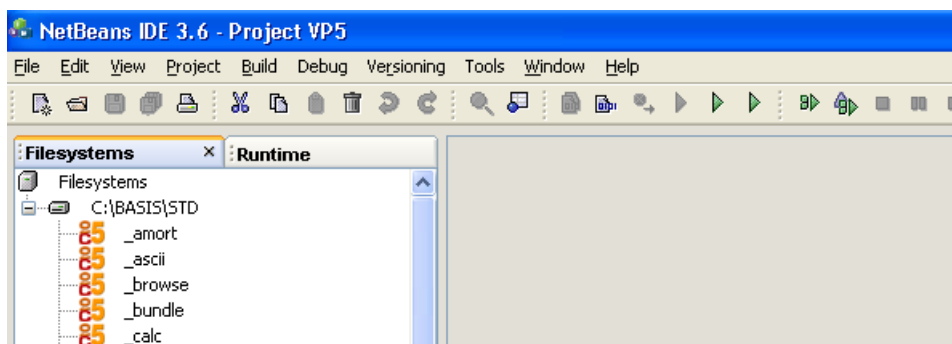
## Creating a (V)PRO/5 Project

### A. Create the (V)PRO/5 project

1. From the Project Menu in the BASIS IDE, select Project Manager.
2. In the Project Manager window, click [New].
3. In the Create New Project window, enter VP5 for the project name. Click [OK].
4. Associate a project directory with the VP5 project by right-clicking **Filesystems**. Select **Mount** from the drop-down menu, then **Local Directory** from the submenu.
5. The Local Directory window displays. Browse to and select a project directory.

**Note:** For evaluation purposes, mount the standard utilities directory <vpro5home>/STD, installed with (V)PRO/5.

Once again, the IDE automatically scans the VP5 project folder and displays the appropriate icons and actions associated with known file types illustrated in **Figure 9**.



**Figure 9.** Compiled BBx programs listed in the Filesystem tree

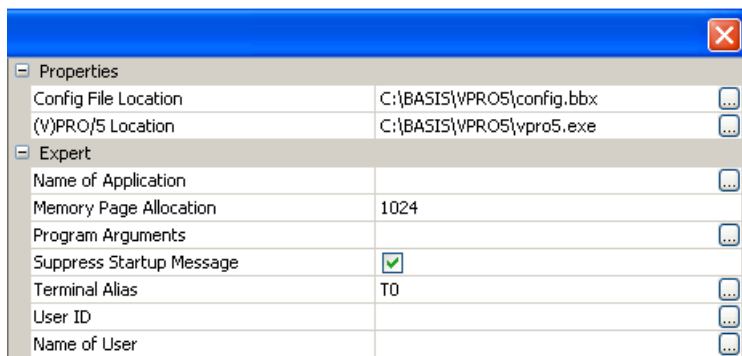
## B. Configure the (V)PRO/5 project

Since the BASIS IDE is not bundled with (V)PRO/5, it is necessary to configure (V)PRO/5 options settings to point to key (V)PRO/5 binaries.

### 1. Set Debugging and Executing Settings

**Note:** If needed, add new file extensions to the VP5 project as discussed in **Configure the BBj Project – Associate New File Extensions**.

- a. From the Tools/Options menu, expand **Debugging and Executing**.
- b. Double-click **(V)PRO/5 Execution Settings**.
- c. Click the ellipse [...] button next to Config File Location and browse to the location of the VPRO/5 config.bbx file.
- d. Click the ellipse [...] button next to (V)PRO/5 location to browse to the location of the vpro5.exe file. See **Figure 10**.



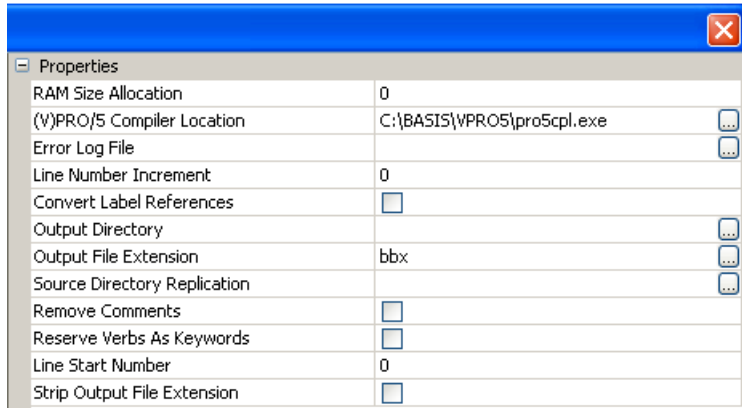
**Figure 10.** (V)PRO/5 Execution Settings property sheet

The Expert properties in this sheet correspond to the following (V)PRO/5 command line arguments:

- Name of Application – an application name for the `-a` argument
- Memory Page Allocation – pages of memory to be allocated with the `-m` argument
- Program Arguments – arguments passed to the (V)PRO/5 program being executed
- Suppress Startup Message – if checked, suppresses the (V)PRO/5 splash screen
- Terminal Alias – default terminal alias defined in the config file with the `-t` argument
- User ID – user account for the `-u` argument
- Name of User – A user name for the `-n` argument

### 2. Set Compiler Building Settings

- a. From the Tools/Options menu, expand **Building**.
- b. Double-click **(V)PRO/5 Compiler Settings**.
- c. Click the ellipse [...] button next to **(V)PRO/5 Compiler Location** to browse to and select the `pro5cpl.exe` file. See **Figure 11**.



**Figure 11.** (V)PRO/5 Compiler Settings property sheet

The properties in this sheet correspond to the following (V)PRO/5 compiler parameters:

- RAM Size Allocation – the number of KB of RAM to hold the program.
- Error Log File – fully qualified path and filename of error log
- Line Number Increment – numerical increment between line numbers
- Convert Label References – if checked, converts label references to line number references
- Output Directory – the directory where compiled files will be placed.
- Source Directory Replication – controls the replication of source directories in the output directory structure
- Remove Comments – if checked, removes comments from compiled files
- Reserve Verbs As Keywords – if checked, verbs will be reserved words
- Line Start Number – the beginning line number is a numbered program
- Strip Output File Extension – if checked, the output file's last extension is removed

### 3. Set Lister Building Settings

- a. Double-click (V)PRO/5 Lister Settings.
- b. Click the ellipse button next to (V)PRO/5 Lister to browse to the location of the `vpro5lst.exe` file.

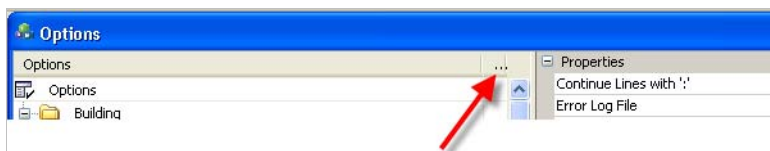
Additional properties in this sheet correspond to the following (V)PRO/5 Lister parameters:

- Continue Lines with ':' – if checked, long lines will use the ':' continuation
- Flow Listing Print Formatting – if checked, produces a flow (indented) listing
- Output Directory – the directory where the text output files will be placed
- Convert Label References – if checked, converts label references to line number references
- Output Directory – the directory where compiled files will be placed
- Source Directory Replication – controls the replication of source directories in the output directory structure
- Pretty Print Formatting – if checked, produces a pretty print formatted listing
- Suppress Line Numbers – if checked, output files will contain no line numbers
- Line Number Width – the number of columns in a line

4. Set Business BASIC Development Settings
  - a. From the Tools/Options menu, expand Business BASIC Development.
  - b. Double-click Business BASIC Types.
  - c. From the drop-down list, next to 'Business BASIC Language,' select (V)PRO/5.
5. Set Level Settings

Now that the VP5 project is configured, it is necessary to ensure that the setting changes just made do not affect the previously configured BBJ project.

- a. Click the view levels ellipse [...] button column at the top of the Tools/Options window shown in **Figure 12**.

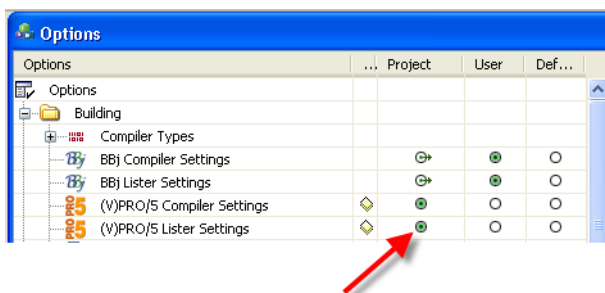


**Figure 12.** The 'View Levels' button, in the Options window

- b. The View Levels column expands to reveal the default level of all options set within the BASIS IDE.

**Note:** By default, most options are set to the 'user' level, which means the changes made will affect all projects within the IDE.

- c. Click the radio button under the **Project** column for each of the following Options Settings to lock the setting to the VP5 project. See **Figure 13**.
  - (V)PRO/5 Compiler Settings
  - (V)PRO/5 Lister Settings
  - (V)PRO/5 Execution Settings
  - Business BASIC Types



**Figure 13.** The expanded View Levels column showing level settings for BBJ and (V)PRO/5 settings.

- d. Close the Options window.

Configuration of the BASIS IDE is complete.