EE218: Computer Org. & Assembly Language

# Lab#02: Installation & Configuration of TextPad Editor

## Agenda

## Installation of TextPad Editor for Assembly language Programming

## Configuring TextPad Editor for Assembly Programming

### 16-Bit Real Mode Programs

### Build 16-Bit MASM

* + - Run 16-Bit MASM

### 32-Bit Protected Mode Programs

### Build 32-Bit MASM

* + - Run 32-Bit MASM
* Getting started with Assembly Programming
  + 16-bit Real Mode Program Vs 32-Bit Protected Mode Program
  + Comments in Assembly Language
    - Multi Line Comments

## Installation of TextPad Editor for Assembly language Programming

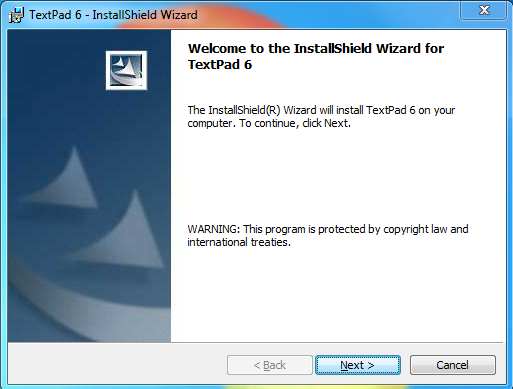
In this short tutorial, we will show you how to add commands to the Tools menu in TextPad, so you can easily assemble and link your assembly language programs.

Download the latest version of TextPad from: <http://www.textpad.com/>

Or get it from the lectures server.

In this tutorial we will be uisng TextPad6.

Run setup.exe, this will install the TextPad6 Editor.



After the installation is complete, the next step is configure it to assemble and link assembly language programs.

## Configuring TextPad4 Editor for Assembly Programming

### 16-Bit Real-Mode Programs

#### Step 1: Build 16-Bit MASM

1. Select Preferences from the Configure menu.
2. Select Tools in the left hand pane, and click the Add button.



1. Select "DOS command..." from the drop-down list.
2. When the popup dialog appears, enter the following and click the OK button:

**C:\Masm615\make16.bat $BaseName**



1. Click the Apply button.
2. Next, you will rename the command to Build 16-Bit MASM. To do this,
3. Click once on Tools in Left Pane, you will see the command entered in the precious step in Right Pane.



* Then click name,
* Wait a second,
* Click again. When it turns blue,
* Retype the command's name and press Enter.

1. Click on Apply button.
2. You will see this command in the Left Pane under Tools Section well as in the Right Pane as in the figure below.



1. Then click on Build 16-Bit MASM command in Left Pane.
2. You will see a panel as in the figure below.



1. In the right-hand pane, if Initial folder entry is empty or written anything else then enter the following:

**$FileDir**

1. Finally click OK button to save changes

**Note**

Uncheck the “Capture Output” check boxes for user friendly compilation and execution as a novice user. If you check the “Capture output” option, then execution and assembler's output messages would be in a TextPad Editor’s window.

#### Step 2: Run 16-Bit MASM

1. As you have already created one command in the Tools menu, we can speed things up a bit.
2. b) Select Preferences from the Configure menu
3. Select Tools in the left hand pane, and click the Add button on the right side same as in above steps.
4. Select” DOS command...” from the drop-down list. When the popup dialog appears, enter the following:

**$BaseName**

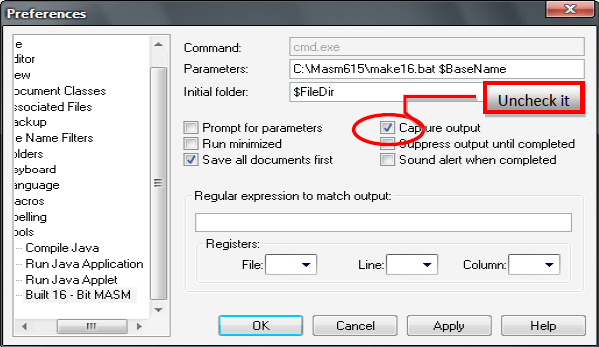
1. Click on OK and Apply button.



1. Next, rename the command to” Run ASM 16 – Bit”. To do this, click once on the name, wait a second, and click again. When it turns blue, retype the command's name and press Enter. Click Apply same as in above steps.



1. Click on the "+" next to Tools in the left-hand pane. When the list expands, select “Run ASM 16 – Bit”. In the right hand pane of the dialog, remove the check mark next to Capture output.



**Note**

Uncheck the “Capture Output” check boxes for user friendly compilation and execution as a novice user.

If you check the “Capture output” option, then execution and assembler's output messages would be in a TextPad4 Editor’s window.

1. Finally click OK button to save changes

Now you have successfully configured “TextPad Editor” for 16 – Bit Programs of Real Mode.

### 32-Bit Protected-Mode Programs

#### Step 1: Build 32-Bit MASM

1. Select Preferences from the Configure menu.
2. Select Tools in the left hand pane, and click the Add button.



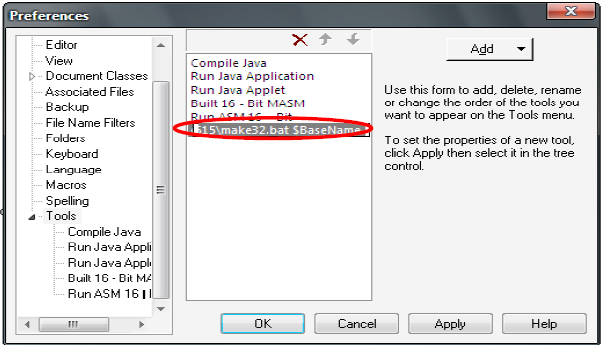
1. Select "DOS command..." from the drop-down list.
2. When the popup dialog appears, enter the following and click the OK button:

**C:\Masm615\make32.bat $BaseName**



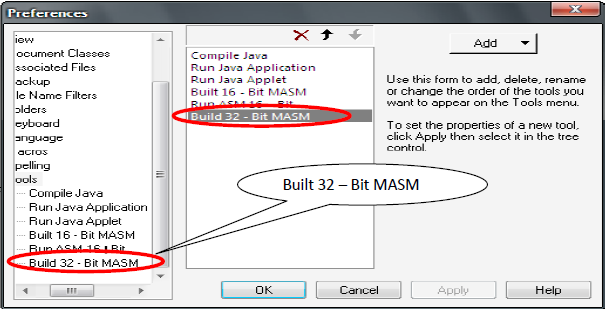
1. Click the Apply button.
2. Next, you will rename the command to Build 32-Bit MASM. To do this,

* Click once on Tools in Left Pane, you will see the command entered in the precious step in Right Pane.



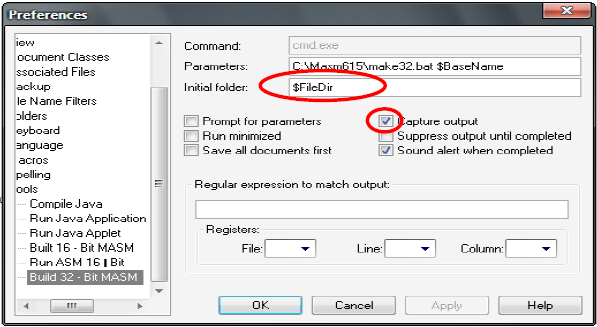
* Then click name,
* Wait a second,
* Click again. When it turns blue,
* Retype the command's name and press Enter.

1. Click on Apply button.
2. You will see this command in the Left Pane under Tools Section well as in the Right Pane as in the figure below.



1. Then click on Build 32-Bit MASM command in Left Pane.
2. You will see a panel as in the figure below.
3. In the right-hand pane, if Initial folder entry is empty or written anything else then enter the following:

**$FileDir**



1. Finally click OK button to save changes

**Note**

Uncheck the “Capture Output” check boxes for user friendly compilation and execution as a novice user.

If you check the “Capture output” option, then execution and assembler's output messages would be in a TextPad Editor window.

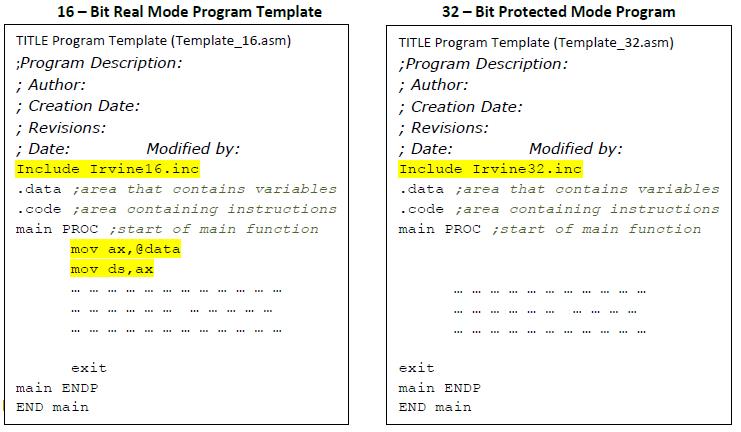
#### Step 2: Run 32-Bit MASM

Same procedure for running 16-Bit MASM programs.

## Getting Started with Assembly Programming

### 16 – Bit Real Mode Program vs. 32 – Bit Protected Mode Program

As there is a difference between Real Mode and Protected Mode, in the same way there is structural difference between these both mode programs using Irvine’s Libraries.



These are templates for both types of programs. Highlighted lines are different in the both type of programs. This structure has to follow for corresponding modes. These templates are prepared by using Irvine Libraries.

**Note**

Semicolon (;) is used to comment the statement.

#### Comment in Assembly Language

Comments are good and recommended in programming for best understanding and readability of the code. You can explain the program's purpose, creation date and many more.

Single Line Comments

Semicolon (;) is used for single line comment

##### **Multi-Line Comments**

It begins with COMMENT directive and a programmer-chosen character end with the same programmer-chosen character. E.g.

COMMENT @

«««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Hello World In Assembly using TextPad \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_This is Lab # 02 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Creation Date. 03.09.2013 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««««

@

## Tasks

Task # 1. Show the successful configuration of TextPad Editor with Masm.

Task # 2. Assemble & Link some sample examples