Standard C Preprocessors

In C programming, a preprocessor directive is a command that instructs the C preprocessor to perform specific tasks before the actual compilation of the source code. Preprocessor directives start with the **#** symbol and are processed by the preprocessor before any compilation occurs. They are not part of the regular C syntax and do not end with a semicolon.

Here are some common preprocessor directives in C:

1. **#define Directive:**

   - **Purpose:** Defines a macro, which is a symbolic name representing a sequence of code or a value.
   - **Example:**

   ```
   #define PI 3.14159
   ```

2. **#include Directive:**

   - **Purpose:** Includes the contents of another file in the current source file.
   - **Example:**

   ```
   #include <stdio.h>
   ```

3. **#ifdef, #ifndef, #else, #endif Directives:**

   - **Purpose:** Conditional compilation based on whether a macro is defined (**#ifdef**), not defined (**#ifndef**), or for an alternative block of code (**#else** and **#endif**).
   - **Example:**

   ```
   #ifdef DEBUG
       // Debugging code
   #else
      // Release code
   #endif
   ```

4. **#undef Directive:**

   - **Purpose:** Undefines a previously defined macro.
   - **Example:**

   ```
   #define PI 3.14159
   #undef PI
   ```

5. **#pragma Directive:**

- **Purpose:** Provides implementation-specific instructions to the compiler.
- **Example:**

  ```
  #pragma warning(disable : 1234)
  ```

6. **Stringizing Operator (#):**
   - **Purpose:** Converts a macro parameter into a string literal.
   - **Example:**

     ```
     #define STRINGIZE(x) #x
     printf("Value of x: %s\n", STRINGIZE(42));
     ```

7. **Token Pasting Operator (##):**
   - **Purpose:** Concatenates two tokens into a single token.
   - **Example:**

     ```
     #define CONCAT(a, b) a ## b
     int xy = CONCAT(10, 20); // Results in int xy = 1020;
     ```

8. **#error Directive:**
   - **Purpose:** Generates a compilation error with a specified error message.
   - **Example:**

     ```
     #ifndef SUPPORTED_COMPILER
             #error This compiler is not supported
     #endif
     ```

Preprocessor directives play a crucial role in code organization, configuration management, and conditional compilation in C programs. They are processed by the preprocessor before the compiler proper starts its work, allowing developers to control various aspects of the compilation process and customize the behavior of their programs.