

# Archived:Creating temporary files in Open C

 Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template `{{ReviewForRemovalFromArchive|user=~~~~~|write your reason here}}`.

## Overview

Programs often need to create temporary files just for the life time of the program. In Open C `tmpnam()` and `tmpfile()` functions exist to assist in this task.

- `tmpnam()` generates file names that can be used for a temporary file.
- `tmpfile()` creates a temporary file and opens a corresponding stream to the created file.

The `tmpnam()` and `tmpfile()` functions return a pointer to a file name on success or NULL pointer in case of an error.

**Note:** In order to use this code, you need to install the [Open C plug-in](#).

This snippet can be self-signed.

## MMP file

The following libraries are required:

```
LIBRARY libc.lib
```

## Source file

```
#include <stdio.h> //fprintf, tmpnam, tmpfile, FILE
#include <sys/stat.h> //S_IWUSR

int main (void)
{
    char *tmp_pathname = 0;
    char buffer[100];

    FILE *tmp_fileptr = 0;
    FILE *tmp_filestream = 0;

    /* create the tmp directory */
    mkdir("c:\\tmp", S_IWUSR);

    /* - tmpnam - */
    if (!(tmp_pathname = tmpnam(NULL)))
    {
        perror("Error creating temporary filename!");
        abort();
    }

    fprintf(stdout, "Temporary pathname %s\n", tmp_pathname);

    if (!(tmp_fileptr = fopen(tmp_pathname, "w")))
    {
        perror("Error opening temporary file");
        abort();
    }
}
```

```
    }
else
    {
        fclose(tmp_fileptr);
        remove(tmp_pathname);
    }

/* - tmpfile - */
if (!(tmp_filestream = tmpfile()))
    {
        perror("Error generating temporary stream!");
        abort();
    }

/* write to temporary file */
fprintf(tmp_filestream, "Temporary stream created by PID[%ld]", (long)getpid());
fflush(tmp_filestream);

/* read from temporary file */
rewind(tmp_filestream);
fgets(buffer, 100, tmp_filestream);
fprintf(stdout, "Temporary stream: %s\n", buffer);

if(tmp_filestream)
    fclose(tmp_filestream);

return 0;
}
```

## Postconditions

---

Two temporary files are created. The `tmpfile()` created file is automatically deleted by the OS when all references to the file are closed.