

entry:

```
%dest1 = getelementptr inbounds %struct.jpeg_compress_struct,  
... %struct.jpeg_compress_struct* %cinfo, i64 0, i32 5  
%0 = bitcast %struct.jpeg_destination_mgr** %dest1 to  
... %struct.my_destination_mgr**  
%1 = load %struct.my_destination_mgr*, %struct.my_destination_mgr** %0,  
... align 8, !tbaa !3  
%buffer = getelementptr inbounds %struct.my_destination_mgr,  
... %struct.my_destination_mgr* %1, i64 0, i32 2  
%2 = load i8*, i8** %buffer, align 8, !tbaa !11  
%outfile = getelementptr inbounds %struct.my_destination_mgr,  
... %struct.my_destination_mgr* %1, i64 0, i32 1  
%3 = load %struct._IO_FILE*, %struct._IO_FILE** %outfile, align 8, !tbaa !15  
%call = tail call i64 @fwrite(i8* %2, i64 1, i64 4096, %struct._IO_FILE* %3)  
%cmp = icmp eq i64 %call, 4096  
br i1 %cmp, label %if.end, label %if.then, !prof !16
```

T

F

if.then:

```
%err = getelementptr inbounds %struct.jpeg_compress_struct,  
... %struct.jpeg_compress_struct* %cinfo, i64 0, i32 0  
%4 = load %struct.jpeg_error_mgr*, %struct.jpeg_error_mgr** %err, align 8,  
... !tbaa !17  
%msg_code = getelementptr inbounds %struct.jpeg_error_mgr,  
... %struct.jpeg_error_mgr* %4, i64 0, i32 5  
store i32 36, i32* %msg_code, align 8, !tbaa !18  
%error_exit = getelementptr inbounds %struct.jpeg_error_mgr,  
... %struct.jpeg_error_mgr* %4, i64 0, i32 0  
%5 = load void (%struct.jpeg_common_struct)*, void  
... (%struct.jpeg_common_struct)** %error_exit, align 8, !tbaa !20  
%6 = bitcast %struct.jpeg_compress_struct* %cinfo to  
... %struct.jpeg_common_struct*  
tail call void %5(%struct.jpeg_common_struct* %6) #3  
br label %if.end
```

if.end:

```
%7 = bitcast i8** %buffer to i64*  
%8 = load i64, i64* %7, align 8, !tbaa !11  
%9 = bitcast %struct.my_destination_mgr* %1 to i64*  
store i64 %8, i64* %9, align 8, !tbaa !21  
%free_in_buffer = getelementptr inbounds %struct.my_destination_mgr,  
... %struct.my_destination_mgr* %1, i64 0, i32 0, i32 1  
store i64 4096, i64* %free_in_buffer, align 8, !tbaa !22  
ret i32 1
```

CFG for 'empty_output_buffer' function