# Internal Function - Unit Testing

**name get_nft_name_from_number(uint64_t nft_number, name suffix);**
**-- version 10b (2023-09-30)**

```
name nftscribe::get_nft_name_from_number(uint64_t nft_number, name suffix) {

    // Ensure the suffix is valid
    check(suffix.length() >= 1 && suffix.length() <= 6, "Invalid suffix length. Must be between 1 and 6 characters.");

    // Ensure nft_number is not past maximum allowed
    check(nft_number <= 28629150, "get_nft_name_from_number received too high of a value for nft_number, max is 28629150. ");

    // Helper function to convert a value into its corresponding character
    auto value_to_char = [](uint64_t value) -> char {
        if (value >= 1 && value <= 5) return '0' + value;
        if (value >= 6 && value <= 30) return 'a' + value - 6;
        if (value == 0) return 'z';  // Special case for 0
        return ' ';  // Default, shouldn't happen with checks in place
    };

    string result_name = "";  // The string representation of the resulting name without the suffix

    do {
        uint64_t remainder = nft_number % 31;  // Get remainder of nft_number divided by 31
        result_name.insert(result_name.begin(), value_to_char(remainder));  // Convert remainder to char and prepend to result_name
        nft_number /= 31;  // Divide nft_number by 31 to get the next digit
    } while (nft_number > 0);

    // Now we concatenate the name with the provided suffix
    string full_name_string = result_name + "." + suffix.to_string();
    name full_name(full_name_string);

    return full_name;
}
```

**PASSED** Name Resolution Test (961, name("nft"))

```
name name_test = get_nft_name_from_number(961, name("nft"));
check(false, "name_test=" + name_test.to_string());
```

```
assertion failure with message:
name_test=1zz.nft
```

**PASSED** Max Name Test(28629150, name("nft"))

```
name name_test = get_nft_name_from_number(28629150, name("nft"));
check(false, "name_test=" + name_test.to_string());
```

```
assertion failure with message:
name_test=yyyyy.nft

  "code"  500
  "message"  "Internal Service Error"
  "error"
```

**PASSED** Min Name Test(0, name("nft"))

```
name name_test = get_nft_name_from_number(0, name("nft"));
check(false, "name_test=" + name_test.to_string());
```

```
assertion failure with message:
name_test=z.nft
```

**FAILED & FIXED** 1 Past Max Test (28629151, name("nft"))

```
name name_test = get_nft_name_from_number(28629151, name("nft"));
check(false, "name_test=" + name_test.to_string());
```

Result Error

```
assertion failure with message:
name_test=1zzzzz.nft

  "code"  500
  "message"  "Internal Service Error"
```

Redrafted function to 10b with additional max number allowed

```
// Ensure nft_number is not past maximum allowed
check(nft_number <= 28629150, "get_nft_name_from_number received too high of a value for nft_number, max is 28629150. ");
```

**PASSED** 1_Past_Max_Test_(28629151,_name("nft"))

```
name name_test = get_nft_name_from_number(28629151, name("nft"));
check(false, "name_test=" + name_test.to_string());
```

```
assertion failure with message:
get_nft_name_from_number received too high
of a value for nft_number, max is
28629150.
{
  "code": 500,
  "message": "Internal Service Error",
```