

# Configuring Authorship

In this section:

- [Overview](#)
- [About Authorship Configuration Priority](#)
- [Configuring How Authorship is Computed](#)
- [Creating Authorship XML Map Files](#)

## Overview

You can configure C/C++test to collect authorship data during analysis to facilitate task assignment. The data can be sent to the DTP server where additional analysis components, such as the Process Intelligence Engine (PIE), can be leveraged to facilitate defect remediation and development optimization.

You can configure C/C++test to assign authorship based on information from source control, XML files that directly map sources to authors, and/or the current local user.

## About Authorship Configuration Priority

Authorship priority is determined by reading the settings in the .properties configuration file from top to bottom. If multiple authorship sources are used, the following order of precedence is used:

1. information from source control
2. XML map file
3. current user

If one of the selected options does not determine an author, authorship will be determined based on the next option selected. If an author cannot be determined, the user is set as "unknown". Likewise, if none of these options is selected, the user is set as "unknown."

## Configuring How Authorship is Computed

Modify the .properties configuration file to specify how authorship is determined:

```
scope.local=[true or false]
scope.scontrol=[true or false]
scope.xmlmap=[true or false]
```

## Additional Authorship Configurations

By default, author names are case-sensitive, but you can disable case sensitivity by setting the `authors.ignore.case` property to true:

```
authors.ignore.case=true
```

You can set the user name, email, and full name for a user with the `authors.user[identifier]` setting. For example, your configuration may resemble the following:

```
authors.user1=john,john.doe@company.com,John Doe
```

If a user is no longer on the team or must transfer authorship to another user, you can configure the `authors.mapping[x,y]` property:

```
authors.mapping1=old_user,new_user
```

If you are transferring authorship between users, the author-to-author mapping information can be stored locally or in a shared XML map file:

```
authors.mappings.location=[local or shared]
```

If the mapping file is shared, you must specify the location of the shared XML file:

```
authors.shared.path=[path to file]
```

## Creating Authorship XML Map Files

The `<authorship>` element contains indicates the beginning of the mapping information.

The `<file />` element is placed inside the `<authorship>` element and takes two properties, `author` and `path` to map users to files or sets of files:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE authorship (View Source for full doctype...)>
<authorship>
  <!-- assigns all files named: "foo/src/SomeClass.java" to "author1" -->
  <file author="author1" path="foo/src/SomeClass.java" />
</authorship>
```

| Wildcard Expression        | Description  |
|----------------------------|--|
| <code>?oo/src/Foo.c</code> | Assigns all files that have names starting with any character (except /) and ends with "oo/src/"           |
| <code>**cs</code>          | Assigns all *.cs files in any directory  |
| <code>**src/**</code>      | Assigns every file whose path has a folder named "src"   |
| <code>src/**</code>        | Assigns all files located in directory "src"   |
| <code>src/**/Test*</code>  | Assigns all files in directory "src" whose name starts with "Test" (e.g., "src/some/other/dir/TestFile.c") |



### The mapping order matters!

The mapping file is read from top to bottom, so beginning with the most specific mapping ensures that authorship will map to the correct files.