# CWE Detail – CWE-462

## Description

Duplicate keys in associative lists can lead to non-unique keys being mistaken for an error.

## Extended Description

A duplicate key entry -- if the alist is designed properly -- could be used as a constant time replace function. However, duplicate key entries could be inserted by mistake. Because of this ambiguity, duplicate key entries in an association list are not recommended and should not be allowed.

## Threat-Mapped Scoring

Score: 0.0

Priority: Unclassified

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Quality Degradation, Varies by Context — Notes:

## Potential Mitigations

**•** Architecture and Design: Use a hash table instead of an alist. (Effectiveness: N/A)

**•** Architecture and Design: Use an alist which checks the uniqueness of hash keys with each entry before inserting the entry. (Effectiveness: N/A)

## Applicable Platforms

**•** C (Class: None, Prevalence: Undetermined)

**•** C++ (Class: None, Prevalence: Undetermined)

**•** Java (Class: None, Prevalence: Undetermined)

**•** C# (Class: None, Prevalence: Undetermined)

## Demonstrative Examples

**•** Since basename is not necessarily unique, this may not sort how one would like it to be.