ONE PAGE SUMMARY

IT & Software › CI/CD › Artifact Management

# Introduction to Artifact Management

Mastering Artifact Management for Efficient CI/CD Workflows

## DEFINITION

> Artifact management is the process of storing, organizing, and managing software artifacts. Artifacts include binaries, libraries, and documentation generated during the software development lifecycle.

## KEY CONCEPTS

### Artifacts

Artifacts are files produced during the software development process, such as compiled code or documentation.

### Repository

A repository is a storage location for artifacts, allowing teams to access and manage them easily.

### Versioning

Versioning tracks changes to artifacts over time, ensuring teams can revert or reference previous versions.

### Dependency Management

This involves managing the libraries and components that artifacts depend on to function correctly.

### Continuous Integration

Continuous integration is a practice where code changes are automatically tested and integrated, often using artifacts.

## EXAMPLES

→ Storing compiled Java binaries in a Maven repository.

→ Using Docker images as artifacts for containerized applications.

→ Managing library versions with npm for JavaScript projects.

## MEMORY TIPS

★ Think of artifacts as 'building blocks' of software.

★ Remember 'RAVD' for Repository, Artifacts, Versioning, Dependency management.

★ Visualize a library where each book is an artifact version.

## COMMON MISTAKES

✗ Neglecting to version artifacts, leading to confusion.

✗ Not cleaning up old artifacts, causing storage issues.

✗ Failing to document dependencies, resulting in integration problems.

## QUICK RECAP

Artifact management is crucial for organizing and maintaining software artifacts in CI/CD pipelines. Understanding repositories, versioning, and dependency management helps streamline development and deployment processes.