

ONE PAGE SUMMARY

IT & Software > CI/CD > Artifact Management

Introduction to Artifact Management

Streamline your software delivery with effective artifact management.

DEFINITION

Artifact management is the process of storing, organizing, and controlling software artifacts. It ensures that all components needed for software development and deployment are easily accessible and versioned correctly.

KEY CONCEPTS

Artifacts

Artifacts are files produced during the software development process, such as binaries, libraries, and documentation.

Repository

A repository is a centralized location where artifacts are stored and managed, allowing teams to share and retrieve them easily.

Versioning

Versioning is the practice of assigning unique identifiers to different iterations of artifacts, helping to track changes and maintain compatibility.

Dependency Management

Dependency management involves handling the libraries and components that software artifacts rely on, ensuring they are compatible and up-to-date.

Continuous Integration

Continuous Integration (CI) is a practice where developers frequently integrate code changes into a shared repository, often triggering automated builds and tests.

EXAMPLES

- Storing compiled binaries in a central repository for easy access.
- Using version tags to track changes in software libraries.
- Automating the retrieval of dependencies during the build process.

MEMORY TIPS

- ★ Think of artifacts as building blocks of software; they need to be organized.
- ★ Remember 'RVD' for Repository, Versioning, Dependency management.
- ★ Visualize a library where each book (artifact) has a unique ISBN (version).

COMMON MISTAKES

- ✗ Neglecting to version artifacts, leading to confusion over changes.
- ✗ Failing to clean up old artifacts, causing storage issues.
- ✗ Not using a central repository, which complicates collaboration.

QUICK RECAP

Artifact management is crucial for organizing and controlling software components. It involves using repositories and versioning to ensure smooth development and deployment. Remember to manage dependencies and integrate changes continuously for best results.