

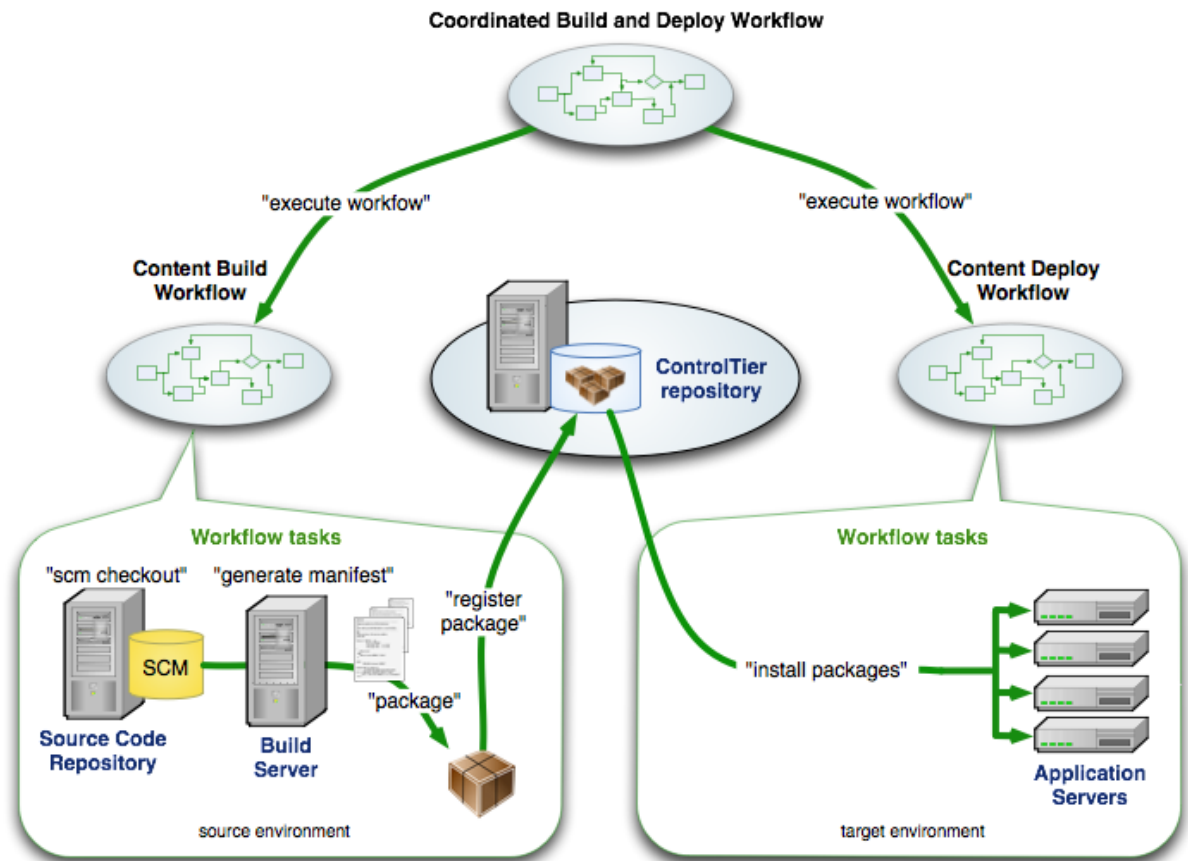
# Welcome to "Content Deployment Library"

This library provides a simple framework that automates the promotion of content from one environment to another. This automation extracts content from a source code repository, packages it, and deploys it to target environments using one standardized end-to-end process.

## What is it?

This is a library of command modules that build on the ControlTier automation base types and establish an end-to-end build and deployment process. This process is defined in terms of command workflows. The figure below describes there are three primary workflows:

1. **Coordinated Build and Deploy:** This workflow defines the end-to-end process and a single command to execute it called "BuildAndDeploy".
2. **Content Build:** This workflow is responsible for checking for changes in the SCM repository, checking out or updating files from the repository and creating a structured package to deploy the files.
3. **Content Deploy:** The deploy workflow manages the distribution and installation of the content packages.



### Why use it?

This library builds on the ControlTier service provisioning platform to offer a number of notable features for organizations where content migration and promotion is an important part of the application lifecycle and is a frequent and critical process:

- *End-to-end automation:* This framework ties together many individual procedures that are often run independently. Building on the ControlTier base types, the Content Library offers a single command that: checks code out (both full checkout and incremental updates), packages the content, manages package dependencies and deploys packages to multiple hosts.
- *Rollback:* All file distributions are packaged and versioned. You can rollback to any previous version. Files that will be overwritten by new versions are automatically backed up in their own package and stored in the repository and can be used to restore a previous state.
- *Task delegation and self-service:* You end up with schedulable "jobs" letting a less

- knowledable person run the checkout-deploy process on demand or at defined periods.
- *Standardized operational interface:* End users of the Content Library can rely on simple and predictable commands no matter what environment, source code module, or application.
  - *Structured Packaging:* The content of each package is declared in a manifest file. This file is used during the installation process to ensure integrity, and facilitate rollback.
  - *Reporting:* Every deployment done via the Content Library is logged. Reports show who deployed what where and when. The full output of the job is saved and can be used for auditing later.
  - *Graphical interfaces:* All operational tasks can be done via a graphical interface. ControlTier's JobCenter is used to run commands while Workbench can be used to review the current deployment environment dependencies.
  - *Security:* Using the ControlTier access control infrastructure, you can control who updates what when and where.
  - *Extendability:* The library offers a working end to end process but it is not monolithic. You can override various parts of the process via sub-typing.

### **Why not use a shell script?**

You might already have a scripted procedure to check out source files and use a utility like scp, rsync or rdist to distribute the files. Why not continue to follow that approach? If you have a simple environment where the process does not change then you probably don't need this library. But, if you have to run these kinds of processes for multiple applications and would benefit from some of the features listed above, then it is worth the effort to give it a try.

We have noticed several disadvantages of the typical scripted approach:

- **Hard to delegate:** It may be difficult to delegate running the script to someone not technical or familiar with the shell environment.
- **Hard to generalize:** The original script might be sufficient for the initial use case, but uses a lot of hard coded values that make it hard to use for new use cases.
- **Unweildly:** It might be a single script that is hard to customize with special behavior as needs evolve.
- **Lacks enterprise features:** Simple shell scripts are typically minimal bare bones implementations. It is difficult to imagine investing a great deal of time and effort in them so they include essential enterprise features (eg, reporting, logging, security) offered by a specialized and advanced framework.

### **Getting started**

You can start using the this library by following the steps of the documentation pages listed below:

1. Install: Describes how to download and install the library
2. Configure: Describes how to configure the library to define new "BuildAndDeploy" job
3. Run: Explains how to run the job either via the graphical JobCenter application or by command line.

[Next: Install #](#)