

ControlTier Solution Platform Overview

The Integrated ControlTier Platform: Control Dispatching Framework + Web Execution + Data Management Tools + Reporting + Solution Libraries

ControlTier automation libraries together with the full suite of ControlTier tools serve as the platform - or environment - on which ControlTier automation solutions are built. This platform provides the essential components you would otherwise have to create to build an enterprise automation system - components such as a security model, consistent package lifecycle, distributed application runstate control and coordination, configuration data registry, standardized yet customizable metadata model, centralized logging and reporting, web execution, and a myriad of repository management and automation services.

Control Dispatching Tool: CTL

All of the automation code is executed via the [CTL](#) control dispatching tool and framework. CTL along with a ControlTier plugin that integrates CTL to the platform repository, make accessing all the artifacts, modules and metadata stored in the repository possible. CTL's distributed "node dispatch" capability allow automation actions to be coordinated across hosts and managed objects. This control dispatching framework is installed on any host where the provisioning automation tasks are executed.

ControlTier Automation Libraries: The Process Building Blocks

Automation Libraries are sets of ControlTier modules that provide fundamental building blocks for an automation solution you wish to develop. Building an automation solution specific to your enterprise is primarily achieved through configuring workflow commands that are already defined in a library's automation modules. The general idea is that the functionality that comes prebuilt with the ControlTier automation libraries does most of the heavy lifting for you.

Modules and demo projects are available at moduleforge.controltier.org.

The following diagram shows an example of an automation library provides a coordinated [application service provisioning](#) process that spans from build through to deployment. In this diagram, library modules interact to provide and end-to-end workflow that automates packages being created, the organized storage of those packages in the ControlTier repository, and their consumption by various Deployments and Services hosted on Nodes in the target environment. This is a pretty standard pattern that most ControlTier automation solutions follow.

Object-oriented flexibility

In ControlTier, automation modules are not monolithic scripts but are object-oriented components. Each module defines a type and a set of commands (akin to class methods), and attributes (akin to class properties). Each step of the provisioning process is defined as a command in a type. These commands are then combined together in workflow commands to establish higher level processes.

The following chart shows the ControlTier object types and the commands each type provides.

Users can customize the behavior of any part of the process through sub-typing - adding new commands and overriding existing ones from the appropriate base type. This object-oriented approach facilitates process refactoring, a very important requirement for those that must keep up with a changing application.

Platform Development Tools: Workbench and ProjectBuilder

ControlTier provides two tools to develop your own solutions. Workbench provides a graphical development environment to design and develop automation modules. Workbench gives an integrated graphical interface to your provisioning automation code, data model and view and control of the repository. Inside Workbench, you can define types, commands, workflows, edit the metadata, inspect the artifacts in the repository and generate reports.



ProjectBuilder is an alternative to Workbench, providing a command-line oriented environment, wherein XML specifications are used to define the automation modules and data model and these specifications are fed through a series of command utilities to build, package and deploy them to the repository. Development is done in a traditional developer's edit, build, test cycle from source files. The artifacts of the ProjectBuilder are loaded into the

operational repository via repository services. ProjectBuilder is useful when one needs to really scale up their use of ControlTier and or prefer a command-line/source-file approach to maintaining automation module code and data.

The Platform Process Administrative Tool: Job Center

All the key processes that drive your automation solution are exposed as schedulable jobs in the ControlTier Job Center. Job Center gives the administrative staff a single interface to schedule, run and track the execution of commands and workflows. Job Center is also highly useful for delegating tasks to non-technical staff.

Job Center acts as the operational dashboard to drive your automation processes.

The Activity Tracking Tool: Report Center

All the key processes that drive your automation solution are recorded and tracked by the ControlTier Report Center. Report Center gives staff and their managers a single interface to querying and reporting on the operational activities that drive the application life cycle.

ReportCenter is a standalone repository that collects activity events from various ControlTier components including commands run from CTL, jobs run through JobCenter, and changes to the data model made in Workbench.

The Repository

The platform includes a structured repository that houses artifacts from the build process as well as the module code and data that drives the application service provisioning process. Workbench provides a graphical interface to browsing the content of the repository. The repository is based on a standard web technology, called WebDAV, and can be made accessible to other tools via HTTP and WebDAV clients.

The diagram below describes the three kinds of repository content:

For deployment or provisioning solutions, an essential facility of the platform is a central repository wherein all the packaged build artifacts produced by the build process are registered and maintained. During the deployment phase of the process, application components with package dependencies retrieve their required artifacts from the repository.

A registry of application configuration, package dependencies, host deployment, and key configuration settings is based on the same type/object model defined by the automation

modules. This data model is used to drive the commands and workflows at runtime.

When Workbench or ProjectBuilder is used to develop modules, the modules are automatically versioned, packaged, and stored in the repository.

[Next: Platform architecture #](#)