

Commands Reference

- [Builder](#): An interface to a software builder.
 - [Build](#): Run the build cycle.
 - [Export](#): Export a set of packages.
 - [generateBuildstampExcludes](#): Generate a list of buildstamps to avoid purging the most recent packages of builders of the given type, updating the model if required
 - [Purge](#): Purge unused and not-current packages.
 - [repoExport](#): Export package files from the repository.
 - [repoFind](#): Find package objects in the repository.
 - [repoFindAndPurge](#): Use repoFind to find unused and non-current packages, and then use repoPurge to remove them from the Repository.
 - [repoImport](#): Import packages into the repository.
 - [repoPurge](#): Remove package objects from the repository.
 - [runBuildScript](#): Invoke the build tool.
 - [scmCheckout](#): Run the checkout command.
 - [setBuildstamp](#): Optionally automatically generate the buildstamp, updating the model if required
- [Deployment](#): An interface to an application deployment
 - [Change-Dependencies](#): Query model and switch package dependencies to the specified version or buildstamp.
 - [Configure](#): Run the configuration cycle for the deployment.
 - [Deploy](#): Run the deployment cycle, coordinating package installation and configuration.
 - [Docs-Generate](#): Generate the configured template-based documents.
 - [Docs-Verify](#): Verify the documents generated by Docs-Generate.
 - [exec](#): executes a command
 - [Get-Process-View](#): Get the process view XML for a command invocation
 - [Get-Properties](#): Get the properties for the deployment.
 - [Install](#): Install the deployment, and its module and properties to AntDepo.
 - [Packages-Install](#): Install the configured package dependencies for the deployment.
 - [Properties](#): Print deployment properties to the console.
 - [Register](#): Register the deployment as an object to the repository.
 - [Register-Dependency](#): Register an object-to-object dependency to the repository.
 - [runChangeDependencies](#): Query model and switch Package dependencies to specified version or buildstamp.

- [Update](#): Change dependencies and run the deployment cycle.
- [Managed-Entity](#): Represents an abstract manageable object.
 - [Get-Process-View](#): Gets the process view file for the object.
 - [Get-Properties](#): Gets the properties for the object.
 - [Get-Supers](#): Gets the supertype module.
 - [Install](#): Initializes object workspace.
 - [Install-Module](#): Installs the specified module.
 - [object-deployment-uninstall](#): Removes the object.
 - [Properties](#): Prints out entity properties.
 - [Set-Properties](#): Sets the properties for the object.
- [Mediator](#): Mediates actions across a set of Deployments
 - [Change-Dependencies](#): Query model and switch package dependencies to the specified version or buildstamp.
 - [Configure](#): Run the configuration life cycle.
 - [Deploy](#): Run the coordinated deployment cycle across the configured Sites.
 - [dispatchCmd](#): Dispatch the specified command to child dependencies that match the specified type and name.
 - [Install](#): Install the deployment, its modules and properties to CTL, as well as any child dependency deployments.
 - [Packages-Install](#): Install the configured package dependencies for the subordinate deployments.
 - [Properties](#): Print deployment properties to the console.
 - [Update](#): Changes package dependencies and runs the coordinated deployment cycle across the configured Sites.
- [Node](#): A machine hosting application deployments
 - [Get-Properties](#): Get the properties for the deployment.
- [Package](#): An interface to a software package.
 - [assertPackageIsInstalled](#): Confirm the package process is installed.
 - [assertPackageIsUnInstalled](#): Confirm the package process is installed.
 - [assertPackageIsVerified](#): Verify the package installation.
 - [create](#): Create a package archive.
 - [extract](#): Extract the package archive.
 - [finish](#): Perform post package extraction finish step.
 - [get](#): Get the package from the repository.
 - [Get-Properties](#): Get the properties for the deployment.
 - [Install](#): Install object and run the package installation cycle.
 - [installDependencies](#): Install package's child dependencies.
 - [installPackage](#): Run the package installation cycle.
 - [prepare](#): Perform package installation preparation step.

- [purge](#): removes the file from the repository
- [register](#): Register the package to the repository.
- [remove](#): Remove the installed package.
- [upload](#): Upload the file to the repository.
- [ProjectBuilder](#): ProjectBuilder is a utility type to assist developing and loading of source-based projects in the ControlTier system.
 - [add-tag](#): Add a tag to the specified objects.
 - [archive-objects](#): Create an archive of objects matching the specified type and name.
 - [build-library](#): Build a type library.
 - [build-object](#): Generate entity.xml.
 - [build-type](#): Build and package a type defined in the type.xml.
 - [convert-rdf](#): Convert type and module rdf to type.xml.
 - [create-object](#): Create a bootstrap entity.xml.
 - [create-project](#): Deprecated: Create a new project in the repository.
 - [create-type](#): Create a new type source directory and initial type.xml.
 - [find-jobs](#): Query jobcenter for a list of matching jobs, and optionally store the result as an XML file.
 - [find-objects](#): Query project for objects matching query params.
 - [generate-forrest-docs](#): Generate forrest docs from type.xml files.
 - [generate-jobs](#): Generate an initial set of job data files from the type templates.
 - [generate-objects](#): Generate an initial set of object data files from the type templates.
 - [load-jobs](#): Load jobs into jobcenter via an XML file.
 - [load-library](#): Load a library into the repository.
 - [load-objects](#): Load object data from a file into the project model on the server.
 - [load-types](#): Upload type module jar files to the repository.
 - [purge-objects](#): Delete objects in the repository from the results of find-objects.
 - [refactor-rename](#): Replace text within matching files of a directory.
 - [remove-tag](#): Remove a tag from the specified objects.
- [Service](#): An interface to an application service deployment
 - [assertServiceIsDown](#): Confirm the service process is down.
 - [assertServiceIsUp](#): Confirm the service process is running.
 - [Deploy](#): Run the service deployment cycle, stopping installing package dependencies, configuring and the starting it.
 - [Prepare](#): Prepare the service for the update.
 - [Restart](#): Restart the service process.
 - [Start](#): Start the service process.
 - [startService](#): Start the service process.
 - [Status](#): Get the status of the service.
 - [Stop](#): Stop the service process.

- [stopService](#): Stop the service process.
- [Update](#): Run the service update cycle, stopping installing package dependencies, configuring and the starting it.
- [Site](#): An interface to a set of related application services.
 - [Prepare](#): Run the platform preparation cycle for the Services in the Site.
 - [Restart](#): Run the application re-start process for the Services in the Site.
 - [Start](#): Start the Services configured for the Site.
 - [Status](#): Call the Status command for the Services configured for the Site.
 - [Stop](#): Stop the Services configured for the Site.
- [Updater](#): Provides a coordinated build and deployment process.
 - [Build](#): Run the coordinated build cycle across the configured Builders
 - [BuildAndUpdate](#): Run the coordinated end-to-end the build and deployment processes across the configured Builders and Sites.
 - [Prepare](#): Run the platform preparation cycle across the configured Sites.
 - [runTestScript](#): Invoke the test tool.
 - [Test](#): Run the test cycle.
- [bin](#): A self extracting shar package
 - [extract](#): extracts the package by executing the self installer
- [coretests](#): Test suite for coreutils code
 - [okTest](#): base success test
 - [testAll](#): Run all tests
 - [testDavMacrosResolveDavUrl](#): test resolve-dav-url macro
 - [testDavutil](#): basic 'davutil' tests
 - [testFileutil](#): tests all fileutil commands
 - [testFileutilAvailable](#): Tests available command
 - [testFileutilCopy](#): Tests copy command
 - [testFileutilExecutable](#): test executable command
 - [testFileutilLink](#): Tests link command
 - [testFileutilLs](#): test ls command
 - [testFileutilMkdir](#): test mkdir command
 - [testFileutilMove](#): test move command
 - [testFileutilNewer](#): test newer command
 - [testFileutilOlder](#): test older command
 - [testFileutilReadable](#): test readable command
 - [testFileutilRemove](#): test remove command
 - [testFileutilRmdir](#): test rmdir command
 - [testFileutilTouch](#): test touch command
 - [testFileutilWriteable](#): test writeable command
 - [testNetutil](#): test all 'netutil' commands

- [testNetutilListening](#): test listening command
- [testNetutilReachable](#): test reachable command
- [testShellutil](#): test all 'shellutil' commands
- [testShellutilEnv](#): test env command
- [testShellutilPs](#): test ps command
- [testShellutilWhoami](#): test whoami command
- [testTextutil](#): test all 'textutil' commands
- [testTextutilCat](#): test cat command
- [testTextutilChecksum](#): test checksum command
- [testTextutilHead](#): test head command
- [testTextutilReplace](#): test replace command
- [testTextutilTail](#): test tail command
- [testTextutilWc](#): test wc command
- [davutil](#): various WebDAV utilities
 - [copy](#): Copy a resource
 - [delete](#): Delete a resource
 - [extract](#): Retrieve an archived resource and extract it
 - [mkcol](#): Make resource collection
 - [move](#): Moves/renames a WebDAV-resource.
 - [put](#): Put a file in the repository
- [ear](#): An ear package
 - [create](#): Create ear package.
 - [extract](#): Extract the package archive.
- [file](#): A package consisting of a single file
- [fileutil](#): various file utilities
 - [available](#): test if file available
 - [copy](#): copy file
 - [executable](#): test if file executable
 - [link](#): make a link
 - [ls](#): list directory
 - [mkdir](#): make directory
 - [move](#): move file
 - [newer](#): test if file newer than other file
 - [older](#): test if file older than other file
 - [readable](#): test if file readable
 - [remove](#): remove file
 - [rmdir](#): remove directory
 - [touch](#): touch a file
 - [writeable](#): test if file writeable

- [jar](#): A jar package
 - [create](#): Create jar package archive.
 - [extract](#): Extracts jar package archive.
- [logicutil](#): various command flow logic utilities
 - [timeout](#): Limit execution to specified time.
 - [try](#): try catch finally.
- [modelutil](#): a set of model management utilities
 - [deployment-get](#): Retrieve deployment data
 - [deployment-set](#): Define deployment data
 - [exists](#): Checks if an object exists
 - [find](#): Find objects matching name and type
 - [node-get](#): Retrieve node data
 - [node-set](#): Set node data
 - [package-get](#): Retrieve package data
 - [package-set](#): Set package data
 - [remove](#): Remove an object from the model
 - [resource-add](#): Add a resource relationship
 - [resource-remove](#): Remove a resource relationship
 - [setting-get](#): Retrieve setting data
 - [setting-set](#): Define setting data
 - [tag-add](#): Add a tag to an object
 - [tag-remove](#): Remove a tag from an object
- [netutil](#): various network utilities
 - [listening](#): checks for the existence of a TCP/IP listener at the specified host and port.
 - [reachable](#): test if server or URL is reachable
 - [traceroute](#): print the route packets take to network host
- [pkg](#): A SVR4 package
 - [create](#): Create package archive.
 - [extract](#): Extract package archive.
 - [remove](#): Remove the installed package.
- [rpm](#): A binary RPM package type
 - [assertPackageIsInstalled](#): Confirm the package is installed.
 - [assertPackageIsVerified](#): Verify installed package against the RPM database.
 - [create](#): Create a package archive. Runs rpmbuild.
 - [extract](#): Extract the package.
 - [remove](#): Remove the installed package. Runs rpm -e.
- [shellutil](#): various shell utilities
 - [env](#): get environment info

- [exec](#): executes a command
- [kill](#): kill process
- [pkill](#): kill named process
- [ps](#): get process info
- [whoami](#): print the user name
- [textutil](#): various text utilities
 - [cat](#): concatenate the file
 - [checksum](#): checksum a file
 - [contains](#): Checks if file contains pattern
 - [head](#): display first lines of a file
 - [nl](#): show file content with line numbers
 - [replace](#): replace string
 - [sort](#): sort file
 - [tail](#): display last lines of a file
 - [uniq](#): uniq file
 - [wc](#): ord, line, character, and byte count
- [tgz](#): A tgz package
 - [create](#): Create package archive.
 - [extract](#): Extract the package.
- [war](#): A war package
 - [create](#): Create package archive.
 - [extract](#): Extract the package archive.
- [zip](#): A zip package
 - [create](#): Create package archive.
 - [extract](#): Extract the package archive.