Open API Changes

Changes from 2017.1 to 2017.2

- With the introduction of the ability to attach a build configuration to multiple templates the following changes have been made:
  - In BuildTypeSettings:
    - getTemplate() and getTemplateId were deprecated
    - getTemplates() and getTemplateIds() were introduced instead
  - in SBuildType:
    - attachToTemplate(BuildTypeTemplate) and detachFromTemplate() were deprecated
    - addTemplate(BuildTypeTemplate, boolean), and removeTemplates(Collection<? extends BuildTypeTemplate>, boolean) were introduced instead
    - also setTemplates(List<? extends BuildTypeTemplate>, boolean) and setTemplatesOrder(List<String>) were added
- With the introduction of default templates the following changes have been made:
  - in SProject:
    - getDefaultTemplate(boolean) has been added
  - in SBuildType:
    - getOwnTemplates() has been added
  - in BuildTypeTemplate:
    - getUsagesAsDefaultTemplate() and getNumberOfUsagesAsDefaultTemplate() have been added

Changes from 10.0 to 2017.1

- ArtifactsURLProvider is renamed to ArtifactAccessor. Several methods were added to support artifact retrieval using this interface instances.
- ServerVersionInfo.getDisplayVersionMajor and ServerVersionInfo.getDisplayVersionMinor now return int instead of byte

Changes from 9.1 to 10.0

- UptodateValue:TimeToLiveProvider#getTimeToLiveMillis since 10.0 has a parameter - value to be cached, so the cache time can depend on the cached value
- AgentLifeCycleListener and AgentLifeCycleAdapter have two new methods:
  - dependenciesDownloaded - called when all artifact dependencies of the build have been successfully resolved and downloaded
  - preparationFinished - called when all preparations for the build are finished (sources checkout, personal patch, artifact dependencies, free disk space requirement, etc)
- jetbrains.buildServer.web.openapi.healthStatus.suggestions.ProjectSuggestion class can be used as base class for project-level suggestions
- Service messages-related classes are no longer available in common-api.jar. serviceMessages.jar is now essential part of Common API.

  This change only affects compile time (already compiled binaries will work as is). To fix compile-time "NoClassDefFound" errors, add serviceMessages.jar to your project's library.
- Methods SRunningBuild.addBuildMessage and SRunningBuild.addBuildMessages are deprecated and cannot be used in version 10.

  Plugins, using these methods should be rewritten to use jetbrains.buildServer.serverSide.buildLog.BuildLog methods (see SRunningBuild.getBuildLog).
- BuildServerListener.messageReceived event will not work in two-node configuration, and will be removed in the future.

  Consider using some other approach. For instance, a plugin can obtain an Iterator from build log (BuildLog.getMessagesIterator) and tail it periodically in background.
- Signature of jetbrains.buildServer.messages.BuildMessagesTranslator.translateMessages method has been changed,

  now it accepts list of messages instead of single message: List<BuildMessage1> translateMessages(SRunningBuild build, List<BuildMessage1> messages)
- jetbrains.buildServer.BuildAgent has new method int getAgentPoolId()
- jetbrains.buildServer.clouds.CloudImage has new method Integer getAgentPoolId() - it represents the agent pool that instances from this image will fall into.

The value is nullable, which means that agent pool for the instances can be configured manually in Agents->Pools UI

Changes from 9.0 to 9.1

Issue Trackers integration API changes

- `IssueProviderType` has been extracted as a separate class. It serves as issue tracker integration descriptor and contains id, display name and URLs to controller and issue rendering pages
- `AbstractIssueProviderFactory` now takes `IssueProviderType` as an argument rather type as a String
  To be compatible with 9.1, existing plugins must implement `IssueProviderType` and change the corresponding provider factory according to base class interface. This change in Github integration plugin can be taken as an example

Changes from 8.1.x to 9.0

Server API changes

- `jetbrains.buildServer.serverSide.dependency.Dependent#getDependencyReferences` and `jetbrains.buildServer.serverSide.dependency.Dependent#getNumberOfDependencyReferences` were moved to `jetbrains.buildServer.serverSide.SBuildType`
- `jetbrains.buildServer.issueTracker.IssueProviderFactory#getDisplayName` display name for issue tracker in UI.

Changes from 7.1.x to 8.0

External ID -related changes

- `jetbrains.buildServer.serverSide.dependency.DependencyFactory#createDependency` now accepts external ID instead of internal one, use `jetbrains.buildServer.serverSide.dependency.DependencyFactory#createDependencyByInternalId` for internal ID.
- `jetbrains.buildServer.serverSide.ArtifactDependencyFactory#createArtifactDependency` now accepts external ID instead of internal one, use `jetbrains.buildServer.serverSide.ArtifactDependencyFactory#createArtifactDependencyByInternalId` for internal ID.

Common API changes

- `SimpleCommandLineProcessRunner.RunCommandEvents => SimpleCommandLineProcessRunner.ProcessRunCallback`
- added `jetbrains.buildServer.serverSide.CachePaths` for plugins to get cache directory on server
- `jetbrains.buildServer.serverSide.statistics.ValueType#getFormat` now returns String constant representing format style
- `jetbrains.buildServer.serverSide.statistics.ValueType#getColor` now returns String containing Web Color

Server API changes

- Added `jetbrains.buildServer.serverSide.SProject#getPluginDataDirectory` that returns per-project plugin data directory
- `jetbrains.buildServer.serverSide.BuildTypeSettings#addBuildRunner` not accepts `jetbrains.buildServer.serverSide.BuildRunnerDescriptor` instead of *S*BuildRunnerDescriptor
- `jetbrains.buildServer.serverSide.TeamCityProperties` no longer contains static methods to compute TeamCity Data Directory. Use `jetbrains.buildServer.serverSide.ServerPaths` spring bean instead
- `jetbrains.buildServer.serverSide.buildDistribution.AgentsFilterContext` now contains `getCustomData` and `setCustomData` methods. Agent filters can now store data there to be used during distribution/filtering process
- added `jetbrains.buildServer.serverSide.buildDistribution.DefaultAgentsFilterContext`. Contains default implementation of custom data storage

Authentication API changes

- Changes in `jetbrains.buildServer.serverSide.auth.LoginConfiguration` class:

    - `registerLoginModule(LoginModuleDescriptor)` method is deprecated, use `registerAuthMethodType(AuthMethodType)` instead
    - `getSelectedLoginModuleDescriptor()` method is deprecated, use `getConfiguredLoginModules()` instead
    - `createJAASConfiguration()` method is deprecated, use `createJAASConfiguration(AuthMethod)` instead
    - `getAuthType()` method now always returns the value `"mixed"` and is deprecated, use `getConfiguredAuthMethods(Class)` or `isAuthMethodConfigured(Class)` instead
- Changes in `jetbrains.buildServer.serverSide.auth.LoginModuleDescriptor` class:
    - `jetbrains.buildServer.serverSide.auth.LoginModuleDescriptorAdapter` class was added, extend your implementation from this class to not depend on future changes in `LoginModuleDescriptor`
    - `getOptions()` method is deprecated, you need to implement `getJAASOptions(Map)` method
    - `LoginModuleDescriptor` interface now extends `jetbrains.buildServer.serverSide.auth.AuthMethodType` interface and it contains some new methods you need to implement (or just use the adapter mentioned above)
- Changes in `javax.security.auth.spi.LoginModule` class:
    - Message from `javax.security.auth.login.FailedLoginException` thrown from `javax.security.auth.spi.LoginModule` is now visible to user as is on login page
    - Login module should now store own user name in TeamCity user's properties if it can differ from TeamCity's login. On login attempt login module must find existing user with the specified value of that property and return TeamCity's login for that user or return own user name if user does not exist yet. Use `jetbrains.buildServer.serverSide.auth.LoginModuleUtil#getUserModel(Map)` to get `jetbrains.buildServer.users.UserModel` in login module.
- You need now call `jetbrains.buildServer.serverSide.auth.ServerPrincipal#setCreatingNewUserAllowed(true)` if you want TeamCity to create the specified user in case he/she does not exist yet.

VCS API changes

General

- Non-required VcsManager::registerVcsSupport method have been removed.
- tests-related constructors from `jetbrains.buildServer.vcs.ModificationData` were moved to `jetbrains.buildServer.vcs.ModificationDataForTest`
- most methods from `jetbrains.buildServer.vcs.VcsSupportUtil` moved to parent class `jetbrains.buildServer.vcs.utils.VcsSupportUtil`
- `VcsException` class no longer have `setRoot`, `getRoot`, `prependMessage` methods that are not designed to be used for vcs-plugins, in core-related tasks use `jetbrains.buildServer.vcs.VcsRootVcsException`
- added method `jetbrains.buildServer.vcs.VcsSupportContext#getVcsExtension` for Vcs plugin context, override this method to provide additional services from plugin
- `jetbrains.buildServer.vcs.VcsSupport#ignoreServerCachesFor` no longer be called, please migrate to post TeamCity 4.5 API

Patch building

- `jetbrains.buildServer.vcs.patches.PatchBuilder` no longer extends `jetbrains.buildServer.vcs.patches.PatchBuilderBase`. All methods from `jetbrains.buildServer.vcs.patches.PatchBuilderBase` were moved into `jetbrains.buildServer.vcs.patches.PatchBuilder`
- `jetbrains.buildServer.vcs.patches.PatchBuilderEx#setTimeStamp` was removed, use `jetbrains.buildServer.vcs.patches.PatchBuilder#setLastModified`
- `jetbrains.buildServer.vcs.patches.PatchBuilder` code was covered with `@NotNull/@Nullable` annotations

Access to VCS Services

Vcs API is split into two parts: VCS plugin api, which is used to implement VCS services in TeamCity, and VCS usage api, or just VCS API, which is used to work with VCS services from within TeamCity.

- `jetbrains.buildServer.vcs.VcsManager#getAllVcs` is replaced with `jetbrains.buildServer.vcs.VcsManager#getAllVcsCore`
- Introduced `jetbrains.buildServer.vcs.VcsRootInstance#findService` method to obtain a VcsService
- `jetbrains.buildServer.vcs.VcsManager#getVcsUsernames` return type has changed from VcsSupportContext to VcsSupportCore in the key of the returned Map.

Changes from 7.0 to 7.1

- new API calls `AgentRunningBuild#stopBuild` and `AgentRunningBuild#getInterruptReason()`. (Those methods were in `AgentRunningBuildEx` since 6.5)

- Responsibility API changes:

- Added:
  - jetbrains.buildServer.responsibility.ResponsibilityEntry
    - enum RemoveMethod
  - jetbrains.buildServer.responsibility.ResponsibilityEntry
    jetbrains.buildServer.serverSide.ResponsibilityInfo
    jetbrains.buildServer.serverSide.ResponsibilityInfoData
    jetbrains.buildServer.tests.TestResponsibilityData
    - getRemoveMethod()
  - jetbrains.buildServer.responsibility.ResponsibilityEntryFactory
    - createEntry(BuildType)
  - jetbrains.buildServer.responsibility.impl.BuildTypeResponsibilityEntryImpl
    - constructor(BuildType)
  - jetbrains.buildServer.web.functions.user.ResposibilityFunctions
    - isUserResponsible(ResponsibilityEntry, User)
- Changed:
  - jetbrains.buildServer.responsibility.impl.BuildTypeResponsibilityEntryImpl
    - constructor(BuildType, State, User, User, Date, String, RemoveMethod)
  - jetbrains.buildServer.responsibility.ResponsibilityEntryFactory
    - createEntry(BuildType, State, User, User, Date, String, RemoveMethod)
    - createEntry(TestName, long, State, User, User, Date, String, String, RemoveMethod)
  - jetbrains.buildServer.BuildType
    - getResponsibilityInfo() now returns ResponsibilityEntry
  - jetbrains.buildServer.serverProxy.RemoteBuildServer
    - updateResponsibility(Vector, String, String, String, String, String)
- Removed (deprecated):
  - jetbrains.buildServer.serverSide.ResponsibilityInfo
    - createInactive()
    - createInactive(String, boolean, User)
    - getSince()
    - getUser()
    - getUserWhoPerformsTheAction()
    - isActive()
    - isFixed()
    - setUser(User)
  - jetbrains.buildServer.serverSide.ResponsibilityInfoData
    - isActive()
    - isFixed()
  - jetbrains.buildServer.BuildType
    - removeResponsible(boolean, User, String)
    - setResponsible(User, String)
  - jetbrains.buildServer.serverProxy.RemoteBuildServer
    - removeResponsible(String, boolean, String)
    - removeResponsible(String, boolean, String, String)
    - resetResponsible(String, String)
    - resetResponsible(Vector, String, boolean, String, String, String)
    - setIsFixed(String, String, String)
    - setResponsible(String, String, String)
    - setResponsible(String, String, String, String)
    - setResponsible(Vector, String, String, String, String)
  - jetbrains.buildServer.serverSide.BuildServerListener
  - jetbrains.buildServer.serverSide.BuildServerAdapter
    - responsibleChanged(SBuildType, ResponsibilityInfo, ResponsibilityInfo, boolean)
  - jetbrains.buildServer.responsibility.SBuildTypeResponsibilityFacade
  - jetbrains.buildServer.responsibility.STestNameResponsibilityFacade
- Removed:
  - jetbrains.buildServer.serverSide.problems.BuildProblem and all implementations
  - jetbrains.buildServer.serverSide.problems.BuildProblemsProvider and all implementations
  - jetbrains.buildServer.serverSide.problems.BuildProblemsVisitor
  - jetbrains.buildServer.serverSide.SBuild
    - getBuildProblems()
    - visitBuildProblems(BuildProblemsVisitor)

- JavaScript: Activator is now BS.Activator and its source file has been moved from js/activation.js to js/bs/activation.js

Changes from 6.5 to 7.0

- new API calls: `BuildStatistics.findTestBy(TestName)` and `BuildStatistics.getAllTests()`
- event-method `projectCreated` of `j.b.serverSide.BuildServerListener` and `j.b.serverSide.BuildServerAdapter` now receives two parameters: `projectId` and `user`.

- no longer publish `AntTaskExtension*, AntUtil, TestNGUtil, ElementPatch, JavaTaskExtensionHelper` classes to openapi package. Those classes can still be found in <teamcity>/webapps/ROOT/WEB-INF/plugins/ant/agent/antPlugin.zip!antPlugin/ant-runtime.jar
- `Notificator` interface: methods `notifyResponsibleChanged` and `notifyResponsibleAssigned` changed second parameter from `j.b.serverSide.ResponsibilityInfo` to `j.b.responsibility.ResponsibilityEntry` (due to ResponsibilityInfo deprecation).
- `j.b.serverSide.BuildServerListener` - we've deprecated `responsibleChanged` method which used `j.b.serverSide.ResponsibilityInfo` parameter and added a similar method which uses `j.b.responsibility.ResponsibilityEntry`
- new API calls: `j.b.agent.AgentRunningBuild.getBuildFeatures()` and `j.b.agent.AgentRunningBuild.getBuildFeaturesOfType(String)`. With help of these methods you can access build features enabled for the current build with all parameters properly resolved.
- new API calls: `j.b.serverSide.BuildTypeSettings.isEnabled(String)` and `j.b.serverSide.BuildTypeSettings.setEnabled(String, boolean)`. These calls allow to enable / disable a setting with specified id (build runner, trigger or build feature), or check if it is enabled.
- Classes from serviceMessages.jar no longer depend on `j.b.messages.Status` class. If you used some of the classes (for example, `j.b.messages.serviceMessages.BuildStatus` class) and want to make your code compatible with TeamCity versions 6.0 - 7.0, please use `j.b.messages.serviceMessages.ServiceMessage.asString(...)` methods.
- new API extension point to filter all build messages: `j.b.messages.BuildMessagesTranslator`
- `j.b.serverSide.BuildServerListener` - we've removed `beforeBuildFinish(SRunningBuild, boolean)` method which was deprecated since TeamCity 3.1, there is another method `beforeBuildFinish(SRunningBuild)` which can be used instead.

Changes from 6.0 to 6.5

- Classes `j.b.serverSide.TestBlockBean`, `j.b.serverSide.TestInProject`, `j.b.serverSide.FailedTestBean`, `j.b.TestNameBean` are removed from the Open API. Interfaces `j.b.serverSide.STest`, `j.b.serverSide.STestRun` should be used instead.
- `j.b.serverSide.ShortStatistics.getFailedTests()`, `j.b.serverSide.BuildStatistics.getIgnoredTests()` and `j.b.serverSide.BuildStatistics.getPassedTests()` return the list of `j.b.serverSide.STestRun` accordingly.
- Classes `j.b.tests.TestName` and `j.b.tests.SuiteTestName` are combined together into `j.b.tests.TestName`.

Changes from 5.1.2 to 6.0

- `j.b.vcs.TriggerRules` class was removed from Open API as part of API cleanup. Please let us know if your plugin is affected by the change.

New responsibility event methods added:

- `j.b.serverSide.BuildServerListener.responsibleChanged(SProject, Collection<SuiteTestName>, ResponsibilityEntry, boolean)`.
- `j.b.notification.Notificator.notifyResponsibleChanged(Collection<SuiteTestName>, ResponsibilityEntry, SProject, Set<SUser>)`, `j.b.notification.Notificator.notifyResponsibleAssigned(Collection<SuiteTestName>, ResponsibilityEntry, SProject, Set<SUser>)`.
- `j.b.notification.NotificationEventListener.responsibleChanged(SProject, Collection<SuiteTestName>, ResponsibilityEntry, boolean)`.
- `j.b.messages.ServiceMessageTranslator` is reworked to allow binding to arbitrary message type by name instead of only known types

Most methods in `j.b.agent.AgentLifeCycleListener` interface were extended to receive `j.b.agent.BuildRunnerContext`.

`j.b.agent.AgentLifeCycleListener#runnerFinished(...)` method added. It is called after build step is finished.

`j.b.agent.duplicates.DuplicatesReporter` and `j.b.duplicator.DuplicateInfo` are added for reporting code duplicates on agent side.

Build Agent changes:

- `j.b.agent.AgentRunningBuild` does not extend `j.b.agent.AgentBuildInfo`, `j.b.agent.ResolvedParameters`. All methods from those interfaces were inlined into AgentRunningBuild interface.

Most methods from `j.b.agent.AgentRunningBuild` were splitted into `j.b.agent.BuildRunnerContext` and `j.b.agent.Build`

Context. We have added

Parameters required for build runner are represented with j.b.agent.BuildRunnerContext interface.
Every time AgentRunningBuild and BuildRunnerContext return resolved parameters back.

j.b.agent.BuildRunnerContext represents the context of current build runner. All add* methods modifies context for the runner. Those changes will be reverted whene context is switched to next runner.

j.b.agent.AgentRunningBuild provides a context of a build (i.e. shared between all runners). All addShared* methods modifies the build context (and thus all build runner contexts).

j.b.agent.BuildAgentConfiguration now contains getBuildParameters() and getConfigParameters() methods to access parameters. Configuration parameters here are formed from properties from buildAgent.properties that does not start from 'system.' or 'env.' prefix. All parameters are returned with all references resolved.

j.b.agent.AgentBuildRunner#createBuildProcess method signature has been changed to receive j.b.agent.BuildRunnerContext.

j.b.agent.CommandLineBuildService#initialize(...) method signature has been changed to receive j.b.agent.BuildRunnerContext.

j.b.agent.CommandLineBuildService#getRunnerContext(...) added

j.b.agent.CommandLineBuildService#afterProcessSuccessfullyFinished() added

j.b.agent.BuildServiceAdapter is added to simplify as proposed base class for commandline base build runner service.

Changes from 5.0 to 5.1

Web extensions:

- deprecated method removed: j.b.web.openapi.WebControllerManager.addPageExtension(final WebPlace addTo, final WebExtension extension, Anchor<WebExtension> anchor)
- deprecated class removed: j.b.serverSide.Anchor
- deprecated class removed: j.b.notification.TemplatePatternProcessor; j.b.notification.TemplateProcessor added instead, see Extending Notification Templates Model
- method removed: j.b.notification.TemplateMessageBuilder.setPatternProcessor()
- several methods in j.b.serverSide.SBuildType now return boolean instead of void. You will probably need to recompile your plugins that use the interface.

Changes from 4.5.5 to 5.0

Parameters

j.b.serverSide.parameters.AbstractBuildParameterReferencesProvider is renamed to j.b.serverSide.parameters.AbstractBuildParametersProvider
j.b.serverSide.parameters.BuildParameterReferencesProvider is renamed into j.b.serverSide.parameters.BuildParametersProvider
BuildParameterReferencesProvider.getParameters(@NotNull final SBuild build) changed signature to getParameters(@NotNull final SBuild build, final boolean emulationMode)
j.b.agent.BuildAgentConfiguration#getCacheDirectory now receives String as argument
j.b.serverSide.buildDistribution.StartBuildPrecondition#canStart second parameters (Map<QueuedBuildInfo, BuildAgent>) may contain null values for some queued builds

Miscellaneous

Added new build server events:
j.b.serverSide.BuildServerListener.vcsRootRemoved(SVcsRoot),
j.b.serverSide.BuildServerListener.responsibleChanged(SProject, TestNameResponsibilityEntry, TestNameResponsibilityEntry, boolean)

Added three notification methods:
j.b.notification.Notificator.notifyResponsibleAssigned(SBuildType, Set<SUser>),

`j.b.notification.Notificator.notifyResponsibleChanged(TestNameResponsibilityEntry, TestNameResponsibilityEntry, SProject, Set<SUser>), j.b.notification.Notificator.notifyResponsibleAssigned(TestNameResponsibilityEntry, TestNameResponsibilityEntry, SProject, Set<SUser>)`

Changes prior to 4.5.5

Not documented

preparationFinished