

GeneXus™
by **Globant**

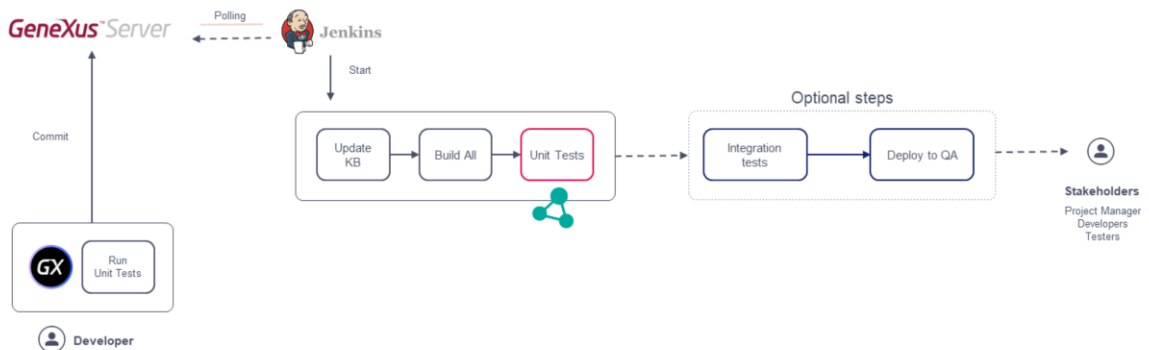
Pipeline

GeneXus[™]
© 2018

Creating the first pipeline

GeneXus™
by Globant

Trunk/Dev pipeline



Now that we have the Unit Test capability, let's see how to set the unattended tests execution in a basic Continuous Integration pipeline. Note that this is a basic continuous integration pipeline, it is possible to add more automated tests.

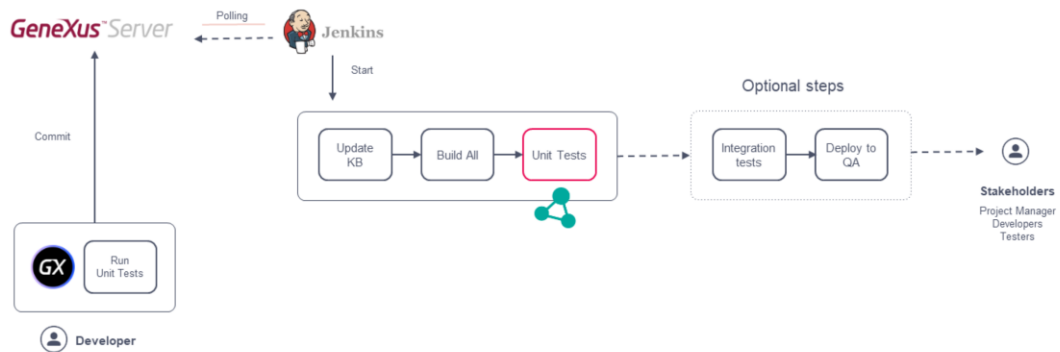
In this case, we will automate the pipeline in Trunk branch of the ebanking application to trigger some tasks when GXserver changes was detected. In this case, then we will configure the following tasks:

First, the update latest changes from GXserver (Trunk Version). This task will enable you to update the artifact to build (KB) to the latest commit.

Then, we will set the Build All task by which will be run a build all to specify and compile the KB.

Finally, we will set the Unit Tests execution

Trunk/Dev branch pipeline



Now that we have the Unit Test capability, let's see how to set the test execution in a Continuous Integration pipeline. Note that this is a basic continuous integration pipeline, it is possible to add more automated tests.

In this case, we will automate the pipeline in Trunk branch of the ebanking application to trigger some tasks when GXserver changes are detected. In this case, then we will configure the following tasks:

First, the update latest changes from GXserver (Trunk Version). This task will enable you to update the artifact to build (KB) to the latest commit.

Then, we will set the Build All task by which will be run a build all to specify and compile the KB.

Finally, we will set the Unit Tests execution

Try a Search

All ▾

Recents Automated Testing

DEVOPS IN GENEXUS

Agile Development

Typical GeneXus development cycle for Agile

▾ Continuous Integration

▾ Built-in

CI integrated to GeneXus and GXserver

CI: Create Pipeline

CI: Pipeline Execution

CI: User Permission

How to configure GeneXus Server for Continuous Integration

Continuous Integration Under the Hood

▾ Custom

▾ Continuous Delivery

Continuous Deployment

Operation and Monitoring the DevOps process

<How to configure GeneXus Server for Continuous Integration

This documentation is valid for:

[GeneXus Server 17 Help](#) [GeneXus 18 Help](#) [GeneXus 17 Help](#)

These are the necessary software requirements to be able to work with continuous integration pipelines from GeneXus and GeneXus Server:

- [GeneXus Server 17](#) installation
- One or more [GeneXus 17](#) installations
- [Jenkins](#) server
- Copy of the [GeneXus-Jenkins Integration](#) repository

The machines where GeneXus Server and Jenkins are located must have HTTP and HTTPS connectivity between them. It is highly recommended that they be separate machines.

It is necessary install and configure Jenkins to work with CI pipelines and catalog credentials in Jenkins to integrate to GeneXus Server. Follow [these instructions](#) if you are interested in installing a Jenkins server.

If you want to perform these steps manually please refer the following documents:

- [How to install and configure Jenkins to work with CI pipelines](#)
- [How to configure credentials to work with CI pipelines](#)

<https://wiki.genexus.com/commwiki/servlet/wiki?746996,How+to+configure+GeneXus+Server+for+Continuous+Integration>

To set the pipeline you just follow the documentation guide in the article "How to configure GeneXus server for Continuous Integration".

GeneXus server

- DASHBOARD
- MY ACCOUNT
- KNOWLEDGE BASES
- CONFIGURATION
- SECURITY
- CONTINUOUS INTEGRATION
 - Pipelines
 - Provider configuration**
- LICENSE

Provider configuration

Jenkins connection info

Url:

User Name:

User Token:

Jenkins configuration

GXserver URL (as seen from Jenkins):

GXserver credentials on Jenkins:

Pipelines folder:

Pipelines initial configuration

GeneXus installation:

MSBuild installation:

SQL Server for Knowledge Bases:

SQL Server credentials on Jenkins:

Deploy To Cloud:

Pipelines Execution Parameters defaults

Force Rebuild:

Run Tests:

<https://wiki.genexus.com/commwiki/servlet/wiki/746996,How-to-configure-GeneXus-Server-for-Continuous-Integration>

After the Jenkins installation and other simple configurations, you will go to GXserver and easily create the trunk pipeline with the option “Provider configuration”.

For that, you just have to complete a kind of form with all information requested and “Save”.

In the picture you can see the ebanking pipeline configuration.

GeneXus server admin * English * Version: 17.6.143577.0311

DASHBOARD

- MY ACCOUNT
- KNOWLEDGE BASES
 - All
 - Recently Used
 - Recently Changed
- CONFIGURATION
- SECURITY
- CONTINUOUS INTEGRATION
 - Pipelines**
 - Provider configuration
- LICENSE

Pipelines

Create Refresh Search Show only mine

Status	Name	Knowledge Base	Version	Environment	Run	Last Run	Next Run			
Success	DemoEbankingGXtest	DemoEbankingGXtest	DemoEbankingGXtest	NETSQLServer1	32	Nov. 9, 2022 05:13 PM	Nov. 9, 2022 06:13 PM	Run	Edit	Remove

<https://wiki.genexus.com/commwiki/servlet/wiki?46996,How-to-configure-GeneXus-Server-for-Continuous-Integration>

After saving, it is possible to see the created pipeline in the “Continuous Integration -> Pipeline” menu.

Let’s run the pipeline clicking “Run” option



Clicking “Run” option, you can see how it is possible run the pipeline from GXserver.

The pipeline automatically starts to update, build, and after that run the unit tests. You can see the different stages opening the Blue Ocean plugin in Jenkins URL.

When the execution is finished, you can see the executed test in the Test tab. In this case we just execute the CheckBalanceForTransferUnitTest.

If the test execution is successful, the pipeline status is successful in GXserver, otherwise, the status is Fail.

The screenshot displays the GeneXus CI/CD interface for a pipeline named 'GeneXusServer17U11 / DemoEbankingGXtest'. The interface is divided into several sections:

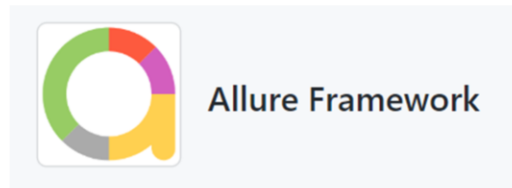
- Header:** Shows the pipeline name, 'No changes', and 'Started by user gxtest training'. It also includes navigation tabs for 'Pipeline', 'Changes', 'Tests', and 'Artifacts', along with a 'Logout' button.
- Branch/Commit:** Displays 'Branch: -' and 'Commit: -'.
- Timing:** Shows the pipeline duration as '2m 12s' and the start time as '2 minutes ago'.
- Test Results:** A central message box states 'All tests are passing' with a checkmark icon and the text 'Nice one! All 1 tests for this pipeline are passing.' Below this, a list of test results shows 'Passed - 1' with a sub-entry for 'Tests.CheckBalanceForTransferTest - Standalone test results'.

In the Tests tab you can see which tests were executed, but not the execution detail.

It is possible to use the Allure Report to visualize the execution detail of tests

Visualize test results with Allure

GeneXus[™]
by Globant



GXtest exports the execution results in Allure format to be able to view them in the Allure plugin in Jenkins, for example.



<https://github.com/allure-framework>

Allure is a flexible multi-language test report tool to show you a detailed representation of what has been tested.

The screenshot displays the Allure test results interface. On the left is a dark sidebar with navigation options: Overview, Categories, Suites, Graphs, Timeline, Behaviors, and Packages. The main area is titled 'Suites' and shows a table of test results. A single test, '#1 Tests.CheckBalanceForTransferTest', is highlighted in yellow, indicating it passed. The table columns are 'order', 'name', 'duration', and 'status'. The test's duration is 678ms. To the right of the table, a detailed view for the selected test is shown. It includes the title 'Standalone test results: Tests.CheckBalanceForTransferTest', a 'Passed' status, and tabs for 'Overview', 'History', and 'Retries'. The 'Overview' tab is active, showing 'Severity: normal' and 'Duration: 678ms'. Under the 'Execution' section, the 'Test body' is expanded to show three assertions, all of which passed:

- AssertBoolEquals(false, false, '1 Expected:Success: ') 0s
- AssertBoolEquals(true, true, '2 Expected:Success: ') 0s
- AssertBoolEquals(false, false, '3 Expected:Success: ') 0s

In our trunk pipeline, you can see the detail execution for each test.

In this case, the details for CheckBalanceForTransferUnitTest.

GeneXus[™]
by **Globant**

training.genexus.com
wiki.genexus.com