

Running the application for the first time

GeneXus[™]

Prototyping modes

Full version

- Local
- Cloud
- Any language

Prototyping

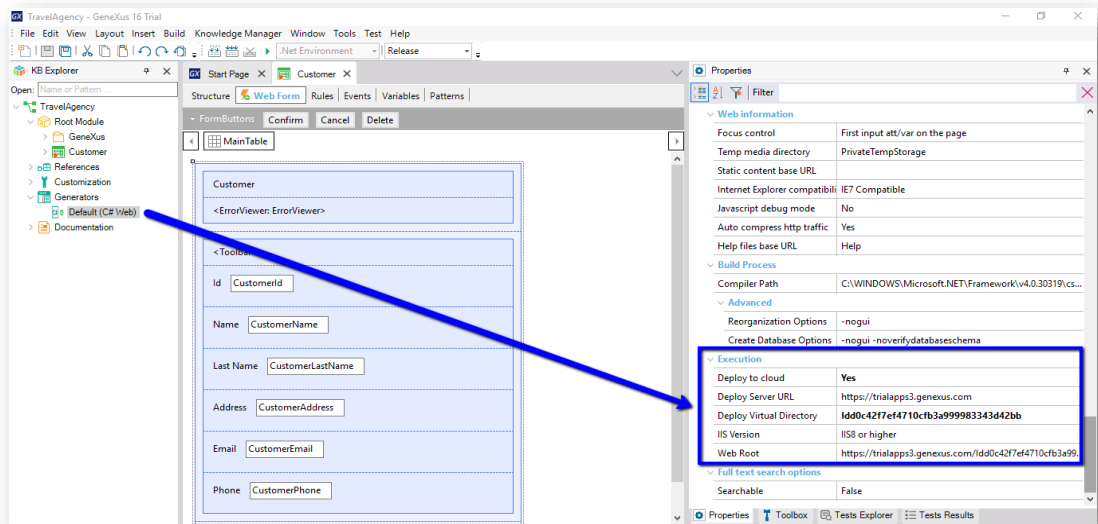
Trial version

- Cloud
- .Net

Now we want to test the transaction we created in the previous steps. To do so, we'll have GeneXus build the application, create the database, and do everything necessary to get the programs ready to be run and tested. This test cycle that developers undertake before the application is ready is called prototyping.

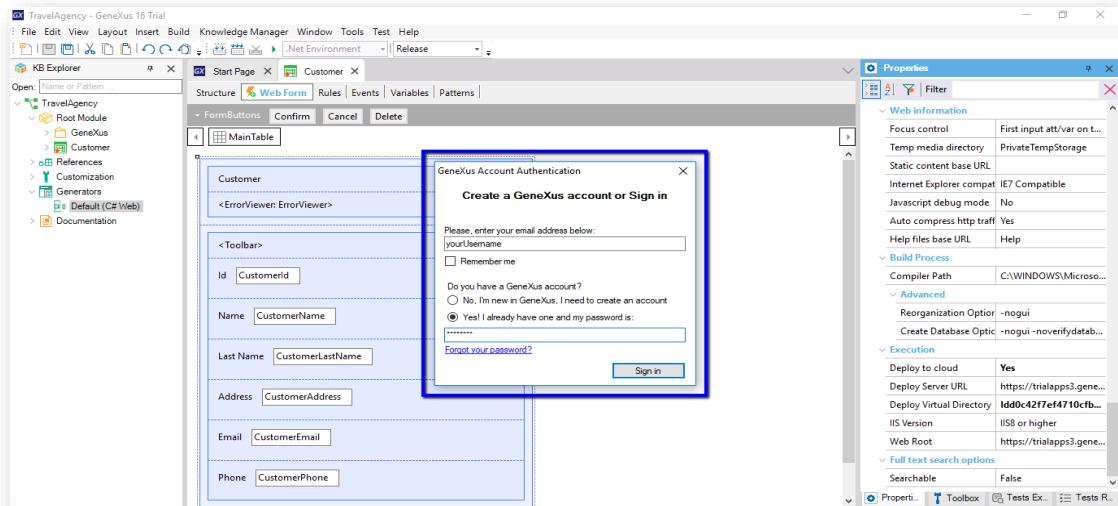
The full version of GeneXus allows prototyping an application locally (that is to say, the programs and the database will be hosted in our network servers) and in a cloud that is available to GeneXus users. In addition, we can prototype in any of the programming languages supported.

Prototyping mode: In the Cloud



On the other hand, the trial version only allows prototyping in the cloud and in .Net language. We can see it here, by opening the Properties window and confirming that we can only prototype in the cloud.

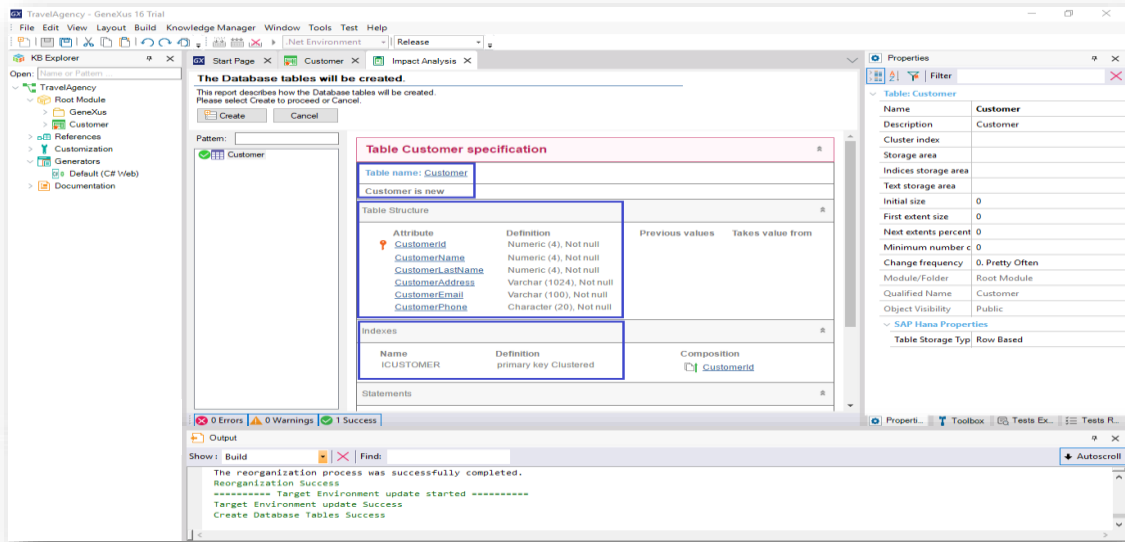
Prototyping mode: In the Cloud



We'll do it right now, by pressing the function key F5.

Because it is the first run, we're asked to log in with our GeneXus username. If we don't have one yet, we're offered to create one.

Impact Analysis



Once we log in, GeneXus starts working...

In the end, it displays a report called "Impact Analysis."

In this report, GeneXus examines the impact caused by the new definitions made in the knowledge base. Also, it indicates which additions or structural changes need to be made to the database.

If we examine the report... we can see that a new table called Customer will be created...

Also, that this table structure will have the following attributes:

As we can see, GeneXus will also create an index by CustomerId automatically. We will explain this concept later on.

Since we agree to have this table created with this definition in the database, we click on the Create button...

So, GeneXus starts to create the necessary programs to create the database (which didn't exist) and the Customer table, with this structure, in this database.

Next, GeneXus executes those programs...

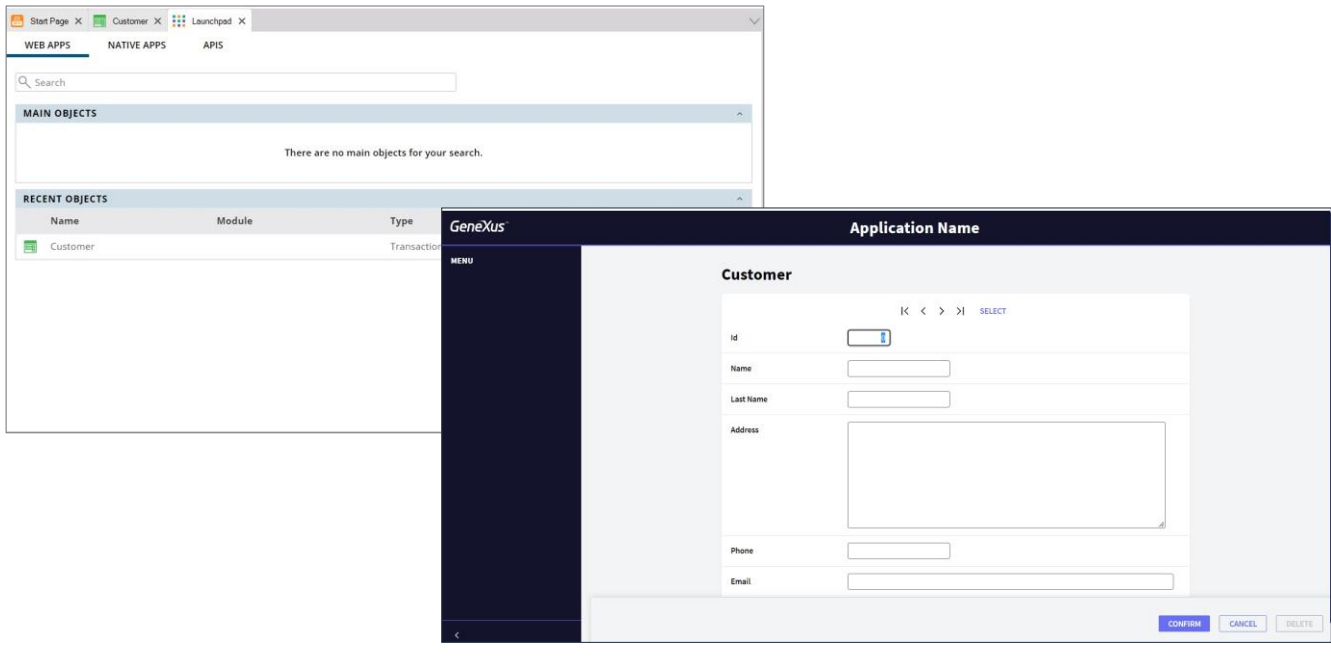
And the database is created. Next, GeneXus generates other programs (that is to say, all the necessary lines of code in the selected programming language, so that our application does what we want, which in this case is to enter customer details)... Also, we are informed if the result was successful, or if there were any errors or warnings.

Running the application

- In GeneXus, create everything that has been analyzed so far.
- Execute the application for the first time (F5).
 1. Impact Analysis: GeneXus analyzes the impact caused by our definitions, and informs any creations or structural changes it detects to be performed in the DB.
 2. If we agree and proceed → GeneXus creates automatically:
 - the Database
 - the Customer table
 - the program associated to the Customer Transaction + programs that provide supplementary functionalities
- Perform entry, modification and deletion of customers.



Running the application



Next, this launchpad appears that allows you to select the object to execute, so we select the Customer transaction. And we have the application running: Here we are looking at a page that allows us to enter, modify and delete clients. This menu that we see here offers access to transactions and other objects that we will see later.

We enter the details of our first customer...

This customer will be identified with number 1.

His name is John. His last name is Smith. He lives on Fifth Avenue. His phone number is: 1111 and his email address is jsmith@hotmail.com

We click on Confirm, and a message is displayed to inform us that the data has been successfully added; meanwhile, the form is cleared and ready for us to enter another customer:

Let's add the second customer... we identify her with number 2:

Her name is Susan. Her last name is **"Brown"**. Her address is on Seventh Avenue. Her phone number is 2222 and her email address is sbrown@gmail.com

We click on Confirm: once again, a message is displayed to inform us that the data has been successfully added.

To look for **John's** details, we enter his customer number (one) and press the Tab key.

We see that John's full details are displayed, and that from here we could change some information, such as his phone number.

Also, we could even delete customer John from the customer records using the Delete button.

We can also use the upper section buttons to browse customers by going to the next or the previous one, the first or the last one.

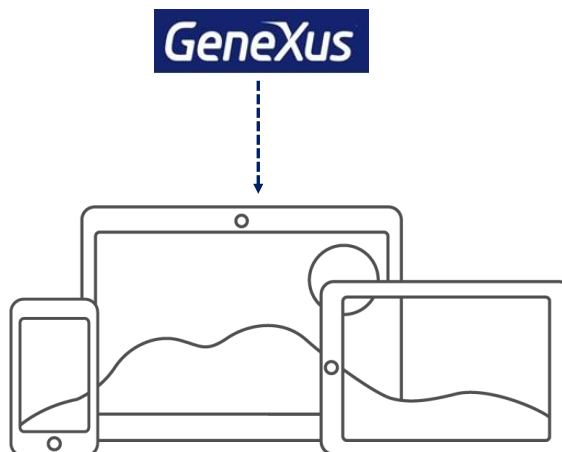
In addition, we can look for a specific customer using this window.

This window was automatically created by GeneXus and it corresponds to the Web Panel shown.

Note that if we change the screen size, the controls are adjusted to the space available.

Responsive web application

- Provide optimal visualization, fine browsing experience and excellent use of screen spaces, regardless of their size.

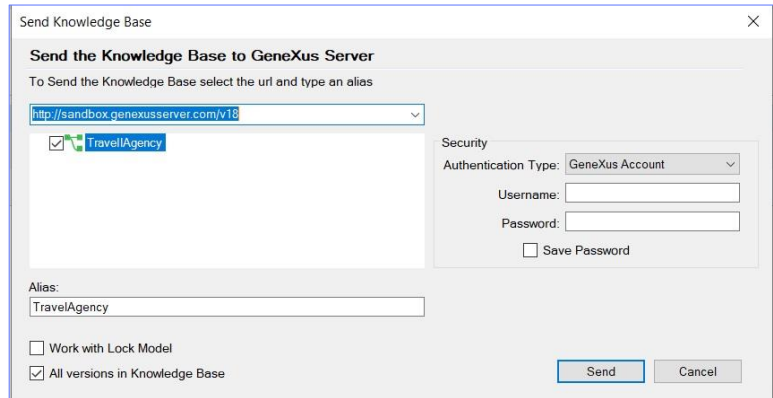
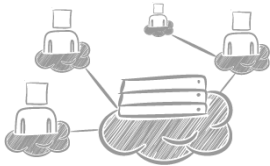
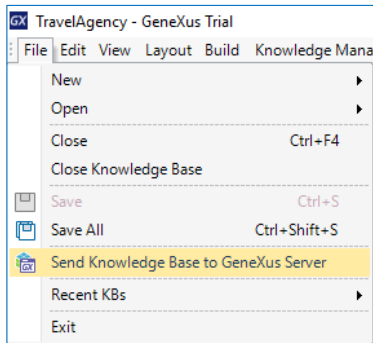


What does it mean? GeneXus generates responsive web applications, or sensitive applications, which offer optimum viewing, good navigation and excellent use of screen space.

This is achieved with design features that allow viewing all the information in the screen, using the functions provided by GeneXus to this end. They allow us to show different layouts depending on the screen size of the device.

Now that **we've** seen everything that was automatically generated by GeneXus using the Customer transaction that we've created... **let's** go back to the development environment.

Send Knowledge Base to GeneXus Server



Now we will upload this knowledge base to a GeneXus server in the cloud. Even though it's not mandatory, we should use GXserver to keep a history log of the activities performed as part of the project if there is more than one person working on the same knowledge base.

GeneXus Server allows centralizing our knowledge base (KB) and thus facilitates management if we work together on it, even if the users are in different geographical locations. Each developer works on his own local copy of the KB, which in turn is connected to the centralized KB managed by GXserver.

Having a central KB managed by GeneXus Server allows any authorized developer to send and receive modifications from their own workplace.

This allows us to share development tasks between several users, with each developer working independently, but as if they were working in a centralized way, while GeneXus Server consolidates the changes made to the project by each developer.

To upload our KB, in the File menu we select Send Knowledge Base to GXserver.

Note that the following server address is offered by default to upload our KB.

This server is free to use and the knowledge bases hosted there are shared with the entire GeneXus community; they are automatically deleted after 15 days of inactivity.

Here we enter our GeneXus username and press Send. A KB based on our KB will be created in the Server, with the name indicated here. If a KB with the same name already exists in the Server, the attempt fail. We must make sure that the name is unique.

KB Visualizer

The image shows two overlapping screenshots of the GeneXus Server web interface. The background screenshot is the login page, titled "GeneXus Server" and "Team development for GeneXus". It features a "Collaboration and versioning for GeneXus™ Projects" banner and a login form with fields for "Authentication Type" (set to "GeneXus Account"), "Username" (containing "diegomm"), and "Password" (masked with asterisks). There is also a "Remember me on this computer" checkbox.

The foreground screenshot is the "My Activity" dashboard. The top navigation bar shows "GeneXus server", the user "diegomm", and the language "English". The left sidebar contains a "DASHBOARD" menu with options: "MY ACCOUNT" (with sub-items "Favorite KBs", "Activity", and "Information"), "KNOWLEDGE BASES" (with sub-items "All", "Recently Used", and "Recently Changed"), and "CONFIGURATION". The main content area is titled "My Activity" and includes a search bar and a table of activity logs.

Knowledge Base	Operation	Date	Comment
TravelAgency	Publish	07/22/20 04:44 PM	Send KB

From the browser, we can access a KB viewer on the server using the address provided to send the knowledge base.

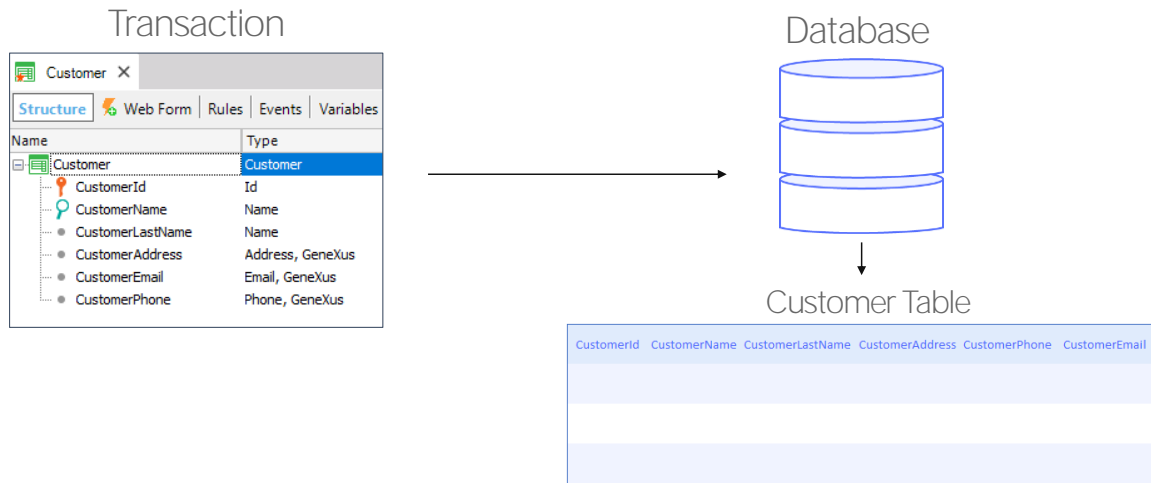
We enter with our GeneXus username and password.

To view the KB we just uploaded, we open the “My account” menu and select “Activity.”

There we see our knowledge base. We can view the type of operation performed on it, the date the action was performed, and the comments made in case we have assigned one.

In summary, we have seen:

- Upon analyzing the transactions defined, GeneXus determines the physical tables to be created.



Summing up what we have seen: from the transaction object GeneXus builds a program to create the table that will store in the database the information that will be handled by that transaction.

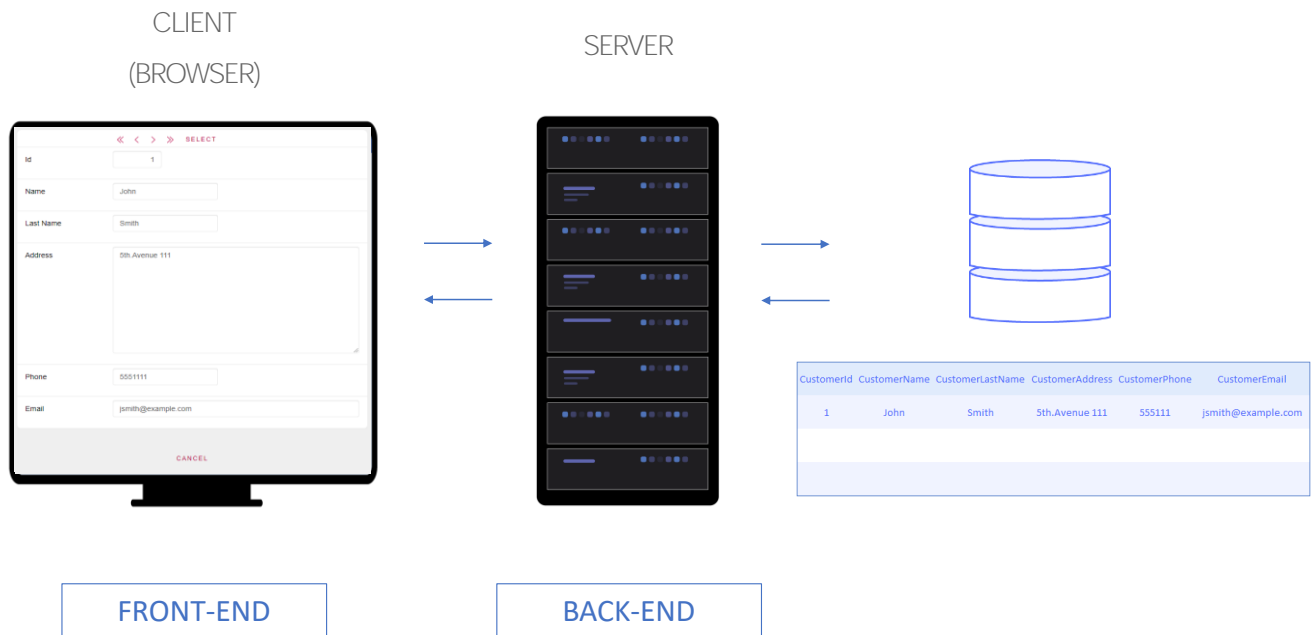
When we press Create, the program is executed.

Since we prototype in the cloud, the web server and database manager will not be local to our machine, but in that cloud. Therefore, when we press Create the creation program will be sent to the cloud and will be executed there, in order to have that table created in a database in the cloud.

In addition to that, GeneXus has sent to the server the program it built (on the chosen platform) to run the transaction. When that program is run, part of it will be running on the server and the other part will be running on the browser.

When we open the transaction, we see the user interface, which allows us to perform actions such as entering data from a new client.

The interface we currently see running corresponds to this part of the GeneXus transaction [show Form]. We could customize this form, and we will see it later.



When you enter data of a new client and press Confirm, the part of the program running on the Browser communicates with the part of the program running on the Server, to whom it sends the data; the program on the Server communicates with the database, asking it to enter that new client in the table Customer that had been created before.

More precisely, when the user enters a value in the identifier, the part of the transaction running on the Browser communicates with the part of the transaction running on the Server, which in turn communicates with the database to find out if that client already exists, or is new. If it exists, the database sends the client's data to the Server, which then sends it to the Browser, which displays it on the form.

Therefore, one part of the transaction object will be programmed in the back end (and, therefore, will be run on the server), and one part of the object will be the front end (and, therefore, will be run on the client).

*GeneXus*TM

training.genexus.com
wiki.genexus.com