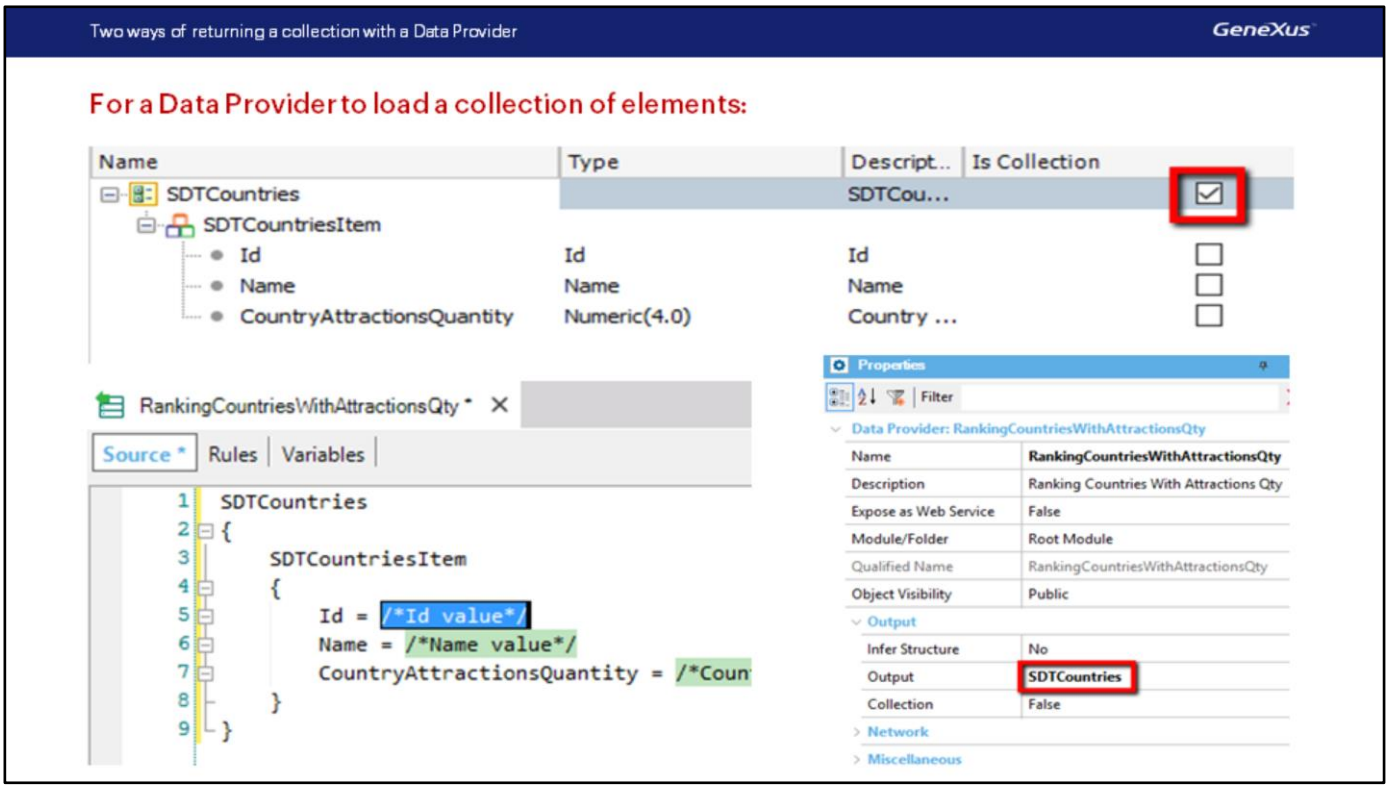
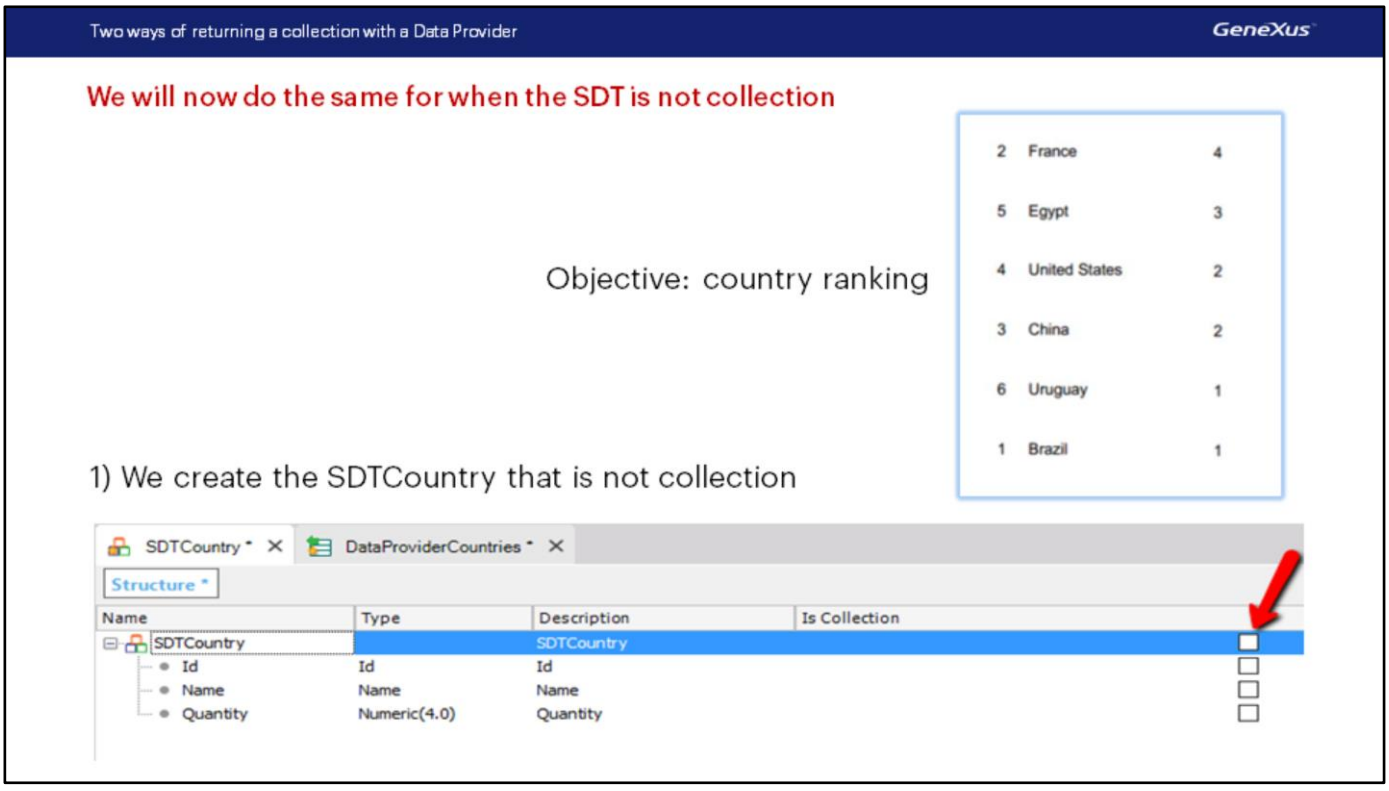
Two ways of returning a collection with a Data Provider

GeneXus 16



Data Providers are versatile objects that facilitate, with a declarative language, the loading of structures – both simple items and collections of items.

As we saw in a previous video, when we want a Data Provider to load a collection of elements, we define a variable based on a structured data type that is collection and then we drag that variable to the source of the data provider. This causes its output to be set up automatically for loading that data type.

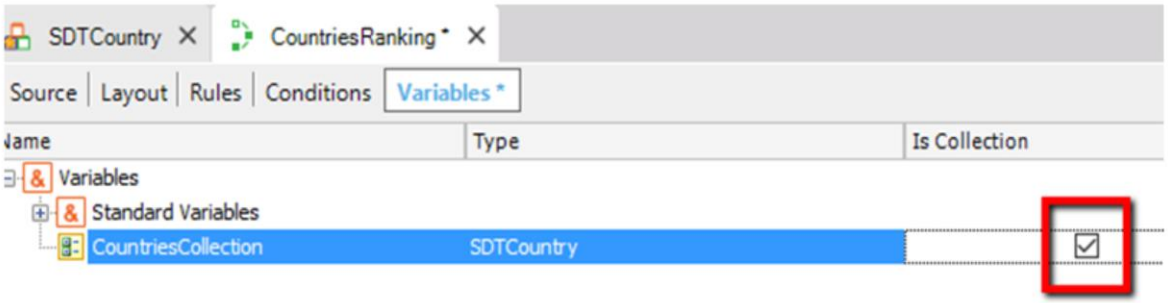


Now we will see the possibility for a Data Provider to return a collection of elements when the structured data **is not a collection.**

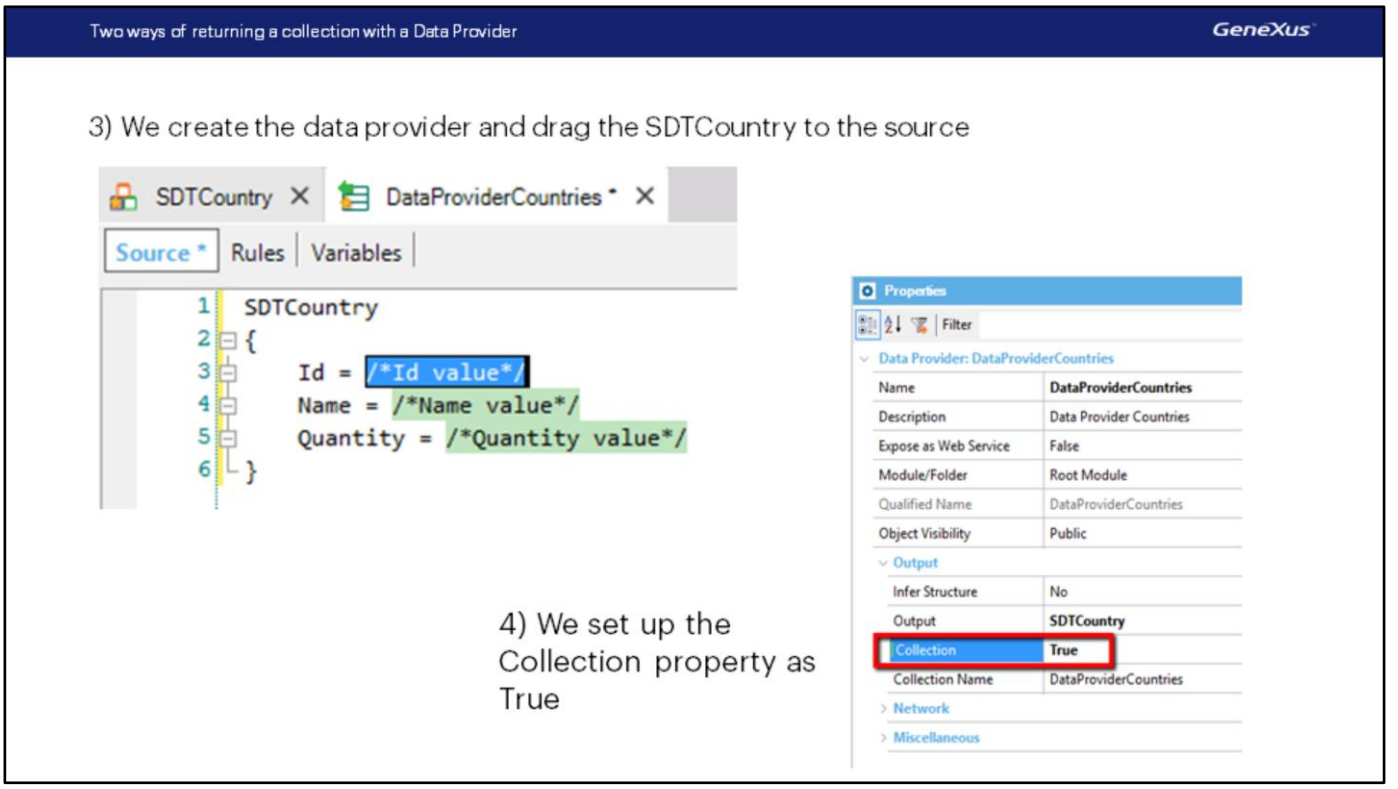
The travel agency requested a report showing a ranking of countries with more tourist attractions. Our example implies creating a collection of countries where each element is of a structured data type that stores the data of only one country.

We first create a structured data type called SDTCountry, with an ID item, a Name item, and a Quantity item where we store the number of tourist attractions for that country. We do not mark this data type as collection, so it will be able to store the data of a single country.

2) We define a variable based on SDTCountry and mark it as collection



Then, in the procedure object that implements the ranking we define a CountriesCollection variable of the SDTCountry type and mark it as collection.



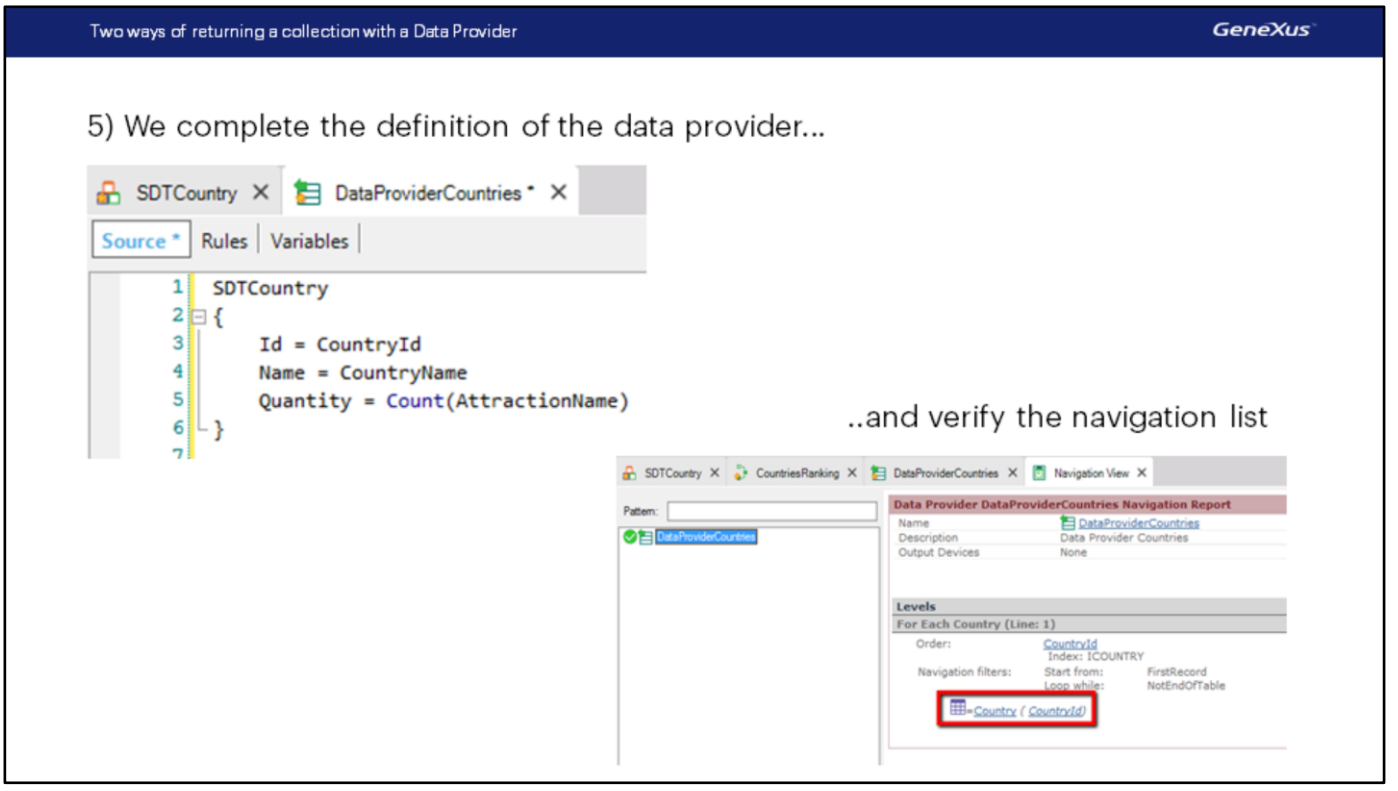
In order to obtain the data of all countries for defining the ranking, we create a data provider object called DataProviderCountries; and then we drag the SDTCountry to the source of the data provider.

We will see that a code was automatically created. It reflects the structure of SDTCountry. However, we don't have any repetitive group, because there is no group defined by curly brackets inside another one. So the definition that is written in the source will only be able to load the data of one country. But we want to load the data of all the countries...

Fortunately, the data provider has a solution for this problem, since it builds the collection for us. To do this we go to the data provider properties and in the Output group we set the **Collection** property with True value.

Note that when we drag SDTCountry to the source of the data provider, the Output property is automatically configured with the name of the SDT, which, as we know, stores a single element.

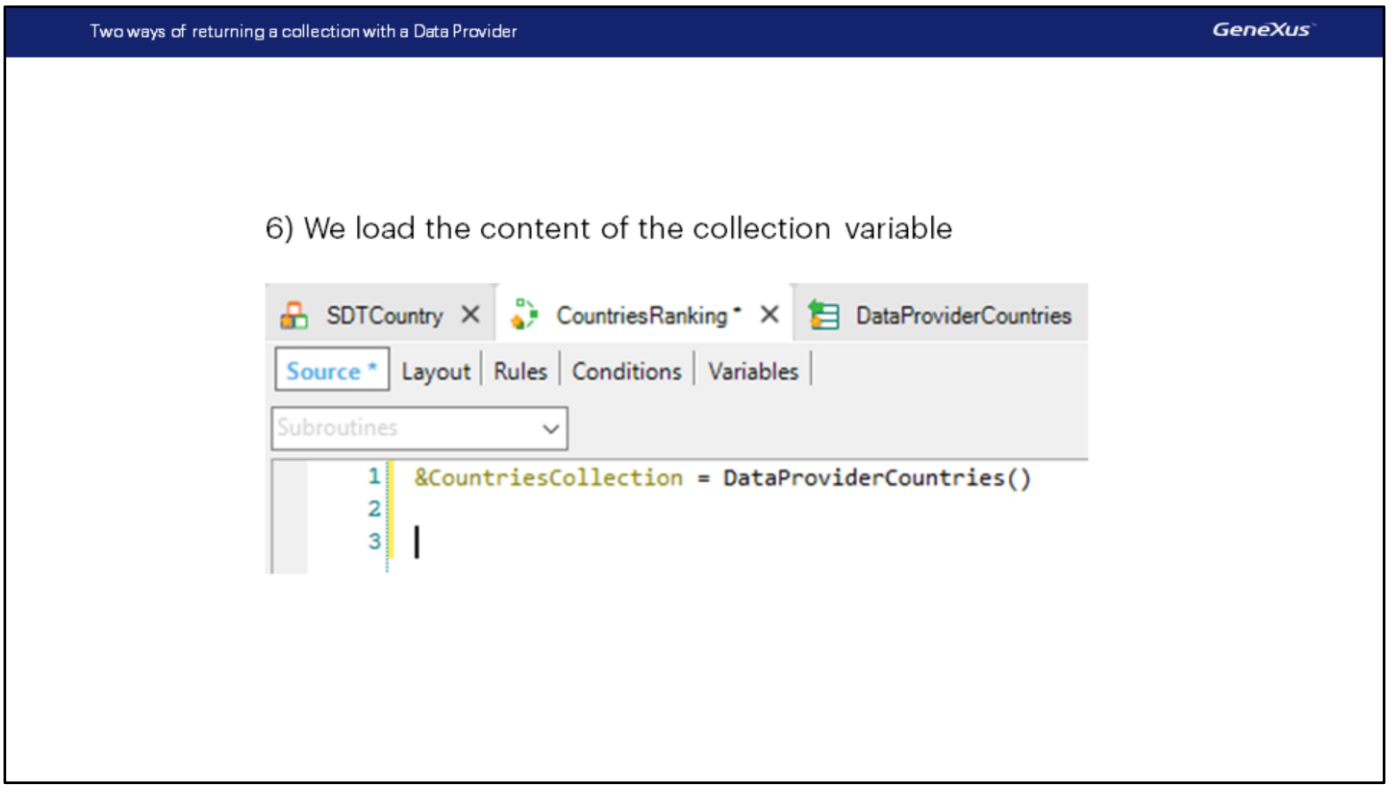
But when we assign the Collection property in True, we are indicating to the data provider that we want it to return a collection of elements of the type of SDTCountry. We can also see that the Collection Name property appears and that a name is automatically assigned to the collection.



To complete the definition, we write where we want the data provider to obtain the data, which in our case is from the countries table. So we assign, to the right of the assignments, the attributes CountryId for the Id, and CountryName for the Name. Then we assign a Count formula to Quantity for counting the number of attractions in the country.

Our data provider is then ready to be run.

When we right click on the name tab of the data provider and select View Navigation, we will see that GeneXus specifies the object and shows the navigation report, where we can see that it will go over the Country table to obtain the data of countries, as we wanted.

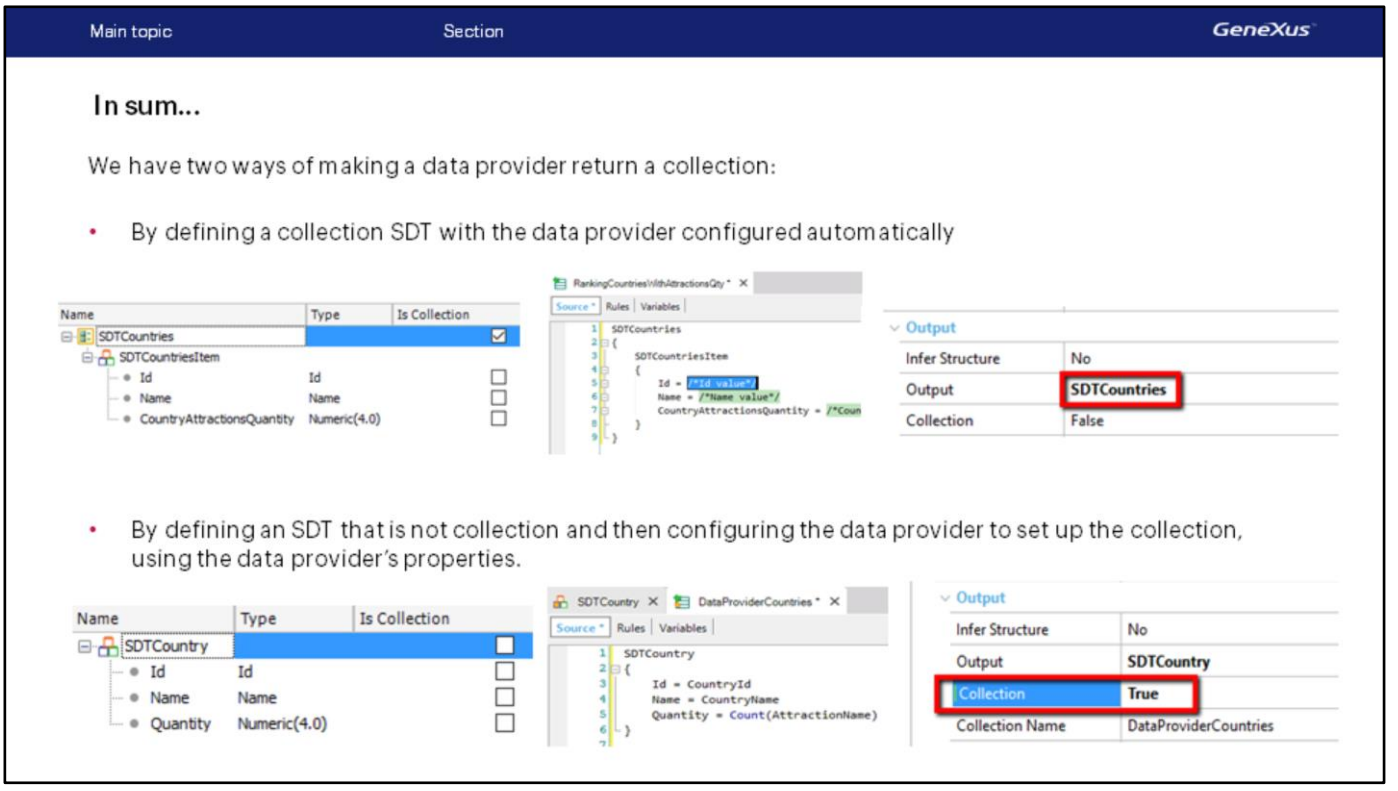


We now define what we need to load the contents of the CountriesCollection collection variable.

We must first add the &CountriesCollection variable to the source and assign to it DataProviderCountries.

When we invoke the data provider, it will run through all the records of the Country table and it will return a loaded collection where each element is of the SDTCountry type. That collection will be stored in the &CountriesCollection collection variable that we also defined as collection of elements of the same type.

Note that the syntax with which the data provider is invoked remains unchanged, except that the variable that receives the result of the data provider, instead of being based on an SDT that is already collection will be based on a simple SDT, and we make the variable a collection by marking the checkbox.



In sum, we have two ways of making a data provider return a collection of elements:

- by defining a structured data type, and when we drag it to the source of the data provider it is automatically configured for returning a collection of that type,

- or by defining a structured data type that is not collection, and then configuring, through the properties of the data provider, that the data provider itself define the collection.

GeneXus™

The power of doing.

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications