

Overview of Natural Business Services

A Natural business service provides a business perspective of a grouping of Natural subprograms. This perspective includes such things as the:

- Business service description
- Service methods and method descriptions

Retrieval algorithms are available to allow quick and easy searches of the business service repository. This allows a business analyst to quickly determine if a particular service currently exists or if one must be created. In addition, you can use the business service repository explorer to deploy business services from one environment to another and/or apply security at a domain, business service, and/or method level.

Note:

The Natural subprograms used for the supplied business services are located in the SYSBIZDE library on the server.

Each method defined for a business service is typically associated with a different Natural subprogram. If a method does not have an associated subprogram, a default subprogram for the entire service is required. In addition, every subprogram that is exposed through a business service requires a subprogram proxy.

The attributes for each method are determined by the subprogram invoked for that method. These attributes can be used to describe the business service or as input, output, input/output, or state values for the service. They can be acted upon and/or changed based on which methods have been defined for a service. For example, the Customer service in the demo application contains the Update method and the Name and Phone Number attributes, which allows the service to be used to change the phone number for a customer.

Once a business service is created, it can be consumed by client proxies generated in Visual Studio or Eclipse. You can also use Visual Studio or Eclipse to generate Web services.

You can quickly develop business services using the Business Service wizard supplied with the Natural Business Services plug-ins. This wizard recognizes the style of service you require based on your answers to simple questions asked on the wizard panels.

Although you can create a business service for any Natural subprogram that does not contain user interface code (i.e., a Natural subprogram that was not generated using Natural Construct and does not contain user interface code), this type of subprogram would only have one method associated with it — DEFAULT, which executes the associated subprogram as it normally would in Natural. An example of this type of business service is the ErrorMessageTesting service in the DEMO domain. This service invokes the FLIPSTY subprogram proxy, which in turn executes the FLIPSTR subprogram.

For more functionality, you can wrap several subprograms into one subprogram and create additional methods. An example of this type of service is the CalculatorAdvance service in the DEMO domain. When this service was created, the Business Service wizard recognized that the developer wanted to use more than one subprogram. The wizard prompted the developer to define the additional methods based on which subprograms to call and in what order to call them. The methods were enhanced by adding user exit code between the calls to these subprograms.

While this functionality is possible without using Natural Construct and Predict, you can enhance the functionality by generating business services based on files that have been defined in Predict. Standard methods, such as Delete, Next, Update, Browse, and FindBy*, can be automatically created based on input to the Business Service wizard and the file definitions set up in Predict.

Note:

If you would like to include the additional functionality provided by Natural Construct and Predict, but your applications were created without using these products, you should consider Natural re-engineering. Re-engineering analyzes existing monolithic applications and uses Natural Construct models to convert them into client/server style applications.