Introduction

Natural Business Services allows you to create and maintain business services. Each business service combines a group of methods related to a common business entity, such as a customer or order. Processing for the methods is supplied by either existing or wizard-generated Natural subprograms. The definition for each service is stored in the business service repository and identifies the associated methods.

As data typically comes from different platforms and uses different character sets, each business service requires a subprogram proxy to retrieve data. All business code is contained in the subprogram that is called by the proxy.

During generation, the Business Service wizard:

- Creates the subprogram proxy
- Populates the business repository with information about the methods used by the service
- Provides the domain, service, and version specifications
- Adds the method and service descriptions to the business repository explorer

For information on how Natural Business Services creates business services, see Understanding Natural Business Services.

This section covers the following topics:

- Create a Web Service for Your Business Service
- Use Unicode Parameters for Your Business Service

Create a Web Service for Your Business Service

Natural Business Services provides different techniques you can use to create Web services for existing business services. Depending on the product you prefer and your development requirements, you have several choices. The following table describes three techniques to create Web services:

Component/Location	Technology	Description
Eclipse plug-in	Java, Apache AXIS	The Natural Business Services Eclipse plug-in generates Web service classes and descriptor files that are understood by the Apache AXIS framework. The plug-in provides a wizard to generate a client proxy (which is used to translate Natural data formats to the appropriate language formats) and, optionally, a Web service for an existing business service. Once the proxy has been created, the features of Eclipse can be used. In addition, the Eclipse plug-in provides an option to generate default tests based on all the methods used by a business service. Once generated, you can deploy the Web service using the WS-Stack plug-in.
Visual Studio add-in (ASMX method)	Natural Development Server (NDV) and EntireX	The Natural Business Services Visual Studio add-in generates .NET client proxy classes you can use in an ASP.NET Web service project. You can then expose the properties and methods to a Web service class. This method requires coding on your part (for an example of this method, see SAMPLES in the Natural Business Services Installation folder). This add-in provides two wizards: one allows you to select an existing business service from the Business Service repository explorer and then generate a client proxy (which is used to translate Natural data formats to the appropriate language formats) and another generates a Web service for an existing business service. Once the proxy has been created, the features of Visual Studio can be used. For example, .NET's Intellisense can provide the attributes for the business service. In addition, the .NET add-in provides an option to generate default tests based on all the methods used by a business service.
	.NET runtime and IIS (Internet Information Server)	In addition to generating business services, the Visual Studio add-in allows you to create a Web service that uses the IIS (Internet Information Server) .NET runtime to execute the Web Service Engine (WSE). Although the business service is written in Natural, you can code in .NET within user exits. The add-in also allows you to send and receive SOAP messages from a Web service and generate the basic components of a web application (for example, HTML, JavaScript, or XSL). **Note:** For more sophisticated web application development, ask about Software AG's web application development tools.

Note:
Although the Natural Development Server (NDV) communication is available for the Visual Studio add-in and can be used at runtime, it is not recommended. NDV is intended for a development environment with relatively few users, while EntireX communication is intended for production applications that have many users.

Use Unicode Parameters for Your Business Service

If your Natural environment handles Unicode, the Business Service wizard can generate and invoke services that use Unicode parameters.

According to their implementation in Natural, Unicode values must be stored in fields with U format. This means that subprograms that implement your business services will have one or more U format parameters in their parameter data areas (PDAs).

Natural Business Services does not require special settings to generate business services containing Unicode format fields. However, your runtime environment must use the supplied Unicode dispatcher to consume Unicode services. To allow this, run the CSRLOAD utility and enter "Y" in the Unicode field. CSRLOAD creates the Natural Business Services server definitions.

Note:
Although the Unicode dispatcher (implemented via SPSDISPU) can process both Unicode and non-Unicode requests and replace the standard dispatcher (implemented via SPSDISP), it will only run in a Unicode-enabled Natural environment.

For more information on the CSRLOAD utility, see:

- Mainframe platforms: Load Default Repository Data
- Unix platforms: Load Default Repository Data

For more information on Unicode support, refer to the Natural documentation.