OpenEdge. Getting Started:
GUI for .NET Primer
Third party acknowledgements — See the table of contents for the "Third Party Acknowledgements" appendix.
<table>
<thead>
<tr>
<th>Sections</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>7</td>
</tr>
<tr>
<td>1. Object-oriented Programming in ABL</td>
<td>17</td>
</tr>
<tr>
<td>Classes</td>
<td>18</td>
</tr>
<tr>
<td>Data Members</td>
<td>19</td>
</tr>
<tr>
<td>Properties</td>
<td>20</td>
</tr>
<tr>
<td>Methods</td>
<td>20</td>
</tr>
<tr>
<td>Class events</td>
<td>20</td>
</tr>
<tr>
<td>Constructor</td>
<td>21</td>
</tr>
<tr>
<td>Destructor</td>
<td>21</td>
</tr>
<tr>
<td>Inheritance</td>
<td>21</td>
</tr>
<tr>
<td>Interfaces</td>
<td>23</td>
</tr>
<tr>
<td>Garbage collection</td>
<td>24</td>
</tr>
<tr>
<td>Essential ABL elements</td>
<td>25</td>
</tr>
<tr>
<td>2. GUI for .NET Environment</td>
<td>29</td>
</tr>
<tr>
<td>.NET Framework</td>
<td>30</td>
</tr>
<tr>
<td>Tools</td>
<td>33</td>
</tr>
<tr>
<td>3. Basic Types of Forms and Application Design</td>
<td>35</td>
</tr>
<tr>
<td>What is a form?</td>
<td>36</td>
</tr>
<tr>
<td>Main form</td>
<td>36</td>
</tr>
<tr>
<td>MDI and MDI child forms</td>
<td>36</td>
</tr>
<tr>
<td>Multiple, non-modal forms</td>
<td>37</td>
</tr>
<tr>
<td>Dialog forms</td>
<td>37</td>
</tr>
<tr>
<td>Creating and initializing forms</td>
<td>38</td>
</tr>
<tr>
<td>Using the WAIT-FOR statement</td>
<td>39</td>
</tr>
<tr>
<td>Waiting on a main or MDI form.</td>
<td>39</td>
</tr>
<tr>
<td>Waiting on multiple, non-modal forms</td>
<td>39</td>
</tr>
<tr>
<td>Modal dialogs</td>
<td>40</td>
</tr>
<tr>
<td>Form inheritance</td>
<td>41</td>
</tr>
<tr>
<td>Form and window coexistence</td>
<td>42</td>
</tr>
<tr>
<td>Embedding an ABL window in a .NET form</td>
<td>43</td>
</tr>
</tbody>
</table>
Contents

4. **Controls in Forms** ................................................................. 45
   Control sources ................................................................. 46
      Microsoft .NET Windows Forms UI controls ........................ 46
      OpenEdge Controls ........................................................... 48
      OpenEdge Ultra Controls for .NET ................................. 48
      Third-party controls ......................................................... 49
      ABL user controls ............................................................ 49
      ABL-derived controls ......................................................... 49
   Working with .NET forms and controls ............................... 51

5. **Events** ............................................................................ 53
   .NET event loop ................................................................... 54
   Event handlers ................................................................. 55

6. **Data Binding** ................................................................. 57
   Binding to a .NET control ...................................................... 58
   Binding to a data source ....................................................... 60
      Buffer .............................................................................. 60
      Query ............................................................................. 60
      ProDataSet ................................................................. 60
   Defining ProBindingSource tables and fields ....................... 61

7. **Service Layers** ............................................................... 63
   Business Entities ............................................................... 64
   Service Adapters and Service Interfaces ............................... 65

A. **Third Party Acknowledgements** ........................................ 67
## Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>.NET Class Inheritance Stack</td>
<td>30</td>
</tr>
<tr>
<td>Figure 2</td>
<td>OpenEdge GUI for .NET Architecture</td>
<td>31</td>
</tr>
<tr>
<td>Figure 3</td>
<td>.NET-ABL Data Binding Model</td>
<td>58</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Service Adapter and Service Interface</td>
<td>65</td>
</tr>
</tbody>
</table>
Preface

This Preface contains the following sections:

- Purpose
- Audience
- Organization
- Using this manual
- Typographical conventions
- Examples of syntax descriptions
- Sample procedures
- OpenEdge messages
Purpose

OpenEdge® supports the ability to access Microsoft .NET objects from ABL much as if they were native ABL classes. ABL access to .NET objects is designed to support the OpenEdge GUI for .NET, which provides an alternative Windows desktop user interface for ABL applications, based on .NET forms and controls. ABL allows your application to use the GUI for .NET together with the OpenEdge GUI, and provides specific support for using .NET forms and ABL windows together in the same ABL session.

For today's modern applications, OpenEdge makes available an in-the-box .NET control set, OpenEdge Ultra Controls for .NET (Ultra Controls), that provide a feature-rich GUI. Ultra Controls includes a number of full-featured .NET controls from Infragistics NetAdvantage. GUI applications can be built using either the Progress Developer Studio for OpenEdge Visual Designer, the OpenEdge ABL text editor, or, at the minimum, any text editor.

ABL supports .NET objects and Ultra Controls entirely within the ABL context by incorporating the .NET class hierarchy as a subset of the ABL class hierarchy. ABL can interact with .NET objects using enhanced ABL features without, itself, becoming a native .NET language. This allows you to continue to use ABL exactly as you have in previous OpenEdge releases that do not provide access to .NET objects.

Much has already been written about working with .NET forms and objects in ABL from a bottom-up, how-to perspective. This manual, however, takes an alternate view, providing a top-down, conceptual look at why one builds a GUI for .NET application using one approach or another, and includes the pros and cons of each approach.

This documentation does not replace and, in fact, depends upon both Microsoft and third-party vendor documentation for understanding the .NET classes that vendors provide.

In addition to this manual, which provides a guide to choosing among different options for creating a new GUI for .NET application, as well as transforming an existing OpenEdge application to use the GUI for .NET, you can find additional information on this feature in the following manuals:

- **OpenEdge Development: ABL Reference** — The complete reference to the ABL, including all language elements that support the GUI for .NET.
- **OpenEdge Development: GUI for .NET Programming** — A guide that describes how to access .NET objects using ABL to implement the OpenEdge GUI for .NET in your OpenEdge applications.
- **OpenEdge Development: GUI for .NET Mapping Reference** — A short reference to ABL elements and features of the GUI for .NET, both as they map to the terminology and C# language syntax of the Microsoft .NET Framework, and as they map to ABL elements and features of the traditional OpenEdge GUI.
- **OpenEdge Deployment: Managing ABL Applications** — A guide to managing and deploying ABL applications, with sections on requirements and features for deploying GUI for .NET applications.
Audience

This manual is written for the ABL developer who is thoroughly familiar with programming in ABL. For more information about ABL, see *OpenEdge Getting Started: ABL Essentials* and *OpenEdge Development: ABL Reference*.

It is also best if the reader is familiar with the Microsoft .NET Framework, object-oriented programming concepts, and has an understanding of another object-oriented programming language, such as C#. For more information on working with .NET in ABL, see *OpenEdge Development: GUI for .NET Programming*. For more information about object-oriented programming in ABL, see *OpenEdge Development: Object-oriented Programming*.

Having a working knowledge of the Progress Developer Studio for OpenEdge Visual Designer tool greatly enhances your productivity in building OpenEdge GUI for .NET applications. For more how-to information about using the Visual Designer, see *OpenEdge Getting Started: Introducing the Progress Developer Studio for OpenEdge Visual Designer*.

Organization

Chapter 1, “Object-oriented Programming in ABL”

Describes extensions to ABL that provide a cohesive and standard object-oriented programming model, while continuing to support the programming model available in previous releases of OpenEdge.

Chapter 2, “GUI for .NET Environment”

Describes the ability to access Microsoft .NET objects from ABL much as if they were native ABL objects.

Chapter 3, “Basic Types of Forms and Application Design”

Describes the types of .NET forms for building different types of ABL applications.

Chapter 4, “Controls in Forms”

Describes the .NET Windows Forms UI controls and non-UI controls, the built-in ABL controls, and the optional OpenEdge Ultra Controls for .NET that can be added to your application.

Chapter 5, “Events”

Describes the .NET event loop and the ABL event handlers that are used when your application uses GUI for .NET.

Chapter 6, “Data Binding”

Describes data binding in the context of the connection between an ABL data source, which can be a ProDataSet, a query on a database table or temp-table, or a buffer, and a .NET UI component.
Chapter 7, “Service Layers”

Describes the advantages of incorporating Business Entities, Service Layers, and Service Interfaces into your application.

Appendix A, “Third Party Acknowledgements”

Using this manual

OpenEdge provides a special purpose programming language for building business applications. In OpenEdge documentation, the formal name for this language is ABL (Advanced Business Language). With few exceptions, all keywords of the language appear in all UPPERCASE, using a font that is appropriate to the context. All other alphabetic language content appears in mixed case.

For the latest documentation updates see the OpenEdge Product Documentation page on PSDN: http://communities.progress.com/pcom/docs/DOC-16074.

References to ABL compiler and run-time features

ABL is both a compiled and an interpreted language that executes in a run-time engine. The OpenEdge documentation refers to this run-time engine as the ABL Virtual Machine (AVM). When the documentation refers to ABL source code compilation, it specifies ABL or the compiler as the actor that manages compile-time features of the language. When the documentation refers to run-time behavior in an executing ABL program, it specifies the AVM as the actor that manages the specified run-time behavior in the program.

For example, these sentences refer to the ABL compiler’s allowance for parameter passing and the AVM’s possible response to that parameter passing at run time: “ABL allows you to pass a dynamic temp-table handle as a static temp-table parameter of a method. However, if at run time the passed dynamic temp-table schema does not match the schema of the static temp-table parameter, the AVM raises an error.” The following sentence refers to run-time actions that the AVM can perform using a particular ABL feature: “The ABL socket object handle allows the AVM to connect with other ABL and non-ABL sessions using TCP/IP sockets.”

References to ABL data types

ABL provides built-in data types, built-in class data types, and user-defined class data types. References to built-in data types follow these rules:

- Like most other keywords, references to specific built-in data types appear in all UPPERCASE, using a font that is appropriate to the context. No uppercase reference ever includes or implies any data type other than itself.
- Wherever integer appears, this is a reference to the INTEGER or INT64 data type.
- Wherever character appears, this is a reference to the CHARACTER, LONGCHAR, or CLOB data type.
• Wherever `decimal` appears, this is a reference to the `DECIMAL` data type.

• Wherever `numeric` appears, this is a reference to the `INTEGER`, `INT64`, or `DECIMAL` data type.

References to built-in class data types appear in mixed case with initial caps, for example, `Progress.Lang.Object`. References to user-defined class data types appear in mixed case, as specified for a given application example.

Typographical conventions

This manual uses the following typographical conventions:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bold</td>
<td>Bold typeface indicates commands or characters the user types, provides emphasis, or the names of user interface elements.</td>
</tr>
<tr>
<td>Italic</td>
<td>Italic typeface indicates the title of a document, or signifies new terms.</td>
</tr>
<tr>
<td>SMALL, BOLD CAPITAL LETTERS</td>
<td>Small, bold capital letters indicate OpenEdge key functions and generic keyboard keys; for example, <code>GET</code> and <code>CTRL</code>.</td>
</tr>
<tr>
<td>KEY1+KEY2</td>
<td>A plus sign between key names indicates a simultaneous key sequence: you press and hold down the first key while pressing the second key. For example, <code>CTRL+X</code>.</td>
</tr>
<tr>
<td>KEY1 KEY2</td>
<td>A space between key names indicates a sequential key sequence: you press and release the first key, then press another key. For example, <code>ESCAPE H</code>.</td>
</tr>
<tr>
<td>Syntax:</td>
<td></td>
</tr>
<tr>
<td>Fixed width</td>
<td>A fixed-width font is used in syntax boxes, code examples, system output, and filenames.</td>
</tr>
<tr>
<td>Fixed-width italics</td>
<td>Fixed-width italics indicate variables in syntax boxes.</td>
</tr>
<tr>
<td>Fixed-width bold</td>
<td>Fixed-width bold indicates variables with special emphasis.</td>
</tr>
<tr>
<td>UPPERCASE fixed width</td>
<td>Uppercase words are ABL keywords. Although these are always shown in uppercase, you can type them in either uppercase or lowercase in a procedure.</td>
</tr>
<tr>
<td>![multi-step procedure]</td>
<td>This icon (three arrows) introduces a multi-step procedure.</td>
</tr>
<tr>
<td>![single-step procedure]</td>
<td>This icon (one arrow) introduces a single-step procedure.</td>
</tr>
</tbody>
</table>
## Examples of syntax descriptions

In this example, the outer (small) brackets are part of the language, and the inner (large) brackets denote an optional item:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period (.)</td>
<td>All statements except <strong>DO, FOR, FUNCTION, PROCEDURE, and REPEAT</strong> end with a period. <strong>DO, FOR, FUNCTION, PROCEDURE, and REPEAT</strong> statements can end with either a period or a colon.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Large brackets indicate the items within them are optional.</td>
</tr>
<tr>
<td>[ ]</td>
<td>Small brackets are part of ABL.</td>
</tr>
<tr>
<td>{ }</td>
<td>Large braces indicate the items within them are required. They are used to simplify complex syntax diagrams.</td>
</tr>
<tr>
<td>{ }</td>
<td>Small braces are part of ABL. For example, a called external procedure must use braces when referencing arguments passed by a calling procedure.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>Ellipses indicate repetition: you can choose one or more of the preceding items.</td>
</tr>
</tbody>
</table>

### Syntax

**ACCUM aggregate expression**

**FOR** is one of the statements that can end with either a period or a colon, as in this example:

```abl
FOR EACH Customer NO-LOCK:
    DISPLAY Customer.Name.
END.
```

In this example, **STREAM stream, UNLESS-HIDDEN, and NO-ERROR** are optional:

**Syntax**

```
DISPLAY [ STREAM stream ] [ UNLESS-HIDDEN ] [ NO-ERROR ]
```

In this example, **ACCUM** is a keyword, and **aggregate** and **expression** are variables:
A called external procedure must use braces when referencing compile-time arguments passed by a calling procedure, as shown in this example:

Syntax

INITIAL [ constant [, constant ] ]

In this example, EACH, FIRST, and LAST are optional, but you can choose only one of them:

Syntax

{ &argument-name }

In this example, you must include two expressions, and optionally you can include more. Multiple expressions are separated by commas:

Syntax

MAXIMUM ( expression , expression [, expression ] ... )

In this example, you must specify MESSAGE and at least one expression or SKIP [ (n) ], and any number of additional expression or SKIP [ ( n ) ] is allowed:

Syntax

MESSAGE { expression | SKIP [ ( n ) ] } ...

In this example, you must specify {include-file, then optionally any number of argument or &argument-name = "argument-value", and then terminate with }:

Syntax

{ include-file
  [ argument | &argument-name = "argument-value" ] ... }

Long syntax descriptions split across lines

Some syntax descriptions are too long to fit on one line. When syntax descriptions are split across multiple lines, groups of optional and groups of required items are kept together in the required order.
In this example, `WITH` is followed by six optional items:

**Syntax**

\[
\begin{align*}
\text{WITH} & \quad [\text{ACCUM} \ max-length ] \ [ \text{expression} \ DOWN ] \\
& \quad [ \text{CENTERED} ] \ [ n \ \text{COLUMNS} ] \ [ \text{SIDE-LABELS} ] \\
& \quad [ \text{STREAM-IO} ] 
\end{align*}
\]

**Complex syntax descriptions with both required and optional elements**

Some syntax descriptions are too complex to distinguish required and optional elements by bracketing only the optional elements. For such syntax, the descriptions include both braces (for required elements) and brackets (for optional elements).

In this example, `ASSIGN` requires either one or more `field` entries or one `record`. Options available with `field` or `record` are grouped with braces and brackets:

**Syntax**

\[
\begin{align*}
\text{ASSIGN} & \quad \{ \ [ \text{FRAME} \ frame ] \ [ \text{field} \ [ = \ \text{expression} ] ] \} \\
& \quad [ \ \text{WHEN} \ \text{expression} ] \} \ ... \\
& \quad | \ \{ \ \text{record} \ [ \text{EXCEPT} \ \text{field} \ ... \ ] \} 
\end{align*}
\]

**Sample procedures**

The optional OpenEdge Ultra Controls for .NET installation includes a number of Infragistics sample procedures that have been implemented in ABL. Each sample is contained in its own folder and can be run independently. Each sample folder contains these files:

- `readme.out` — Describes sample code
- `runner.p` — Launches sample code

You can access the sample files and details for installing the samples from the following locations:

- A self-extracting Documentation and Samples file available on the OpenEdge download page of the Progress Software Download Center
- The OpenEdge Product Documentation Overview page on PSDN:

Once installed, you can locate the example files for this manual in the following path under the OpenEdge Documentation and Samples installation directory:

```
src/prodoc/dotnetobjects
src/samples/GuiForDotNet/UltraControls
```

To compile and run these sample classes and procedures:

1. Install the samples from either location.
2. In the installation directory, locate the samples for this book under the following relative directory path:

```
src\samples\advancedgui
```
3. Start a Proenv window.
4. Change to the directory with runner.p and run this command:

```
prowin32.exe -p runner.p
```

**OpenEdge messages**

OpenEdge displays several types of messages to inform you of routine and unusual occurrences:

- **Execution messages** inform you of errors encountered while OpenEdge is running a procedure; for example, if OpenEdge cannot find a record with a specified index field value.

- **Compile messages** inform you of errors found while OpenEdge is reading and analyzing a procedure before running it; for example, if a procedure references a table name that is not defined in the database.

- **Startup messages** inform you of unusual conditions detected while OpenEdge is getting ready to execute; for example, if you entered an invalid startup parameter.

After displaying a message, OpenEdge proceeds in one of several ways:

- Continues execution, subject to the error-processing actions that you specify or that are assumed as part of the procedure. This is the most common action taken after execution messages.

- Returns to the Procedure Editor, so you can correct an error in a procedure. This is the usual action taken after compiler messages.

- Halts processing of a procedure and returns immediately to the Procedure Editor. This does not happen often.

- Terminates the current session.
OpenEdge messages end with a message number in parentheses. In this example, the message number is 200:

```
** Unknown table name table. (200)
```

If you encounter an error that terminates OpenEdge, note the message number before restarting.

**Obtaining more information about OpenEdge messages**

In Windows platforms, use OpenEdge online help to obtain more information about OpenEdge messages. Many OpenEdge tools include the following Help menu options to provide information about messages:

- Choose **Help ➔ Recent Messages** to display detailed descriptions of the most recent OpenEdge message and all other messages returned in the current session.

- Choose **Help ➔ Messages** and then type the message number to display a description of a specific OpenEdge message.

- In the Procedure Editor, press the **HELP** key or **F1**.

On UNIX platforms, use the OpenEdge pro command to start a single-user mode character OpenEdge client session and view a brief description of a message by providing its number.

**To use the pro command to obtain a message description by message number:**

1. Start the Procedure Editor: `OpenEdge-install-dir/bin/pro`

2. Press **F3** to access the menu bar, then choose **Help ➔ Messages**.

3. Type the message number and press **ENTER**. Details about that message number appear.

4. Press **F4** to close the message, press **F3** to access the Procedure Editor menu, and choose **File ➔ Exit**.
Object-oriented Programming in ABL

Before getting into a discussion about OpenEdge GUI for .NET, it helps to understand the object-oriented (OO) foundation that supports it all. This section introduces key terms and concepts of object-oriented programming in ABL.

This chapter includes the following sections:

- Classes
- Interfaces
- Garbage collection
- Essential ABL elements
Classes

Beginning with OpenEdge® Release 10.1, ABL includes support for classes. Classes allow an ABL application to be built from built-in classes, as well as user-defined class-based objects that can be defined and organized for use by an application at compile time. Class-based objects are defined and managed using standard features of object-oriented programming available in ABL, and are similar to other programming languages such as Java. Procedure-based and class-based objects can coexist in a single application.

Object-oriented programming support in ABL extends the language to provide a cohesive and standard object-oriented programming model, while continuing to support the programming model available in previous releases of OpenEdge. That is, ABL provides support for programming with classes in addition to its support for programming with procedures with full interoperability between the two. This object-oriented language support is a natural basis for writing applications that conform to the OpenEdge Reference Architecture (OERA) introduced with OpenEdge Release 10.

You define a class using the `CLASS` statement and define its members within the class block terminated by the `END CLASS` statement. A new instance of a class is created using the `NEW` function. This is comparable to running a persistent procedure using the `RUN` statement, and returning a handle to the procedure with the `SET` option.

In OpenEdge releases prior to Release 10.1A, ABL persistent procedures allow you to create and manage objects in which most of the relationships between them are created and managed at run time. ABL classes, on the other hand, contain relationships that you create at compile time. Unlike a persistent procedure, a class defines a well-structured, strong type (data type) that ABL recognizes and verifies at compile time and that can be realized as an object at run time. A persistent procedure is weakly typed and therefore can only be verified at run time. The data type of a class is defined by its members. That is, as long as the interface defined by the members of a class does not change, the data type of the class remains the same no matter how much you change its implementation. This has significance for determining when you have to recompile a given class or procedure when a class upon which it depends changes.

Class members are themselves strongly typed elements that provide data and behavior for the class, similar to the variables, handle-based objects, internal procedures, and user-defined functions of a persistent procedure. Again, as strongly typed elements, the members of a class are verified at compile time, while the similar elements of a procedure can only be verified at run time. In addition, classes are built in strict hierarchies, where any given user-defined class inherits strongly-typed members from another class as well as defining members of its own, all of which can be inherited by yet another user-defined class. This strong typing of classes and their inheritance relationships allows for robust error checking at compile time, when an application is in the development cycle, and long before it is available to end users.

In terms of the static DEFINE versus dynamic CREATE ABL perspective, all OO programming in ABL is dynamic. You instantiate objects explicitly at run time and you set their properties much as you do with the ABL CREATE statement.

ABL and .NET organize their class-based type hierarchies using different mechanisms. ABL organizes user-defined types into packages, which are logical pathnames that correspond to physical directory paths relative to PROPATH. It then stores the types in class definition (.cls) files that reside in the corresponding directories.
On the other hand,.NET organizes its object types into namespaces, which are logical pathnames that are defined along with the object types associated with them. .NET namespaces are completely logically defined and have no correspondence to physical storage of any kind. Instead,.NET stores object type definitions, along with their defined namespaces, in assembly files. An assembly can be a Windows dynamic link library (.dll) or executable (.exe) file that is formatted to store one or more types associated with their individual namespaces. These assemblies are then stored in designated locations, which can include locally configureable directories or a standard location defined by .NET known as the Global Assembly Cache (GAC). OpenEdge supports options for specifying the location of the assemblies you access as part of the GUI for .NET.

Thus, a .NET namespace is analogous to an ABL package in that it provides a means to organize and uniquely identify types. But instead of being associated with an actual directory structure where the type definition is stored, a namespace is a logical construct that is associated with each type that is stored in an assembly.

Both .NET namespaces and ABL packages are hierarchical. The representation of each type of hierarchy in ABL uses the same dot-separated notation, whether it is logical for namespaces or physical directory-based for ABL classes.

Note that this and other OpenEdge product documentation does not replace and, in fact, depends on both Microsoft and third-party vendor documentation for understanding the .NET classes that vendors provide. OpenEdge supplies code samples that provide examples of using classes. These code samples can be found on the self-extracting Documentation and Samples file available on the OpenEdge download page of the Progress Software Download Center under the following relative directory path:

```
src/prodoc/getstartoop
```

### Data Members

Variables defined within the main block of a class are known as data members. Like variables defined in the main block of a procedure, in an internal procedure, or in a user-defined function, data members can be defined as a scalar or an array type, such as CHARACTER, INTEGER, DATE, DECIMAL, LOGICAL, and so on.

In keeping with strong typing, the compiler checks assignments that involve data members at compile time. Assignments involving data members that are considered invalid, for example, performing an integer operation on a character string, are not allowed and raise ERROR. Strong typing allows for catching many potential type errors at compile time. Data members are PRIVATE by default.

Like a variable defined within the main block of a procedure, a data member of a class is scoped to the entire class and can be used freely in methods of the class. Unlike a variable defined within the main block of a procedure, a data member of a class can be defined with an access mode that determines where it can be read or updated. For example:

- **PUBLIC** — A public data member can be read or updated by the defining class, any inheriting class, or any class that instantiates the class.
• **PROTECTED** — A protected data member can be read or updated by the defining class and any inheriting class. An instance can access the protected member of a second instance that is at the same level or in a superclass in the class hierarchy.

• **PRIVATE** — A private data member can be read or updated by the defining class, which is the default access mode. An instance can access the private data member of another instance if both instances are from the same class.

### Properties

Similar to ABL variables, *properties* in class files maintain data or state. Like a data member in a class, a property can be defined with an access mode (PUBLIC, PROTECTED, PRIVATE) that determines where it can be read or updated. Properties are PUBLIC by default.

But unlike ABL variables or class data members, properties can be defined with a GET accessor and/or a SET accessor for reading and writing a property’s value, respectively. These accessors, which are blocks of ABL statements, can contain code which is executed when the property is read (GET) or assigned (SET). If either the GET accessor or the SET accessor is omitted, the property is defined as write-only or read-only, respectively.

While there is no direct equivalent to properties in traditional ABL, properties can be simulated in persistent procedures, for example in the ADM2, by defining user-defined functions that control the reading and writing of an ABL variable.

### Methods

Similar to internal procedures and user-defined functions in procedure files, *methods* provide behavior in class files. Methods can have parameters and can define a return type. A method can also have a return type of VOID, which means that it does not return a value, and is therefore comparable to an internal procedure, as opposed to a function.

Like data members and properties, methods can be defined with an access mode, either PUBLIC, PROTECTED, or PRIVATE, that determines where and how the method is called. Methods are PUBLIC by default.

As with internal procedures and user-defined functions, you can raise ERROR within a method or property GET accessor or SET accessor using RETURN ERROR or UNDO, THROW. This statement allows you to raise (THROW) an error as an object. An error object, part of the structured error handling feature introduced in OpenEdge Release 10.1C, is a class-based object that can be trapped with custom error handlers.

### Class events

Similar to named events in procedure files, *class events* provide a way to subscribe a method or procedure that executes in response to application conditions. Unlike a named event, which is defined when you publish the event to execute its event handlers, you define class events as members of a class independent of any event handlers that might execute for them. A class event is thus defined with a name, and can have parameters and a return type (similar to a method), which define the signature that must match the signature of any handler for this event. The return type for a class event and its handler is always VOID. Finally, unlike named events that can
have only internal procedures as handlers, class events can have both internal procedures and methods as handlers, as long as the handler signatures match the signatures of their respective events.

Like data members, properties, and methods, events can be defined with an access mode, either PUBLIC, PROTECTED, or PRIVATE, that determines where and how the event can have a handler subscribed for it. Events are PUBLIC by default. However, note that events can only be published from within the class where they are defined, regardless of their access mode.

A class event can have multiple handlers subscribed for it in different classes and procedures of an application, and these handlers can also be unsubscribed. ERROR can also be raised for a class event on the statement that publishes the event when any one of its subscribed handlers raises ERROR.

Constructor

Similar to a Main Block in an ABL persistent procedure, a constructor is a block of code that is executed when a class-based object is instantiated. Constructors are special methods that can have parameters, but do not have a return type. In a hierarchy, a constructor in each class of the hierarchy is executed.

Destructor

A destructor is a special type of method that is executed when a class-based object is deleted, either as a result of automatic garbage collection, or programmatically using the DELETE OBJECT statement. Destructors do not have parameters and do not have a return type. In a hierarchy, all destructors of the classes in the hierarchy are executed. This is a good place to put cleanup code that you want to ensure is run before the class-based object is deleted. That is, unless the entire session ends abnormally, in which case the destructor might not run.

Inheritance

Inheritance of common data and behavior is at the core of object-oriented programming. A class inherits the non-private data members, properties, methods, and events of its super class (also referred to as the base or parent class). The inheriting class can also extend the interface of its super class by defining its own members, making all of its inherited and defined non-private members available for inheritance by another class (a subclass). In this way, each class in a class hierarchy derives its interface based on the super classes above it.

One of the key benefits of inheritance is code reuse, coupled with defined relationships between classes. The compiler cross-checks the contents of super classes and subclasses in a class hierarchy at compile time, which is not done for procedures. In an inheritance hierarchy of ABL classes, a method in a subclass can also override and redefine a method of the same name and signature in a super class. In this case the most-derived (inherited) version of the overridden method is invoked at run time.
Compared to internal procedures and user-defined functions which have no inherent relationships among them, object-oriented inheritance provides for organizing classes and factoring out common elements of related classes. You can also define a form of inheritance with procedure objects using super procedures. A set of related super procedures can act in much the same way as a class hierarchy, but their relationships are validated at run time.

These are some notable ABL elements for working with inherited classes:

- **INHERITS option, CLASS statement** — The `INHERITS` option specifies the super class from which your derived class inherits. If you do not specify a super class, all ABL user-defined classes derive from the top-level super class (*root class*), `Progress.Lang.Object`, by default. Thus, the interface defined by every ABL class includes members inherited from this ABL root class.

- **METHOD, DEFINE PROPERTY, DEFINE EVENT, DEFINE VARIABLE, and various other DEFINE statements** — These statements allow you to extend the inherited interface with additional methods, properties, events, and data members of various types.

- **OVERRIDE option, METHOD statement** — When an ABL class inherits a class, it can modify behavior of the base or parent class by overriding public and protected instance methods in the inherited class hierarchy with its own ABL `OVERRIDE` methods.

- **SUPER statement** — The `SUPER` statement invokes a constructor for the immediate super class as the first statement in a constructor of the defining class.

- **THIS-OBJECT statement** — The `THIS-OBJECT` statement invokes a constructor defined in the current class definition.

- **SUPER system reference** — Using the `SUPER` system reference, a subclass can call the PUBLIC and PROTECTED instance methods of its super class in the inherited class hierarchy. If the specified method definition is not found in the subclass's immediate super class, the ABL compiler repeatedly looks to the next super class in the inherited class hierarchy until it finds the definition.
Interfaces

Interfaces, which have long existed in OO languages such as C#, are also supported in ABL. An interface is an object type, like a class, that consists of one or more method, property, or event prototypes that a class must implement. That is, the methods, properties, and events defined in an interface are empty and do not contain any executable code. Any class that includes an interface in its definition must implement all the methods, properties, and events of that interface. Note that an ABL interface never defines prototypes for data members that a class must implement. However, it can include temp-table and ProDataSet definitions to specify parameters for method or event prototypes that define temp-tables and ProDataSets as parameters.

You define an ABL interface using the `INTERFACE` statement and define its members within the interface block terminated by the `END INTERFACE` statement. Like a class type, you can define an interface type in a package. Similarly, a .NET interface, like a .NET class, can be defined in a namespace.

To implement one or more interfaces in a user-defined class, you specify them with the `IMPLEMENTS` option of the `CLASS` statement. By implementing an interface, you assure any other code (class or procedure) that uses your class that the methods, properties, and events defined by the interface have been implemented. You can use a class object reference defined as an interface to access the interface methods, properties, and events, even for object references of different class types, as long as the given class implements the specified interface.

Defining and implementing interfaces is useful when you want to build classes with a standard set of methods, properties, and events. The implementation of interfaces is validated at compile time.
**Garbage collection**

ABL developers are familiar with the use of the `DELETE OBJECT` statement for destroying handle-based objects, such as a widget, a procedure, a server, or a socket. Explicitly deleting class-based objects is not necessary because of garbage collection. Class-based objects are deleted during garbage collection when all references to that object have been removed, that is, when its reference count goes to zero.

As long as you need an object, you must ensure that you maintain an object reference to that object. You can always assign the object reference to another variable before it goes out of scope, or you can pass it to another procedure, where you can continue to manage the object until either you delete it or it is deleted by garbage collection.

Garbage collection does not delete a class-based object with a circular reference. For example, Object A has a reference to Object B and Object B has a reference to Object A. To delete an object with a circular reference you must use the `DELETE OBJECT` statement.
Essential ABL elements

The OpenEdge GUI client provides access to .NET forms (analogous to ABL WINDOW widgets) and controls using object-oriented extensions to ABL. Using ABL, you create forms, add controls, access data, subscribe to events and write event handlers for .NET controls. Accessing and using .NET objects is very similar to working with ABL objects using classes, however, there are some exceptions, due to the differences between supported constructs in .NET and ABL.

The following are examples of ABL elements that are used when working with .NET forms and controls:

- **USING** — The USING statement makes code easier to read and less verbose by allowing you to abbreviate the full class or object type name in code. For example:

  ```abl
  USING Progress.Windows.Form.
  USING System.Windows.Forms.*.
  USING Infragistics.Win.UltraWinListView.*.
  ```

- **CLASS** — The CLASS statement defines a user-defined class, representing a user-defined data type. Its characteristics are defined by a set of data members, including properties, that define class data and methods that define class behavior. For example:

  ```abl
  CLASS MyUserControl:
    ...
  END CLASS.
  ```

- **INHERITS** — When defining a class, the INHERITS option specifies the super class whose state and behavior this class inherits. For example:

  ```abl
  CLASS MyUserControl INHERITS Progress.Windows.UserControl:
    ...
  END CLASS.
  ```

- **IMPLEMENTS** — Classes can implement one or more interfaces, each of which are defined as a separate .cls file starting with an INTERFACE statement. An interface declares a common public mechanism to access behavior that one or more classes can define, even though these classes do not inherit from another common class. Interfaces allow you to define and manage common behavior that might be implemented differently in different classes and for different purposes. For example:

  ```abl
  CLASS MyUserControl IMPLEMENTS IUpdatable:
    ...
  END CLASS.
  ```
Chapter 1: Object-oriented Programming in ABL

- **PUBLIC** — The **PUBLIC** access mode specifies that the object, code block, or data member can be accessed by the defining class, any of its inheriting classes, and any class or procedure that instantiates that class (through an object reference), or has access to the object reference. In this example, **VOID** indicates that the method does not return a value:

  ```abl
  METHOD PUBLIC VOID addRecord( ):
  ...
  END METHOD.
  ```

- **PROTECTED** — The **PROTECTED** access mode specifies that the object, code block, or data member can be accessed by the defining class and any of its inheriting classes. An instance can access a protected member of a second instance that is at the same level or in a super class in the class hierarchy. For example:

  ```abl
  DEFINE PROTECTED VARIABLE cName AS CHARACTER NO-UNDO.
  ```

- **PRIVATE** — The **PRIVATE** access mode specifies that the object, code block, or data member can be accessed by the defining class. An instance can access the private member of another instance if both instances are from the same class. For example:

  ```abl
  DEFINE PRIVATE VARIABLE bindSrc1 AS Progress.Data.BindingSource NO-UNDO.
  ```

- **CONSTRUCTOR** — Similar to a Main Block in an ABL structured procedure (.p or .w), a **CONSTRUCTOR** is a block that is executed when a class-based object is instantiated. A constructor of a class is a special method that is called when a class is instantiated, for example, using the **NEW** function. A constructor is named the same as the class. For example:

  ```abl
  CONSTRUCTOR PUBLIC PurchaseOrderForm( ):
  ...
  END CONSTRUCTOR.
  ```

- **DESTRUCTOR** — The **DESTRUCTOR** is a code block that is executed when a class-based object is deleted, either as a result of garbage collection or by using the **DELETE OBJECT** statement. This block is analogous to an **ON CLOSE OF THIS-PROCEDURE DO: block** in a persistent procedure. A destructor is named the same as the class. For example:

  ```abl
  DESTRUCTOR PUBLIC PurchaseOrderForm( ):
  ...
  END DESTRUCTOR.
  ```
• **METHOD** — The **METHOD** is a code block that provides class behavior. Methods are similar to internal procedures and user-defined functions, in that they encapsulate object behavior. For example:

```
METHOD PRIVATE VOID InitializeComponent():
...
END METHOD.
```

• **NEW** — **NEW** function creates an instance of a class and returns an object reference to that instance. The new or derived class is a subclass of its super class parent, inheriting its **PUBLIC** and **PROTECTED** data members, methods, and properties. All classes in ABL (including .NET classes) have the root super class `Progress.Lang.Object`. For example:

```
DEFINE VARIABLE bindSrc1 AS Progress.Data.BindingSource NO-UNDO.
bindSrc1 = NEW Progress.Data.BindingSource{ }.
```

• **THIS-OBJECT** — An object reference to the class instance currently running. **THIS-OBJECT** is useful as a qualifier when the class member reference name is a reserved ABL keyword. The primary use of **THIS-OBJECT** is to pass a reference of the current object to another routine and to improve readability. For example:

```
THIS-OBJECT:Display{ }.
```

• **VALID-OBJECT** — The **VALID-OBJECT** function returns **TRUE** if the object reference points to a valid ABL or .NET object instance. For example:

```
VALID-OBJECT(MyObjectRef).
```

• **CAST** — **CAST** function returns a new object reference to the same class instance as an existing object reference, but with a different data type. The underlying object does not change, but the reference to it has changed. For example:

```
CAST(MyObjectRef, MySubClass).
```
GUI for .NET Environment

The GUI for .NET environment is comprised of extensions to the OpenEdge GUI client, ABL elements, and tools that support the ability to access Microsoft .NET objects from ABL much as if they were native ABL objects.

This chapter contains the following sections:

- .NET Framework
- Tools
The OpenEdge GUI client supports the ability to access Microsoft .NET objects from ABL much as if they were native ABL objects. ABL access to .NET objects is designed to support the OpenEdge GUI for .NET, which provides an alternative Windows desktop user interface for ABL applications based on .NET Windows Forms forms and controls. ABL enables your application to use the GUI for .NET together with the traditional OpenEdge GUI, and provides particular support for using .NET forms and ABL windows together in the same ABL session.

ABL provides access to objects in a .NET class library. Figure 1 shows that all class-based objects in ABL are derived from the `Progress.Lang.Object` root class.

![Figure 1: .NET Class Inheritance Stack](image)

ABL supports .NET controls within ABL. ABL can interact with .NET objects using enhanced ABL features without, itself, becoming a native .NET language. This allows you to continue to use ABL as you have in previous OpenEdge releases. References to .NET controls in ABL are made using object-oriented (OO) syntax. You can make those references from procedural or class-based code or any combination.

Figure 2 represents the OpenEdge GUI for .NET architecture at a high level. In support of .NET objects and events, ABL incorporates the ABL Virtual Machine (AVM) and the .NET Common Language Runtime (CLR) within a single OpenEdge client process. The CLR is the core run-time engine for executing applications in the Microsoft .NET Framework.

What makes communication between the ABL and .NET possible is the bridge that provides the transformation by allowing information to go back and forth between the AVM and the CLR. When the CLR gets a request, such as to run a .NET method, the bridge converts the data between the CLR and the AVM.
On the AVM side you define your components, do your data access, do I/O blocking (a version of the \texttt{WAIT-FOR} statement), and write event handlers in ABL. On the .NET side, the CLR handles the visualization of the .NET UI. The user interacts with the CLR, because that is where the user interface is running.

![OpenEdge GUI for .NET Architecture](image)

Figure 2: OpenEdge GUI for .NET Architecture

Figure 2 depicts a simplified sample flow of control between the AVM and the CLR:

1. In ABL, a new .NET form is created. The form subscribes an ABL event handler method to its FormClosing event, and the form is run.
2. The request is passed over the bridge to the CLR to create the .NET form.
3. The form is displayed using a .NET method specified by the \texttt{WAIT-FOR} statement.
4. The user does some work and closes the form.
5. The FormClosing event message is sent to all subscribers from .NET.
6. The bridge sends the message to the AVM and causes the event handler method to run.

There is no one application architecture supported by GUI for .NET. You can either keep your existing architecture or follow a modularized design such as the OERA. There is no requirement to change your application architecture, though some applications might require more modifications than others to provide a GUI for .NET interface that communicates with existing data management and business logic. There might be compelling reasons to update parts of your application architecture to take advantage of advanced features that GUI for .NET provides. Many architectural topologies are supported, including fat client, client/server, and n-tier. GUI for .NET does not require you to change from one topology to another.

Just as there is no one UI technology that is applicable for all of today’s applications, an application’s primary use often dictates what UI technology works best. To put GUI for .NET development in perspective, it helps to compare this technology with other OpenEdge UI technologies.
Table 1 lists the various OpenEdge UI technologies, their primary use, and what development tools are best suited for building these types of business applications. These are suggestions; the table is not meant to be all-inclusive and there is overlap between technologies, their primary use, and tools.

### Table 1: OpenEdge UI technologies

<table>
<thead>
<tr>
<th>UI Technology</th>
<th>Primary Use</th>
<th>Tools to Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI for .NET</td>
<td>Build modern, state-of-the-art GUIs for Microsoft Windows in ABL</td>
<td>Progress Developer Studio for OpenEdge Visual Designer</td>
</tr>
<tr>
<td>OpenEdge GUI</td>
<td>Build Windows 95 user interface look and feel</td>
<td>OpenEdge Studio, Progress Developer Studio for OpenEdge AppBuilder, Editor</td>
</tr>
<tr>
<td>Open Client</td>
<td>Build alternate UIs (.NET &amp; Java™) to OpenEdge applications, requires AppServer</td>
<td>Open Client Toolkit</td>
</tr>
<tr>
<td>WebSpeed®</td>
<td>Build web-browser based applications with zero client footprint</td>
<td>WebSpeed Workshop</td>
</tr>
<tr>
<td>Character</td>
<td>Build traditional green-screen user interfaces</td>
<td>OpenEdge Studio, Progress Developer Studio for OpenEdge Editor, 4GL Development System</td>
</tr>
</tbody>
</table>

WebClient can be used as a deployment platform for OpenEdge GUI, GUI for .NET, and Character UIs.
Tools

The primary tool to use for GUI for .NET application development is Progress Developer Studio for OpenEdge, an Eclipse-based Integrated Development Environment (IDE) that includes a number of plug-ins, for example the OpenEdge Visual Designer, the OpenEdge Editor, and the OpenEdge Class Browser:

- **OpenEdge Visual Designer** — The Visual Designer is a WYSIWYG tool that supports visual building of complex GUIs in ABL based on .NET objects. You create user interfaces with Visual Designer by dragging and dropping .NET controls onto a Design Canvas. The Visual Designer generates ABL code written as classes that encapsulates the creation of the WYSIWYG design code. You can get some hands-on experience working with the Visual Designer in *OpenEdge Getting Started: Introducing the Progress Developer Studio for OpenEdge Visual Designer*.

- **OpenEdge Editor** — The Editor is a full-featured, color-coded editor for working with UI class files. For example, you run the Editor when you want to add methods that implement event handlers. Complementary to the OpenEdge Visual Designer.

- **OpenEdge Class Browser** — The Class Browser is an Eclipse view that lists the members of either an ABL or a .NET class, showing the signature for each member in ABL syntax.
ABL supports several types of .NET forms for building several types of applications. A built-in form class can be used to create standard, borderless, and floating windows. You can also use this class to create modal windows such as a dialog. A special form, the Multiple Document Interface (MDI) form, can contain other forms as child forms, including ABL windows that you created in OpenEdge version 11 or earlier versions.

This chapter includes the following sections:

- What is a form?
- Using the WAIT-FOR statement
- Form inheritance
- Form and window coexistence
- Embedding an ABL window in a .NET form
What is a form?

Your GUI for .NET application will require the use of one or more types of forms. A form is a representation of any window displayed in your application. ABL supports several types of forms for building different types of applications. The OpenEdge .NET Progress.Windows.Form class can be used to create standard, borderless, and floating windows. You can also use this class to create modal windows, such as a dialog. A special kind of form, the Multiple Document Interface (MDI) form, can contain other forms as child forms, including ABL windows that you might have created in OpenEdge version 11 or earlier.

The form types you choose to meet your application requirements determine the objects, controls, methods, and event handlers necessary to tie it all together.

In ABL a WINDOW widget is created using the CREATE WINDOW statement. In this example the variable custWin contains a handle to the newly created created window:

```
DEFINE VARIABLE custWin AS HANDLE NO-UNDO.
CREATE WINDOW custWin ASSIGN ...
```

A .NET form is similar to an ABL WINDOW widget. In ABL a .NET form is typically an instance of the Progress.Windows.Form class and is created using the NEW function. In this example, FormObjectRef contains an object reference to the new .NET form:

```
DEFINE VARIABLE FormObjectRef AS CLASS Progress.Windows.Form NO-UNDO.
FormObjectRef = NEW Progress.Windows.Form( ).
```

Main form

In this architecture a main, controlling form acts as a container for the objects, controls, event handlers, and other components that make up the application. A main form is analogous to an ABL WINDOW with an associated FRAME widget, which is either defaulted or explicitly defined.

The main form type is useful for dashboard applications. All other application forms are children of the main form. To construct your own application-specific forms use the Visual Designer.

MDI and MDI child forms

A Multiple Document Interface (MDI) form is a window that acts as the background of an application and is the container for MDI child forms. An application can have only one MDI form object but many MDI child forms. You can construct your own application-specific MDI form and child forms using the Visual Designer. If an MDI child form has menus, the child form's menu bar is merged with the MDI form object's menu bar when the MDI child form is active at run time. Your application determines how the menu merge will be performed.
The closest thing to MDI using ABL widgets is an ABL WINDOW containing multiple FRAME widgets. Even though each FRAME is parented to the same WINDOW, the similarity to true MDI ends there. While a WINDOW can contain a MENU-BAR widget, a FRAME cannot, so the idea of merging a child FRAME’s menu with its parent WINDOW menu does not apply.

A minimized MDI child form is displayed as an icon within the MDI form. MDI child forms are designed independent of the MDI form, but are always contained within the MDI form at run time.

In ABL, you typically build a .NET MDI using different instances of Progress.Windows.Form. The difference between the MDI main form and the MDI child forms are in the property settings of each form instance.

You can also embed an existing ABL window in a special OpenEdge .NET MDI child form (Progress.Windows.MDIChildForm). When embedded in this way, only the client area of the original ABL window is displayed. The embedded window’s title, menu, status area, message area, and border are not displayed. Embedding an ABL window is discussed in more detail in the “Embedding an ABL window in a .NET form” section on page 43.

**Multiple, non-modal forms**

An application with multiple, non-modal forms consists of several equal forms which are displayed at the same time or separately. This form type is useful when a set of windows is needed to perform a task and when the forms to display are determined at run time depending on the user and the task. To construct your own application-specific forms use the OpenEdge Visual Designer.

**Dialog forms**

Dialogs are modal forms that inherit from Progress.Windows.Form and that block for user input. Dialog forms are used to interact with the user and retrieve information, in a similar way to an ABL FRAME defined WITH VIEW-AS DIALOG-BOX.

You can construct your own application-specific dialog in the OpenEdge Visual Designer and add controls such as Label, Textbox, and Button to customize the dialog form to your specific needs. The .NET Framework also includes predefined dialog forms, such as File Open and message boxes, which you can use programmatically in your own applications.

You can use ABL dialog-boxes and .NET dialog forms in the same application. The ABL MESSAGE statement is similar to the System.Windows.Forms.Application:MessageBox( ) method, both of which display similar types of alert boxes which represent a different type of dialog using a dedicated event loop. The ABL SYSTEM-DIALOG COLOR statement is comparable to the .NET System.Windows.Forms.ColorDialog or System.Windows.Forms.FileDialog.
Creating and initializing forms

The OpenEdge Visual Designer, available as an Progress Developer Studio for OpenEdge plug-in, is the WYSIWYG designer tool for building GUI for .NET applications. While visually designing a form, an ABL class file is generated for the form. The Visual Designer keeps track of all .NET assemblies and resources that you use in a project.

Within a GUI for .NET application, you can plan a gradual migration by converting some of your main windows to .NET forms and leaving the rest of the UI as ABL windows. This is possible because you can display ABL windows alongside .NET forms. OpenEdge allows parenting of forms to windows and windows to forms using the OpenEdge PARENT attribute on the window handle.
Using the WAIT-FOR statement

For an application that uses non-modal .NET forms, whether or not it also uses ABL windows, you must use a .NET event loop controlled by a single WAIT-FOR statement.

ABL supports a version of the WAIT-FOR statement that instructs the AVM to stop executing the current block and remain in a wait state until a .NET method that it calls returns. The AVM continues to respond to all incoming ABL events as well as .NET events, and it executes any associated triggers, event procedures, or .NET event handlers while in this wait state. To accommodate the different .NET form types, the WAIT-FOR statement supports multiple syntax versions. If your application uses non-modal .NET forms, including MDI forms, this .NET-oriented WAIT-FOR statement must be the first, non-modal WAIT-FOR of any kind in your application.

Waiting on a main or MDI form

Many GUI applications have a single, main form for the application. When the application starts, the form is displayed and when the form is closed, the application terminates. When the application waits on a main or MDI form, the ABL WAIT-FOR statement must specify Application:Run() with the main form passed in as a parameter. This starts a .NET message loop which is used to capture events and run ABL event handlers.

This is an example of the WAIT-FOR statement for an application that has a single, main form:

```
WAIT-FOR Application:Run( FormObjectRef ).
```

When the main form FormObjectRef is specified in the WAIT-FOR statement, the form is displayed when the message loop is started.

Closing the main form terminates the WAIT-FOR. This continues execution in the current block in your program immediately after the WAIT-FOR. The application can close the main form by calling the Close() method on FormObjectRef.

Waiting on multiple, non-modal forms

Waiting on a GUI application with multiple, independent, non-modal forms is set up differently than waiting on a main form. In this case, no form is specified on the WAIT-FOR statement. None of the forms are automatically displayed, so the Show() method must be called explicitly to display a form. To display forms when the application is started the Show() method is called on each form prior to the WAIT-FOR statement. The actual display of any forms is delayed until the WAIT-FOR statement executes. For example:

```
Form1ObjectRef:Show( ).
Form2ObjectRef:Show( ).
WAIT-FOR Application:Run( ).
```

Calling the Close() method on all the forms does not end the WAIT-FOR. The application must end the WAIT-FOR by calling the Application:Exit() method.
Modal dialogs

The `ShowDialog()` method on the form is used in the `WAIT-FOR` statement to display and wait for a modal .NET dialog box. If you display the modal dialog box within an existing .NET event loop, the nested `WAIT-FOR` must be satisfied before returning to the outer `WAIT-FOR`. The application ends the `WAIT-FOR` by calling the `Close()` method on the modal form object.

In this example, the `rDateForm` form is created and displayed as a modal dialog using the `ShowDialog()` method:

```plaintext
rDateForm = NEW Progress.Examples.MyDialog.
...  
WAIT-FOR rDateForm:ShowDialog().
```

This is an example of the `WAIT-FOR` statement for a modal dialog in which the parent form is specified:

```plaintext
WAIT-FOR FormObjectRef:ShowDialog( ParentFormObjectRef ).
```

ParentFormObjectRef is an object reference to the dialog’s parent form. This version of the `ShowDialog()` method allows the dialog to display centered over the parent form.
Form inheritance

GUI for .NET supports inheritance from .NET forms. All of your application-specific forms typically inherit from the Progress.Windows.Form class. You might want to create your own ABL form class and then derive forms from that. This technique is a good idea for enforcing a company standard across all forms in your application. One advantage of putting common behavior in a super class is that feature changes or bug fixes to that class are propagated to all derived classes.

However, putting too much functionality in a super form class might result in having to write workaround code (overrides) in derived forms to get the particular instance behavior you want. So, try to put in a super class only functionality that is required by all derived classes.

OpenEdge applications can support both ABL windows and .NET forms. .NET forms can be mixed with ABL windows in the same application. For Windows desktop applications that already use the OpenEdge GUI, you can add new GUI for .NET forms without changing the ABL windows of the existing user interface. The windows and forms can run side-by-side and one can parent the other. Each window and form is an independent entity in the application.
Form and window coexistence

To allow forms and windows to coexist in an application, several ABL enhancements work to tie them together. A FormProxy object is a wrapper that is created for every ABL window. Similar to a form, the FormProxy allows the window to reside on the form chain, which is a list of all running forms.

Except for the default window, a FormProxy is created for windows that are explicitly created in the session, not for windows created internally by the AVM. The AVM maintains form proxies, automatically creating and destroying them. Even though you do not work with FormProxy objects directly, it is helpful to know about them, since they exist on the form chain.

A shadow window widget, which is created for every form, provides a means of parenting a window to a form. The shadow window has no visualization.
Embedding an ABL window in a .NET form

For Windows desktop applications that already use the OpenEdge GUI, you can add new GUI for .NET forms without changing the ABL windows of the existing user interface. Another way to integrate ABL windows with GUI for .NET forms is to embed the client area of ABL windows into a .NET form.

There are several options for embedding OpenEdge windows in a .NET form. Existing ABL windows can be shown embedded in:

- A bounded area in a form, inherited from `Progress.Windows.WindowContainer`. The developer determines the size and location of the container.

- An MDI child form, inherited from `Progress.Windows.MDIChildForm`. The size and location of the embedded window are determined by the size of the MDI child form client area.

The embedded window continues to exist on the ABL session window chain.
Controls in Forms

ABL forms are containers for .NET Windows Forms UI controls and non-UI controls, built-in ABL controls, OpenEdge Ultra Controls for .NET, and other third-party controls.

This chapter is described in the following sections:

- Control sources
- Working with .NET forms and controls
Control sources

ABL-derived forms can be containers for .NET Windows Forms UI and non-UI controls. .NET contains a variety of controls that you can add to forms—controls that display text boxes, buttons, drop-down boxes, radio buttons, and even Web pages. From a coding perspective a .NET form is made up of a set of controls.

GUI for .NET controls can also come from several different areas, the difference being in how you obtain the control:

- **Microsoft .NET UI Controls.** This is the OpenEdge name for the .NET Framework controls that are installed as visual design components.
- **OpenEdge Ultra Controls for .NET (Ultra Controls),** based on Infragistics NetAdvantage for Windows Forms.
- **OpenEdge .NET controls / components:**
  - Window Container control (`Progress.Windows.WindowContainer`).
  - ProBindingSource control (`Progress.Data.ProBindingSource`) for binding data to a control.
- **Other 3rd party .NET Windows Forms controls.**
- **ABL-derived .NET custom controls:** user controls and inherited controls.

Control vendors normally provide detailed documentation and help on using their controls. While this documentation is often written assuming the use of a .NET language, the description of a control and its properties and events are all relevant for its use in ABL. You can access this documentation in the following ways:

- Visit the vendor’s website. For example, to obtain more information on Microsoft .NET UI Controls and all .NET object types available with the .NET Framework version 4.0, see the Class Library reference documentation at the following location:


- Open the control in the OpenEdge Class Browser to see control definitions using a modified ABL syntax.
- Press F1 after you place the control on the form within Progress Developer Studio for OpenEdge Visual Designer. The online help opens on the respective vendor site.

**Microsoft .NET Windows Forms UI controls**

Microsoft Windows Forms supports a variety of controls that you can add to forms creating Windows-based applications that take full advantage of the rich user interface features available in the Microsoft Windows operating system. Windows Forms controls support a variety of features, from the most basic to features that emulate the functionality in high-end applications like Microsoft Office.
Menus and toolbars

Windows Forms contains a rich set of controls to create custom toolbars and menus with modern appearance and behavior. The ToolStrip, MenuStrip, ContextMenuStrip, and StatusStrip controls can be used to create toolbars, menus, context menus, and status bars, respectively.

Controls

Some Windows Forms controls are designed for data entry within the application, such as TextBox and ComboBox controls. The TextBox is similar to the ABL EDITOR and FILL-IN widgets. Other controls display application data, such as Label and ListView. There are controls, such as Button, for invoking commands within the application. The WebBrowser control and managed HTML classes, such as HtmlDocument, let you display and manipulate HTML pages within your Windows Forms application. The MaskedTextBox control is an advanced data entry control that lets you define masks that accept or reject user input.

Layout

Several important controls in Windows Forms define the layout of controls on a display surface, such as a form or user control. FlowLayoutPanel lays out all the controls it contains in a serial manner, and TableLayoutPanel lets you define cells and rows for laying out controls in a fixed grid. SplitContainer divides your display surface into two or more adjustable parts.

Data viewing

Windows Forms defines a rich architecture for binding to data sources such as databases and XML files. The DataGridView control provides a customizable table for displaying data, and lets you customize cells, rows, columns, and borders. The DataGridView functions like the ABL BROWSE widget. Using the OpenEdge ProBindingSource control, you can also use these controls to view ABL data sources.

Components

Besides controls, there are other classes that provide visual or non-visual features to a Windows-based application. These classes are referred to as components. Components, such as ToolTip and ErrorProvider, enhance control capabilities or provide information to the user. With the Help and HelpProvider Components, you can display Help information to a user of your applications.

Common dialog boxes

Microsoft provides several common dialog boxes that you can use to give your application a consistent user interface when performing tasks such as opening and saving files, manipulating the font or text color, or printing files. The OpenFileDialog and SaveFileDialog classes provide the functionality to display a dialog box that lets the user locate and enter the name of a file to open or save. The FontDialog class displays a dialog box to change elements of the Font used by your application. The PageSetupDialog, PrintPreviewDialog, and PrintDialog classes display dialog boxes that enable the user to control aspects of printing documents.
OpenEdge Controls

OpenEdge provides four notable controls, OpenEdge Window Container (Progress.Windows.WindowContainer class), OpenEdge ProBindingSource (Progress.Data.BindingSource, ABL User Control (Progress.Windows.UserControl), and ABL Inherited Control:

- **OpenEdge Window Container** — The Window Container is an OpenEdge .NET container control used for embedding an ABL window in a .NET form. This control, which is available in the Visual Designer Toolbox, can serve as an aid to migrating your existing GUI to GUI for .NET. This control is placed on a form and is designed to fill the area in which it is placed. The Window Container is similar to the ABL FRAME.

- **ABL User Control** — The User Control is the base class for building ABL custom user controls.

- **OpenEdge ProBindingSource** — The ProBindingSource is an OpenEdge .NET, non-visual control used to bind ABL data to one or more .NET UI controls that support data binding.

- **ABL Inherited** — The ABL Inherited control is used to create a customized control from a given class.

**Note:** If you do not find ABL User Control and ABL Inherited Control on the design canvas, add it from the Toolbox from File -> New options.

OpenEdge Ultra Controls for .NET

The OpenEdge Ultra Controls for .NET (Ultra Controls) include additional controls that can be used for developing OpenEdge GUI for .NET applications. These Ultra Controls are a set of supported Infragistics NetAdvantage Windows Forms controls that provide built-in features different from those provided by the Microsoft controls. These built-in features allow developers to create forms with sophisticated user interfaces primarily by setting a rich set of properties.

OpenEdge supplies samples that provide examples of using the Ultra Controls. They are found on the self-extracting Documentation and Samples file available on the OpenEdge download page of the Progress Software Download Center under the following relative directory paths:

```
src/prodoc/dotnetobjects
src/samples/GuiForDotNet/UltraControls
```
Infragistics provides detailed documentation and help on these controls. While this documentation is written assuming use of a .NET language, the description of a control and its properties and events are all relevant for its use in ABL. You can access this documentation in the following ways:

- Visit the vendor’s web site. For more information on OpenEdge Ultra Controls, see the API Reference Guide under Windows Forms at the following location:
  
  http://help.infragistics.com/NetAdvantage/WinForms

- Open the control in the OpenEdge Class Browser to see control definitions using a modified ABL syntax.

- Press F1 after you place the control on the form within Progress Developer Studio for OpenEdge Visual Designer. The online help on the respective vendor site opens.

**Third-party controls**

There are many .NET Windows Forms control vendors providing full control sets or specialized controls, such as graphing and reporting. GUI for .NET supports third-party, Windows Forms controls. Prior to choosing a third-party control for your application, it is recommended that you search the OpenEdge Knowledge Base to see if there are known issues with the control. This information can be found at:

http://www.progress.com/openedge/support/index.ssp

**ABL user controls**

Similar to an ABL FRAME, an ABL user control is a container for .NET controls that inherits from Progress.Windows.UserControl, which in turn derives from System.Windows.Forms.UserControl. You can select and place existing .NET controls onto your user control. User controls can be added to a form just like any other .NET control.

User controls combine a group of controls. Examples of user controls are label and editbox combined, address block, table grid, and standard forms containing individual fields.

User controls are best designed with reuse in mind, to implement a common behavior, or for standard UI look-and-feel.

**ABL-derived controls**

An ABL-derived control extends an existing .NET control by customization. ABL-derived controls can be created in the Visual Designer and added to the Visual Designer Toolbox through PROPATH. ABL-derived controls can be added to a form just like any other .NET control.
An example of an ABL-derived control can be something as simple as an OK button. The OK button can be created as a class that inherits from the `System.Windows.Forms.Button` class. You can then set properties to change the button's shape, color, font, and so on. The button subscribes to the `Click` event which is then assigned to an event handler method. When dropped onto a .NET form, you can set the form's `AcceptButton` property to the OK button object reference.

If you are familiar with the object-oriented extensions to ABL, working with .NET objects looks very familiar since the syntax is the same. All .NET objects are referenced by specifying the namespace followed by the .NET object name.
Working with .NET forms and controls

In the code example that follows, the first line defines an object reference for a form as the `Progress.Windows.Form` class. The `Form` class is contained in the `Progress.Windows` namespace, as shown:

```abl
DEFINE VARIABLE FormObjectRef AS Progress.Windows.Form NO-UNDO.
ASSIGN
  FormObjectRef = NEW Progress.Windows.Form( )
  helloBtn = NEW System.Windows.Forms.Button( )
  helloBtn:Text = "Hello World".
FormObjectRef:Controls:Add(helloBtn).
MESSAGE helloBtn:Text VIEW-AS ALERT-BOX.
```

In this example two .NET objects are created, one for a form (`FormObjectRef`) and one for a .NET button (`helloBtn`). The `NEW` statement is used to create an instance of the .NET form and button objects and assign them to variables. The “Hello World” string is assigned to the button’s `Text` property. Finally, the `helloBtn` button object is added to the form’s control collection. Even though `helloBtn` is a .NET object, it can be used in ABL, for example displaying its property in a `MESSAGE` statement. The fact that `helloBtn` is a .NET object is transparent when working in ABL.
5

Events

A GUI for .NET, event-driven application blocks for input and processes events as they are received. ABL supports several event models, including the use of triggers and call-backs for handle-based objects, and the use of class events for class-based objects.

The main communication between .NET and ABL is the invocation of ABL event handlers. These event handlers are associated with specific .NET events. .NET forms and controls fire events when an action occurs. ABL allows you to subscribe and unsubscribe event handler methods or procedures to .NET control events.

This chapter includes the following sections:

- .NET event loop
- Event handlers
Chapter 5: Events

.NET event loop

An application must use the .NET event loop when there is a chance it might use GUI for .NET. Incorporating the .NET event loop into your ABL application is Step 4 in the following list of steps that are required to create and manage a .NET form:

1. Instantiate the form and its controls using the NEW function for classes
2. Initialize the form, together with its controls
3. Subscribe handlers to appropriate .NET events
4. Execute a .NET WAIT-FOR statement, which calls an appropriate .NET input-blocking method for the form
5. After the form closes and the WAIT-FOR statement returns, do any post-form-closing tasks, such as resource clean-up, that your application might require

For more information about blocking on .NET forms, see OpenEdge Development: GUI for .NET Programming.
Event handlers

.NET supports an event model for classes. Like ABL class-based objects, .NET objects define events as members. An ABL application can respond to any .NET event using an ABL event handler. These event handlers are associated with specific .NET events. .NET forms and controls fire events when an action occurs. As a result all subscribed handlers get called.

ABL allows you to subscribe and unsubscribe event handler methods or internal procedures to .NET events. You can use the `Subscribe()` method on a .NET event of a particular .NET class instance to subscribe event handlers written in ABL. .NET supports multiple subscribers attached to one event. Normally two parameters are passed to the event handler—the object firing the event and an object that provides event-specific arguments as properties.

In this example, the `FormObjectRef` object subscribes to the owner of the `Load` event, passing it the name of the event handler, the `FormObjectRefLoad()` method:

```
FormObjectRef:Load:Subscribe(FormObjectRefLoad).
```

```
METHOD PRIVATE VOID FormObjectRefLoad
    ...
END METHOD.
```

You can use the `Unsubscribe()` method to detach an event handler from a control.

A .NET event handler can be either a method in a class (.cls) or an internal procedure. The main difference between using methods and internal procedures as .NET event handlers is that ABL verifies method event handler signatures at compile time (using strong typing), and verifies internal procedure event handler signatures at run time.
Data Binding

From a UI standpoint, data binding is an integration between UI components (controls) and data sources. As such, data binding establishes the connection between the ABL data source, which can be a ProDataSet, a query on a database table or temp-table, or a buffer, and the .NET UI component. We all have worked with data binding in ABL, but may not realize it since in many cases it is implied. Data binding with .NET is a bit more deliberate and more explicit. OpenEdge provides a built-in data binding object, the ProBindingSource, a class used as a conduit between the ABL data source and the .NET controls, and used by the .NET controls to access ABL data.

This chapter includes the following sections:

- Binding to a .NET control
- Binding to a data source
- Defining ProBindingSource tables and fields
Chapter 6: Data Binding

Binding to a .NET control

From a UI standpoint, data binding is an integration between UI components (controls) and data sources. As such, data binding establishes the connection between the ABL data source, which can be a ProDataSet, a query on a database table or temp-table, or a buffer, and the .NET UI component. Examples of tight data binding in ABL include:

- Queries and browses
- Formats applied to UI widgets
- Code to display data
- Code to update data
- Statements that handle data transfer

For bound controls, data is supplied to the control by the binding source. The binding source interface provides methods and properties that a bound control can access and it also provides events to notify a bound control. Think of a binding source as a view to your data.

A critical component of a GUI for .NET application is the ability to access and update ABL data from the UI. .NET defines an API that all bound controls use to obtain their data. The OpenEdge data binding Progress.Data.BindingSource class (referred to as the ProBindingSource) is an OpenEdge .NET class used by .NET controls to access ABL data. This component, which is available in the Visual Designer Toolbox, is the best mechanism for binding your ABL data to your user interface.

The ProBindingSource is a non-visual component that is used at design time and run time. Once placed on a form, it is made available to all .NET controls as a data source. A binding source can be shared by multiple controls.

Figure 3 is a graphical representation of where the ProBindingSource fits between the .NET control and the data source, in this case a ProDataSet. UI controls are bound to the ProBindingSource which acts as a data source (#1). The ProBindingSource exposes an API to .NET controls to access the data.

At run time the ProBindingSource is then bound to a buffer, a query, or a ProDataSet (#2). The ProBindingSource spans the bridge (#3) between the OpenEdge AVM and the .NET CLR to pass data to the UI control and back again.

![Figure 3: .NET-ABL Data Binding Model](image-url)
The ProBindingSource provides ABL data-centric behavior to .NET controls, including the common behaviors of Creating, Reading, Updating, and Deleting data records (CRUD). Key benefits of using the ProDataSet as a data source for a ProBindingSource include:

- Automatic data synchronization
- Updating
- Batching
- Currency

A currency manager is built-in, controlling the position of the record cursor. This is in comparison to .NET, in which data source and currency management are handled separately.

**Note:** Data binding is not limited to visual objects. For example, the Ultra Controls scheduling controls can be data bound using an intermediary control, which then provides data to the visual objects.
**Binding to a data source**

The ProBindingSource supports three types of ABL data source objects:

- **Buffer**
- **Query**
- **ProDataSet**

**Buffer**

An ABL buffer acting as a data source supplies one record at a time to its bound, single-value control, such as a text box or a toggle box. The ProBindingSource Position property is always 0 when using a buffer in this way.

**Query**

An ABL query acting as a data source can be either static or dynamic. The query can also represent a join between multiple tables. A query that is part of a ProDataSet is also a valid data source. Key benefits of using an ABL query as a data source are that it provides automatic updating, batching and currency control.

When binding a control to a query, PSC recommends use of the PRESELECT option on the query, which is an optimized means of providing the binding source with the query record count. For dynamic queries, the AVM enforces use of the PRESELECT option.

PSC also recommends that a query used as a data source be defined using the SCROLLING option. This option specifies that you can jump to a location within the list of records that satisfy the query by using the REPOSITION statement.

**ProDataSet**

A ProDataSet, acting as a data source to the ProBindingSource, can provide a hierarchy of tables to the bound .NET control. The ProDataSet’s parent buffer controls what tables are available through the ProBindingSource. If the ProDataSet contains more than one top-level table and the parent buffer table is not specified, the first top-level buffer acts as the parent table.

You should only bind to a ProDataSet directly (as opposed to one of its queries), if you want to show more than one level of hierarchy in a single control, such as an UltraGrid or TreeView.
Defining ProBindingSource tables and fields

Bound .NET controls have no direct understanding of the bound ABL data source objects. The .NET controls cannot access the schema of a bound ABL data source object to find the default format, the validation phrase, or the help text for a field. The .NET controls can only access what the binding interfaces in the Progress.Data.BindingSource class make available.

The task of binding a data source to the ProBindingSource control includes the identification of data source tables and fields that will be exposed to the bound .NET control. Since the data source is provided at run time, the binding source may provide a design-time schema that matches the schema used at run time. So, how is the schema supplied to the binding source? The ProBindingSource Designer tool, available in the Visual Designer’s Design view, is used to define what data source tables and fields are available to be bound to a control. Schema information can either be entered manually or imported from an XML Schema file. Similar to an OpenEdge Data Definition file (.df), in this context an XML Schema file defines table name, field name, field data type, field display format, field label, and so on as XML data.

You can create an XML Schema file containing the definition of a ProDataSet, a temp-table, or a temp-table buffer object using the ABL WRITE-XMLSCHEMA( ) method. This method uses the industry-standard XML Schema Definition (XSD) language. The XML Schema file that is produced may contain more fields than you need. After importing an XML Schema file, you can eliminate unwanted fields in the ProBindingSource control by removing them with the ProBindingSource Designer. Alternately, you can allow the ProBindingSource to contain all the data source fields, so it can be reused and bound to a number of controls.
Implementing an OpenEdge GUI for .NET does not mean that you have to rewrite your application code. You should review your application architecture and decide which, if any, components could be modified to take full advantage of the Ultra Controls and the underlying object-oriented extensions to ABL. Materials on PSDN and presentations given at Exchange user conferences and other mediums over the past several years have described an approach to building applications called the OpenEdge Reference Architecture (OERA). Most of those existing materials focus on the data management side of an application, describing data management components such as Business Entities that encapsulate access to a particular set of related application data.

This chapter includes the following sections:

- Business Entities
- Service Adapters and Service Interfaces
Business Entities

A Business Entity is a reusable, modular component that encapsulates all access to a particular set of related application data identified by its use within your application, for example, a Customer or an Order. A Business Entity becomes a type of service, one that can be called from any part of your application needing to retrieve or update data.

One advantage of encapsulating access to your data along with all the business logic that applies to it is that you can substitute a new user interface without having to change your business logic in any way. For this reason, when you consider how to integrate a new generation of user interface based on the GUI for .NET with your existing data and business logic, you should consider how best to keep the data and logic that you already have, while separating that logic from the specifics of the UI. While there is no requirement to re-architect your application to use the GUI for .NET, you should take advantage of the opportunity where you can to make a clean separation of the "client" side of your application (where the user interface is defined), from the "server" side of your application (where the data and business logic are encoded). This is true whether your application is actually divided between OpenEdge client sessions and AppServers, or whether it uses OpenEdge client-server support within a single session.

Many of the coding examples you will be seeing in documentation materials for OpenEdge 10.2 show GUI for .NET forms and their supporting code interacting with Business Entities and other data management components. This is referred to in OERA terminology as the Presentation Layer. For examples of GUI for .NET forms and their supporting code, see *OpenEdge Getting Started: Introducing the Progress Developer Studio for OpenEdge Visual Designer*. 
Service Adapters and Service Interfaces

You will also see examples that refer to the concept of a **Service Adapter** and a **Service Interface**. The Service Adapter is simply a layer of indirection between the Presentation on the client side, and Business Entities and other data management components on the server side. A Service Adapter takes responsibility for knowing how and where to make a request for data, so that this connection is not hard-wired into a particular user interface, which is only concerned with the definition of the data as it is displayed and updated, and not where the data comes from.

Likewise, a Service Interface defines an entry point for each accessible operation in each Business Entity or other business component. A Service Adapter on the client side can make a call to a Service Interface on the server side (again, whether the application is physically separated into client and server sessions or not). The Service Interface takes responsibility for identifying and calling the business component, so that any details of how this is done are kept independent of the call itself. These different application layers maximize the reusability, flexibility, and ease of maintenance of all the different parts of your application logic.

**Figure 4** shows where the Service Interface and the Service Adapter fit in this layered architecture.

![Service Adapter and Service Interface Diagram](image)
Third Party Acknowledgements

One or more products in the Progress OpenEdge v11.3 release includes third party components covered by licenses that require that the following documentation notices be provided:

Progress OpenEdge v11.3 may incorporate ANT v1.5.4. Such technology is subject to the following terms and conditions: The Apache Software License, Version 1.1, applies to all versions of up to ant 1.6.0 included. The Apache Software License, Version 1.1 - Copyright (C) 2000-2003 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and where any such third-party acknowledgments normally appear. 4. The names "Ant" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
Appendix A: Third Party Acknowledgements

ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions
made by many individuals on behalf of the Apache Software Foundation. For more
information on the Apache Software Foundation, please see

Progress OpenEdge v11.3 may incorporate Apache Avalon 2002/08/06, Apache Batik
v1.1, Apache Jakarta POI v1.1, Apache POI 2003/05/14, Regexp (Pure Java Regular
Expression) v1.2, and XMLTask v1.15.1 from The Apache Software Foundation. Such
technologies are subject to the following terms and conditions: The Apache Software
License, Version 1.1 Copyright (c) 2000 The Apache Software Foundation. All rights
reserved. Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met: 1.
Redistributions of source code must retain the above copyright notice, this list of
conditions and the following disclaimer. 2. Redistributions in binary form must
reproduce the above copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution. 3. The
end-user documentation included with the redistribution, if any, must include the
following acknowledgment: "This product includes software developed by the Apache
Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment
may appear in the software itself, if and wherever such third-party acknowledgments
normally appear. 4. The names "Apache" and "Apache Software Foundation" must not
be used to endorse or promote products derived from this software without prior written
permission. For written permission, please contact apache@apache.org. 5. Products
derived from this software may not be called "Apache", nor may "Apache" appear in
their name, without prior written permission of the Apache Software Foundation. THIS
SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions
made by many individuals on behalf of the Apache Software Foundation. For more
information on the Apache Software Foundation, please see
<http://www.apache.org/>. Portions of this software are based upon public domain
software originally written at the National Center for Supercomputing Applications,
University of Illinois, Urbana-Champaign.

Progress OpenEdge v11.3 may incorporate Apache JAXP v1.3.04 from The Apache
Software Foundation. Such technology is subject to the following terms and conditions:
The Apache Software License, Version 1.1 Copyright (c) 1999-2003 The Apache
Software Foundation. All rights reserved. Redistribution and use in source and binary
forms, with or without modification, are permitted provided that the following conditions
are met: 1. Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer. 2. Redistributions in binary form must
reproduce the above copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution. 3. The
end-user documentation included with the redistribution, if any, must include the
following acknowledgment: "This product includes software developed by the Apache
Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment
may appear in the software itself, if and wherever such third-party acknowledgments
normally appear. 4. The names "Xalan" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ´´AS IS´´ AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, Lotus Development Corporation, http://www.lotus.com. For more information on the Apache Software Foundation, please see http://www.apache.org/.

Progress OpenEdge v11.3 may incorporate Crimson v1.1.3 from Progress Extensions for Eclipse v2.2.1. Such technology is subject to the following terms and conditions:
The Apache Software License, Version 1.1 Copyright (c) 1999-2003 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ´´AS IS´´ AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, International Business Machines, Inc., http://www.ibm.com. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.
Appendix A: Third Party Acknowledgements

Progress OpenEdge v11.3 may incorporate jfor v0.7.2 from jfor. Such technology is subject to the following terms and conditions: jfor Apache-Style Software License. Copyright (c) 2002 by the jfor project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the jfor project (http://www.jfor.org)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The name "jfor" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact info@jfor.org. 5. Products derived from this software may not be called "jfor", nor may "jfor" appear in their name, without prior written permission of info@jfor.org. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE JFOR PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate Apache LogKit v1.2 from The Apache Software Foundation. Such technologies are subject to the following terms and conditions: The Apache Software License, Version 1.1 -- Copyright (C) 1997-2003 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "LogKit", "Jakarta" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress OpenEdge v11.3 may incorporate Xerces for Java XML Parser v2.6.2. Such technology is subject to the following terms and conditions: The Apache Software License, Version 1.1 Copyright (c) 1999 The Apache Software Foundation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org. 5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation. THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, International Business Machines, Inc., http://www.ibm.com. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress OpenEdge v11.3 may incorporate SOAP v2.3.1 from Apache Foundation. Such technology is subject to the following terms and conditions: The Apache Software License, Version 1.1

Copyright (c) 1999 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
Appendix A: Third Party Acknowledgements

3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)."

Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "SOAP" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation. For more information on the Apache Software Foundation, please see <http://www.apache.org/>.

Progress OpenEdge v11.3 may incorporate ANTLR (Another Tool for Language Recognition) v2.7.6. Such technology is subject to the following terms and conditions: ANTLR 3 License [The BSD License] Copyright (c) 2003-2006, Terence Parr All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate Castor v0.9.3 from castor.org. Such technology is subject to the following terms and conditions: Original Intalio license -
Copyright 1999-2004 (C) Intalio Inc., and others. All Rights Reserved. Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The name "ExoLab" must not be used to endorse or promote products derived from this Software without prior written permission of Intalio Inc. For written permission, please contact info@exolab.org. 4. Products derived from this Software may not be called "Castor" nor may "Castor" appear in their names without prior written permission of Intalio Inc. Exolab, Castor and Intalio are trademarks of Intalio Inc. 5. Due credit should be given to the ExoLab Project (http://www.exolab.org/). THIS SOFTWARE IS PROVIDED BY INTALIO AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL INTALIO OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate Decimal Conversion Code (dtoa.c; g_fmt.c; md_prod.s; desctrof.c; desctrstf.c; desctc.c; gfmt.c; gfmtimp.c; gethex.c; g_init.c; ginit.c; smisc.c; sum.c; ulp.c). Such technologies are subject to the following terms and conditions: dtoa.c License: The author of this software is David M. Gay. Copyright (c) 1991, 2000, 2001 by Lucent Technologies. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. g_fmt.c License: The author of this software is David M. Gay. Copyright (c) 1991, 1996 by Lucent Technologies. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE.

md_prod.s License: The author of this software is David M. Gay. Copyright (c) 1991 by Lucent Technologies. Permission to use, copy, modify, and distribute this software for any purpose without fee is hereby granted, provided that this entire notice is included in all copies of any software which is or includes a copy or modification of this software and in all copies of the supporting documentation for such software. THIS SOFTWARE IS BEING PROVIDED "AS IS", WITHOUT ANY EXPRESS OR IMPLIED WARRANTY. IN PARTICULAR, NEITHER THE AUTHOR NOR LUCENT MAKES ANY
Appendix A: Third Party Acknowledgements

REPRESENTATION OR WARRANTY OF ANY KIND CONCERNING THE MERCHANTABILITY OF THIS SOFTWARE OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. decstrtof.c License: The author of this software is David M. Gay. Copyright (C) 1998-2001 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

dcstrtof.c License: The author of this software is David M. Gay. Copyright (C) 1998, 2000 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

dmisc.c License: The author of this software is David M. Gay. Copyright (C) 1998 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

gdtoa.c License: The author of this software is David M. Gay. Copyright (C) 1998, 1999 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY
SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. gdtoa.h License: The author of this software is David M. Gay. Copyright (C) 1998 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. gdtoaimp.h License: The author of this software is David M. Gay. Copyright (C) 1998-2000 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. gethex.c License: The author of this software is David M. Gay. Copyright (C) 1998 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. gmisc.c License: The author of this software is David M. Gay. Copyright (C) 1998 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT
SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL,
INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER
RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION
OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
hd_init.c License: The author of this software is David M. Gay. Copyright (C) 2000 by
Lucent Technologies All Rights Reserved Permission to use, copy, modify, and
 distributes this software and its documentation for any purpose and without fee is hereby
granted, provided that the above copyright notice appear in all copies and that both that
the copyright notice and this permission notice and warranty disclaimer appear in
supporting documentation, and that the name of Lucent or any of its entities not be
used in advertising or publicity pertaining to distribution of the software without specific,
written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD
TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS
ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL
 DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE,
DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE
USE OR PERFORMANCE OF THIS SOFTWARE. misc.c License: The author of this
software is David M. Gay. Copyright (C) 1998, 1999 by Lucent Technologies All Rights
Reserved Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted, provided that the
above copyright notice appear in all copies and that both that the copyright notice and
this permission notice and warranty disclaimer appear in supporting documentation,
and that the name of Lucent or any of its entities not be used in advertising or publicity
pertaining to distribution of the software without specific, written prior permission.
LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS.
IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY
SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS
ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE. smisc.c License: The author of this software
is David M. Gay. Copyright (C) 1998, 1999 by Lucent Technologies All Rights
Reserved Permission to use, copy, modify, and distribute this software and its
documentation for any purpose and without fee is hereby granted, provided that the
above copyright notice appear in all copies and that both that the copyright notice and
this permission notice and warranty disclaimer appear in supporting documentation,
and that the name of Lucent or any of its entities not be used in advertising or publicity
pertaining to distribution of the software without specific, written prior permission.
LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE,
INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS.
IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY
SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS
ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE. strtodg.c License: The author of this software
is David M. Gay. Copyright (C) 1998-2001 by Lucent Technologies All Rights Reserved
Permission to use, copy, modify, and distribute this software and its documentation for
any purpose and without fee is hereby granted, provided that the above copyright
notice appear in all copies and that both that the copyright notice and this permission
notice and warranty disclaimer appear in supporting documentation, and that the name
of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. strord.c License: The author of this software is David M. Gay. Copyright (C) 1998, 2000 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. sum.c License: The author of this software is David M. Gay. Copyright (C) 1998 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. ulp.c License: The author of this software is David M. Gay. Copyright (C) 1998, 1999 by Lucent Technologies All Rights Reserved Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the name of Lucent or any of its entities not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. LUCENT DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL LUCENT OR ANY OF ITS ENTITIES BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
Appendix A: Third Party Acknowledgements

Progress OpenEdge v11.3 may incorporate DOM4J v1.6.1 from MetaStuff, Ltd. Such technology is subject to the following terms and conditions: Copyright 2001-2005 (C) MetaStuff, Ltd. All Rights Reserved. Redistribution and use of this software and associated documentation ("Software"), with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain copyright statements and notices. Redistributions must also contain a copy of this document. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The name "DOM4J" must not be used to endorse or promote products derived from this Software without prior written permission of MetaStuff, Ltd. For written permission, please contact dom4j-info@metastuff.com. 4. Products derived from this Software may not be called "DOM4J" nor may "DOM4J" appear in their names without prior written permission of MetaStuff, Ltd. DOM4J is a registered trademark of MetaStuff, Ltd. 5. Due credit should be given to the DOM4J Project - http://www.dom4j.org THIS SOFTWARE IS PROVIDED BY METASTUFF, LTD. AND CONTRIBUTORS "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL METASTUFF, LTD. OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate GraphicsMagick v1.3.14 from GraphicsMagick Group. Such technology is subject to the following terms and conditions:

.. This text is in reStucturedText format, so it may look a bit odd. .. See http://docutils.sourceforge.net/rst.html for details.

====================================================================================

GraphicsMagick Copyrights and Licenses

====================================================================================

This file is part of the GraphicsMagick software distributed by the GraphicsMagick Group.

[*Please note that the legal community considers 15 or more total lines of code or text (not necessarily contiguous) to be significant for the purposes of copyright. Repeated changes such as renaming a symbol has similar significance to changing one line of code.*]

The licenses which components of this software fall under are as follows.

1) In November 2002, the GraphicsMagick Group created GraphicsMagick from ImageMagick Studio's ImageMagick and applied the "MIT" style license: Copyright (C) 2002 - 2012 GraphicsMagick Group
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2) In October 1999, ImageMagick Studio assumed the responsibility for the development of ImageMagick (forking from the distribution by E. I. du Pont de Nemours and Company) and applied a new license:

Copyright (C) 2002 ImageMagick Studio, a non-profit organization dedicated to making software imaging solutions freely available.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files ("ImageMagick"), to deal in ImageMagick without restriction, including without limitation

the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of ImageMagick, and to permit persons to whom the ImageMagick is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of ImageMagick.

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall ImageMagick Studio be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with ImageMagick or the use or other dealings in ImageMagick.

Except as contained in this notice, the name of the ImageMagick Studio shall not be used in advertising or otherwise to promote the sale, use or other dealings in ImageMagick without prior written authorization from the ImageMagick Studio.

3) From 1991 to October 1999 (through ImageMagick 4.2.9), ImageMagick was developed and distributed by E. I. du Pont de Nemours and Company: Copyright 1999 E. I. du Pont de Nemours and Company

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files ("ImageMagick"), to deal in ImageMagick without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of ImageMagick, and to permit persons to whom the ImageMagick is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of ImageMagick.
Appendix A: Third Party Acknowledgements

The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement.

In no event shall E. I. du Pont de Nemours and Company be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with ImageMagick or the use or other dealings in ImageMagick.

Except as contained in this notice, the name of the E. I. du Pont de Nemours and Company shall not be used in advertising or otherwise to promote the sale, use or other dealings in ImageMagick without prior written authorization from the E. I. du Pont de Nemours and Company.

4) The GraphicsMagick Base64Decode() and Base64Encode() functions are based on source code obtained from OpenSSH. This source code is distributed under the following license: Copyright (c) 2000 Markus Friedl. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5) Many of the pattern images in coders/logo.c are derived from XFig, which is distributed under the following license:

| FIG : Facility for Interactive Generation of figures |
| Copyright (c) 1985-1988 by Supoj Sutanthavibul |
| Parts Copyright (c) 1989-2000 by Brian V. Smith |
| Parts Copyright (c) 1991 by Paul King |

Any party obtaining a copy of these files is granted, free of charge, a full and unrestricted irrevocable, world-wide, paid up, royalty-free, nonexclusive right and license to deal in this software and documentation files (the "Software"), including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons who receive copies from any such party to do so, with the only requirement being that this copyright notice remain intact.
6) The documentation for the composition operators is copied from the rlecomp manual page, which is authored by Rod Bogart and John W. Peterson. Rlecomp is part of the Utah Raster Toolkit distributed by the University of Michigan and the University of Utah. The copyright for this manual page is as follows: Copyright (c) 1986, University of Utah

This software is copyrighted as noted below. It may be freely copied, modified, and redistributed, provided that the copyright notice is preserved on all copies.

There is no warranty or other guarantee of fitness for this software, it is provided solely "as is". Bug reports or fixes may be sent to the author, who may or may not act on them as he desires.

You may not include this software in a program or other software product without supplying the source, or without informing the end-user that the source is available for no extra charge.

If you modify this software, you should include a notice giving the name of the person performing the modification, the date of modification, and the reason for such modification.

7) The source code comprising swab.c is originally derived from libtiff which has the following license:

| Copyright (c) 1988-1997 Sam Leffler
| Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

8) The C++ API known as "Magick++", and which resides in the Magick++ directory, is distributed under the following license: Copyright 1999 - 2003 Bob Friesenhahn <bfriesen@simple.dallas.tx.us>

Permission is hereby granted, free of charge, to any person obtaining a copy of the source files and associated documentation files ("Magick++"), to deal in Magick++ without restriction, including without limitation of the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of Magick++, and to permit persons to whom the Magick++ is furnished to do so, subject to the following conditions:

This copyright notice shall be included in all copies or substantial portions of Magick++. The copyright to Magick++ is retained by its author and shall not be subsumed or replaced by any other copyright.
The software is provided "as is", without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and noninfringement. In no event shall Bob Friesenhahn be liable for any claim, damages or other liability, whether in an action of contract, tort or otherwise, arising from, out of or in connection with Magick++ or the use or other dealings in Magick++.

9) The GraphicsMagick HaldClutImagePixels() function in magick/hclut.c is based on source code from the HaldCLUT package by Eskil Steenberg (http://www.quelsolaar.com/technology/clut.html) which is distributed under the following license: Copyright (c) 2005 Eskil Steenberg. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

10) GraphicsMagick makes use of third-party "delegate" libraries to support certain optional features. These libraries bear their own copyrights and licenses, which may be more or less restrictive than the GraphicsMagick license. For convenience, when GraphicsMagick is bundled with (or compiled with) "delegate" libraries, a copy of the licenses for these libraries is provided in a "licenses" directory.

=====================================================================

... |copy|   unicode:: U+000A9 .. COPYRIGHT SIGN
Copyright |copy| GraphicsMagick Group 2002 - 2011

OVERVIEW and LEGAL ISSUES from jpeg.txt file (from GraphicsMagick):

The Independent JPEG Group's JPEG software

=====================================================================

README for release 6b of 27-Mar-1998

=====================================================================
This distribution contains the sixth public release of the Independent JPEG Group’s free JPEG software. You are welcome to redistribute this software and to use it for any purpose, subject to the conditions under LEGAL ISSUES, below.

Serious users of this software (particularly those incorporating it into larger programs) should contact IJG at jpeg-info@uunet.uu.net to be added to our electronic mailing list. Mailing list members are notified of updates and have a chance to participate in technical discussions, etc.

This software is the work of Tom Lane, Philip Gladstone, Jim Boucher, Lee Crocker, Julian Minguillon, Luis Ortiz, George Phillips, Davide Rossi, Guido Vollbeding, Ge' Weijers, and other members of the Independent JPEG Group. IJG is not affiliated with the official ISO JPEG standards committee.

OVERVIEW
========

This package contains C software to implement JPEG image compression and decompression. JPEG (pronounced "jay-peg") is a standardized compression method for full-color and gray-scale images. JPEG is intended for compressing "real-world" scenes; line drawings, cartoons and other non-realistic images are not its strong suit. JPEG is lossy, meaning that the output image is not exactly identical to the input image. Hence you must not use JPEG if you have to have identical output bits. However, on typical photographic images, very good compression levels can be obtained with no visible change, and remarkably high compression levels are possible if you can tolerate a low-quality image. For more details, see the references, or just experiment with various compression settings. This software implements JPEG baseline, extended-sequential, and progressive compression processes. Provision is made for supporting all variants of these processes, although some uncommon parameter settings aren't implemented yet.

For legal reasons, we are not distributing code for the arithmetic-coding variants of JPEG; see LEGAL ISSUES. We have made no provision for supporting the hierarchical or lossless processes defined in the standard.

We provide a set of library routines for reading and writing JPEG image files, plus two sample applications "cjpeg" and "djjpeg", which use the library to perform conversion between JPEG and some other popular image file formats. The library is intended to be reused in other applications.

In order to support file conversion and viewing software, we have included considerable functionality beyond the bare JPEG coding/decoding capability; for example, the color quantization modules are not strictly part of JPEG decoding, but they are essential for output to colormapped file formats or colormapped displays. These extra functions can be compiled out of the library if not required for a particular application. We have also included "jpegtran", a utility for lossless transcoding between different JPEG processes, and "rdjpgcom" and "wrjpgcom", two simple applications for inserting and extracting textual comments in JFIF files.

The emphasis in designing this software has been on achieving portability and flexibility, while also making it fast enough to be useful. In particular, the software is not intended to be read as a tutorial on JPEG. (See the REFERENCES section for introductory material.) Rather, it is intended to be reliable, portable, industrial-strength code. We do not claim to have achieved that goal in every aspect of the software, but we strive for it.
We welcome the use of this software as a component of commercial products.

No royalty is required, but we do ask for an acknowledgement in product documentation, as described under LEGAL ISSUES.

LEGAL ISSUES

=============

In plain English:

1. We don't promise that this software works. (But if you find any bugs, please let us know!)

2. You can use this software for whatever you want. You don't have to pay us.

3. You may not pretend that you wrote this software. If you use it in a program, you must acknowledge somewhere in your documentation that you've used the IJG code.

In legalese:

The authors make NO WARRANTY or representation, either express or implied, with respect to this software, its quality, accuracy, merchantability, or fitness for a particular purpose. This software is provided "AS IS", and you, its user, assume the entire risk as to its quality and accuracy.

This software is copyright (C) 1991-1998, Thomas G. Lane.

All Rights Reserved except as specified below.

Permission is hereby granted to use, copy, modify, and distribute this software (or portions thereof) for any purpose, without fee, subject to these conditions:

(1) If any part of the source code for this software is distributed, then this README file must be included, with this copyright and no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be clearly indicated in accompanying documentation.

(2) If only executable code is distributed, then the accompanying documentation must state that "this software is based in part on the work of the Independent JPEG Group".

(3) Permission for use of this software is granted only if the user accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

These conditions apply to any software derived from or based on the IJG code, not just to the unmodified library. If you use our work, you ought to acknowledge us.

Permission is NOT granted for the use of any IJG author's name or company name in advertising or publicity relating to this software or products derived from it. This software may be referred to only as "the Independent JPEG Group's software".

We specifically permit and encourage the use of this software as the basis of commercial products, provided that all warranty or liability claims are assumed by the product vendor.

ansi2knr.c is included in this distribution by permission of L. Peter Deutsch, sole proprietor of its copyright holder, Aladdin Enterprises of Menlo Park, CA.
ansii2knr.c is NOT covered by the above copyright and conditions, but instead by the usual distribution terms of the Free Software Foundation; principally, that you must include source code if you redistribute it. (See the file ansi2knr.c for full details.) However, since ansi2knr.c is not needed as part of any program generated from the IJG code, this does not limit you more than the foregoing paragraphs do.

The Unix configuration script "configure" was produced with GNU Autoconf.

It is copyright by the Free Software Foundation but is freely distributable.

The same holds for its supporting scripts (config.guess, config.sub, ltconfig, ltmain.sh). Another support script, install-sh, is copyright by M.I.T. but is also freely distributable.

It appears that the arithmetic coding option of the JPEG spec is covered by patents owned by IBM, AT&T, and Mitsubishi. Hence arithmetic coding cannot legally be used without obtaining one or more licenses. For this reason, support for arithmetic coding has been removed from the free JPEG software. (Since arithmetic coding provides only a marginal gain over the unpatented Huffman mode, it is unlikely that very many implementations will support it.)

So far as we are aware, there are no patent restrictions on the remaining code.

The IJG distribution formerly included code to read and write GIF files.

To avoid entanglement with the Unisys LZW patent, GIF reading support has been removed altogether, and the GIF writer has been simplified to produce "uncompressed GIFs". This technique does not use the LZW algorithm; the resulting GIF files are larger than usual, but are readable by all standard GIF decoders.

We are required to state that "The Graphics Interchange Format(c) is the Copyright property of CompuServe Incorporated. GIF(sm) is a Service Mark property of CompuServe Incorporated."

Contents of png.txt file (from GraphicsMagick):

This copy of the libpng notices is provided for your convenience. In case of any discrepancy between this copy and the notices in the file png.h that is included in the libpng distribution, the latter shall prevail.

COPYRIGHT NOTICE, DISCLAIMER, and LICENSE:

If you modify libpng you may insert additional notices immediately following this sentence.

libpng versions 1.0.7, July 1, 2000, through 1.2.0, September 1, 2001, are Copyright (c) 2000 Glenn Randers-Pehrson and are distributed according to the same disclaimer and license as libpng-1.0.6 with the following individuals added to the list of Contributing Authors

Simon-Pierre Cadieux

Eric S. Raymond

Gilles Vollant

and with the following additions to the disclaimer:

There is no warranty against interference with your enjoyment of the library or against infringement. There is no warranty that our efforts or the library will fulfill any of your
particular purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality, performance, accuracy, and effort is with the user.

Libpng versions 0.97, January 1998, through 1.0.6, March 20, 2000, are Copyright (c) 1998, 1999 Glenn Randers-Pehrson, and are distributed according to the same disclaimer and license as libpng-0.96, with the following individuals added to the list of Contributing Authors:

Tom Lane
Glenn Randers-Pehrson
Willem van Schaik

Libpng versions 0.89, June 1996, through 0.96, May 1997, are Copyright (c) 1996, 1997 Andreas Dilger

Distributed according to the same disclaimer and license as libpng-0.88, with the following individuals added to the list of Contributing Authors:

John Bowler
Kevin Bracey
Sam Bushell
Magnus Holmgren
Greg Roelofs
Tom Tanner

Libpng versions 0.5, May 1995, through 0.88, January 1996, are Copyright (c) 1995, 1996 Guy Eric Schalnat, Group 42, Inc. For the purposes of this copyright and license, "Contributing Authors" is defined as the following set of individuals:

Andreas Dilger
Dave Martindale
Guy Eric Schalnat
Paul Schmidt
Tim Wegner

The PNG Reference Library is supplied "AS IS". The Contributing Authors and Group 42, Inc. disclaim all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The Contributing Authors and Group 42, Inc. assume no liability for direct, indirect, incidental, special, exemplary, or consequential damages, which may result from the use of the PNG Reference Library, even if advised of the possibility of such damage. Permission is hereby granted to use, copy, modify, and distribute this source code, or portions hereof, for any purpose, without fee, subject to the following restrictions:

1. The origin of this source code must not be misrepresented.

2. Altered versions must be plainly marked as such and must not be misrepresented as being the original source.
3. This Copyright notice may not be removed or altered from any source or altered source distribution.

The Contributing Authors and Group 42, Inc. specifically permit, without fee, and encourage the use of this source code as a component to supporting the PNG file format in commercial products. If you use this source code in a product, acknowledgment is not required but would be appreciated.

A "png_get_copyright" function is available, for convenient use in "about" boxes and the like:

```c
printf("%s", png_get_copyright(NULL));
```

Also, the PNG logo (in PNG format, of course) is supplied in the files "pngbar.png" and "pngbar.jpg (88x31) and "pngnow.png" (98x31).

Libpng is OSI Certified Open Source Software. OSI Certified Open Source is a certification mark of the Open Source Initiative.

Glenn Randers-Pehrson
randeg@alum.rpi.edu

September 1, 2001

Contents of tiff.txt file (from GraphicsMagick):

Copyright (c) 1988-1997 Sam Leffler

Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics. THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Contents of zlib.txt file (from GraphicsMagick):

zlib 1.1.3 is a general purpose data compression library. All the code is thread safe. The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files ftp://ds.internic.net/rfc/rfc1950.txt (zlib format), rfc1951.txt (deflate format) and rfc1952.txt (gzip format). These documents are also available in other formats from ftp://ftp.uu.net/graphics/png/documents/zlib/zdoc-index.html

All functions of the compression library are documented in the file zlib.h (volunteer to write man pages welcome, contact jloup@gzip.org). A usage example of the library is
given in the file example.c which also tests that the library is working correctly. Another example is given in the file minigzip.c. The compression library itself is composed of all source files except example.c and minigzip.c.

To compile all files and run the test program, follow the instructions given at the top of Makefile. In short "make test; make install" should work for most machines. For Unix: "configure; make test; make install"

For MSDOS, use one of the special makefiles such as Makefile.msc.

For VMS, use Make_vms.com or descrip.mms.

Questions about zlib should be sent to <zlib@quest.jpl.nasa.gov>, or to Gilles Vollant <info@winimage.com> for the Windows DLL version.

The zlib home page is http://www.cdrom.com/pub/infozip/zlib/

The official zlib ftp site is ftp://ftp.cdrom.com/pub/infozip/zlib/

Before reporting a problem, please check those sites to verify that you have the latest version of zlib; otherwise get the latest version and check whether the problem still exists or not.

Mark Nelson <markn@tiny.com> wrote an article about zlib for the Jan. 1997 issue of Dr. Dobb's Journal; a copy of the article is available in http://web2.airmail.net/markn/articles/zlibtool/zlibtool.htm

The changes made in version 1.1.3 are documented in the file ChangeLog.

The main changes since 1.1.2 are:

- fix "an inflate input buffer bug that shows up on rare but persistent occasions" (Mark)
- fix gzread and gztell for concatenated .gz files (Didier Le Botlan)
- fix gzseek(..., SEEK_SET) in write mode
- fix crc check after a gzeek (Frank Faubert)
- fix miniunizip when the last entry in a zip file is itself a zip file
  (J Lilge)
- add contrib/asm586 and contrib/asm686 (Brian Raiter)
  See http://www.muppetlabs.com/~breadbox/software/assembly.html
- add support for Delphi 3 in contrib/delphi (Bob Dellaca)
- add support for C++Builder 3 and Delphi 3 in contrib/delphi2 (Davide Moretti)
- do not exit prematurely in untgz if 0 at start of block (Magnus Holmgren)
- use macro EXTERN instead of extern to support DLL for BeOS (Sander Stoks)
- added a FAQ file

plus many changes for portability.
Unsupported third party contributions are provided in directory "contrib". A Java implementation of zlib is available in the Java Development Kit 1.1
http://www.javasoft.com/products/JDK/1.1/docs/api/Package-java.util.zip.html

See the zlib home page http://www.cdrom.com/pub/infozip/zlib/ for details.

A Perl interface to zlib written by Paul Marquess <pmarquess@bfsec.bt.co.uk> is in the CPAN (Comprehensive Perl Archive Network) sites, such as:

A Python interface to zlib written by A.M. Kuchling <amk@magnet.com> is available in Python 1.5 and later versions, see
http://www.python.org/doc/lib/module-zlib.html

A zlib binding for TCL written by Andreas Kupries <a.kupries@westend.com> is available at http://www.westend.com/~kupries/doc/trf/man/man.html

An experimental package to read and write files in .zip format, written on top of zlib by Gilles Vollant <info@winimage.com>, is available at http://www.winimage.com/zLibDll/unzip.html and also in the contrib/minizip directory of zlib.

Notes for some targets:

- To build a Windows DLL version, include in a DLL project zlib.def, zlib.rc and all .c files except example.c and minigzip.c; compile with -DZLIB_DLL

  The zlib DLL support was initially done by Alessandro Iacopetti and is now maintained by Gilles Vollant <info@winimage.com>. Check the zlib DLL home page at http://www.winimage.com/zLibDll

  From Visual Basic, you can call the DLL functions which do not take a structure as argument: compress, uncompress and all gz* functions.

  See contrib/visual-basic.txt for more information, or get
  http://www.tcfb.com/dowseware/cmp-z-it.zip

- For 64-bit Irix, deflate.c must be compiled without any optimization. With -O, one libpng test fails. The test works in 32 bit mode (with the -n32 compiler flag). The compiler bug has been reported to SGI.

- zlib doesn’t work with gcc 2.6.3 on a DEC 3000/300LX under OSF/1 2.1 it works when compiled with cc.

- on Digital Unix 4.0D (formely OSF/1) on AlphaServer, the cc option -std1 is necessary to get gzprintf working correctly. This is done by configure.

- zlib doesn’t work on HP-UX 9.05 with some versions of /bin/cc. It works with other compilers. Use "make test" to check your compiler.

- gzdopen is not supported on RISCOS, BEOS and by some Mac compilers.

- For Turbo C the small model is supported only with reduced performance to avoid any far allocation; it was tested with -DMAX_WBITS=11 -DMAX_MEM_LEVEL=3

- For PalmOs, see http://www.cs.uit.no/~perm/PASTA/pilot/software.html
Per Harald Myrvang <perm@stud.cs.uit.no>

Acknowledgments:
Appendix A: Third Party Acknowledgements

The deflate format used by zlib was defined by Phil Katz. The deflate and zlib specifications were written by L. Peter Deutsch. Thanks to all the people who reported problems and suggested various improvements in zlib; they are too numerous to cite here.

Copyright notice:

(C) 1995-1998 Jean-loup Gailly and Mark Adler

This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu

If you use the zlib library in a product, we would appreciate *not* receiving lengthy legal documents to sign. The sources are provided for free but without warranty of any kind. The library has been entirely written by Jean-loup Gailly and Mark Adler; it does not include third-party code.

If you redistribute modified sources, we would appreciate that you include in the file ChangeLog history information documenting your changes.

Progress OpenEdge v11.3 may incorporate International Classes for Unicode (International Components for Unicode) v2.4 from IBM. Such technology is subject to the following terms and conditions: ICU License - The ICU project is licensed under the X License (see also the x.org original), which is compatible with GPL but non-copyleft. The license allows ICU to be incorporated into a wide variety of software projects using the GPL license. The X license is compatible with the GPL, while also allowing ICU to be incorporated into non-open source products. License ICU License - ICU 1.8.1 and later COPYRIGHT AND PERMISSION NOTICE Copyright (c) 1995-2003 International Business Machines Corporation and others All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR
ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

-------------------------------------------------------------------------

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

Progress OpenEdge v11.3 may incorporate International Components for Unicode v4.8.0. Such technology is subject to the following terms and conditions: ICU License - ICU 1.8.1 and later COPYRIGHT AND PERMISSION NOTICE Copyright (c) 1995-2011 International Business Machines Corporation and others All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

Progress OpenEdge v11.3 may incorporate java.net args4j v2.0.12 from Kohsuke Kawaguchi. Such technology is subject to the following terms and conditions: Copyright (c) 2003, Kohsuke Kawaguchi All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate Jaxen v1.1-beta-3 from jaxen.org. Such technology is subject to the following terms and conditions: Project License $Id: LICENSE.txt 1128 2006-02-05 21:49:04Z elharo $ Copyright 2003-2006 The Werken Company. All Rights Reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the Jaxen Project nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate Jing 20030619 portion of Progress Extensions for Eclipse v2.2.1. Such technology is subject to the following terms and conditions: Jing Copying Conditions. Copyright (c) 2001-2003 Thai Open Source Software Center Ltd. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the Thai Open Source Software Center Ltd nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE. Third-party JARs - This distribution includes some additional JAR files, which have their own copying conditions:

saxon.jar Comes from the Saxon 6.5.2 distribution and is covered by these conditions
xercesImpl.jar xml-apis.jar Come from the Xerces-J 2.4.0 distribution and are covered by the Apache Software License isorelax.jar Comes from ISO RELAX 2003/01/08 distribution and is covered by the following license: Copyright (c) 2001-2002, SourceForge ISO-RELAX Project (ASAMI Tomoharu, Daisuke Okajima, Kohsuke Kawaguchi, and MURATA Makoto) Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Progress OpenEdge v11.3 may incorporate Sun's Jmaki framework v1.0 and Sun's Jmaki framework v1.0.3 from Sun Microsystems, Inc. Such technology is subject to the following terms and conditions: Copyright 1994-2006 Sun Microsystems, Inc. All Rights Reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission. This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. You acknowledge that this software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

Progress OpenEdge v11.3 may incorporate Jmaki framework v1.8.0 from Sun Microsystems, Inc. Such technology is subject to the following terms and conditions: Copyright 1994-2006 Sun Microsystems, Inc. All Rights Reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: * Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer. *
Appendix A: Third Party Acknowledgements

Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission. This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. You acknowledge that this software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

Progress OpenEdge v11.3 may incorporate JSTL v1.0 from Sun Microsystems, Inc. Such technologies are subject to the following terms and conditions: Code sample License Copyright 1994-2006 Sun Microsystems, Inc. All Rights Reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission. This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. You acknowledge that this software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

Progress OpenEdge v11.3 may incorporate Looks v2.1.2 from JGoodies. Such technology is subject to the following terms and conditions: The BSD License for the JGoodies Looks - Copyright (c) 2001-2007 JGoodies Karsten Lentzsch. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: o Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of JGoodies Karsten Lentzsch nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written
permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate OpenSSL v0.9.8 from The OpenSSL Project. Such technology is subject to the following terms and conditions: OpenSSL License. Copyright (c) 1998-2006 The OpenSSL Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)." 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. 5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. 6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)." THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

OpenSSL includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). Original SSSLey License. Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved. This package is an SSL implementation written by Eric Young eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are aheared to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, Ihash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com). Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author.
of the parts of the library used. This can be in the form of a textual message at program
startup or in documentation (online or textual) provided with the package.
Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met: 1. Redistributions of source
code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution. 3. All advertising materials mentioning features or use of
this software must display the following acknowledgement: "This product includes
cryptographic software written by Eric Young (eay@cryptsoft.com)." The word
'cryptographic' can be left out if the routines from the library being used are not
cryptographic related :-). 4. If you include any Windows specific code (or a derivative
thereof) from the apps directory (application code) you must include an
acknowledgement: "This product includes software written by Tim Hudson
(tjh@cryptsoft.com)." THIS SOFTWARE IS PROVIDED BY ERIC YOUNG `AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED
AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN
ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE. The licence and distribution terms for any
publically available version or derivative of this code cannot be changed. i.e. this code
cannot simply be copied and put under another distribution licence [including the GNU
Public Licence.]

Progress OpenEdge v11.3 may incorporate Quartz Enterprise Job Scheduler v1.3.2
from James House. Such technologies are subject to the following terms and
conditions: Copyright James House (c) 2001-2003 All rights reserved. Redistribution
and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met: 1. Redistributions of source code must
retain the above copyright notice, this list of conditions and the following disclaimer. 2.
Redistributions in binary form must reproduce the above copyright notice, this list of
conditions and the following disclaimer in the documentation and/or other materials
provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE AUTHOR
AND CONTRIBUTORS `AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES,
INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF
MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA,
OR PROFITS; OR BUSINESS INTERRUPTION). HOWEVER CAUSED AND ON
ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR
TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
SUCH DAMAGE. This product uses and includes within its distribution, software
developed by the Apache Software Foundation (http://www.apache.org/).

Progress OpenEdge v11.3 may incorporate relaxngDatatype.jar portion of Progress
Extensions for Eclipse v2.2. Such technology is subject to the following terms and
conditions: Copyright (c) 2001, Thai Open Source Software Center Ltd, Sun
Microsystems. All rights reserved. Redistribution and use in source and binary forms,
with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the names of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate SLF4J v1.6.2 and SLF4J Log4J v1.6.1 from sfl4j.org. Such technology is subject to the following terms and conditions:

Copyright (c) 2004-2008 QOS.ch All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Progress OpenEdge v11.3 may incorporate STAX API (JSR 173) v3.1.1 from Woodstox Project. Such technology is subject to the following terms and conditions:

Copyright (c) 2004-2010, Woodstox Project (http://woodstox.codehaus.org/) All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the Woodstox XML Processor nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate Trang 20030619 portion of Progress Extensions for Eclipse v2.2.1. Such technology is subject to the following terms and conditions: Copyright (c) 2002, 2003 Thai Open Source Software Center Ltd. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of the Thai Open Source Software Center Ltd nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REAGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate xpp3-1.1.3.4.O portion of Progress Extensions for Eclipse v2.2.1. Such technology is subject to the following terms and conditions: Indiana University Extreme! Lab Software License Version 1.1.1 Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment:

"This product includes software developed by the Indiana University Extreme! Lab (http://www.extreme.indiana.edu/)."
Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.

4. The names "Indiana Univeristy" and "Indiana Univeristy Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact http://www.extreme.indiana.edu/.

5. Products derived from this software may not use "Indiana Univeristy" name nor may "Indiana Univeristy" appear in their name, without prior written permission of the Indiana University.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate Xpp3 v1.1.3.4 from Indiana University. Such technology is subject to the following terms and conditions: Indiana University Extreme! Lab Software License Version 1.1.1 Copyright (c) 2002 Extreme! Lab, Indiana University. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Indiana University Extreme! Lab (http://www.extreme.indiana.edu/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. 4. The names "Indiana Univeristy" and "Indiana Univeristy Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact http://www.extreme.indiana.edu/.

5. Products derived from this software may not use "Indiana Univeristy" name nor may "Indiana Univeristy" appear in their name, without prior written permission of the Indiana University. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Appendix A: Third Party Acknowledgements

Progress OpenEdge v11.3 may incorporate XStream v1.3.1 from XStream. Such technology is subject to the following terms and conditions: Copyright (c) 2003-2006, Joe Walnes Copyright (c) 2006-2007, XStream Committers All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of XStream nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate YAJL v0.4.0 from Lloyd Hilaiel. Such technology is subject to the following terms and conditions: Copyright 2007, Lloyd Hilaiel. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of Lloyd Hilaiel nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate GNU ZLIB v1.1.4 and GNU ZLIB v1.2.3 from Jean-loup Gailiy & Mark Alder. Such technology is subject to the following terms and conditions: Copyright notice: (C) 1995-2002 Jean-loup Gailly and Mark Adler This software is provided 'as-is', without any express or implied warranty. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: 1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required. 2. Altered source versions must be plainly marked as such, and must not be
misrepresented as being the original software. 3. This notice may not be removed or altered from any source distribution.

Jean-loup Gailly        Mark Adler
jloup@gzip.org          madler@alumni.caltech.edu

Progress OpenEdge v11.3 may incorporate ZLIB.NET Free v1.0.4 from ComponentAce. Such technology is subject to the following terms and conditions: Copyright (c) 2006-2007, ComponentAce http://www.componentace.com All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of ComponentAce nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Progress OpenEdge v11.3 may incorporate LALR Parser Generator in Java v0.10k. Such technology is subject to the following terms and conditions: CUP Parser Generator Copyright Notice, License, and Disclaimer – Copyright 1996-1999 by Scott Hudson, Frank Flannery, C. Scott Ananian - Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both the copyright notice and this permission notice and warranty disclaimer appear in supporting documentation, and that the names of the authors or their employers not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Progress OpenEdge v11.3 includes the RSA Data Security, Inc. MD5 Message-Digest Algorithm. Copyright ©1991-2, RSA Data Security, Inc. Created 1991. All rights reserved. (MD5 Encryption Library v3.0 and MD5 Encryption vMD5C.C) These technologies are subject to the following terms and conditions: RSA Data Security MD5 message-digest algorithm RSA Data Security, Inc. MD5C.C - RSA Data Security, Inc., MD5 message-digest algorithm Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved. License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function. License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest
Appendix A: Third Party Acknowledgements

Algorithm® in all material mentioning or referencing the derived work. RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind. These notices must be retained in any copies of any part of this documentation and/or software.