

Borland C++ Builder 6 Certification

Study Guide

Version 1.0

Copyright © 2002 - 2003 Borland Software Corporation.

All Rights Reserved.

Introduction

This study guide provides information about the Borland C++ Builder certification exam to help you prepare for taking the exam. The material in this document is organized in the following three sections:

- Exam Overview – describes the rules and the organization of the exam.
- Resources – recommends training courses, books and manuals that can help acquire the knowledge required to pass the exam.
- Exam Sections – lists topics tested in each exam section.
- Sample Questions – provides sample questions and answers.

Exam Overview

General Details

The exam is organized as follows.

- Online exam (hosted by Prometric Online Testing Centers).
- Maximum time allowed for exam is **60 minutes**.
- Exam will contain **60 questions** (taken from a pool of several hundred questions).
- Minimum score required to pass is **80%** (48 or more correct answers required to pass).
- Most questions are of a multiple choice / single answer type. Some questions are multiple choice / multiple answer type. No partial credit is given for the multiple choice / multiple answer questions.
- All exam questions are equally weighted. Each question is weighed equally toward the final score.

Organization of Questions

The exam's questions are grouped into the following sections:

- IDE Environment
- Tool Usage (Advanced)
- VCL
 - Basic
 - Intermediate
 - Advanced
 - Database Support
- Database
 - Basic
 - Intermediate
 - Advanced
- Borland Conventions

- Packages and DLLs
- Language Conventions
- Internet Programming
- Advanced Windows

Each section contains a large pool of questions. At least one question is randomly chosen from each section. Every iteration of the test contains the same number of questions from each section. We do not provide the actual number of questions per section.

Notice that some subject areas are considered advanced topics, *Advanced Database* for instance. This means that a consultant-level understanding and familiarity of advanced Borland C++ Builder topics (as enumerated above) is necessary for successful Borland Product Certification.

Resources

Training Course

- *C++ Builder Application Development*. A 5-day instructor led course available through Borland Training. Delivered by Borland Certified Instructors. See <http://www.borland.com/services/training/classes/bcppbuilder/> for details.

On-line Documentation

- *Borland C++ Builder 6 Documentation*.
<http://www.borland.com/techpubs/bcppbuilder/>
- *Borland C++ Builder FAQ*.
<http://www.borland.com/devsupport/bcppbuilder/faq/>

Reference Books

- John Miano, *Borland C++ Builder How-To*. Waite Group Press, (1997)
- Dan Butterfield, Bob Swart, Jamie Allsop. *C++ Builder Developer's Guide*. Addison-Sams (2000).

Discussion forums

- The Borland Community - <http://community.borland.com/cpp/>
- There are over 20 newsgroups for C++ Builder discussions within the Borland User and Developer community - news://newsgroups.borland.com/orland.public.cppbuilder.* .

Exam Sections

C++ Builder IDE and Tool Usage

You are expected to know the fundamental concepts of C++ Builder. In particular, knowledge of the following is expected:

- Understand the differences, usage, and operations of a form, unit, package, and project. -This understand must extend to the contents of files generated, as well as their associated file types (.BPL, .BPR, .DFM, .OBJ, .EXE).
- Utilize common fields, buttons, and components.
- Recognize the default names of events associated with common component and object usage.
- Understand the life-cycle and use of common component events (OnClick, OnCreate, etc).
- Recognize a valid field name as well as valid property values for enumerated types.
- A demonstrated mastery in the use of the Integrated Development Environment (IDE) and the most commonly used configuration options. This includes the ability to recognize and use all function keys in the IDE (F11, F12, F7, F8, ctrl + arrow, etc).

VCL: Using Objects

- Object Parentage. You should be able to understand the importance of inheritance as it applies to Borland's Control Library. Specifically, you should understand:
 - What the name of the default object is and how it relates to all other objects.
 - Where the division between graphical and non-graphical components occur in Borland's Object Hierarchy (TGraphicControl, etc).
 - What a persistent object is, and how objects like TFileStream operate.
 - How to write custom controls.
- Object Operation: You need to be aware of how to apply graphical and non-graphical objects. Objects might include (but are not limited to):
 - TButton, TLabel, TImage, TImageList.

- TDataSource, TQuery, TDataSet, TField.
- TString, TStringList, TThread.

Database Architecture and Concepts

Expect the exam questions to be *very* database intensive. You should feel confident with the application of Borland's Database Components and related technologies at an advanced level. This include the ability to:

- Create database aliases and use them from their inception to the application of their hosted data elements in data-aware controls.
- Understand and recognize the interpretation of advanced SQL statements.
- Understand database cursor manipulation and operation.
- Feel confident applying database transactions, stored procedures, and triggers.
- Understand what files need to be included when distributing database applications.
- Manage database exceptions as well as their interception and processing.

C++ Concepts, Language, Extensions and Windows Environment

C++ Language Features: Understanding common as well as Borland's extensions to the C++ programming language is important. You should be able to:

- Understand the operation of common C++ inheritance keywords and conventions.
- Be able to utilize the "published" keyword (and related conventions) well enough to author new components for the component pallet.
- Be well-versed in exception creation, processing, handling, as well as Borland's exception hierarchy.
- Understand how to use Run-time Type Identification (RTTI) under C++ Builder (static_cast(), dynamic_cast(), etc.).

Related Tools and Technologies

The ability to configure and use key tools and related Borland Technologies is expected. Anticipate questions that require familiarity with the dialog-level configuration and run-time operations of:

- dbExpress.
- Advanced IDE configuration.

- Borland's Integrated Debugger.
- Database Desktop.
- Advanced package operations.

Sample Questions

Questions

1. The OnKeyPress event fires when which types of keys are pressed?
 1. arrow keys
 2. numeric keys (0-9)
 3. alphabetic keys (A-Z)
 4. function keys (F1-F12)

2. Which are valid lines in a .CPP file associated with a package file?
 1. USERES("DCLUSR35.res");
 2. Contains
 3. Requires
 4. USEPACKAGE("vc150.bpi");
 5. USEPACKAGE("vc150.BPL");

3. TLabel is a descendant of TGraphicControl. What is NOT a published property of the TLabel component?
 1. Visible
 2. OnClick
 3. TabOrder
 4. DragMode
 5. ParentFont

4. You have designed a class that inherits from a TCustomPanel named TMyPanel. Upon creation, an instance of your class creates a TStringList for its internal use and assigns that TStringList instance to a private field as follows: MyList = new TStringList(); Which destructor implementation is correct and will properly deallocate the TStringList?
 1. **TMyPanel::~TMyPanel(){ delete [] MyList;}**
 2. **TMyPanel::~TMyPanel(){ delete MyList;}**
 3. **TMyPanel::~TMyPanel(){ MyList->Free();}**

4. **Destructor TMyPanel::~TMyPanel(){ delete MyList;}**

5. When is a TField component's OnChange event activated, if a data-aware control is linked to the field?

1. when the control receives focus
2. after each keystroke inside the control
3. when the dataset is placed in Edit mode
4. after the control posts the changes into the current record

6. You have written a component that inherits from TTable. Upon creation, an instance of this component needs to create a TQuery and associate that query with the same database with which the Table descendant is associated. You try to use the following code in your constructor:

```
FQuery = TQuery->Create(self);  
FQuery->DatabaseName = self->DatabaseName;
```

But you notice that FQuery->DatabaseName is always an empty string no matter what value you assign to the TTable descendant at design time. Why is FQuery->DatabaseName an empty string?

1. Because FQuery->DatabaseName is defined as protected.
2. Because this->DatabaseName is not readable within a constructor.
3. Because this->DatabaseName is always an empty string at run time.
4. Because this->DatabaseName has not yet been loaded from the DFM information in the resources.

7. When typecasting an object, which construct will throw a Bad_cast_exception?

1. dynamic_cast
2. static_cast
3. SubClassOf
4. Type(Variable)

8. Given an ancestor class declared as:

```

class TAncestor : public TObject {
private:
    TColor *FColor;
    void SetColor(TColor *AColor);
protected:
    virtual void SetColorName(const String Name);
public:
    bool IsTransparent();
    __published:
    __property TColor Color = {read=FColor, write=SetColor};
};

```

Which is a proper polymorphic descendant class procedure declaration?

1. virtual void SetColorName(const String Name);
2. void SetColorName(const strParam String); virtual;
3. void SetColorName(const strParam String); override;
4. void SetDescendantColorName(const strParam String);

9. Which statement describes the most likely situation in which the "Optimization" compiler option should be unchecked?

1. Optimizations may cause spontaneous recursion resulting in stack overflow.
2. The increase in performance resulting from optimizations may cause timing and synchronization problems on older systems.
3. Optimized executables may report an incorrect .EXE checksum causing certain OS utilities to report the file as having been corrupted
4. Optimizations may cause local variables to be eliminated and looping code to be rearranged making it more difficult to use the debugger.

10. What will happen if in the process of creating a new package, you open the Package Options dialog and check only the Design Package under the Usage Options?

1. C++ Builder will create only the design-time package BPL.
2. The applications which utilize this package will not run outside the C++ Builder IDE.
3. C++ Builder will create the run-time .BCP file but will not create a run-time .BPL file.
4. The compiler will generate an error when the application using the package is compiled.

Answers to Sample Questions

C++ Builder Operations

1. The OnKeyPress event fires when which types of keys are pressed?
 1. arrow keys
 2. **numeric keys (0-9)**
 3. **alphabetic keys (A-Z)**
 4. function keys (F1-F12)

2. Which are valid lines in a .CPP file associated with a package file?
 1. **USERES("DCLUSR35.res");**
 2. Contains
 3. Requires
 4. **USEPACKAGE("vcl50.bpi");**
 5. USEPACKAGE("vcl50.BPL");

Using Objects

3. TLabel is a descendant of TGraphicControl. What is NOT a published property of the TLabel component?
 1. Visible
 2. OnClick
 3. **TabOrder**
 4. DragMode
 5. ParentFont

4. You have designed a class that inherits from a TCustomPanel named TMyPanel. Upon creation, an instance of your class creates a TStringList for its internal use and assigns that TStringList instance to a private field as follows: MyList = new TStringList(); Which destructor implementation is correct and will properly deallocate the TStringList?
 1. TMyPanel::~TMyPanel(){ delete [] MyList;}
 2. TMyPanel::~TMyPanel(){ delete MyList;}
 3. **TMyPanel::~TMyPanel(){ MyList->Free();}**
 4. destructor TMyPanel::~TMyPanel(){ delete MyList;}

Database Architecture and Concepts

5. When is a TField component's OnChange event activated, if a data-aware control is linked to the field?

1. when the control receives focus
2. after each keystroke inside the control
3. when the dataset is placed in Edit mode
- 4. after the control posts the changes into the current record**

6. You have written a component that inherits from TTable. Upon creation, an instance of this component needs to create a TQuery and associate that query with the same database with which the Table descendant is associated. You try to use the following code in your constructor:

```
FQuery = TQuery->Create(self);  
FQuery->DatabaseName = self->DatabaseName;
```

But you notice that FQuery->DatabaseName is always an empty string no matter what value you assign to the TTable descendant at design time. Why is FQuery->DatabaseName an empty string?

1. Because FQuery->DatabaseName is defined as protected.
2. Because this->DatabaseName is not readable within a constructor.
3. Because this->DatabaseName is always an empty string at run time.
- 4. Because this->DatabaseName has not yet been loaded from the DFM information in the resources.**

C++ Concepts and Language Extensions

7. When typecasting an object, which construct will throw a Bad_cast_exception?

- 1. dynamic_cast**
2. static_cast
3. SubClassOf
4. Type(Variable)

8. Given an ancestor class declared as:

```

class TAncestor : public TObject {
private:
    TColor *FColor;
    void SetColor(TColor *AColor);
protected:
    virtual void SetColorName(const String Name);
public:
    bool IsTransparent();
    __published:
    __property TColor Color = {read=FColor, write=SetColor};
};

```

Which is a proper polymorphic descendant class procedure declaration?

1. **virtual void SetColorName(const String Name);**
2. void SetColorName(const strParam String); virtual;
3. void SetColorName(const strParam String); override;
4. void SetDescendantColorName(const strParam String);

Related Tools and Technologies

9. Which statement describes the most likely situation in which the "Optimization" compiler option should be unchecked?

1. Optimizations may cause spontaneous recursion resulting in stack overflow.
2. The increase in performance resulting from optimizations may cause timing and synchronization problems on older systems.
3. Optimized executables may report an incorrect .EXE checksum causing certain OS utilities to report the file as having been corrupted
4. **Optimizations may cause local variables to be eliminated and looping code to be rearranged making it more difficult to use the debugger.**

10. What will happen if in the process of creating a new package, you open the Package Options dialog and check only the Design Package under the Usage Options?

1. **C++ Builder will create only the design-time package BPL.**
2. The applications which utilize this package will not run outside the C++ Builder IDE.
3. C++ Builder will create the run-time .BCP file but will not create a run-time .BPL file.
4. The compiler will generate an error when the application using the package is compiled.