AdIns
Advance Innovations

Exceptions & Error Handling

Topics

- Overview
- Where Exceptions Occur
- Exception Handling
- Global Exception Handlers

Overview

- No one is perfect.
- As developers, we create our software in a friendly environment where everything is set up to work perfectly.
- However, users will ultimately run software in a less-than-perfect environment where anything can and will go wrong and flaws in your software will be exposed.
- Unless handled correctly, exceptions cause application errors.

Where Exceptions Occur

- Exceptions can occur anywhere in code.
- The most common thing that can go wrong is the user entering an invalid user name and password.
- There are some common situations you should always design your applications to detect and gracefully handle :
  - Bad input
  - Multiuser conflicts
  - Network outages
  - Corrupt or missing files
  - Crashes
  - Stress

Exception Handling

- The safest bet is to assume you can’t detect and handle every unexpected circumstance and that exceptions will inevitably occur.

Exception Handling (Cont.)

- Some basic do’s and don’ts to handle the exception :
  - DOs
    - write *Try…Catch* or *On Error GoTo* exception handling on all database interactions.
    - report an error to the user explaining briefly what went wrong and what she should do next.
    - give the user information about who to contact when something goes wrong.
    - log full details of the exception.
  - DON’Ts
    - give too much information in the error message to the user.

Exception Handling (Cont.)

- *Try…Catch* or *On Error GoTo*
  - Which method you choose is mainly a matter of personal preference, although *Try…Catch* exception handling has an edge over *On Error GoTo* for several reasons.
  - You can mix and match the two types of exception handling, using *On Error GoTo* in one function and *Try…Catch* in another.
  - However, the Visual Basic compiler will not let you use both *On Error GoTo* and *Try…Catch* in the same function.

Global Exception Handlers

- Exceptions bubble up to the parent method.
- Putting an exception handler in *Sub Main* gives the effect of a global exception handler.
- Because of a problem in Visual Basic .NET 2002 and Visual Basic .NET 2003, some types of exceptions—typically, COM interop– based exceptions—do not bubble up to the parent exception handler.
- Viewing the Event Log Remotely