



- M**anage each user's set of permissions
- A**utomatically adjust the application  
the user connected

## Visual Guard for PowerBuilder

### **What is Visual Guard**

#### **What Visual Guard can do for you? :**

The main functionalities of Visual Guard. How will your application be modified to adapt itself to the user's permissions?

#### **Why choose Visual Guard ?**

Which benefits does Visual Guard provide? Gain of time, money and reliability...

#### **How does it work?**

Visual Guard's operation and its integration within your application

#### **Technical specifications**

#### **Download**

## Visual Guard for .NET

### **What is Visual Guard**

#### **What Visual Guard can do for you? :**

The main functionalities of Visual Guard. How will your application be modified to adapt itself to the user's permissions?

#### **Why choose Visual Guard ?**

Which benefits does Visual Guard provide? Gain of time, money and reliability...

#### **How does it work?**

Visual Guard's operation and its integration within your application

#### **Technical specifications**

#### **Download**

# Visual Guard for PowerBuilder

## What is Visual Guard?

Visual Guard is a tool designed to manage user profiles and permissions in PowerBuilder applications.

With Visual Guard, you determine what each user can do, see and modify.

Administrative tools simplify the daily management of users and their permissions.

No need to change the application's code: Visual guard dynamically changes the application's behaviour according to the user profile.

## What Visual Guard can do for you?

### Grant access to the application:

- Who is allowed to open the application?
- Which part of the application does each user have access to?

Example: Mr Smith belongs to the sales team. He can logon to the application and access the purchase order window. But as a salesperson, he cannot enter Human Resources windows.

Visual Guard will first secure the user login. Then, it will disable some menu options to restrict access to authorized windows only.

### Secure the data

You may define which user can modify or view each data.

Example: Salary is confidential. Only the director can modify it, some managers can review it and most of employee won't see it.

Visual Guard will hide this field or disable modifications depending on the user permissions.

For security or confidentiality reasons, you may also want to filter data users can see and modify.

Example: a salesperson in charge of USA can only access the deals made with American clients whereas the person in charge of Canada will access Canadian deals.

Visual Guard will dynamically filter the deals the user can see or modify.

### **Adapt the graphical interface to the User Profile**

You may need to change the application's GUI to reflect the user permissions.

Example: The customer window contains a button called "new" to create a new customer.

Visual Guard may hide or disable this button the user is not allowed to create customers.

This feature is available for any item of the user interface (button, tab, menu, fields, ...).

### **Parameter business logic**

Business rules may vary from one user to another.

Example: employees can create purchase orders up to \$500 and managers up to \$10,000.

Visual Guard will apply this limitation according to the value defined for the user role.

## ■ Why choose Visual Guard ?

### No coding necessary

No need to write code to define permissions. Just declare them within the Visual Guard Developer Workshop.

Furthermore, you will use Profile Manager to manage the different profiles and users without having to code anything within the application.

The application will be modified automatically at runtime.

Example: a new window is developed with a button that should be hidden from some users.

You will use the Developer Workshop to declare a permission which hides the button and Profile Manager to grant this permission to the appropriate users.

Once the window opens, Visual Guard will hide this button to users who are not allowed to see it. The addition of this permission does not take more than a few minutes and does not change the application code

### Easy implementation

You can integrate Visual Guard with any new or existing application.

The integration process is very simple: you add a few lines of code into your project.

Note: if you already have a permission system, you should probably deactivate it before using Visual Guard.

Our support team will guide you during this integration process. Do not hesitate **contact us!**

### **Release the development team from the daily role and user management:**

When using Visual Guard, profiles and users management does not require any specific development skills.

A non technical person (super-user, administrator, head of department...) can easily manage them.

He may declare profiles, grant them permissions and associate them to users.

It releases the IT department from the daily profile and user management.

### **Strongly reduce the cost of security**

**Development costs:** Without Visual Guard, you have to write and maintain some code (to modify the application according to the users, to manage profiles and permissions...). With Visual Guard, there is nothing to code, nothing to maintain.

**Profile and user management:** Without Visual Guard, developers have to manage profiles and users. With Visual Guard, end-users or administrators can handle it and release the development team.

**Availability:** Without Visual Guard, it takes time to develop and validate the security system. On the contrary, Visual Guard works immediately on any application.

**Update and application deployment:** Without Visual Guard, adding permissions implies code modifications and supplementary deployment.

With Visual Guard, you can add new permissions **without modifying or deploying the application**. These permissions are immediately active.

**Project management:** With Visual Guard, security implementation is totally independent from the rest of the application. You can implement Visual Guard during or after development. This integration can be done by a member of the development team or by someone in charge of security.

### ■ How does it work?

#### Visual Guard is made of:

- A security repository that stores user, profiles and permissions.
- Visual Guard engine, a component included in your application that modifies it dynamically according to the user permissions.
- An application allowing developers to create permissions: Developer Workshop.
- An application aimed at administrators to manage profiles and permissions daily: Profile Manager
- An example of a PowerBuilder application integrating Visual Guard

### **What happens at run time?**

- The user logs in the application
- Visual Guard connects to the security repository and retrieves the user permissions.
- Visual Guard engine then modifies the application according to these permissions.
- Example: when a window opens, Visual Guard may hide buttons and filter data.

More info => [Getting started](#)

### **Implementation process:**

1. Add Visual Guard engine to your project and activate the security (just a few lines of code required).
2. Implement Visual Guard login Window (or use your own login window).
3. Create a security repository and declare your application in this repository.
4. Developers define with the Developer Workshop all the permissions available for this application. As a result, a permission catalog is created. It will be used later (see step 6).
5. The application is tested, compiled and deployed.
6. Administrators use Profile Manager to create profiles and grant them permissions among those defined in the permission catalog. Then those profiles will be associated to users.
7. No technical skills are required: administrators may be super-users, heads of departments...

Note: If you already have a permission system, it is recommended to disable it before using Visual Guard

More info => [Getting started](#)

## ■ Technical specifications

### Technical requirements

- PowerBuilder 5 to 10.5
- VB.NET, C#, ASP.NET or ASP.NET 2.0
- .Net framework 1.1 or 2.0

You can integrate Visual Guard within any new or existing application, no matter how big or complex.

No limitations: any change you want to make in your application and any permission is possible. The only question is: how simple is it?

With visual Guard, there's always a way! Still have a doubt? Just ask us!

# Visual Guard for .NET

## ■ What is Visual Guard?

Visual Guard is designed to manage user **roles** and **permissions** in a .NET application.

With Visual Guard, you determine what each user can do, see and modify.

Management tools make daily management of users, roles and permissions very easy.

No need to change the application's code: Visual guard dynamically changes the application's behaviour according to the user permissions

## ■ What Visual Guard can do for you?

### Grant access to the application:

- Who is allowed to open the application?
- Which part of the application each user has access to?

Example: Mr Smith belongs to the sales team. He can logon to the application and access the purchase order window. But as a salesperson, he cannot enter Human Resources windows.

Visual Guard will first secure the user login. Then, it will disable some menu options to restrict access to authorized windows only.

### Secure the data

- You can define which user can modify or view each data.

Example: Salary is confidential. Only the director can modify it, some managers can review it and most of employee won't see it.

Visual Guard will hide this field or disable modifications depending on the user permissions.

- For security or confidentiality reasons, you may also want to filter data a user has access to.

Example: the salesperson in charge of USA can only access the deals made with American clients whereas the person in charge of Canada will access Canadian deals.

Visual Guard will dynamically filter the deals the user can see or modify.

### **Adapt the User Interface**

You may need to change the application's GUI to reflect the user permissions.

Example: The customer form contains a button called "new" to create a new customer.

Visual Guard may hide or disable this button if the user is not allowed to create customers.

This feature is available for any item of the user interface (button, tab, menu, field...).

### **Parameter business logic**

Business rules may vary from one user to another.

Example: employees can create purchase orders up to \$ 500 and managers up to \$ 10,000.

Visual Guard will apply this limitation according to the value defined for the user role.

## ■ Why choose Visual Guard ?

### **No coding necessary**

No need to write code to define permissions, roles and users. Just declare them within Visual Guard Console and the application will be dynamically modified at runtime.

Example: a new form is developed with a button that should be hidden from some users.

Use the Console to declare a new permission that hides the button and grant this permission to the appropriate roles: Visual Guard will dynamically hide the button at runtime.

Such a declaration only takes a minute and does not change the application code.

### **Easy implementation**

You can integrate Visual Guard with any new or existing application.

The integration process is very simple: you add a few lines of code into your project in order to activate Visual Guard engine.

Our support team can guide you during this integration process. Just contact us!

### **Release the development team from the daily role and user management:**

When using Visual Guard, roles and users management does not require development skills.

A non technical person (super-user, administrator, head of department...) can manage them.

He may declare roles, grant them permissions and associate roles to users.

It releases the IT department from the daily role and user management.

### **Strongly reduce the cost of security**

- **Development costs:** Without Visual Guard, you have to write and maintain some code (to modify the application according to the users, to manage roles and permissions...). With Visual Guard, there is nothing to code, nothing to maintain.

- **Roles and users management:** Without Visual Guard, developers have to manage roles and users every day. With Visual Guard, end-users or administrators can handle it and release the development team.

- **Availability:** without Visual Guard, it takes time to develop and validate the security system, whereas Visual Guard works immediately on any application.

- **Permission Updates:** without Visual Guard, adding permissions implies code modifications and supplementary deployment.

With Visual Guard, you can add new permissions **without modifying or deploying** the application's code. These permissions are immediately active.

- **Project management:** with Visual Guard, security implementation is totally independent from the rest of the application. You can integrate Visual Guard **during or after development**. This integration can be done by a member of the development team or by someone in charge of security

### ■ How does it work?

#### Visual Guard is made of:

- A security repository that stores user, roles and permissions.
- Visual Guard engine, a component included in your application that modify it dynamically according to the user permissions.
- Visual Guard Console that defines users, roles and permissions
- Visual Guard sample applications (windows & web sample code)

#### What happens at run time?

- The user logs in the application
- Visual Guard connects to the security repository and retrieves the user permissions.
- Visual Guard engine then modifies the application according to these permissions.  
Example: when a form opens, Visual Guard may hide buttons and filter data.
- More info => Getting started

### Implementation process:

1. Add Visual Guard engine to your project and activate the security (just a few lines of code required).
2. Implement Visual Guard login Window (or use your own login window).
3. Create a security repository and declare your application in this repository.
4. Developers define with the Console all the permissions available for this application. As a result, a permission catalog is created. It will be used later (see step
5. The application is compiled and deployed.
6. Administrators create roles and grant them permissions among those defined in the permission catalog. Then roles will be associated to users.

No technical skills required: administrators may be super-users, heads of departments...

Note: if you already have a permission system, you should probably deactivate it before using Visual Guard

More info => [Getting started](#)

### ■ Technical specifications

- VB.NET, C#, ASP.NET or ASP.NET 2.0
- .Net framework 1.1 or 2.0
- PowerBuilder 5 à 10.5

You can integrate Visual Guard within any new or existing application, no matter how big or complex.

No limitations: any change you want to make in your application and any permission is possible. The only question is: how simple is it?

With visual Guard, there's always a way! Still have a doubt? Just [ask us](#) !

# Ressources

## ■ Visual Guard for .NET:

- [See a demo for .NET](#)
- [Getting started](#)
- [Visual Guard Architecture](#)
- [How does it work?](#)
- [Receive an evaluation version](#)

## ■ Visual Guard for PowerBuilder:

- [See a demo for .NET](#)
- [Getting started](#)
- [Visual Guard Architecture](#)
- [How does it work?](#)
- [Receive an evaluation version](#)