

# LabVIEW Fondamental 2

## Description

Le cours LabVIEW Fondamental 2 est une extension du cours LabVIEW Fondamental 1 et vous apprend à utiliser des modèles de conception courants pour l'implémentation et la distribution réussies d'applications LabVIEW destinées à la recherche, l'ingénierie et aux environnements de test. Les sujets couverts incluent le contrôle de votre interface utilisateur par programmation, des techniques d'optimisation de la réutilisation du code existant, l'utilisation de fonctions d'E/S sur fichiers, et les outils permettant de créer des exécutables et des installeurs. Ce cours met l'accent sur les fonctionnalités de LabVIEW qui répondent aux besoins de vos applications et vous permet de commencer rapidement le développement d'applications.

## Durée

En salle, avec instructeur : deux (2) jours

En ligne, avec instructeur : trois (3) sessions de 4 heures et travail personnel

## Public

- Nouveaux utilisateurs et utilisateurs souhaitant développer des applications avec LabVIEW ou NI Developer Suite
- Personnes ayant suivi le cours LabVIEW Fondamental 1
- Utilisateurs et responsables techniques qui évaluent LabVIEW ou NI Developer Suite en vue d'un achat
- Utilisateurs souhaitant obtenir la certification CLAD (Certified LabVIEW Associate Developer)

## Pré-requis

- Expérience avec Microsoft Windows
- LabVIEW Fondamental 1 ou expérience équivalente

## Produits NI utilisés pendant le cours

- Système de développement professionnel LabVIEW
- Matériel NI d'acquisition de données (DAQ)
- BNC-2120

## Après avoir suivi ce cours, vous pourrez :

- Utiliser des variables locales pour modifier des commandes de la face-avant ou arrêter des boucles parallèles
- Appliquer des modèles de conception courants qui utilisent des files d'attente et des événements
- Contrôler les objets d'interface utilisateur par programmation
- Évaluer les formats d'E/S sur fichiers et les utiliser dans des applications
- Modifier le code existant pour faciliter son utilisation

## Inscription

Inscrivez-vous en ligne sur [ni.com/training](http://ni.com/training) ou appelez le 01 57 66 24 24 Fax : 01 57 66 24 14  
E-mail : [france.formation@ni.com](mailto:france.formation@ni.com)

Hors de France, contactez votre filiale NI locale.  
Informations sur les filiales : [ni.com/global](http://ni.com/global)

## Numéros de référence

- 910796-xx
- 01 Siège social de NI ou filiale
  - 11 Régional
  - 21 Sur site (dans vos locaux)
  - 69 En ligne, avec instructeur
  - 71 En ligne, à votre rythme, avec un accès d'un an
  - 76 En ligne, à votre rythme, avec un accès de 6 mois

- Préparer, construire, mettre au point et déployer des applications autonomes

## Suggestions de cours à suivre par la suite

- LabVIEW Fondamental 3
- LabVIEW Connectivity
- LabVIEW Performance
- Data Acquisition and Signal Conditioning
- Embedded Control and Monitoring Using LabVIEW
- Autres cours sur le matériel

## Certification recommandée

- Certification CLAD (Certified LabVIEW Associate Developer)

# Programme du cours LabVIEW Fondamental 2

## 1er jour

### Utilisation de variables

Cette leçon vous explique comment utiliser des variables locales pour modifier la valeur des commandes de la face-avant, arrêter des boucles parallèles et contourner les restrictions du flux de données. Les points traités comprennent :

- La communication entre des boucles parallèles
- L'utilisation de variables globales et locales
- L'écriture de commandes et la lecture d'indicateurs
- La compréhension et la prévention de situations de compétition

### Communication de données entre des boucles parallèles

Cette leçon décrit des techniques de communication asynchrone pour la création de code contrôlé par l'IU et qui synchronise les données entre des boucles parallèles. Les points traités comprennent :

- L'utilisation de files d'attente pour transférer des données bufférisées entre des boucles
- L'utilisation de notificateurs pour diffuser des données à plusieurs boucles

### Implémentation de modèles de conception

Cette leçon vous présente les modèles de conception. Vous apprendrez les avantages et les fonctionnalités spécifiques aux modèles de conception et comment ils peuvent être utilisés pour diminuer le temps de développement. Deux catégories de modèles de conception sont étudiées : les modèles de conception à boucle unique et à boucles multiples. Les points traités comprennent :

- L'utilisation de modèles de conception à une boucle, y compris les modèles de conception de machine à états et les variables globales fonctionnelles
- L'utilisation de modèles de conception à boucles multiples, y compris les modèles de conception producteur/consommateur
- La gestion des erreurs
- La génération de codes et de messages d'erreur
- Le cadencement d'un modèle de conception

### Contrôle de l'interface utilisateur

Cette leçon décrit des méthodes permettant de contrôler les attributs des objets de la face-avant par programmation, comme la désactivation temporaire d'une commande. Vous apprenez à utiliser le

VI Serveur pour accéder aux propriétés et aux méthodes des objets de la face-avant. Les points traités comprennent :

- L'architecture du VI Serveur
- L'utilisation de nœuds de propriété
- L'utilisation de nœuds de méthode
- La création et l'utilisation de références de commandes

## 2ème jour

### Techniques d'E/S sur fichiers

Cette leçon décrit différents formats de fichiers pour la collecte et l'enregistrement de données et la manière de sélectionner le format approprié pour vos applications. Vous vous exercez à implémenter du code modulaire qui lit ou écrit des données de mesure. Les points traités comprennent :

- La comparaison des formats de fichiers
- La création de chemins de fichiers et de dossiers
- L'écriture et la lecture de fichiers binaires
- L'utilisation de fichiers texte multivoies à en-têtes
- L'accès aux fichiers au format TDMS (Technical Data Management Streaming) dans LabVIEW et Excel

### Amélioration d'un VI existant

Cette leçon met l'accent sur des méthodes de refactorisation de code LabVIEW hérité et sur la manière d'optimiser la réutilisation du code existant. La refactorisation consiste à remanier le logiciel afin de le rendre plus lisible et plus facile à entretenir sans altérer son comportement observable. Les points traités comprennent :

- La refactorisation de code hérité
- Les problèmes connus lors de la refactorisation de code

### Création et distribution d'applications

Cette leçon décrit le processus de création d'exécutables autonomes et d'installeurs pour les applications LabVIEW. Vous apprendrez à utiliser l'Application Builder dans LabVIEW. Les points traités comprennent :

- La préparation des fichiers
- La création de spécifications de construction
- La création et la mise au point d'une application
- La création d'un installeur