

 **TestStand™** +  **LabVIEW™**

**une association idéale
pour l'automatisation de vos tests**

**Jean-Louis Schricke
MESULOG**

- ➔ **Introduction**
- ➔ **TestStand : vue d'ensemble**
- ➔ **TestStand : cinq phases**
- ➔ **LabVIEW : quelques rappels**
- ➔ **TestStand et LabVIEW : complémentarité**
- ➔ **Conclusion**

➔ Objectif

- ◆ Proposer une vue d'ensemble de TestStand
- ◆ Permettre d'identifier les cas où l'utilisation de TestStand est souhaitable
- ◆ Mettre en évidence la complémentarité de TestStand et LabVIEW

➔ Illustration

- ◆ Un exemple détaillé de mise en œuvre TestStand et LabVIEW est présenté par SAPHIR

➔ MESULOG en bref

- ◆ Activité : Développement logiciel test et mesure
- ◆ Compétence : TestStand, LabVIEW (Windows, RT, PDA)
- ◆ Création : mars 2000
- ◆ Localisation : Grenoble
- ◆ Taille : « humaine » (3 ingénieurs)
- ◆ Partenaire National Instruments
- ◆ Développeur certifié LabVIEW et TestStand



➔ Historique

- ◆ Successeur de NI Test Executive (Add-on LabVIEW ou LabWindows/CVI)
- ◆ De nombreuses évolutions
 - 1999 : Version 1.0
 - 2001 : Version 2.0
 - 2003 : **Version 3.0**
 - 2004 : Version 3.1
 - 2005 : Version 3.5

➔ Un standard de fait

- ◆ TestStand s'impose en tant que plate-forme depuis la version 3.0
- ◆ **14** of the top **15** electronics manufacturers use NI TestStand
(*Electronic Business*, 2004)

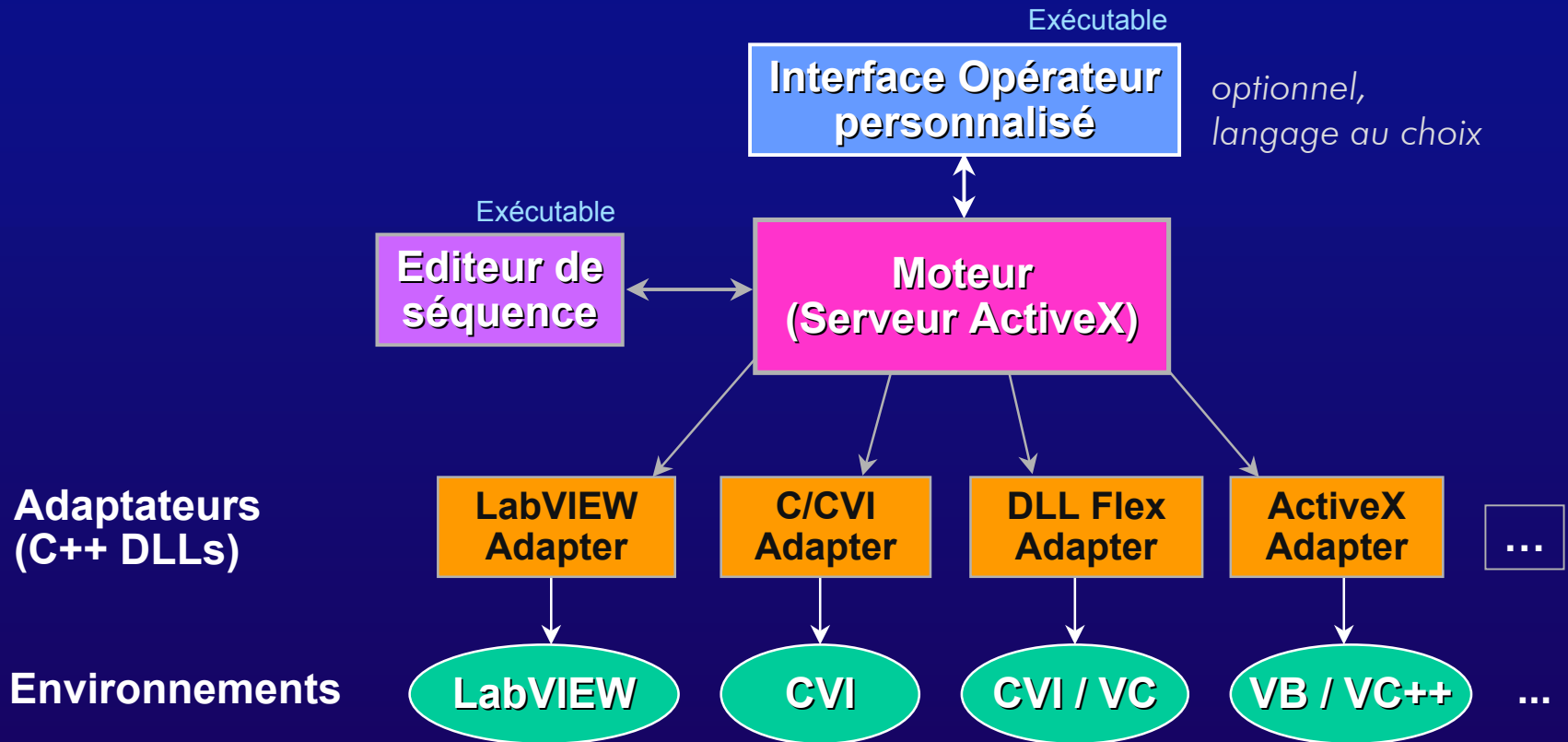
➔ C'est quoi TestStand ?

« TestStand est un séquenceur de test prêt à l'emploi permettant d'organiser, de contrôler et d'exécuter vos systèmes de tests de prototype, de validation ou de production automatisés ».

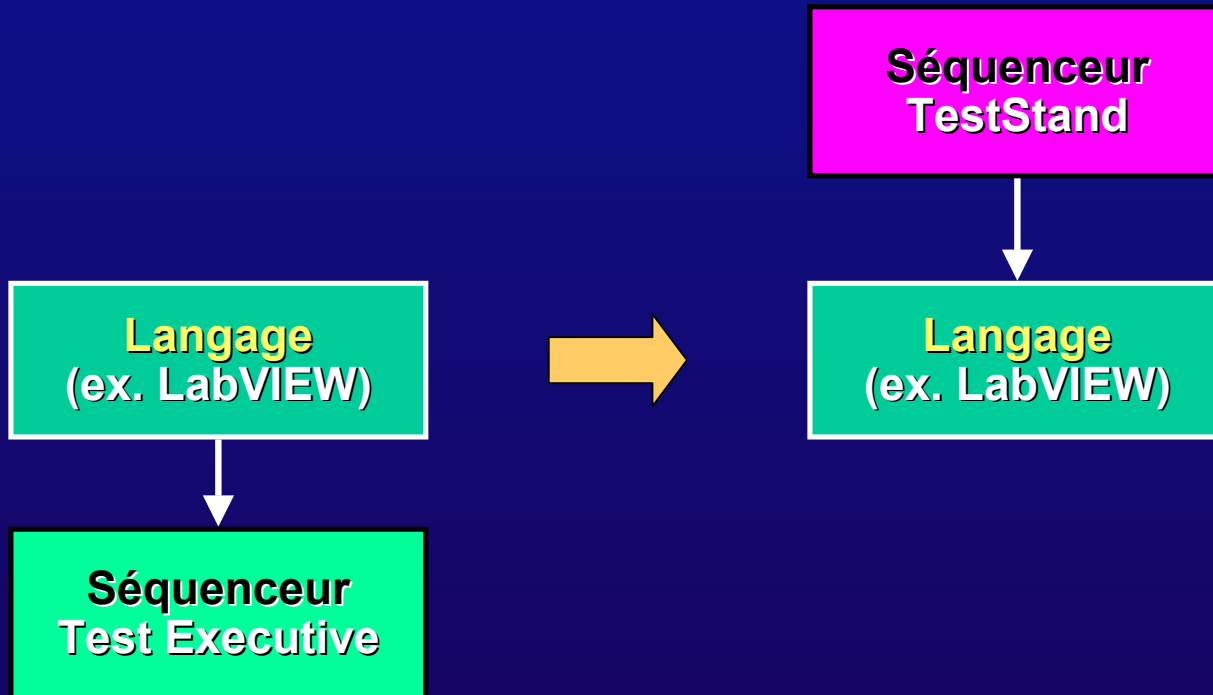
➔ C'est quoi un séquenceur ?

- ◆ Un séquenceur permet de créer, de configurer et d'exécuter des séquences de tests.
- ◆ Une séquence de test est une suite de « pas unitaires » dont l'exécution peut être conditionnée par le résultat des pas précédents.

➔ Architecture de TestStand



➔ Positionnement du séquenceur



➔ En standard dans TestStand

- ◆ Gestion des utilisateurs (avec profils)
- ◆ Identification de l'Objet Sous Test (en anglais : UUT)
- ◆ Génération automatique d'un rapport (trace d'exécution)
- ◆ Sauvegarde des résultats en base de données
- ◆ Outils de branchement et de synchronisation
- ◆ Outils de chargement de conditions de tests (Property Loader)
- ◆ Pilotage d'instruments IVI
- ◆ Traitement des erreurs
- ◆ Deux interfaces opérateur (avec code source)

➔ Phase 1 : Programmation

- ◆ Création de « pas » unitaires configurables (environnement au choix)
- ◆ Chaque pas comprend :
 - un module d'exécution
 - un module de configuration (boîte de dialogue)
- ◆ Intégration des pas créés dans TestStand



Nouvelles « briques » personnalisées venant enrichir la palette TestStand.

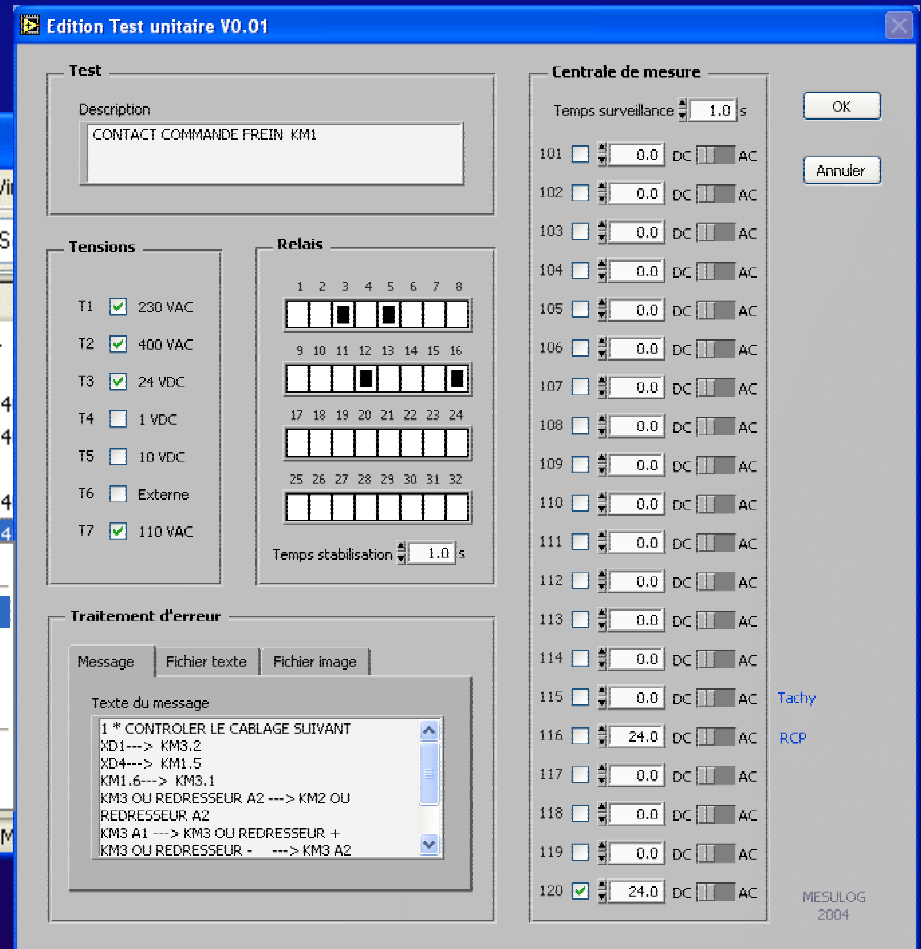
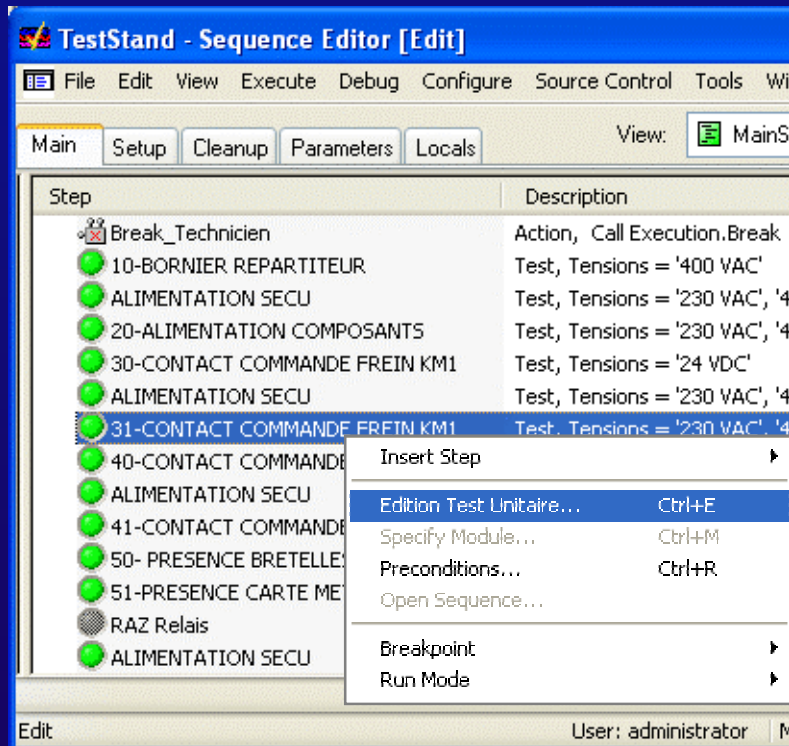
➔ Phase 2 : Édition de séquences

- ◆ Création de séquences et de sous-séquences à l'aide de NI SeqEdit :
 - Insertion de « pas standard » TestStand
 - Insertion de « pas personnalisés génériques » configurables
 - Insertion de « pas personnalisés spécifiques » non configurables
- ◆ Données :
 - Variables locales ou globales
 - Paramètres d'entrées et de sortie
- ◆ Relations entre les pas
 - Exécution conditionnelle
 - Boucles
 - Synchronisation



Mais qui fournit le ciment ?

➔ Phase 2 : Édition de séquences (exemple)



➔ Phase 3 : Création interface opérateur

- ◆ Personnalisation d'un des deux modèles (Simple ou Full-Featured)
 - Ajout d'indicateurs de suivi d'exécution
 - Ajouts de commandes
 - Localisation des messages et indicateurs
- ◆ Mise en place de la communication avec les séquences
 - Évènements de mise à jour (UIMessage)
 - Liens vers un Serveur de données (ex. OPC)
- ◆ Phase optionnelle :
 - si l'on utilise l'éditeur de séquence NI SeqEdit en exécution
 - si l'on utilise une interface opérateur standard

➔ Phase 3 : Interface opérateur (Exemple 1)

Mesulog **SODIMAS : Banc de test SDE**

Logout Reprise Terminer Test Quitter

Etape	Description	Commentaire	Mode Exec.	Status
Message	"Raccordement"			
TestUnit #1	Test, Tensions = 24VDC, 10VDC, 230VAC, <N... Relais = 00010010 01000110 01100001 00000010			
TestUnit #2	Test, Tensions = <NC>, 10VDC, 230VAC, <NC... Relais = 00011010 01000101 01110001 00000010			
End				

Resultat du test:
Produit conforme
OK

Entrées / Sorties Digitales

Tensions
T1: 1 VDC
T2: 10 VDC
T3: 24 VDC
T4: 230 VAC
T5: 400 VAC
T6: Externe

Relais
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32

Sécurité
Etat QF1
Etat QF2
Sécurité
Clé

Voyants
Buzzer
Test en cours
Défaut

Interface raccordement
code 1: 12
code 2: 0

Contrôle
Etat QF1
Etat QF2

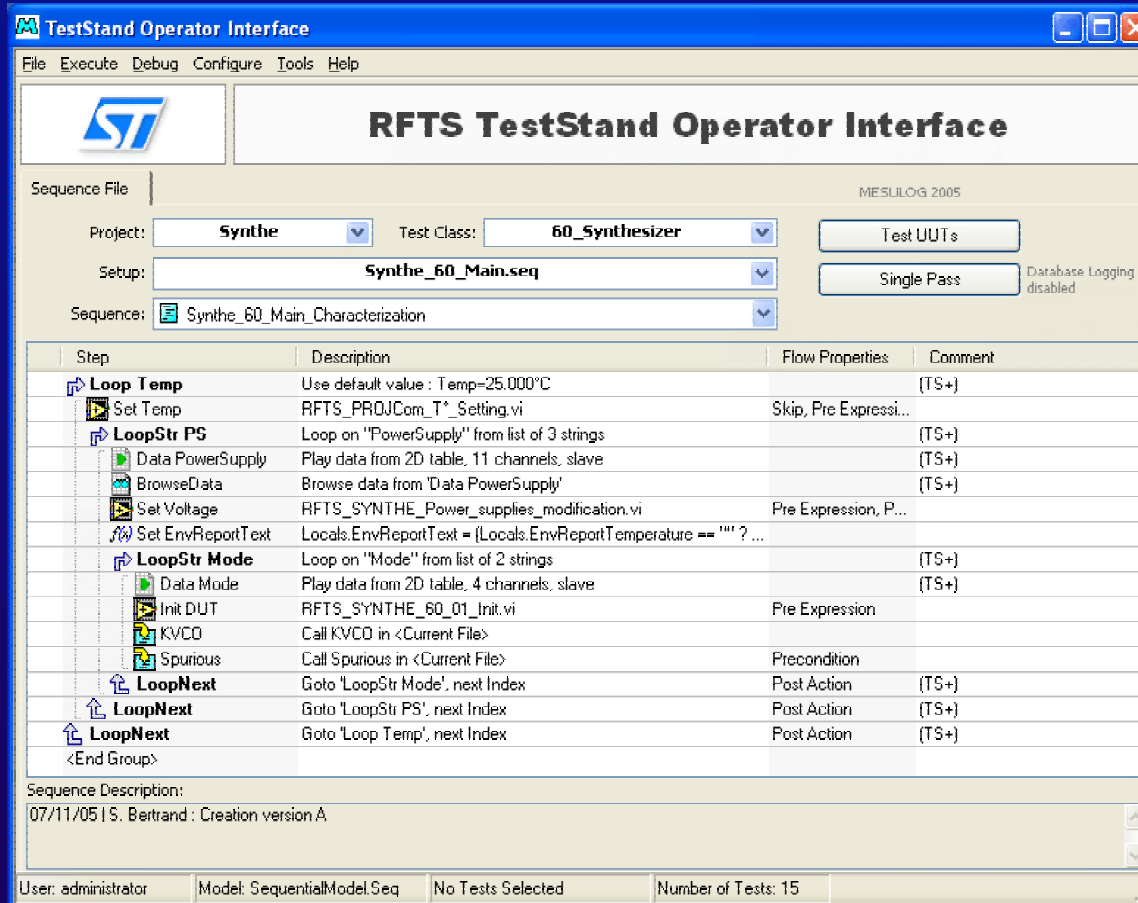
Ordre de Fabrication: 1234567
Scénario: Carte porte et frein
Progression: 20:26:51 12/09/04 Cycle 1 / 3

Centrale de mesure
Agilent 34970A
N° voie: 102 VDC
24,13 V

Utilisateur: Mesulog Nombre d'étapes: 3

*Banc de test
en production*

➔ Phase 3 : Interface opérateur (Exemple 2)



Sequence File

MESULOG 2005

Project: **Synthe** Test Class: **60_Synthesizer** Test UUTs

Setup: **Synthe_60_Main.seq** Single Pass Database Logging disabled

Sequence: **Synthe_60_Main_Characterization**

Step	Description	Flow Properties	Comment
Loop Temp	Use default value : Temp=25.000°C		(TS+)
Set Temp	RFTS_PROJCom_T*_Setting.vi	Skip, Pre Expressi...	
LoopStr PS	Loop on "PowerSupply" from list of 3 strings		(TS+)
Data PowerSupply	Play data from 2D table, 11 channels, slave		(TS+)
BrowseData	Browse data from 'Data PowerSupply'		(TS+)
Set Voltage	RFTS_SYNTHE_Power_supplies_modification.vi	Pre Expression, P...	
Set EnvReportText	Locals.EnvReportText = (Locals.EnvReportTemperature == "" ? ...		
LoopStr Mode	Loop on "Mode" from list of 2 strings		(TS+)
Data Mode	Play data from 2D table, 4 channels, slave		(TS+)
Init DUT	RFTS_SYNTHE_60_01_Init.vi	Pre Expression	
KVCD	Call KVCD in <Current File>		
Spurious	Call Spurious in <Current File>	Precondition	
LoopNext	Goto 'LoopStr Mode', next Index	Post Action	(TS+)
LoopNext	Goto 'LoopStr PS', next Index	Post Action	(TS+)
LoopNext	Goto 'Loop Temp', next Index	Post Action	(TS+)
<End Group>			

Sequence Description:
07/11/05 | S. Bertrand : Creation version A

User: administrator Model: SequentialModel.Seq No Tests Selected Number of Tests: 15

Banc de
caractérisation

➔ Phase 4 : Exécution

- ◆ Login : Profil Opérateur
 - Banc de test fonctionnel (→ PASS / FAIL)
 - Banc de caractérisation (→ exploitation des résultats)
- ◆ Login : Profil Technicien
 - Mise au point, exécution pas à pas
 - Activation/désactivation d'étapes de test

➔ Phase 5 : Évolutions

- ◆ Modification des séquences existantes
 - Changement de l'ordre des tests
 - Modification des conditions limites
 - Duplication et modification de certains pas
 - ◆ Création de nouvelles séquences
 - Duplication et modification d'une séquence existante
 - Réutilisation des pas génériques
- Évolutions possibles sans connaissance poussée de TestStand

➔ LabVIEW en production

- ◆ Génération impérative d'un exécutable (Application builder)
 - Traçabilité
 - Gestion de version
- ◆ Fiabilité et robustesse à soigner :
 - Traitement d'erreur
 - Traitement des interactions utilisateur
- ◆ Fonctionnalités recommandées (à implémenter) :
 - Fichier(s) de configuration
 - Fichier « trace » ou « journal de bord »
 - Fichier « historique des erreurs »

➔ Possibilités de séquençement

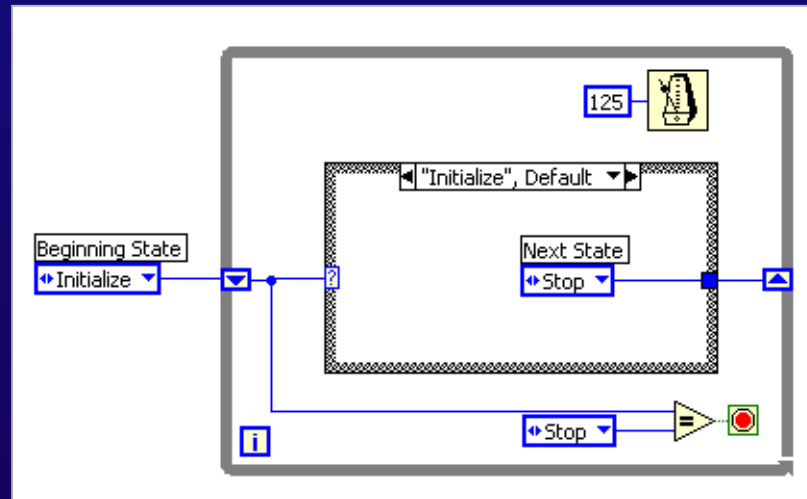
- ◆ Séquence



- ◆ Cheminement d'erreur



- ◆ Machine à états



- ◆ Etc.

➔ LabVIEW en tant que séquenceur ?

- ◆ En production la modification de l'ordre des étapes nécessite :
 - de disposer du code source
 - de disposer de LabVIEW et de son module Application builder
 - de disposer d'un développeur LabVIEW
 - de générer un nouvel exécutable et de tester la nouvelle séquence

OUI

Si la séquence de test :

- est figée
- ne dépend pas du produit testé

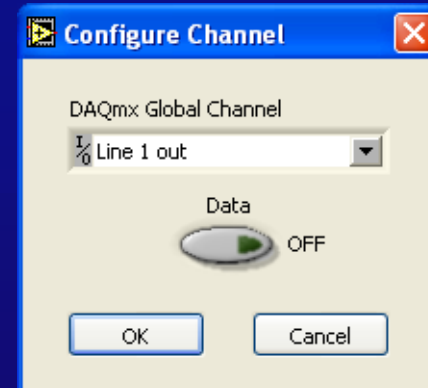
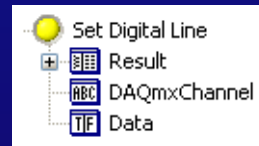
NON

Si la séquence de test :

- doit évoluer
- dépend du produit testé

➔ En phase 1 : Programmation

- ◆ LabVIEW permet de développer les « briques »
 - module d'exécution
 - module de configuration



- ◆ Le passage de paramètres TestStand-LabVIEW est simple

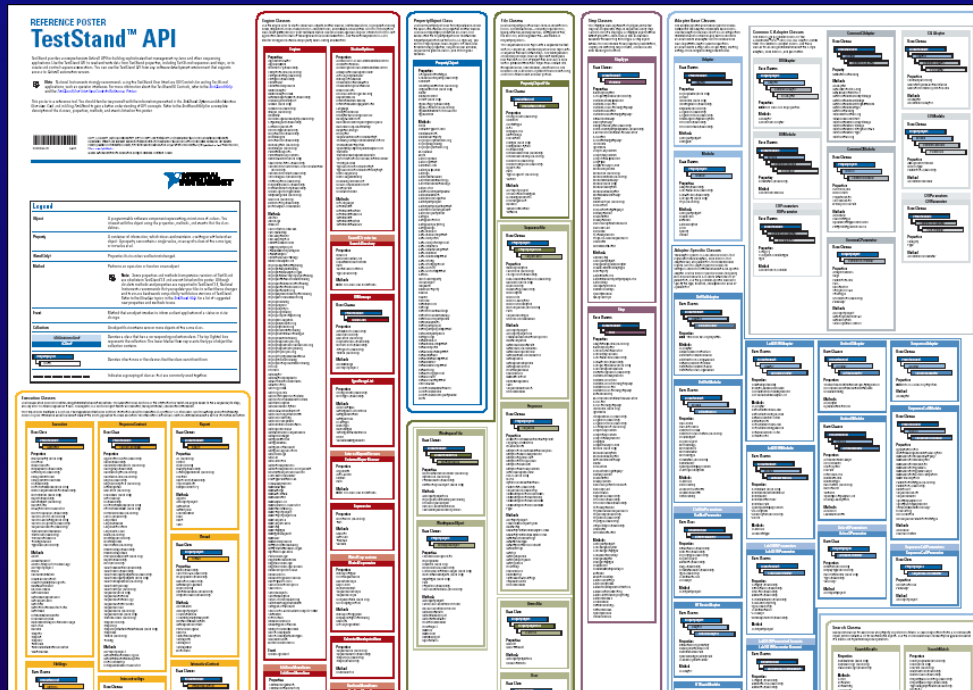
The 'Description and Connector Pane' in LabVIEW. It shows a connector symbol with four terminals: 'Channel' (dotted line), 'data' (dotted line), 'timeout' (solid line), and 'error out' (dotted line). Below the connector is a table of parameters:

Name	Type	In/Out	Default	Value
Channel	ASCII Str	in		Step.DAQmxChannel
data	Boolean	in	<input type="checkbox"/>	Step.Data
timeout	Number (DBL)	in	<input checked="" type="checkbox"/>	10.00000
error out	Container	out		Step.Result.Error

➔ En phase 3 : Création interface opérateur

◆ ActiveX

- LabVIEW est client, serveur et conteneur ActiveX
- Toute la couche API de TestStand est disponible sous LabVIEW

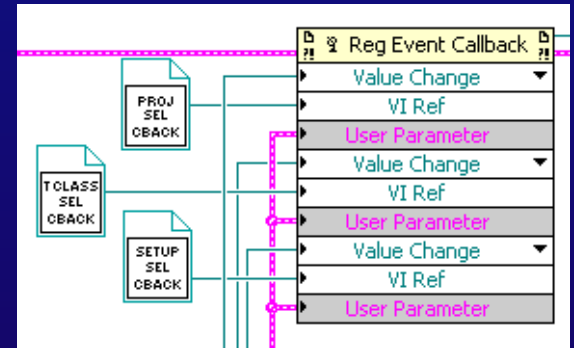


➔ En phase 3 : Création interface opérateur

◆ LabVIEW et le mode évènementiel

- depuis sa version 6.1, les actions utilisateurs peuvent générer des évènements LabVIEW (structure évènementielle).
- depuis sa version 7.0, LabVIEW peut prendre en charge dynamiquement des évènements ActiveX

Déclenchement d'un « VI Callback » par action utilisateur sur un contrôle LabVIEW.



◆ Composants ActiveX TestStand pour interface opérateur

- depuis sa version 3.0, TestStand intègre des UIControls (Composants ActiveX dédiés User Interface)

➔ En phase 4 : Exécution

◆ Performances ?

ADAPTER	CVI-Object	CVI-DLL	DLL	DLL Numeric Limit, Precondition, 4 parameters	ActiveX	LabVIEW Run-Time	LabVIEW
Trace ON	6.30	5.80	4.40	6.30	5.60	4.70	5.20
Trace OFF / Results ON	0.25	0.25	0.14	0.18	0.10	0.09	0.50
Trace OFF / Results OFF	0.16	0.15	0.05	0.14	0.07	0.05	0.47

*Exemple fourni avec TestStand : Benchmark.seq
Temps d'exécution moyen d'un pas en millisecondes*

- ➔ **TestStand et LabVIEW sont complémentaires**
- ➔ **L'utilisation de LabVIEW facilite le développement des pas de test et de l'interface opérateur**
- ➔ **TestStand est un outil puissant mais complexe**
 - ◆ Formation indispensable
 - ◆ Assistance Partenaire NI (si possible certifié TestStand)
 - ◆ Utilisation de pas de test « prêt à l'emploi »
 - StepTypes en téléchargement sur NI Developer Zone
 - MESULOG  sur www.ni.com/teststand



www.mesulog.fr