



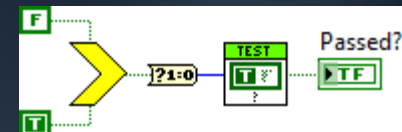
**Tips & Tricks for TestStand development
NI Days 2013**

1. LabVIEW & TestStand:n yhteiskäyttö
2. TestStand arkkitehtuurimalleja LabVIEW pohjaisesti
3. Hyödyllisiä vinkkejä testisekvenssien luontiin

Main (3)	
 Action	Action
 Pass/Fail Test	Pass/Fail Test
 String Value Test	String Value Test

TestStand perinteisesti hoitaa kaikki testirutiinin suoritukseen liittyvät toiminnot, tietokantaoperaatiot sekä raportoinnin

LabVIEW moduulit taas vastaavasti suorittavat jonkin spesifistisen toiminnon esim. yksittäisen mittauksen



Perinteinen lähestymistapa voi johtaa ongelmatilanteisiin:

- Liikaa "älyä" voi siirtyä TestStand:n puolelle
 - Ylläpito ongelmallista, skaalautuvuus huono
 - Huono dynaamisuus
- Sekvenssitiedostojen koko voi kasvaa suureksi
 - Paljon yksittäisiä steppejä tai "MultiPurpose" – tyyppisiä steppejä joissa paljon toimintoja
 - Suoritetaanko steppejä joiden tulokset ovat turhia? Ts. nämä tulokset kuormittavat turhaan tulostietokantaa

Ratkaisuehdotuksia:

- Viedään suurin osa vaativimmista toiminnoista & päättelystä LabVIEW moduulien tehtäväksi
- Yhdistetään TestStand:n sekä LabVIEW:n hyvät puolet ja rakennetaan skaalautuvampi & dynaamisempi arkkitehtuuri

LabVIEW



- Vaativampi päätöksen teko
- Testimoduulien suoritus
- Testimoduulit
- Datan manipulointi

TestStand



- Korkeantason vuonohjaus
- Tulosten tallennus/raportointi
- Toimii korkeantason "moottorina"

*Mallit pätevät varsinkin tilanteissa joissa testisekvenssin kulkuun vaikutetaan ulkopuolelta (Käyttöliittymät)

Ehdotuksia arkkitehtuurimalleiksi:

- Preconditio -pohjainen sekvensseri
- LabVIEW ohjattu tilakonesekvensseri
- TestStand ohjattu dynaaminen alisekvenssejä kutsuva sekvensseri
- LVOOP pohjainen LabVIEW sekvensseri

Preconditio (ennakkoehtoinen) pohjainen sekvensseri

+ Helppo saada aikaan

- Mikä tekniikka ohjaa preconditionia taustalla?
→ "Äly" karkaa monesti sekvenssiin
- Voi syntyä turhien steppien suoritusta
- Synnyttää monimutkaista expressio syntaksia, jota on hankala tulkita
- Dynaamisuus heikko

LabVIEW ohjattu tilakonesekvensseri

- + Dynaamisuus
- + Helppo lisätä ominaisuuksia (Skaalautuva rakenne)
- + Voidaan helposti tehdä "SubSequenceri" jolla oma käyttöliittymä
- Työläämpi toteuttaa
- Testeistä täytyy tehdä Source Distribuutio Runtime ympäristössä jotta LabVIEW löytää kutsuttavat VI't

TestStand ohjattu dynaaminen alisekvenssejä kutsuva sekvensseri

- + Dynaamisuus
 - + Helppo toteuttaa
 - + Työnjako n. 50/50 (TS/LV)
 - + Helppo ylläpitää jos ”polku VI” tehty hyvin
- ”Polku VI” = mekanismi joka kertoo sekvenssitiedostojen polut alisekvenssikutsulle

LabVIEW ohjattu LVOOP -pohjainen sekvensseri

- + Dynaamisuus
 - + Helppo toteuttaa
 - + Jos projekti muutenkin (LabVIEW) oliopohjainen helppo ottaa käyttöön
 - + Suoritettavien testien hallinnointi helppoa ja siistiä
- Jos projektin testit eivät ole oliopohjaisia niin voi olla "liian suuri kustannus" lähteä tieteen tahtoen implementoimaan tätä mallia.

Aina kun referenssien avaamista ynnä muita muistia kuluttavia toimintoja viedään LabVIEW:n puolelle, on muistettava sulkea referenssit sekä hyödyntää LabVIEW:n toimintoja oikein, jotta LabVIEW:n sisäinen muistinhallinta toimii parhaimmalla mahdollisella tehokkuudella

The background of the slide is an abstract composition of dark blue and white. It features a series of concentric, wavy lines that create a sense of depth and movement, resembling a stylized wave or a lens flare effect. The lines are more pronounced on the left side and fade towards the right.

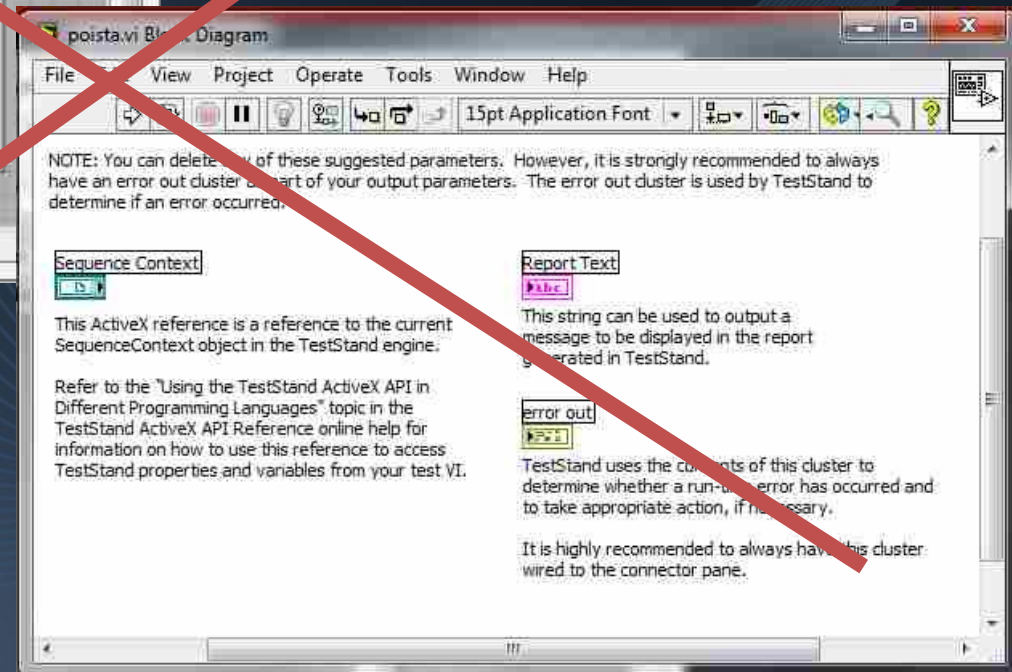
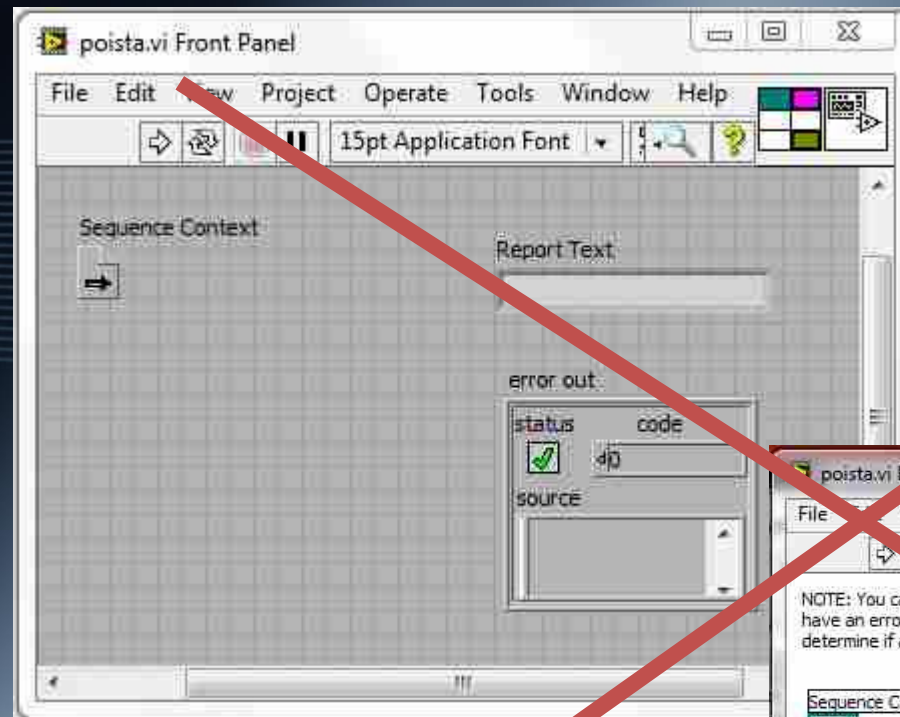
Tips & Tricks

Navigation

Undo / Redo	Ctrl + Z / Ctrl + Y
Navigate Back / Forward	Alt + ← / Alt + →
Types Palette	Ctrl + T
Close Sequence File	Ctrl + F4
Switch Tabs (Sequence Files)	Ctrl + Tab

Debugging / Execution

Execute >> Single Pass	Ctrl + F5
Execute >> Test UUTs	F5
Close Completed Executions	Ctrl + D





Kiitos ajastanne!