DCAF

Supportability Report

|  |  |
| --- | --- |
| Role | Person/People |
| Project Manager | James Wright Gorman |
| Business Owner | Andy Kuker |
| Product Manager | Rita Prather (previously Deborah Burke and Matt Spexarth) |
| Section Manager | Malcolm Borgendale |
| Director | Stefanie Breyer |
| VP – R&D | Kevin Schultz |

Table of Contents

1 Supportability of Product 3

1.1 Ease of Installation or Upgrading 3

1.2 Ease of Configuration 3

1.3 Functionality and Quality of Product 3

1.4 Quality of Shipping Examples 3

1.5 Quality of Documentation 3

2 Support and Maintenance Plan 3

2.1 Special Training Requirements 3

2.2 Web Resources 3

# Supportability of Product

Given that DCAF is meant to help customers avoid problems before they might encounter them (that is, give them a good framework *up front* to reduce problems down the road), we would help customers the most by bringing up DCAF before they start into a large project.

One of the largest impediments to this is a learning curve that is both steeper and longer than many other NI products.

## Ease of Installation or Upgrading

-Installs only through VIPM

-This is not common for official NI software – normal installs are via Web Downloads, NI Update Service, and NIPM

-Distributing solely through VIPM will probably be an issue for customers with restricted internet access. While there is [a way around this](https://support.jki.net/hc/en-us/articles/214135803), it requires VIPM Professional.

-Without a traditional installer, it is more difficult to repair a software package (e.g. Repair in Add/Remove programs or Force Reinstall)

-There are currently 30 DCAF VIPM packages published by National Instruments.

-The way we distribute packages in VIPM could be consolidated, because DCAF Core installs many other VIPM packages. I recommend incorporating some of these packages into DCAF Core (*which* packages is a topic for discussion).

## Ease of Configuration

A large amount of this product is configuration based, and there can be relatively little LabVIEW programming. Therefore, I believe *Ease of Configuration* to be more important for this product than most others.

-Configuration of pre-made modules is non-intuitive and training is helpful (e.g. tags vs mappings vs channels, similar variable names)

-Creating and configuring a static module without training is difficult, but possible. I expect a CLAD would be able to do this after training.

-Creating a dynamic module without training (self-paced or in person) is nearly impossible. With training, a CLD or higher should be able to do this.

In summary, DCAF would be very difficult to configure without some form of training. I recommend we delineate the ability to create dynamic modules to a separate VIPM package. This way, we can caution people to be at the CLD level of competency at a minimum.

**Standard Configuration Editor:**

-It takes a non-trivial amount of time to load (and a significant amount of time the first time)

-When you click the red x to exit, it doesn’t close the VI window. I feel like there must be a way around this.

**Static Module Scripting Utility:**

-Improve documentation on the front panel. As a new user, I cannot make sense of how to complete the tasks I understand the utility was built for (let’s have a meeting).

-(From the front panel of this utility) How would we unload a specified User Control Module from memory? Even if there are multiple ways, pointing the user in the right direction is better than nothing.

## Functionality and Quality of Product

After talking with a user of early DCAF (Platypus), the quality and functionality of this product have greatly increased in the last two years. Some outstanding pain points include:

-Having to configure mappings for larger projects (this process does not scale well)

-Generating the ‘includes’ section can take 15-20 minutes for large projects

-There’s a lot of code to deploy to a target before you can run. This takes a while (normally the first deploy takes a few minutes)

-DCAF is built on LVOOP, which scares many developers. Pollock: “why hasn’t anyone tried OOP?” Nabors employee: “because we have LabVIEW (chuckle)… it has a different paradigm – dataflow”

-Since DCAF is partially crowd-sourced, anyone can create a module to fill a gap they perceive. Each new module brings added functionality to the product

-Target support is vague at best. While DCAF core can be run on Windows, Linux RT, Phar Lap, or VxWorks, not all modules support all targets. I recommend we clearly document expected target compatibility across DCAF core and modules (I’m envisioning additional columns on the “List of Available DCAF Plugins” document).

## Quality of Shipping Examples

There is a LabVIEW Example – Temperature Controller Example.lvproj. It’s similar to what is used in the hands on.

The example is fairly well documented via the two accompanying PDFs.

Most of the ‘example’ code we ship is in the form of Project Templates.

These templates are immensely helpful. Their description in the project template selection window is also helpful.

## Quality of Documentations

Pros: we have lots of documentation

Cons: we have lots of documentation

The volume of documentation we have is intimidating.

We don’t have a clear pathway through it.

“One of the things that makes it difficult is there's no official training. There's a lot of unofficial training...but I don't know where to start” – customer during the beginning of training.

A notable gap in documentation is a clear guideline on which modules are officially NI supported, which modules are partially NI supported, and which modules fall outside of the purview of NI. A logical place for this would be as another column on the “List of Available DCAF Plugins”.

Another large aspect of documentation stems from DCAF not starting as a NPD project. Typically, we have tech writers creating a manual, Marketing creating White Papers, and a PSE creating KBs/WIPs and Known Issues Lists.

This project has none of those.

One effect of this is an inconsistent look and feel across documents and as a part of the NI documentation ecosystem. Systems has done an excellent job creating the current content, and that content outside of the Tech Comm process.

Another effect of this is that all documentation is either on the forum or github. While this inherently is not good or bad, it is a break from the normal NI channels. Customers are not accustomed to this paradigm from NI. This also has a ripple effect for normal support operations – if a customer issue comes in that we want to document, do we create a KB? Or a forum post? We should decide and document a ‘documentation plan’ (update: we now have such a plan).

# Support and Maintenance Plan

## Special Training Requirements

Initially, we won’t expect AEs to support this product and there will be no AE-specific training. If AEs do wish to learn about the product, they will be able to take advantage of all the publicly available training (which currently is all of it).

For customers, Rita (Marketing) and I would like to go through the documents and create a more cohesive and pointed pathway for learning. This might branch into additional pathways (as needed) – for instance, if they have multiple customer personas we plan to serve. This process may also include consolidating some documents together.

A rough vision of this: a clear beginner pathway (Configuration Editor and static modules) for everyone; an additional pathway for those who are advanced users (module creator).

## Web Resources

Currently, the only web resource at NI.com is the forum. This holds a wealth of VCM-like content (tutorials, manuals, introductions, et c.). As part of our documentation plan, we will move a few posts into VCM containers.

Github will be the main technical web resource, and will present a relatively new paradigm for us.

-Customers will be able to pull code at any time.

-Customers will also be able to dig as deep into the code as they wish.

-They could potentially pull the whole stack, make some small internal change for a local branch they use, and run into unintended consequences. This will make troubleshooting more difficult, as we might not know they have modified the standard distribution.

-Customers can submit any thing they deem a bug or feature request

-Customers will be able to see all of our ‘dirty laundry’ and see all open issues.