Integration of SVN into PVSS

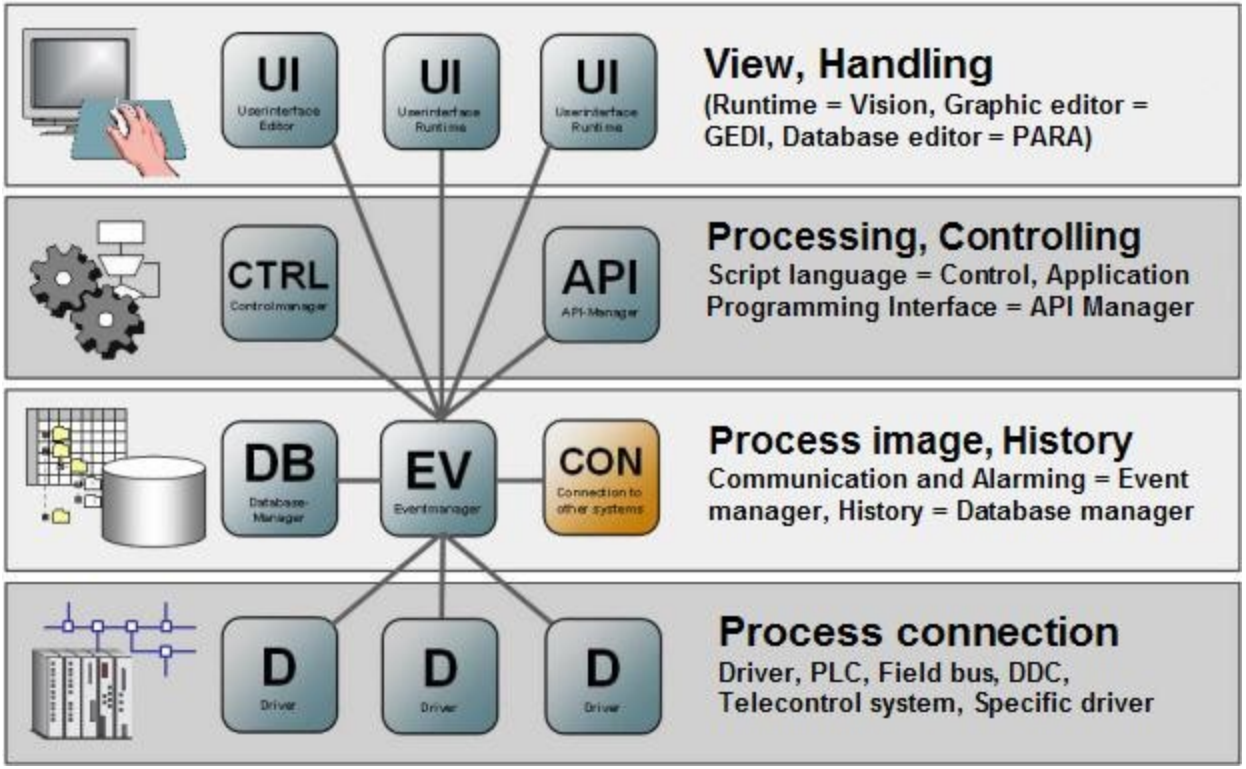
Choices and Challenges

Outline

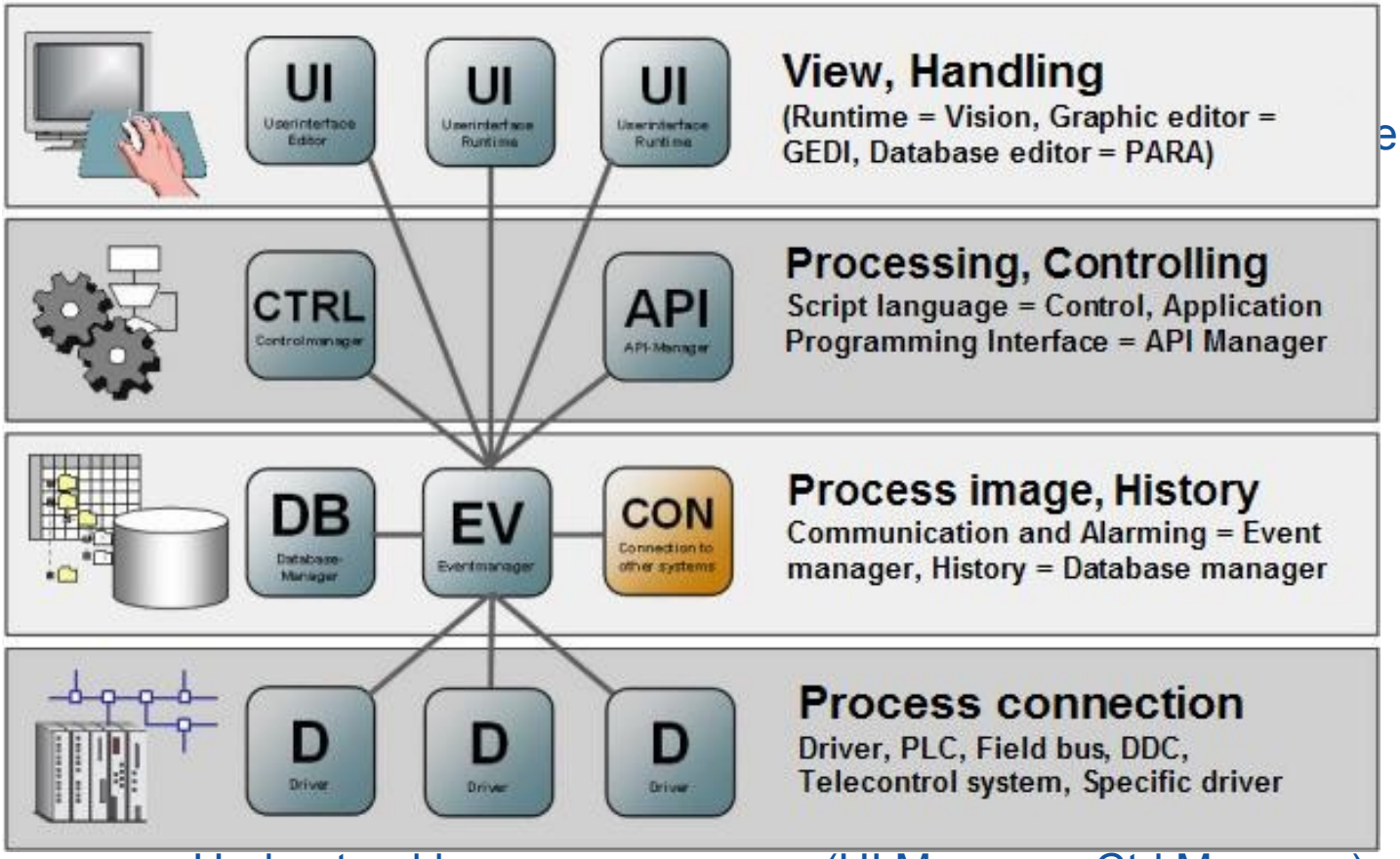
- PVSS Architecture and Basic Concepts
  - Projects
  - Gedi
- Versioning Control in PVSS
  - Strategy
  - Ctrl Interface
  - Integration into GEDI
- Current challenges

PVSS architecture

- PVSS is a SCADA system suitable for Large Distributed Control Systems



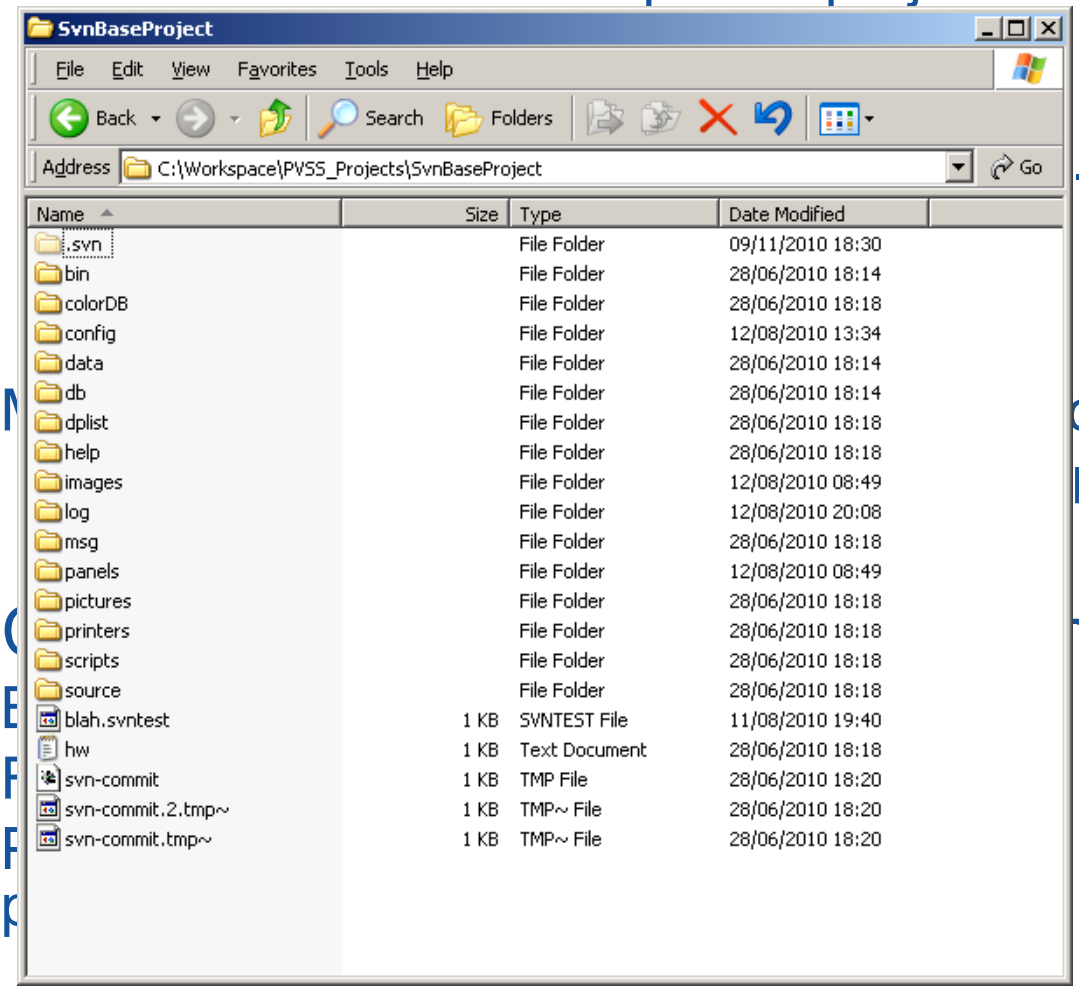
- Basic Concepts



- Understood by some managers (UI Manager, Ctrl Manager)

PVSS concepts

- PVSS Projects
  - Consist of files within a specific project folder:



...)

oject

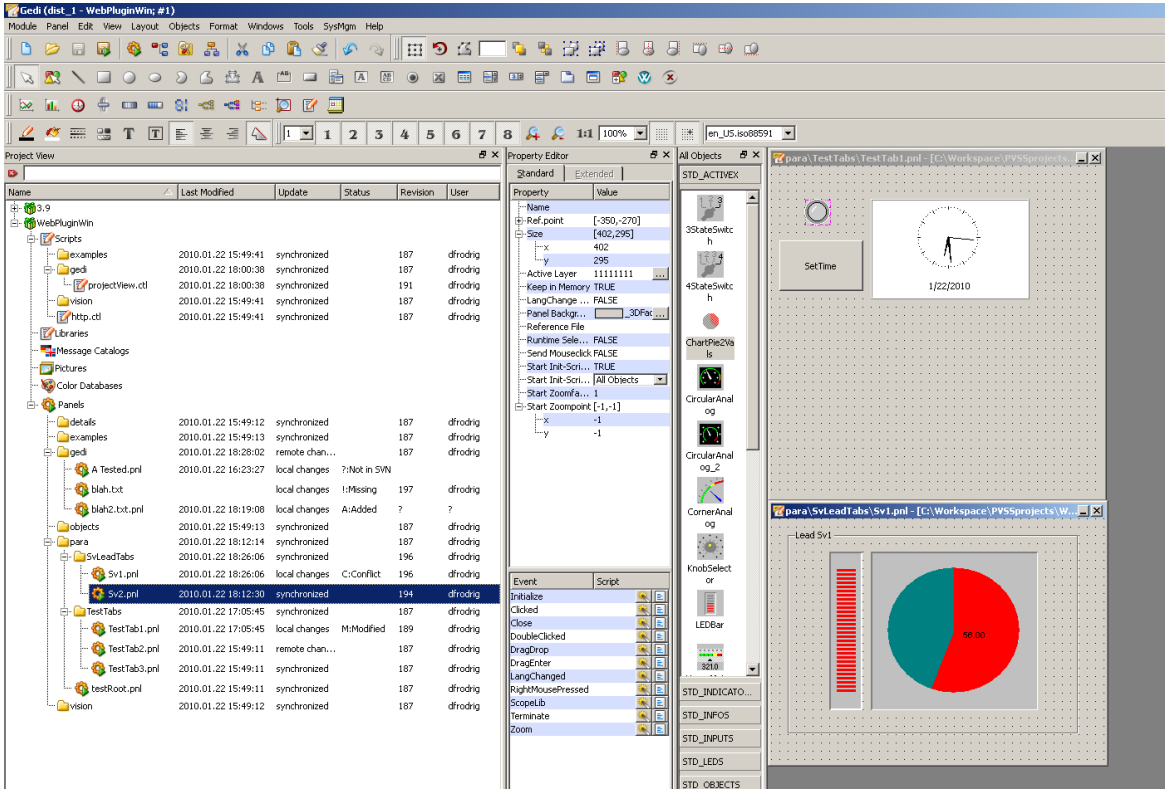
process

nsively

projects

project

- GEDI
  - A PVSS Integrated Development Environment
    - Creation of Panels and Scripts
    - Access to different functionalities (para, system, …)

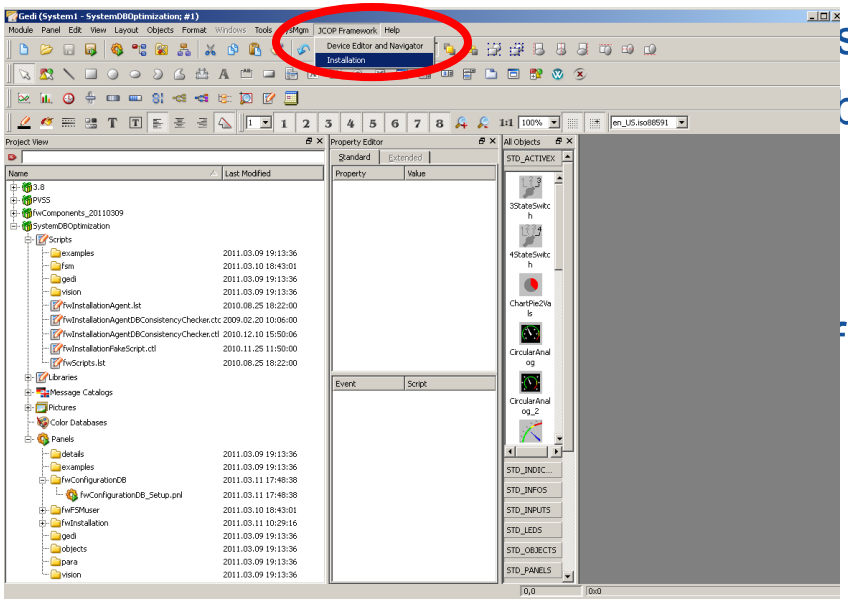


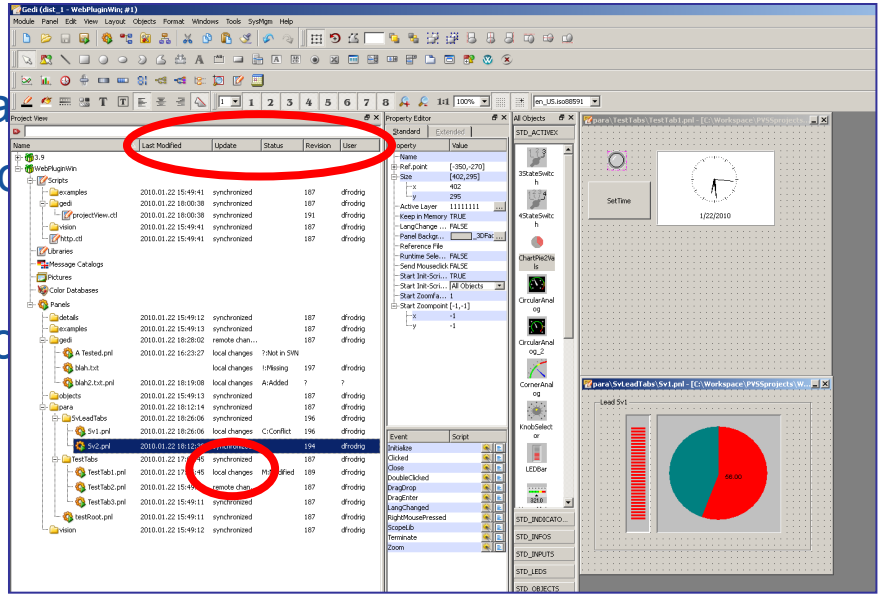
Versioning Control in PVSS

- Benefits of using Version Control are well known
  - Change control, accountability, rollback, team collaboration, backup
- PVSS Projects change over time
  - We want those benefits into PVSS.
  - Panels, scripts, datapoints are candidates
  - not the project database, however
    - Keeps the current project running status
    - Most development is on static components
- GEDI should embed versioning control capabilities

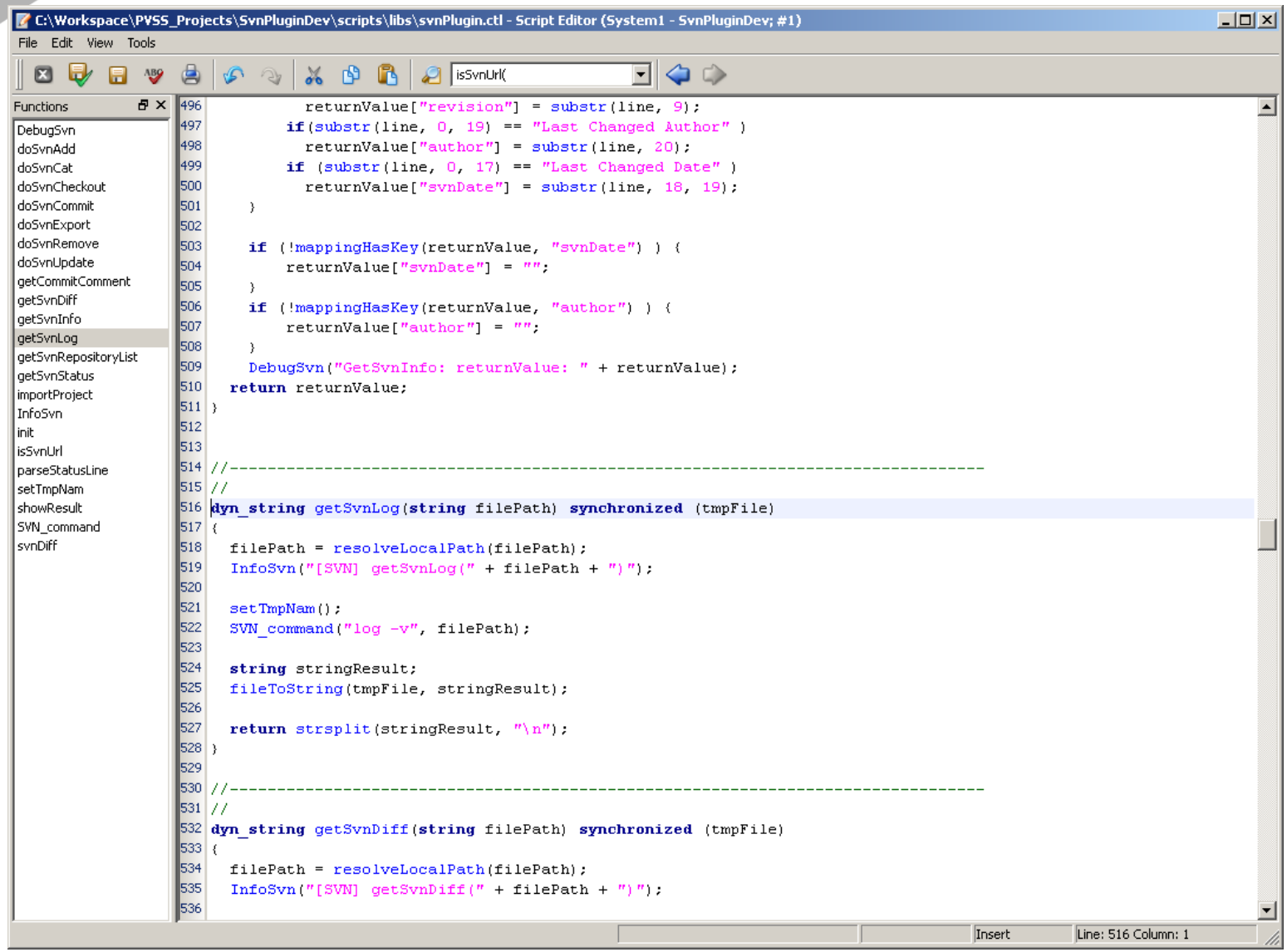
Versioning Control in PVSS

- Strategy
  - Gedi is an UI Manager
    - Understands Scripting Language (CTRL)
  - Before loading, project view runs a ctrl script
    - We can add



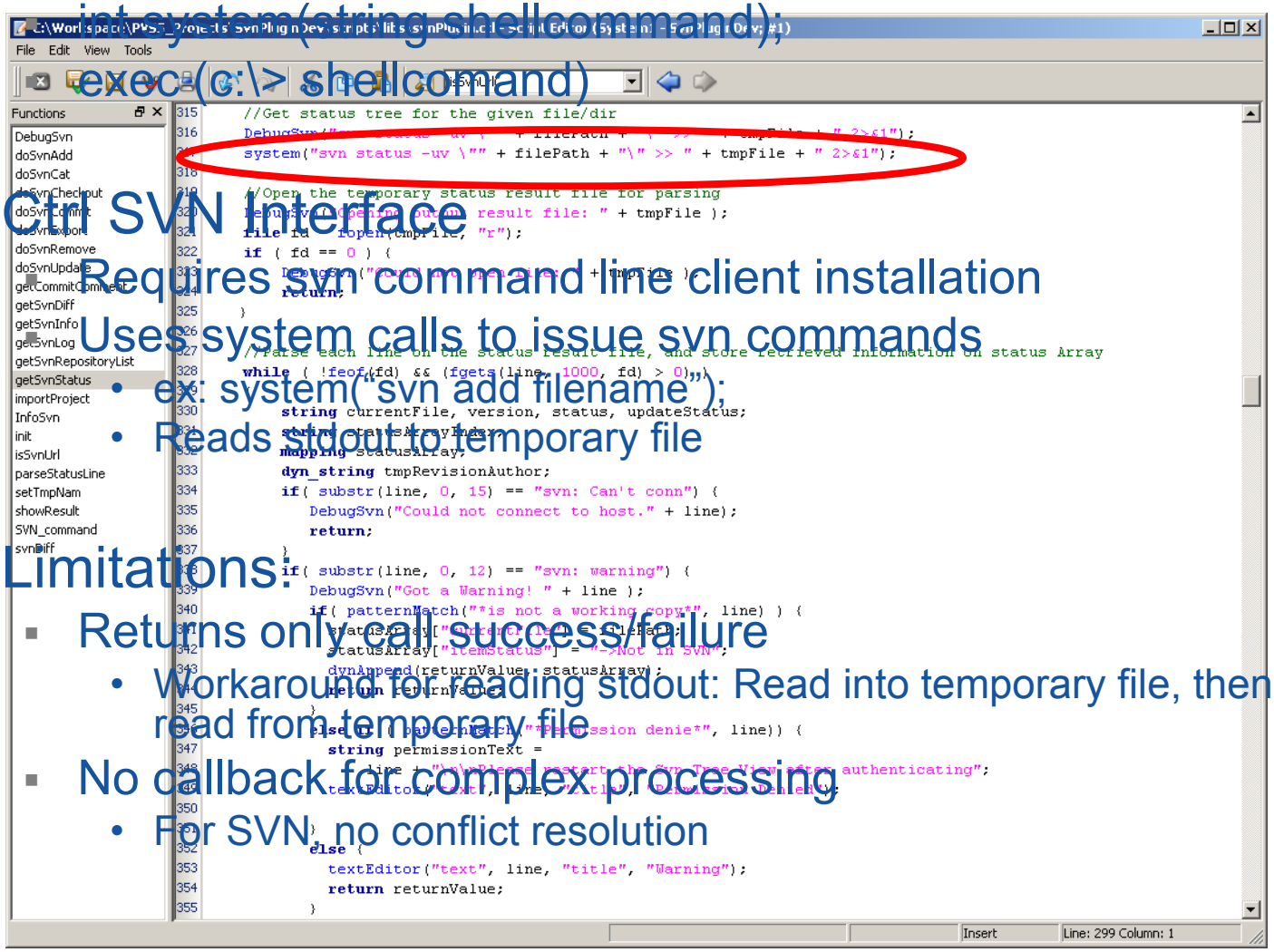


Ctrl SVN interface



Ctrl SVN interface

- CTRL allows for system calls
  - int system(string shellcommand);
  - exec(c:\> shellcomand)
- Ctrl SVN Interface
  - Requires svn command line client installation
  - Uses system calls to issue svn commands
    - ex: system("svn add filename");
    - Reads stdout to temporary file
- Limitations:
  - Returns only call success/failure
    - Workaround for reading stdout: Read into temporary file, then read from temporary file
  - No callback for complex processing
    - For SVN, no conflict resolution



Ctrl SVN interface

- Ctrl SVN library summary
  - Portable library, supporting linux and win32
  - Abstracts SVN command line

Interface

```
dyn_mapping getSvnStatus (string filePath )
dyn_string  getSvnLog    (string filePath )
dyn_string  getSvnDiff   (string filePath )
mapping     getSvnInfo   (string filePath )
string      doSvnRemove  (string filePath, bool commit=False )
string      doSvnUpdate  (string filePath, bool commit=False )
string      doSvnAdd     (string filePath, bool commit=false, string commitComment=" “ )
string      doSvnCommit  (string filePath, string commentText = " “ )
dyn_string  importProject(string projectPath, string projectRepositoryRoot, string commentText = " “ )
dyn_string  doSvnCheckout(string projectPath, string projectSvnUrl )
dyn_string  doSvnExport  (string projectPath, string projectSvnUrl )
dyn_string  doSvnCat     (string fileSvnUrl, string dumpFile = " “ )
dyn_string  getSvnRepositoryList(string svnRepositoryUrl )
```

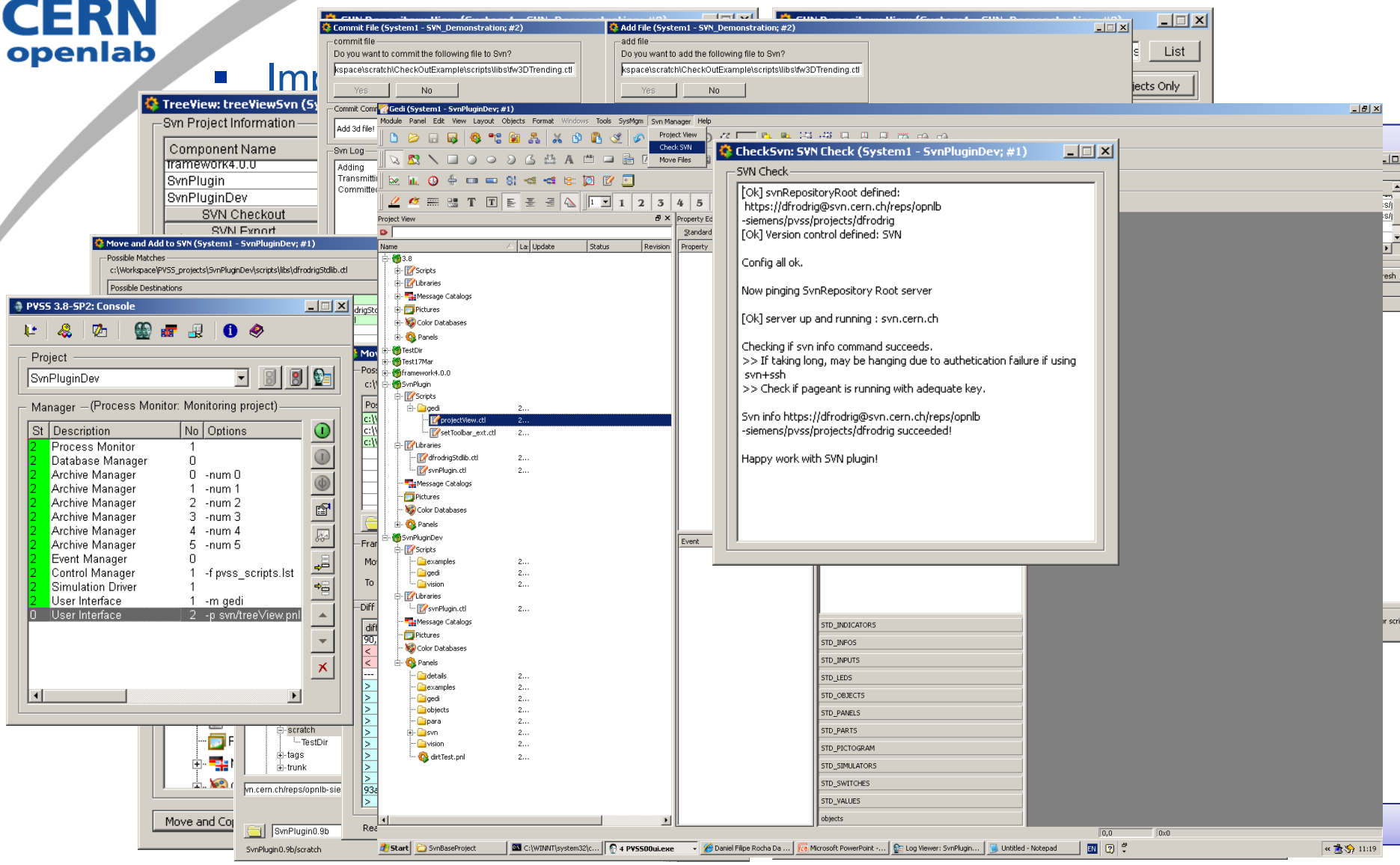
Auxiliar functions

```
dyn_string parseStatusLine( string line )
void       SVN_command( string cmd, string fileName )
void       DebugSvn( string message )
...
```

Integration into GEDI

- Objective
  - Enhance the Project View with SVN capabilities
    - Using the Ctrl SVN interface
- Limitations
  - GEDI Project View has limited hooks
    - Context Menus, toolbar menus
    - Not oriented to work with subprojects
- Solution
  - Prototype project view with panels
    - Panels are read by the UI Manager
    - We can implement additional functionalities
    - UI understands Ctrl, so re-use Ctrl SVN interface!

Integration into GEDI



Current challenges

- There are limitations to the available solution:
  - Dependencies
    - external installation of svn comand line client is required
  - Authentication should be in place
    - Pageant(Putty) + .pkp
    - https certificates
    - Password based authentication not possible
  - Resolving conflicts not trivial
    - Svn command line client requires user input
      - Ex: SVN update with conflicts
    - SvnCTRL library based on system() calls, no interaction
- Some solutions have been explored:

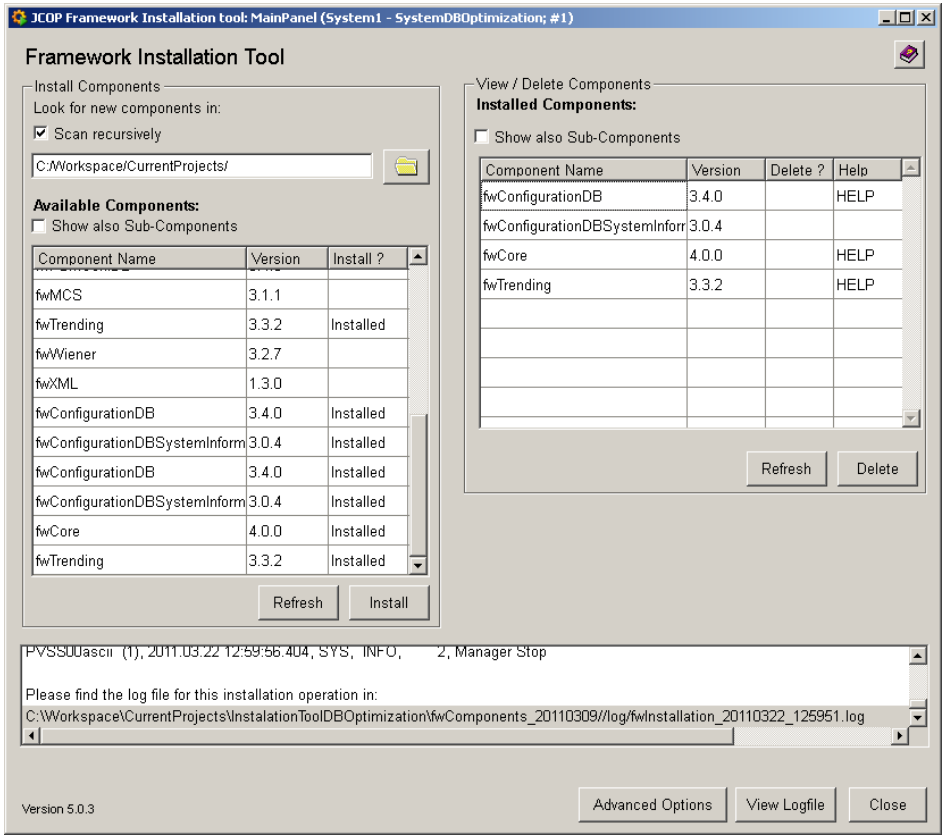
Current challenges

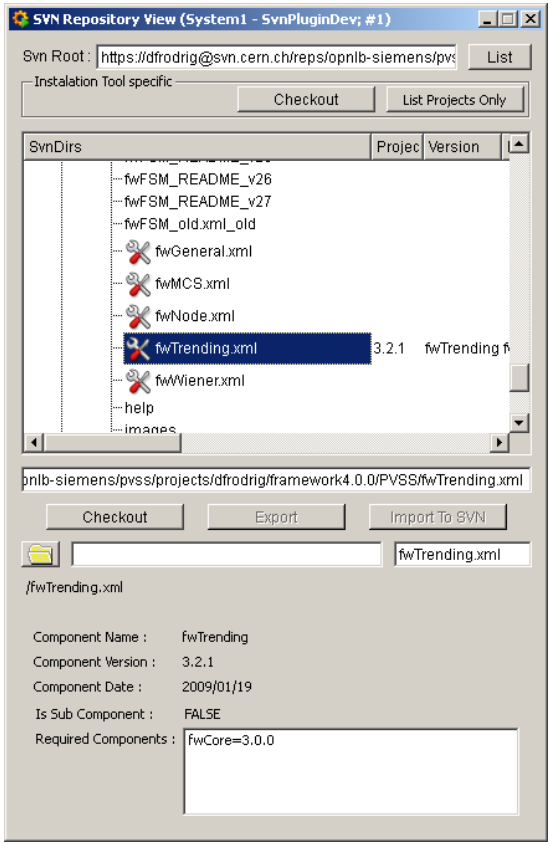
- Write a client based on the SVN C API
  - [+] could be closely binded to PVSS
  - [-] significant amount of coding involved
- Change svn command line client
  - sending callbacks into dialog boxes
  - [+] easier development, Apache License
  - [-] distribution of a specific re-compiled svn command line client required
- Change the system() call on the Ctrl code to allow callbacks
  - [+] not requiring recompilation of svn command line client and distribution
  - [-] first trials not successful due to decoupling between Ctrl and C

Current challenges

- How to use SVN more effectively in GEDI?
  - PVSS projects are developed while running!
    - internal db always changing
    - GEDI is an IDE, but at the same time, UI Manager
      - Depends on a running instance of PVSS, unlike typical IDE's
  - Notion of reusable components is limited
    - Sub projects must contain "static" files/objects
    - Associated datapoints/configs must go in running project
  - CERN developed its own approach for reusable code
    - JCOP framework / Unicos
    - Installation tool
      - Suitable for creation and deployment of components
      - able to pack/unpack code
      - run pre and post processing of datapoints/configs
      - Manages dependencies
- Integration of SVN and Installation Tool
  - Read xml descriptions from SVN repository
  - SVN checkout processes Installation Tool rules
  - SVN commit creates Installation Tool rules

Current challenges





Conclusion

- SVN can improve PVSS
  - Projects are file based as other software
  - Some specificities apply (running database)
- SVN features were integrated into GEDI
  - Using a CTRL SVN client interface
    - Based on svn command line client
    - Using PVSS's ctrl scripting language
  - Graphical tools were created
- There are some challenges to tackle
  - Enhance the Ctrl SVN library
    - Different approaches were analysed
  - Integration with the Installation Tool
    - An adequate solution for deployment
    - Would benefit CERN users directly

Conclusion

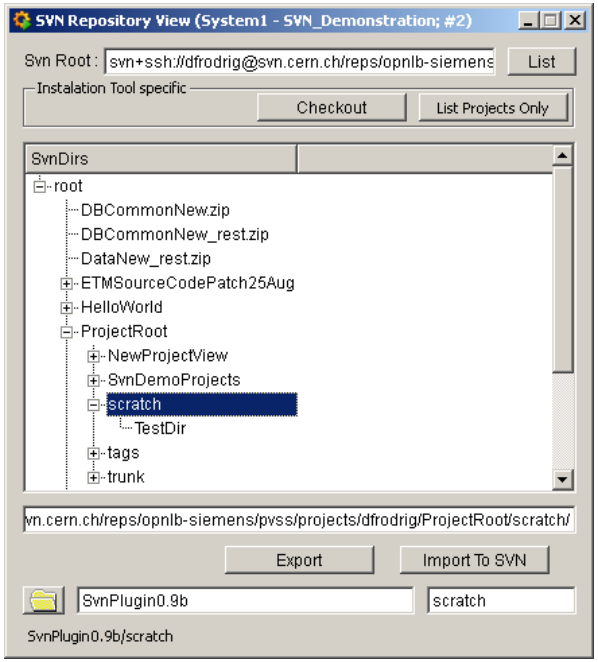
- Thank You for your attention!
- Questions?

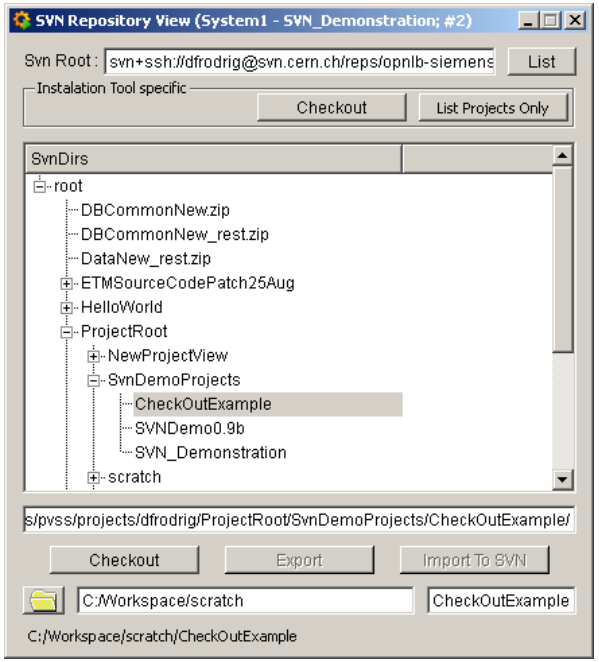
Support slides

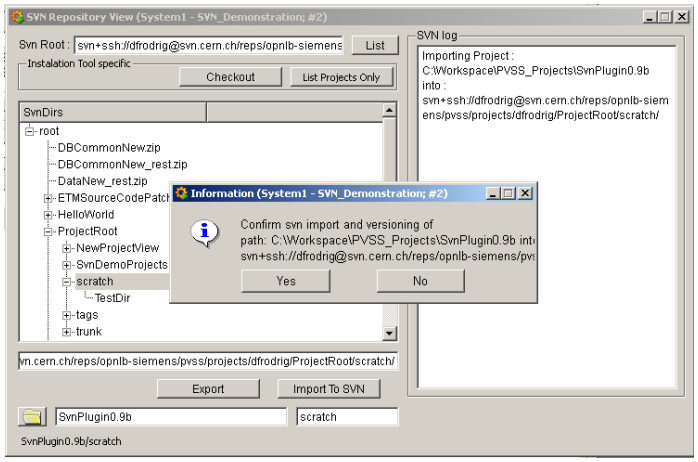
Integration into GEDI

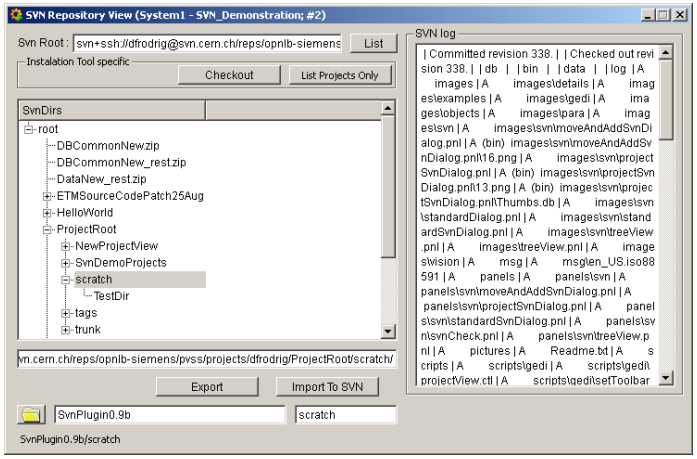
- Implementation of graphical panels
  - Tree View Panel
    - Svn Project Information
    - Project Tree
  - Svn Repository View
    - Checkout / Export / Import SubProjects
  - Standard Dialogs
    - for svn operations and comments
  - Diffing Viewers
  - SVN checking tools
    - Help users detect setup errors
  - Easy Install
    - Add sub project
    - Add config information (default svn url)
    - That's it!
      - panels, scripts, toolbar menus automatically loaded

Integration into GEDI

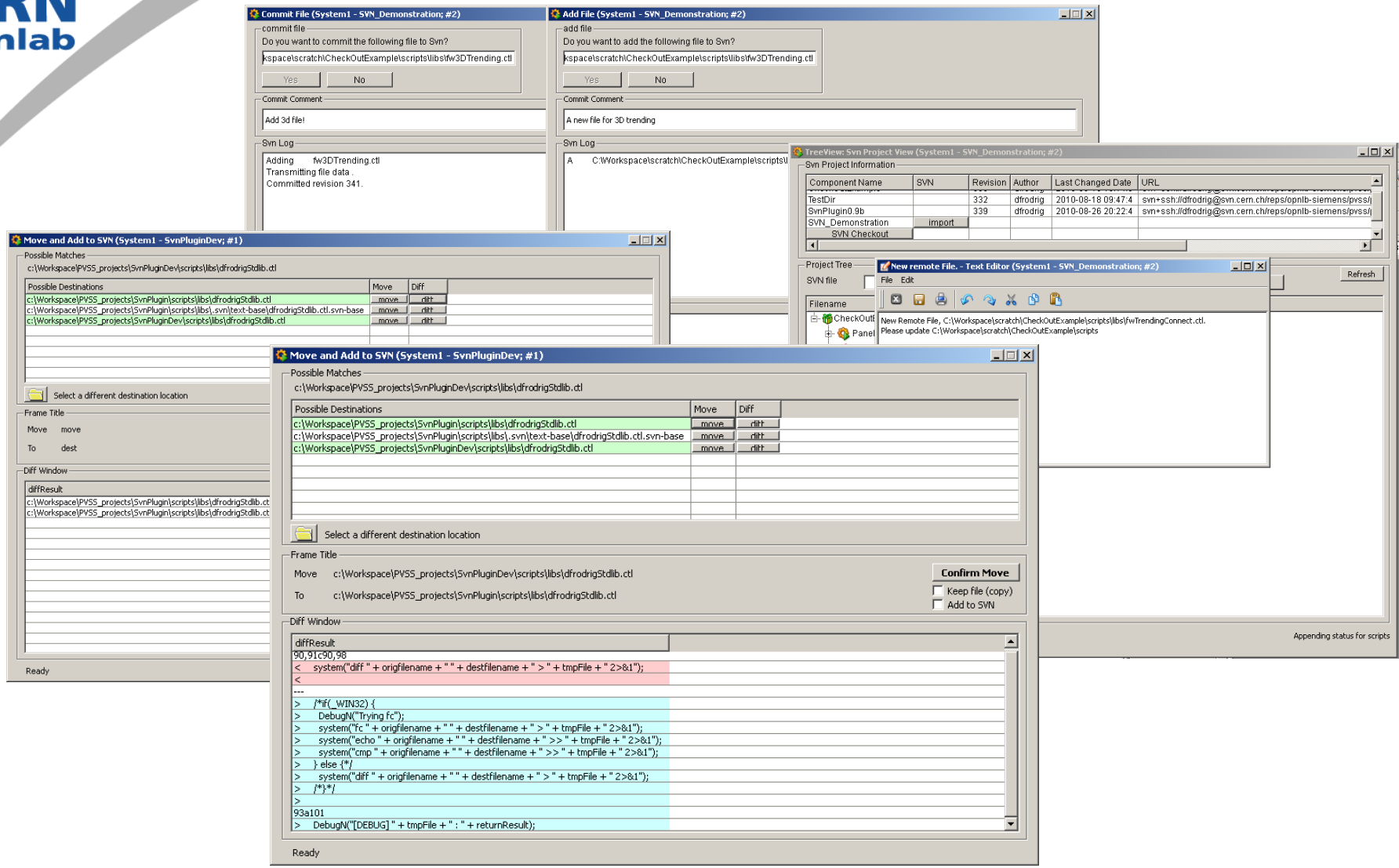
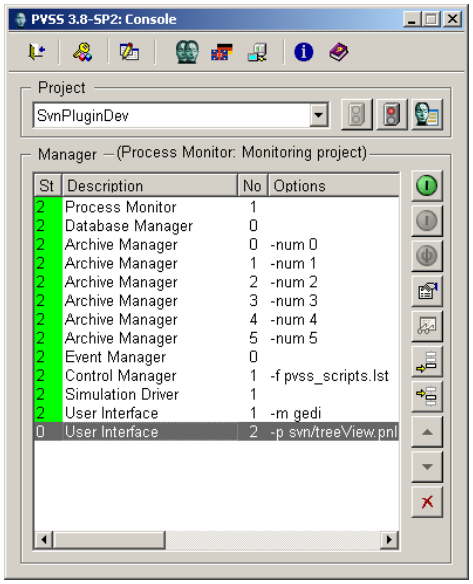
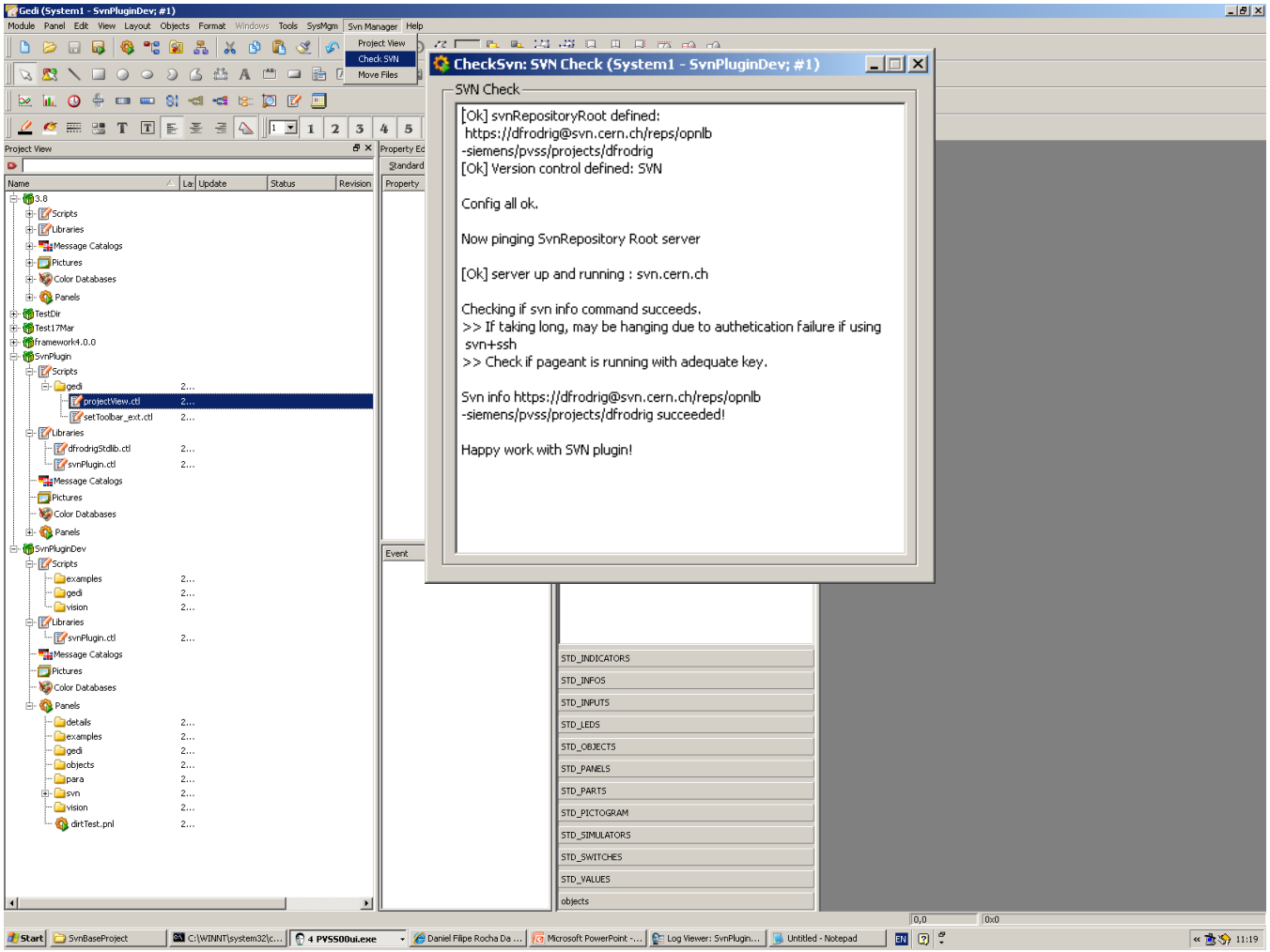








Integration into GEDI



Integration into GEDI