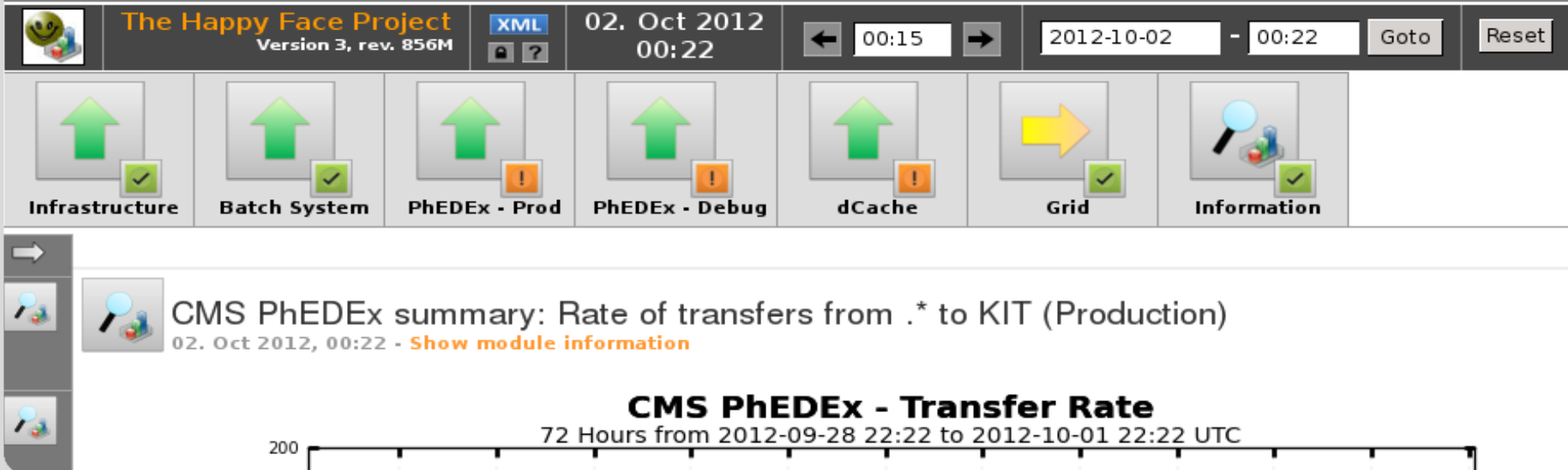


HappyFace3

everything now in Python ;-)

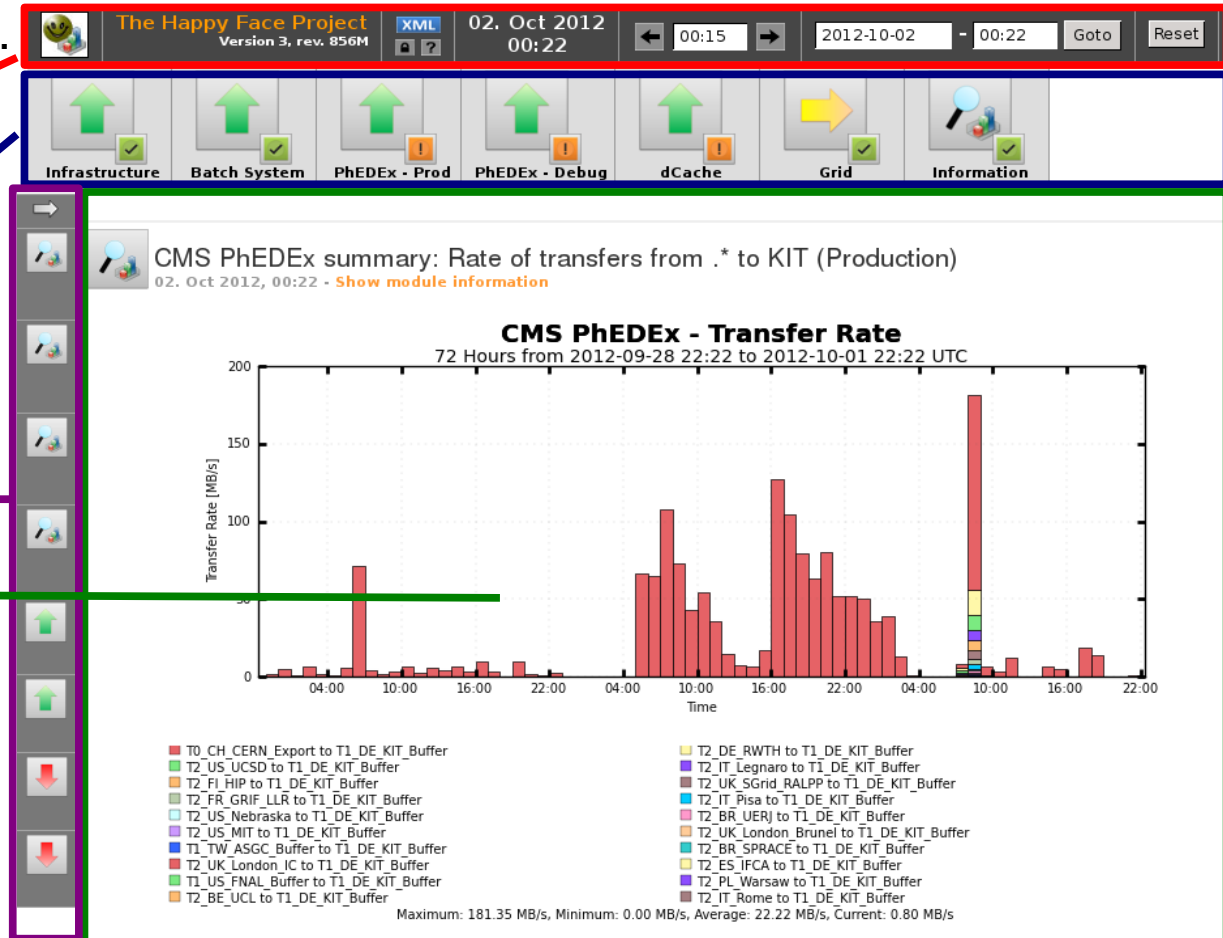
Oliver Oberst, Günter Quast, Steffen Röcker, Marcus Schmitt,
Gregor Vollmer, Stefan Wayand, Marian Zvada
 and the IEKP T1 Shifter-Team

Institut für Experimentelle Kernphysik

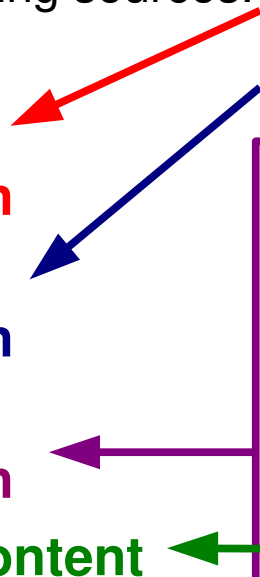


HappyFace – Introduction

- HappyFace is a **meta-monitoring framework**
 - aggregates remote and local site information from different expert monitoring sources.



- History Navigation
- Category Navigation
- Modules Navigation
- Module Content
- Simple Module and category rating system.



HappyFace – Development / Usage

■ HF core was/is developed at KIT to monitor GridKa T1 for CMS

■ Additional module development and usage by:

- DESY
- Uni Hamburg
- RWTH Aachen
- Uni Göttingen

The screenshot displays the HappyFace monitoring interface. At the top, there are several status panels for different components like 'Sam tests', 'Nagios', and 'Hardware', each with an emoji icon. Below these are panels for 'Apel Accounting for G' and 'The Happy Face Project' with various control buttons like 'RecordStart', 'Record', 'SAM', and 'CMS User'. A central panel shows 'Infrastructure', 'Batch System', 'PHEdEx - Prod', 'PHEdEx - Debug', 'dCache', 'Grid', and 'Information' with status indicators. The bottom part of the screenshot features a 'CMS PhEDEx summary' and a 'CMS PhEDEx - Transfer Rate' graph showing transfer rates in MB/s over a 72-hour period. A legend below the graph lists various data sources and their corresponding colors.

DE_DESY	Total
70(515.64)	134.10(515.64)
.78(2.15)	62.16(2.15)
2.48	92.15(0.51)
78(17.29)	64.88(104.01)
71.48	129.31(32.16)
92(145.49)	144.92(154.13)
.85(3.65)	22.32(6.80)
.88(1.39)	46.88(1.39)
86(685.62)	696.72(816.80)

■ **Version 2 in production for over 3 years**

- Very Reliable
- Main tool for T1_DE_KIT shifters and experts
- Used by several sites in Germany (ATLAS+CMS)
- Trial usage by CMS for batch system monitoring of all T1 sites

HappyFace - why do we need a new Version



■ HappyFace 2:

- **Python code with embedded PHP output code.**
- **Structural flaws**
 - Software packaging/release management difficult due to HF2 module/configuration directory structure
 - Some functionality implemented late, not fully integrated in the core (e.g. certificate authentication)
→ **Full HappyFace rewrite required to address all issues**

■ HappyFace 3:

- **Python-only HappyFace core and module code**
 - Uses Python template engine (HTML templates) and CherryPy to replace embedded PHP output code
 - Developers only need to code Python and create HTML templates
→ **Less and simpler code, faster to debug**
 - **Clear separation of configuration and modules code directories**
 - Code documentation in Doxygen style (Sphinx) from the start of development.
 - **Addresses most design flaws of version 2**

HappyFace 3 in Detail

- Started development of HF3 to apply experience gained over the years.
- Pure Python code (CherryPy) for functionality, web output separately designable with HTML templates (python-mako)
- Clearly separated file hierarchy of modules, core and configuration
 - Eases software packaging and release management
- Improved Networking:
 - Categories loaded on separate pages
 - On-demand reloading of details
 - Improved performance on mobile devices
- New Database access scheme
 - Well defined relationships
 - SQLite, PostgreSQL, Oracle, ...
 - Easy automated maintenance
- Support for tool-scripts
 - Database migration / updates
 - Interactive HappyFace shell
 - ...
- New interactive plot generator
- Thorough certificate authorization
 - Weboutput, archived files, plot generator, XML feed

HappyFace 2 vs 3

Code example: Module - CMSPhedexErrorLog



■ HappyFace 2:

■ Python + PHP inline:

■ **602 LOC**

■ HappyFace 3:

■ Python: **208 LOC**

■ **HTML template: 102 LOC**

HappyFace 2 vs 3

Code example: Module - CMSPhedexErrorLog



HappyFace 2:

- Python + PHP inline:

- 602 LOC**

HappyFace 3:

- Python: **208 LOC**

- HTML template: **102 LOC**

This screenshot shows a complex code file for HappyFace 2, where Python and PHP code are mixed together in a single file. The code is dense and difficult to read due to the interleaving of different programming languages.

This screenshot shows the HappyFace 3 code structure, where Python code and HTML templates are separated into distinct files. This separation makes the code more organized and easier to maintain.

**Up to 50%
module code
reduction!!**

HTML:

HappyFace 3 - Documentation

- HappyFace 3 documentation uses Sphinx to generate the code documentation site from in-line comments:

HappyFace v3.0 RC1 documentation »

Table Of Contents

- HappyFace - The Meta Monitoring Framework
 - Documentation Todos
 - Indices and tables
- Next topic**
- 1. Basic Operation and Concepts
- This Page**
- Show Source
- Quick search**

Go

Enter search terms or a module, class or function name.

HappyFace - The Meta Monitoring Framework

- Powerful site specific monitoring system for data from multiple input sources.
- Collects, processes, rates and presents all important monitoring information
- for the overall status and the services of a local or Grid computing site.
- Monitoring data is subdivided in multiple categories.
- Each category is subdivided in multiple modules which corresponds to one single test.
- Simple rating system: -1 = no info / error; status float value = 0.0 .. 1.0 (critical .. fine)
- The overall status of the categories can be calculated from the individual module statii with different algorithms.

- 1. Basic Operation and Concepts
 - 1.1. HappyFace Workflow
 - 1.2. Description of HappyFace Parts
- 2. Installation
 - 2.1. Dependencies
 - 2.2. Getting the Source
 - 2.3. Configuration
 - 2.4. Running HappyFace in an Development Environment
 - 2.5. Setting up HappyFace with Apache2 and mod_wsgi
 - 2.6. Certificate Authorization with Apache2
- 3. Configuration and Site Maintenance
 - 3.1. Core Configuration
 - 3.2. Logfiles
 - 3.3. Modules and Categories
 - 3.4. Updating the Site
 - 3.5. Certificate Authorization
- 4. Module Development
 - 4.1. Module Basics
 - 4.2. Module Class Reference
 - 4.3. HTML Templates, Generating Output
 - 4.4. Step-by-Step Guide
- 5. Core Documentation
 - 5.1. **hf.auth** - Certificate Authorization
 - 5.2. **hf.category** - Category Management and Weboutput
 - 5.3. **hf.configtools** - Config Parser and Startup
 - 5.4. **hf.database** - Database Setup
 - 5.5. **hf.dispatcher** - Root CherryPy Dispatcher
 - 5.6. **hf.downloadservice** - Downloads and Archive

<http://ekphappyface.physik.uni-karlsruhe.de/~happyface/docs/>

HappyFace 3 Release Plan / Status

- Core development almost finished
- RC1 is ready
 - Current work package: improve documentation, minor features, fix errors...
 - Testing phase: Is used by T1_DE_KIT shifters

Milestone: 3.0 RC1

3 months late (09/12/12)



Closed tickets: 30 Active tickets: 0 / Total tickets: 30

First release candidate of HappyFace? 3.0. Ready for testing within the T1_DE_KIT monitoring.

■ Planned stable release: last Friday

Milestone: 3.0

Due in 30 hours (11/30/12)



Closed tickets: 2 Active tickets: 8 / Total tickets: 10

Release of HappyFace? 3.0. A fully rewritten version using cherry.py.

<https://ekptrac.physik.uni-karlsruhe.de/trac/HappyFace/roadmap>

HappyFace 3 Open Issues

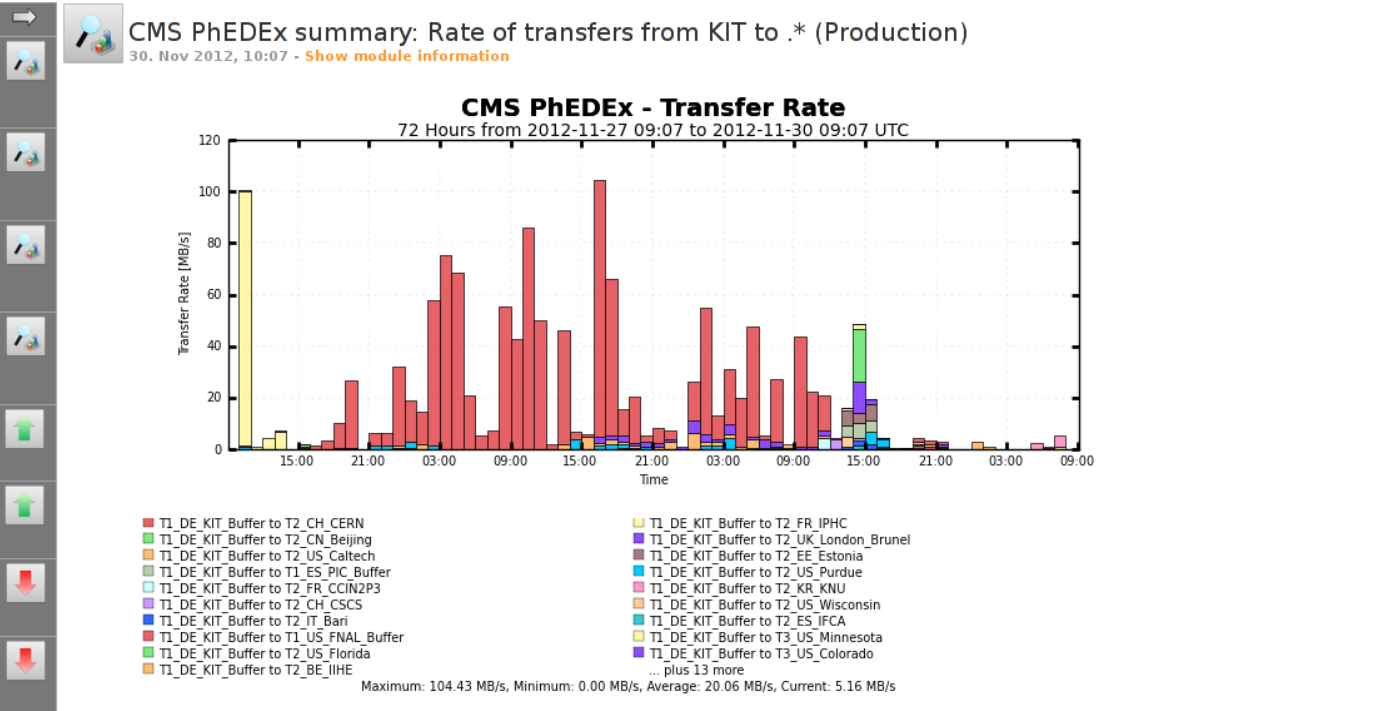
- 2 open module tickets:
 - Review module code
 - Configuration issues in Multiplot module
- 7 open core tickets:
 - Configuration Generator
 - Log rotation and compression of HappyFace logs
 - Documentation updates (two tickets)
 - Database cleanup tool
 - Bug in internal plot generator

HappyFace 3



The Happy Face Project Version 3, rev. 886M XML 30. Nov 2012 10:07 00:15 2012-11-30 - 10:20 Goto Reset

News Infrastructure Batch System PhEEx - Prod PhEEx - Debug dCache Grid Information



PhEEx Stats Production - KIT (Incoming)
30. Nov 2012, 10:07 - Show module information

Start Time Thu Jan 1 00:59:59 1970

<http://ekphappyface.physik.uni-karlsruhe.de/HappyFace/gridka>

Conclusion/Outlook

- HappyFace 2:
 - **Very successful veteran in meta-monitoring used at many sites**
 - **Full HappyFace rewrite was needed to address all issues which accumulated over the years of production usage/development.**

- HappyFace 3:
 - Less and simpler code for module developers
 - **Testing phase started in November**
 - **Release as soon as all major issues are fixed**
 - **Aachen installed HF3 last week**
 - **External non-KIT module developers invited to try it out and start porting of site specific modules, we need your experiences**