# Documentation and Example Codes

Chapel Team, Cray Inc.
Chapel version 1.15
April 6, 2017

## Safe Harbor Statement

## Outline

- Users Guide Improvements
- Issue Tracking
- Quickstart
- Documentation Hierarchy
- Other Documentation Improvements
- Example Code Improvements

## Users Guide Improvements

# Users Guide Improvements

## Background:

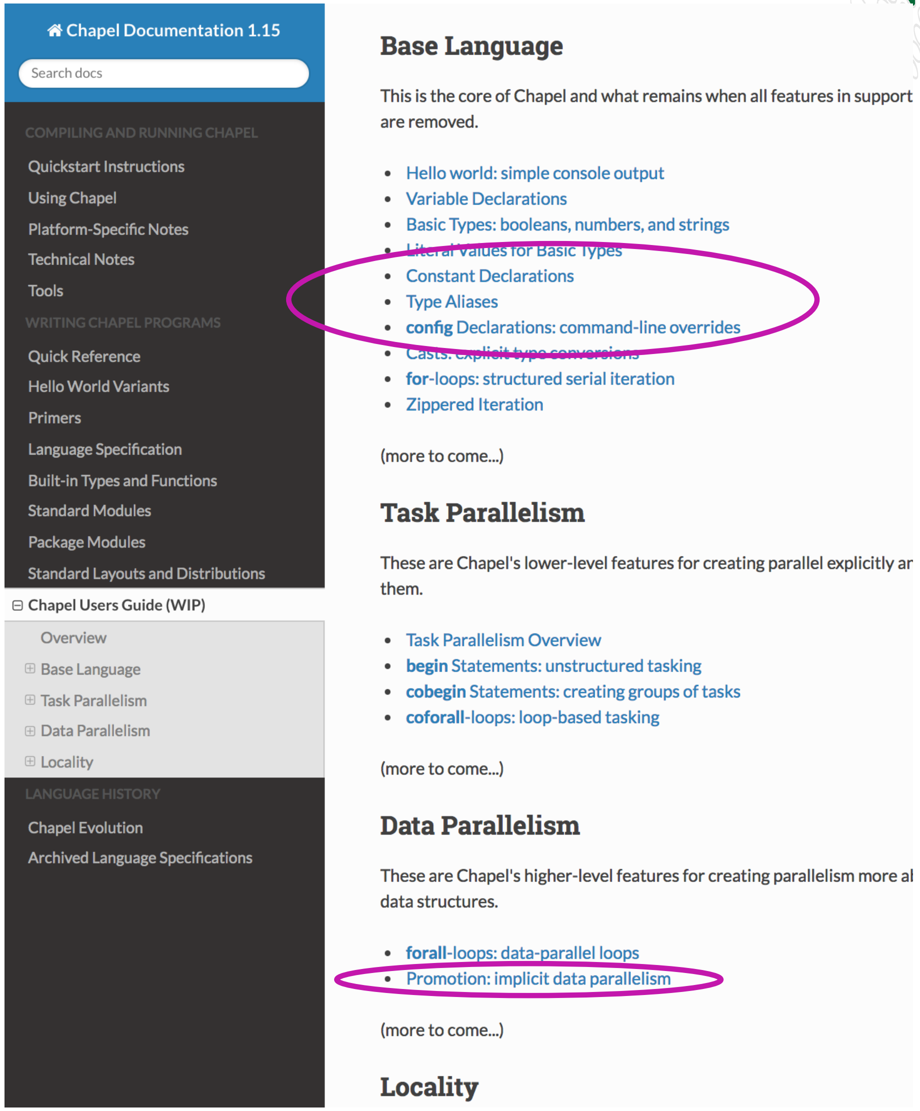
- Chapel 1.13 started including sections for a new users guide
  - each subsequent release has included additional sections

## This Effort:

- added new sections for:
  - promotion
  - constants ('const' and 'param')
  - type aliases
  - 'config' declarations
    - 'const', 'var', 'type', 'param'

## Next Steps:

- continue expanding coverage



**Base Language**

This is the core of Chapel and what remains when all features in support are removed.

- Hello world: simple console output
- Variable Declarations
- Basic Types: booleans, numbers, and strings
- Literal Values for Basic Types
- Constant Declarations
- Type Aliases
- config Declarations: command-line overrides
- Casts: explicit type conversions
- for-loops: structured serial iteration
- Zippered Iteration

(more to come...)

**Task Parallelism**

These are Chapel's lower-level features for creating parallel explicitly ar them.

- Task Parallelism Overview
- begin Statements: unstructured tasking
- cobegin Statements: creating groups of tasks
- coforall-loops: loop-based tasking

(more to come...)

**Data Parallelism**

These are Chapel's higher-level features for creating parallelism more al data structures.

- forall-loops: data-parallel loops
- Promotion: implicit data parallelism

(more to come...)

**Locality**

## Issue Tracking

## Issue Tracking: Background

- **Chapel's supported bug-reports/tracking in several ways:**
  - chapel-bugs@lists.sourceforge.net
  - JIRA
    - read-only for non-core developers
  - Bugzilla
    - Cray customers
  - STATUS.devel / future tests in repository

- **The options for reporting bugs were lacking**
  - Typical users could only report bugs via mailing list
    - Required transcription over to JIRA
  - More advanced users could open a PR for a future test
  - Cray users could use Bugzilla if they preferred
    - Required transcription over to JIRA when appropriate

## Issue Tracking: This Effort

- **Enabled GitHub Issues for the chapel-lang/chapel repo**
- **GitHub offers many advantages as an issue-tracker:**
  - Integration with existing GitHub repository
    - Cross-referencing Issues / PR
    - Tagging GitHub IDs
  - Low barrier of entry
    - GitHub has a large user-base already familiar with Issues
  - High visibility
    - Easily discovered from the chapel-lang/chapel repository
  - Batteries included
    - Very little setup overhead necessary
    - Has almost all the features we need out of the box
- **Started migrating JIRA issues and STATUS.devel**

## Issue Tracking: Impact

- **Increased transparency**
  - More discussions are happening in public via Issues
    - Increasing the involvement of non-core developers in discussions
  - Issues are also tracking tasks, illuminating future plans for the project

- **Better archiving**
  - GitHub search has been a vast improvement for our team
    - Compared to JIRA and the mailing list archives
  - Finding duplicates and cross-referencing them is much easier

- **‘easy’ labels provide starting point for new contributors**
  - this year’s Google Summer of Code applicants have used these

# Issue Tracking: Status and Next Steps

## Status: Active and well-received by team and community

- Most developers are finding it more productive
- Total JIRA issues: **297**
  - Started March, 2015
- Total GitHub issues: **260**
  - Started December, 2016

## Next Steps: Improve processes surrounding issue-tracking

- Migrate remaining public issues from JIRA and STATUS.devel file
- Possibly discontinue using JIRA in favor of private GitHub Issues
- Set up a long-term strategy for detecting new user-issues
- Continue to tune process
  - Milestones
  - Labels
  - Associating futures with issues

## Quickstart

# Quickstart

## Background: QUICKSTART.rst was a top-level document

- Contained many sphinx cross-references that looked odd on GitHub
- Complicated docs-build, since we included it in the html-docs
  - ‘make docs’ resulted in copying the file into the docs-directory
  - Relative file paths lost context in the docs-directory version

## This Effort: QUICKSTART.rst moved into the doc-directory

- The top-level README points to the online URL (preferred)
- The doc-level README points to restructured text file path

## Impact: Improved QUICKSTART.rst and simplified docs build

- Users are now consistently pointed to the preferred online version
  - Users are never pointed to the GitHub-rendered version
- QUICKSTART.rst does not get moved or copied in ‘make docs’

## Documentation Hierarchy

## Documentation Hierarchy

### Background: Documentation hierarchy was asymmetric

- In release, HTML documentation did not mirror text-documentation

```
doc/
├── *.rst
├── platforms/
├── technotes/
└── html/
```

- Also, developer and user views of documentation hierarchy differed
  - GitHub / developer:

```
doc/
├── release/          // things intended for the release
│   └── foo.rst
├── developer/        // things not intended for the release
└── …/                // etc.
```

  - Release / user:

```
doc/
└── foo.rst           // release contents moved to top-level
```

## Documentation Hierarchy

### This Effort: Documentation hierarchy is now symmetric

- HTML documentation mirrors text-documentation
- Developer view is now consistent with user view

```
doc/
├── README.rst
├── rst/
│   └── foo.rst
└── html/
    └── foo.html
```

### Impact: Documentation is more organized and consistent

## Other Documentation Improvements

## Other Documentation Improvements

- **Online documentation**
  - bugs.rst now refers users to GitHub Issues
  - Improved Docker README information
  - Reorganized and categorized platform-specific documentation
  - Unified references to download.html rather than install.html
  - Improved some entries in “Quick Reference” document
- **Library documentation**
  - Documented ‘dim()’ and ‘dims()’ on arrays
  - Fixed ‘string.strip()’ documentation
  - Updated documentation w.r.t. memory management of files/channels
  - Added indication that IO module is used by default

## Example Code Improvements

# Example Code Improvements

- **learnChapelInYMinutes**
  - Added examples, fixed an incorrect one
  - Various comment, style, and formatting improvements
- **Updated example codes to:**
  - Use new ‘RandomStream()’ initializer
  - Use ‘deinit()’ rather than destructors
  - Reflect better const-checking
  - Reflect that ‘Barrier’ is now a record
- **Makefiles**
  - Fixed parallel build of examples directory
  - Fixed FFTW build target
  - Added missing Makefile to examples/patterns directory

## Legal Disclaimer

CRAY CHAPEL

CRAY

THE SUPERCOMPUTER COMPANY