

Design Document Template

1 Motivation

As Drill design grows more complex we would like to make sure that the designs of new features go thru a more rigorous process to ensure quality of design and implementation. In addition, documenting the design makes it easier for future contributors to easily understand the motivations behind, and the details of, the implementation of particular features. While the Concept Document [1] provides details of the feature from the end user perspective, the design document provides details from the development perspective.

The motivation section **MUST** refer to a JIRA that is specified in the Apache JIRA system.

The motivation section describes the problem statement for this feature. This could be a general problem statement and if the design doc is addressing a specific aspect of the bigger problem it should be stated here. For example, "*this document focuses on setting an upper bound on memory used per query*".

Example:

1 Motivation

[DRILL-263333] Drill needs coffee.

The Drill development and user communities have discovered that consuming coffee while using Drill leads to great improvement in Drill user performance. In particular, it is believed that coffee will allow users to run many queries in parallel by using multiple instances of sqlline. The perceived query completion time is also expected to improve. Since Drill is an extensible execution engine, this feature proposes to implement a Drill coffee plugin that will prepare coffee for the end user via a SQL interface.

2 Requirements

The design document **MUST** be in English.

The design document **MUST** be detailed and contain implementation details. The section on Implementation details provides more information on what the document **SHOULD** contain.

The design document **SHOULD** use the keywords specified in RFC 2119 [2] to describe requirements.

The Requirements section contains the specific requirements that **MUST** be satisfied by the design document and **SHOULD** include a subsection that includes use cases.

The design document SHOULD provide specific use cases that will be addressed by the implementation. A use case will be a concrete example of the feature as it will be used. The description of the use case MAY include details from customer cases.

2.1 Use Cases

The Design Document Template is a general purpose document that can be used for any software design document. The most common use case for the template is going to be the design documents for Drill.

The design document MUST contain use cases where relevant. Design reviews SHOULD check if the design adequately covers the use cases.

Example:

2 Requirements

The Drill coffee module must be designed to satisfy the following requirements -

A user should be allowed to brew more than one cup of coffee in a single query.

The coffee storage plugin will provide access to two tables : coffee and beans.

A user can brew a cup of coffee by writing the appropriate query using the two tables.

The module should allow user defined coffee recipes. This should be possible at any time after a drill cluster has started up. It should not be necessary to restart drillbits to add new coffee types.

2.1 Use Cases

The following canonical query will brew one cup of the Drill team's favourite coffee -

```
Select cappuchino from coffee, beans where coffee.bean = beans.id and beans.name = 'Java' and coffee.name = 'Cappuchino' limit 1
```

3 Background and Research

This section is OPTIONAL.

The background section of a design document will contain any background material that may be needed to understand the design. Research may be summarised, and results of prototypes and experiments may be listed here.

Example:

3 Background

Coffee can refer to a drink made from the coffee bean or to the bean itself. In the context of the Drill Coffee storage plugin the term coffee will refer to the drink.

The design and implementation of the coffee storage plugin is based on techniques and methods described in [3], and [4]. The American Brewing Standard [5], describes the basic brewing control chart for maximum efficacy in brewing coffee.

4 Design Overview

This section is REQUIRED.

The design document must provide a high level design overview. Details of actual implementation can be covered in the following section on Implementation. An architecture picture of the design would be useful here.

This section SHOULD also include alternative design ideas and explain why the particular design was chosen. If the design is not finalized and some initial feedback is required, this section SHOULD state that.

This section SHOULD mention the changes to any external interfaces, APIS, and user settable options.

This section SHOULD also mention the limitations of the feature.

Example

4 Design Overview

The Coffee module will be implemented as a storage plugin and will follow the standard design for a storage plugin. The plugin will support filter pushdown and ensure that the coffee generated is appropriately filtered.

5 Implementation Details

This section is REQUIRED

The following sections are a list of items the design must consider. If relevant, the design document MUST provide details. Some of the sections, though REQUIRED, may not be relevant in which case the document MAY include a line that says the section is not relevant.

5.1 Algorithms (and Data Structures)

This section is REQUIRED.

This section

- SHOULD describe the main approach or algorithm and any special data structures to be used.
- SHOULD specify any architectural patterns used (e.g. sync/async).

- SHOULD provide a state diagram or sequence diagram.
- MUST Include what components are affected.
- SHOULD include any inter-dependencies between components
- SHOULD state any assumptions on setup/shutdown of components

5.2 APIs and Protocols

This section documents changes to any APIs or to the RPC protocols.

Public API changes MUST be documented here with complete signatures and descriptions.

Internal API changes of core components MUST be documented here.

5.3 Performance

This section is REQUIRED.

This section SHOULD consider the improvements expected or any possible impact on response time, resources (cpu, memory, network) usage, and concurrency

5.4 Error and Failure handling

This section is REQUIRED

This section

- SHOULD specify specific handling of failure of a drillbit or hardware
- SHOULD specify handling of normal or common failures like network errors, and file system errors
- SHOULD specify error handling of invalid input, and user errors
- SHOULD specify errors propagated to end user

5.5 Deployment

This section is OPTIONAL

This section SHOULD specify how the feature is deployed. In particular the design document must specify if there are any steps that are needed to upgrade as well as any steps needed for a new install.

5.6 Memory management

This section is REQUIRED

The design document MUST include the impact of the design on Drill's memory usage. In particular, any increase in either Heap or Direct memory settings or usage should be introduced if and only if there is no alternative or if the feature is not possible without it.

5.7 Availability Implications

This section is OPTIONAL

This section SHOULD specify the impact of the feature on the availability of Drill, if any.

5.8 Scalability Issues

This section is REQUIRED

This section SHOULD specify the impact of the feature on the scalability of Drill. In particular, any feature that introduces a blocking operation in the execution pipeline or adds to planning time SHOULD consider the impact on scalability.

5.9 Backward Compatibility

This section is OPTIONAL

In the case of changing APIs or any end user facing behavior, the design document MUST consider providing backward compatibility.

For internal APIs and the RPC protocol, considering backward compatibility is REQUIRED as it is a goal to provide rolling upgrades for the Drill backend.

5.10 Security and Authentication impact

This section is OPTIONAL

The design document MUST include the security and authentication implications of any feature that affects the way data is accessed.

The document MUST include the security impact of any changes to the RPC protocol between the user and the server

5.11 UI changes

This section is OPTIONAL

This section SHOULD include any changes to the Web UI or to the command line.

5.12 Options and metrics

This section is OPTIONAL

This section SHOULD include any new user configurable options and any new metrics added by the feature.

5.13 Debugging

This section is OPTIONAL

5.14 Testing implications

This section is REQUIRED

This section SHOULD specify at a high level what are the functional testing and performance testing implications. Something to give a QA person an idea of what to start thinking about.

5.15 Tradeoffs and Limitations

This section is REQUIRED.

The design document must document the tradeoff being made as a result of the choice of design. The document must also specify the limitations of the design and must clearly state what the implementation will not do.

Example

5 Implementation Details

The coffee plugin is a standard plugin that also supports filtering.

5.1 Algorithm

The coffee plugin will use one of two methods to brew the coffee. If a filter condition is pushed down, the coffee will be brewed using a standard drip brew method. If there is no filter, then the higher speed espresso method will be used. The decision will be made by the plugin. The coffee bean data will primarily be kept in a standard *Coffea Arabica* tree structure [6].

5.2 APIs and protocols

There are no changes to any internal or external APIs. There is no change to any RPC protocol

5.3 Performance

As described in 5.1, the performance is expected to depend on the query. We currently do not have a baseline for the expected performance. The initial implementation will therefore establish a baseline for future releases.

5.4 Error and Failure Handling

The coffee module will always produce coffee. Really. Except when it runs out of beans or water or milk. But then you can't have everything. In future releases the plugin will order more beans or download water and milk from the web (the user might have to wait a bit though).

5.5 Deployment

The coffee plugin will be disabled by default and will have to be enabled explicitly. There are no upgrade requirements and no additional work needs to be done to deploy the coffee storage plugin.

5.6 Memory management

The coffee plugin needs no memory.

5.7 Availability

The coffee module is available only during working hours.

5.8 Scalability

The coffee preparation module is a blocking operation. The module scales as the size of the cluster increases, but will limit the number of other Drill operations (as everyone waits) while the coffee is being brewed.

5.9 Tradeoffs and Limitations

There are no tradeoffs and limitations of the coffee plugin.

5.16 Other Items

Other Subsections MAY be added to this section, as needed

6 Implementation Plan

For complex features that require many subtasks, this section SHOULD include a breakdown of the stages in which the implementation will be done. The stages SHOULD include a high level breakdown of the tasks.

6 Example

1. Initial coffee plugin implementation without filtering
2. Filter pushdown
3. Failure conditions
4. Performance testing

7 Open items

The design document MAY have a section that lists any items that have not been resolved. The unresolved items MAY include requirements that need clarifications or design elements where additional research or feedback from reviewers is desired.

Example:

7 Open Items

The coffee module does not have the support for fair trade coffee. It is not clear how this can be supported as the module has no control over the actual beans provided by the end user.

8 Notes

The Design Document Template is self-referential and is itself a Design Document. This note is also self referential.

Example:

8 Notes

Many coffees have fruity notes, which is not surprising considering that coffee beans are seeds of a fruit (coffee cherries)

9 References

[1] <Link to Concept Document Template>

[2] <https://www.ietf.org/rfc/rfc2119.txt>

[3] https://en.wikipedia.org/wiki/Coffee_preparation

[4] Coffee: Recent Developments; Ronald Clarke and O. G. Vitzthum; April 15, 2008; John Wiley & Sons [ISBN: 9780470680216]

[5] Brewing: American Standard: <http://www.mountaincity.com/brewing-1.html>

[6] <http://www.coffeeresearch.org/agriculture/coffeeplant.htm>

10 Document History

Date	Author	Version	Description
2016-06-13	Parth S. Chandra	0.1	Initial Draft