

# An Asynchronous Call Graph for JavaScript

Dominik Seifert  
d01922031@ntu.edu.tw  
National Taiwan University  
Taiwan, Taipei

Jane Hsu  
yjhsu@csie.ntu.edu.tw  
National Taiwan University  
Taiwan, Taipei

Michael Wan  
b07201003@ntu.edu.tw  
National Taiwan University  
Taiwan, Taipei

Benson Yeh  
pcyeh@ntu.edu.tw  
National Taiwan University  
Taiwan, Taipei

## CCS CONCEPTS

• Software and its engineering → Concurrent programming structures.

### ACM Reference Format:

Dominik Seifert, Michael Wan, Jane Hsu, and Benson Yeh. 2022. An Asynchronous Call Graph for JavaScript. In ?. ACM, New York, NY, USA, 1 page. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

TODO: Move missing sections into the technical report and clean it up a little.

## 1 CROSS THREAD DATA DEPENDENCIES

### 1.1 Promise

```
▼ CrossThreadDataDependencies
  producing = 0 producer_consumer_base.js:39
  lastProducingItem = 0 producer_consumer_base....
  buffer producer_consumer_base.js:47
  key seedrandom.js:183
  producingBuffer producer_consumer_base.js:113
  consumerQueue = [] producer_consumer_promis...
  nItems = 0 producer_consumer_base.js:35
  consumingBuffer producer_consumer_base.js:69
  consuming = 0 producer_consumer_base.js:38
  producerQueue = [] producer_consumer_promise...
```

Figure 1: Cross Thread Data Dependencies-Promise

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
ICSE SEIP '22, ?, ?

© 2022 Association for Computing Machinery.  
ACM ISBN ?...\$15.00  
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

### 1.2 Async

```
▼ CrossThreadDataDependencies
  producing = 0 producer_consumer_base.js:39
  lastProducingItem = 0 producer_consumer_base....
  buffer producer_consumer_base.js:47
  key seedrandom.js:183
  producingBuffer producer_consumer_base.js:113
  consumerQueue = [] producer_consumer_async.j...
  nItems = 0 producer_consumer_base.js:35
  consuming = 0 producer_consumer_base.js:38
  consumingBuffer producer_consumer_base.js:69
  producerQueue = [] producer_consumer_async.js:...
```

Figure 2: Cross Thread Data Dependencies-Async

### 1.3 Callback

```
▼ CrossThreadDataDependencies
  producing = 0 producer_consumer_base.js:39
  lastProducingItem = 0 producer_consumer_base....
  buffer producer_consumer_base.js:47
  key seedrandom.js:183
  producingBuffer producer_consumer_base.js:113
  nItems = 0 producer_consumer_base.js:35
  consuming = 0 producer_consumer_base.js:38
  consumingBuffer producer_consumer_base.js:69
```

Figure 3: Cross Thread Data Dependencies-Callback

## REFERENCES