

```

D4D4D4;--ch-t-background: #1E1E1E;--ch-t-lighter-
inlineBackground: #1e1e1ee6;--ch-t-editor-background:
#1E1E1E;--ch-t-editor-foreground: #D4D4D4;--ch-t-editor-
rangeHighlightBackground: #ffffff0b;--ch-t-editor-
infoForeground: #3794FF;--ch-t-editor-
selectionBackground: #264F78;--ch-t-focusBorder:
#007FD4;--ch-t-tab-activeBackground: #1E1E1E;--ch-t-
tab-activeForeground: #ffffff;--ch-t-tab-
inactiveBackground: #2D2D2D;--ch-t-tab-
inactiveForeground: #ffffff80;--ch-t-tab-border: #252526;--
ch-t-tab-activeBorder: #1E1E1E;--ch-t-editorGroup-
border: #444444;--ch-t-editorGroupHeader-
tabsBackground: #252526;--ch-t-editorLineNumber-
foreground: #858585;--ch-t-input-background: #3C3C3C;-
-ch-t-input-foreground: #D4D4D4;--ch-t-icon-foreground:
#C5C5C5;--ch-t-sideBar-background: #252526;--ch-t-
sideBar-foreground: #D4D4D4;--ch-t-sideBar-border:
#252526;--ch-t-list-activeSelectionBackground: #094771;--
ch-t-list-activeSelectionForeground: #ffffffe;--ch-t-list-
hoverBackground: #2A2D2E; }

```

On-chain Tracking

We aim to understand better and recognise our key contributors who are driving the adoption of smart accounts within our ecosystem.

Implementing a Safe on-chain identifier enables tracking of complex data, such as whether a Safe transaction is executed via our SDK or another, whether it originates from a platform like a Safe App or widget (for example, the CoW Swap widget in our Safe interface), the tool version, the project, and more.

By submitting your on-chain identifier through the form provided at the end of this page, you will help us accurately attribute activity and allow us to return value to our Ecosystem Partners in the future.

â”‚, On-chain tracking is supported starting from [Protocol Kit](#) v5.2.0 and [Relay Kit](#) v3.4.0 .

On-chain identifier format

The identifiers used to track Safe deployments and transactions are 50 bytes in length and follow the format below:

Safe 00 6363643438383836663461336661366162653539 646561 393238 653366

Check the last 50 bytes of the data field in this [example transaction \(opens in a new tab\)](#) to see how the identifier appears after the transaction is executed.

Prefix hash

- Type:

- 2 bytes
- Example:
- 5afe

Static prefix to identify the Safe on-chain identifier.

Version hash

- Type:
- 1 byte
- Example:
- 00

Version number of the Safe on-chain identifier format.

Project hash

- Type:
- 20 bytes
- Example:
- 6363643438383836663461336661366162653539

Truncated hash of the project's name (for example, "Gnosis", "CoW Swap").

Platform hash

- Type:
- 3 bytes
- Example:
- 646561

Truncated hash of the platform's name (for example, "Web", "Mobile", "Safe App", "Widget").

Tool hash

- Type:
- 3 bytes
- Example:
- 393238

Truncated hash of the tool's name (for example, "protocol-kit", "relay-kit", or any custom tool built by projects).

Tool version hash

- Type:
- 3 bytes
- Example:
- 653366

Truncated hash of the tool's version (for example, "1.0.0", "1.0.1").

Steps

The on-chain identifier allows tracking the deployment of Safe accounts, the execution of Safe transactions, and the execution of Safe user operations:

Generate an on-chain identifier

Feel free to skip this section if you use the Protocol Kit or Relay Kit from the Safe{Core} SDK, as this is handled internally.

To create an on-chain identifier with the format described above, you need to implement a function that receives the `project`, `platform`, `tool`, and `toolVersion` used; and returns the correct identifier after hashing, truncating, and concatenating all these parameters.

```
_20 function generateOnChainIdentifier({ _20 project, _20 platform = 'Web', _20 tool, _20 toolVersion _20 }:
OnChainIdentifierParamsType): string { _20 const identifierPrefix = '5afe' // Safe identifier prefix _20 const identifierVersion
= '00' // First version _20 const projectHash = generateHash(project, 20) // Take the last 20 bytes _20 const platformHash =
generateHash(platform, 3) // Take the last 3 bytes _20 const toolHash = generateHash(tool, 3) // Take the last 3 bytes _20
const toolVersionHash = generateHash(toolVersion, 3) // Take the last 3 bytes _20 _20 return {identifierPrefix}{identifierVersion}
```

```
{projectHash}{platformHash}{toolHash}{toolVersionHash} _20 } _20 _20 function generateHash(input: string, size: number): string { _20
const fullHash = keccak256(toHex(input)) _20 return toHex(fullHash.slice(-size)).replace('0x', '') // Take the last X bytes _20 }
This identifier will be added to all your Safe transactions and become searchable on-chain.
```

Track Safe deployments

Safe deployments can be tracked by concatenating the on-chain identifier at the end of the deployment transactiondata . This way Safe deployment transactions will include the identifier.

If you use the [Protocol Kit](#) or the [Relay Kit](#) to deploy a Safe, adding theonchainAnalytics property to the initialization method will automatically handle this.

If you use a custom implementation, remember to manually add the on-chain identifier at the end of the deployment transactiondata .

```
Protocol Kit Relay Kit _13 import Safe, { OnchainAnalyticsProps } from '@safe-global/protocol-kit' _13 _13 const
onchainAnalytics: OnchainAnalyticsProps = { _13 project: 'YOUR_PROJECT_NAME' // Required. Always use the same
value for your project. _13 platform: 'CURRENT_PLATFORM' // Optional _13 } _13 _13 const protocolKit = await Safe.init({
_13 // ... _13 onchainAnalytics _13 }) _13 _13 // Execute the deployment
```

Track Safe transactions

Safe transactions can be tracked by concatenating the on-chain identifier at the end of the transactiondata or user operationcallData properties. This way Safe transactions will include the identifier.

If you use the [Protocol Kit](#) or the [Relay Kit](#) to execute the Safe transactions, adding theonchainAnalytics property to the initialization method will automatically handle this.

If you use a custom implementation, remember to manually add the on-chain identifier at the end of the transactiondata /callData .

```
Protocol Kit Relay Kit _13 import Safe, { OnchainAnalyticsProps } from '@safe-global/protocol-kit' _13 _13 const
onchainAnalytics: OnchainAnalyticsProps = { _13 project: 'YOUR_PROJECT_NAME' _13 platform:
'CURRENT_PLATFORM' // Optional _13 } _13 _13 const protocolKit = await Safe.init({ _13 // ... _13 onchainAnalytics _13 })
_13 _13 // Execute the transaction
```

Get the on-chain identifier

If you use the Protocol Kit or the Relay Kit, call thegetOnchainIdentifier method from an initialized instance of the Protocol Kit to get the current Safe on-chain identifier.

```
_10 const onchainIdentifier = protocolKit.getOnchainIdentifier()
```

Submission Form

You can fill out the form by clicking[this link\(opens in a new tab\)](#) or using the form below:

Loadingâ€¦

[Web3Auth Onramp](#) Was this page helpful?

[Report issue](#)