

# The Move Book

To guarantee the safety and security of the network, Sui has certain limits and restrictions. These limits are in place to prevent abuse and to ensure that the network remains stable and efficient. This guide provides an overview of these limits and restrictions, and how to build your application to work within them.

The limits are defined in the protocol configuration and are enforced by the network. If any of the limits are exceeded, the transaction will either be rejected or aborted. The limits, being a part of the protocol, can only be changed through a network upgrade.

The size of a transaction is limited to 128KB. This includes the size of the transaction payload, the size of the transaction signature, and the size of the transaction metadata. If a transaction exceeds this limit, it will be rejected by the network.

The size of an object is limited to 256KB. This includes the size of the object data. If an object exceeds this limit, it will be rejected by the network. While a single object cannot bypass this limit, for more extensive storage options, one could use a combination of a base object with other attached to it using dynamic fields (eg Bag).

The size of a single pure argument is limited to 16KB. A transaction argument bigger than this limit will result in execution failure. So in order to create a vector of more than ~500 addresses (given that a single address is 32 bytes), it needs to be joined dynamically either in Transaction Block or in a Move function. Standard functions like `vector::append()` can join two vectors of ~16KB resulting in a ~32KB of data as a single value.

The maximum number of objects that can be created in a single transaction is 2048. If a transaction attempts to create more than 2048 objects, it will be rejected by the network. This also affects [dynamic fields](#), as both the key and the value are objects. So the maximum number of [dynamic fields](#) that can be created in a single transaction is 1000. The limitation applies to dynamic object fields as well.

The maximum number of dynamic fields that can be accessed in a single transaction is 1000. If a transaction attempts to access more than 1000 dynamic fields, it will be rejected by the network.

The maximum number of events that can be emitted in a single transaction is 1024. If a transaction attempts to emit more than 1024 events, it will be aborted.

## Transaction Size

The size of a transaction is limited to 128KB. This includes the size of the transaction payload, the size of the transaction signature, and the size of the transaction metadata. If a transaction exceeds this limit, it will be rejected by the network.

The size of an object is limited to 256KB. This includes the size of the object data. If an object exceeds this limit, it will be rejected by the network. While a single object cannot bypass this limit, for more extensive storage options, one could use a combination of a base object with other attached to it using dynamic fields (eg Bag).

The size of a single pure argument is limited to 16KB. A transaction argument bigger than this limit will result in execution failure. So in order to create a vector of more than ~500 addresses (given that a single address is 32 bytes), it needs to be joined dynamically either in Transaction Block or in a Move function. Standard functions like `vector::append()` can join two vectors of ~16KB resulting in a ~32KB of data as a single value.

The maximum number of objects that can be created in a single transaction is 2048. If a transaction attempts to create more than 2048 objects, it will be rejected by the network. This also affects [dynamic fields](#), as both the key and the value are objects. So the maximum number of [dynamic fields](#) that can be created in a single transaction is 1000. The limitation applies to dynamic object fields as well.

The maximum number of dynamic fields that can be accessed in a single transaction is 1000. If a transaction attempts to access more than 1000 dynamic fields, it will be rejected by the network.

The maximum number of events that can be emitted in a single transaction is 1024. If a transaction attempts to emit more than 1024 events, it will be aborted.

## Object Size

The size of an object is limited to 256KB. This includes the size of the object data. If an object exceeds this limit, it will be rejected by the network. While a single object cannot bypass this limit, for more extensive storage options, one could use a combination of a base object with other attached to it using dynamic fields (eg Bag).

The size of a single pure argument is limited to 16KB. A transaction argument bigger than this limit will result in execution failure. So in order to create a vector of more than ~500 addresses (given that a single address is 32 bytes), it needs to be joined dynamically either in Transaction Block or in a Move function. Standard functions like `vector::append()` can join two vectors of ~16KB resulting in a ~32KB of data as a single value.

The maximum number of objects that can be created in a single transaction is 2048. If a transaction attempts to create more than 2048 objects, it will be rejected by the network. This also affects [dynamic fields](#), as both the key and the value are objects. So the maximum number of [dynamic fields](#) that can be created in a single transaction is 1000. The limitation applies to dynamic object fields as well.

The maximum number of dynamic fields that can be accessed in a single transaction is 1000. If a transaction attempts to access more than 1000 dynamic fields, it will be rejected by the network.

The maximum number of events that can be emitted in a single transaction is 1024. If a transaction attempts to emit more than 1024 events, it will be aborted.

## Single Pure Argument Size

The size of a single pure argument is limited to 16KB. A transaction argument bigger than this limit will result in execution failure. So in order to create a vector of more than ~500 addresses (given that a single address is 32 bytes), it needs to be joined dynamically either in Transaction Block or in a Move function. Standard functions like `vector::append()` can join two vectors of ~16KB resulting in a ~32KB of data as a single value.

The maximum number of objects that can be created in a single transaction is 2048. If a transaction attempts to create more than 2048 objects, it will be rejected by the network. This also affects [dynamic fields](#), as both the key and the value are objects. So the maximum number of [dynamic fields](#) that can be created in a single transaction is 1000. The limitation applies to dynamic object fields as well.

The maximum number of dynamic fields that can be accessed in a single transaction is 1000. If a transaction attempts to access more than 1000 dynamic fields, it will be rejected by the network.

The maximum number of events that can be emitted in a single transaction is 1024. If a transaction attempts to emit more than 1024 events, it will be aborted.

## Maximum Number of Objects (and dynamic fields) created

The maximum number of objects that can be created in a single transaction is 2048. If a transaction attempts to create more than 2048 objects, it will be rejected by the network. This also affects [dynamic fields](#), as both the key and the value are objects. So the maximum number of [dynamic fields](#) that can be created in a single transaction is 1000. The limitation applies to dynamic object fields as well.

The maximum number of dynamic fields that can be accessed in a single transaction is 1000. If a transaction attempts to access more than 1000 dynamic fields, it will be rejected by the network.

The maximum number of events that can be emitted in a single transaction is 1024. If a transaction attempts to emit more than 1024 events, it will be aborted.

## Maximum Number of Dynamic Fields accessed

The maximum number of dynamic fields that can be accessed in a single transaction is 1000. If a transaction attempts to access more than 1000 dynamic fields, it will be rejected by the network.

The maximum number of events that can be emitted in a single transaction is 1024. If a transaction attempts to emit more than 1024 events, it will be aborted.

## Maximum Number of Events

The maximum number of events that can be emitted in a single transaction is 1024. If a transaction attempts to emit more than 1024 events, it will be aborted.