



## Gateway Bulletin GB-2015-01

### Gateway Advisory: Reliable Transaction Submission

January 5, 2014

## Overview

The purpose of this bulletin is to introduce best practices for reliably submitting transactions to the Ripple network. Gateways and back end applications should implement these best practices to ensure that transactions are validated or rejected in a verifiable and timely fashion. The best practices are intended for gateways and applications interfacing with trusted rippled servers (e.g. a locally operated instances).

The term *reliable transaction submission* describes applications that submit transactions to the Ripple network while achieving:

1. [Idempotency](#) - Transactions will be processed once and only once, or not at all.
2. Verifiability - Applications can determine the final result of a transaction.

Applications which fail to implement best practices are at risk of the following errors:

1. Submitting transactions which are inadvertently never executed.
2. Mistaking provisional transaction results for their final, immutable results.
3. Failing to find authoritative results of transactions that have been applied to the ledger.

This bulletin details the circumstances under which these errors can arise. For a complete description of best practices and instructions for implementing this feature, gateways are encouraged to refer to the developer documentation on [Reliable Transaction Submission](#).

## Background

The Ripple protocol provides a ledger shared across all nodes in the network. Through a process of consensus and validation, the network agrees on the order in which transactions are applied to (or omitted from) the ledger.

The timeline of a single transaction is as follows.

1. A transaction is created and signed by account owner.
2. That transaction is submitted to the network as a candidate transaction.
  - a. Malformed or nonsensical transactions may be rejected immediately.
  - b. Well formed transactions may provisionally succeed, then later fail.
  - c. Well formed transactions may provisionally fail, then later succeed.
3. Through consensus and validation, the transaction is applied to the ledger. Or never applied, in the case of errors.
4. The validated ledger includes the transaction, and its effects are reflected in the ledger state.
  - a. Transaction results are no longer provisional, success or failure is now final and immutable.

*Note that when submitting a transaction via `rippled` or `ripple-lib`, a successful status code returned from a `submit` command indicates the `rippled` server has received the candidate transaction, and does **not** indicate the transaction will be finally applied to the ledger.*

Ripple APIs may return *provisional* results based on candidate transactions. Applications must not confuse these with the final, *immutable*, results of a transaction. Immutable results are found only in validated ledgers. Applications may need to query the status of a transaction repeatedly, until the ledger containing the transaction results is validated. The steps necessarily to verify final results are detailed in the [reliable transaction submission best practices](#).

Transactions submitted to trusted Ripple network nodes are usually validated or rejected in a matter of seconds. There are cases, however, in which a transaction is neither validated nor rejected this quickly. One specific example can occur if the global base [transaction fee](#) increases after an application sends a transaction. If the fee increases above what has been specified in the transaction, the transaction will not be included in the next validated ledger. If at some later date the global base fee decreases, the transaction may become viable again, if an expiration is not provided.

To ensure that transactions are included or rejected from the ledger in a predictable time frame, applications must use the `LastLedgerSequence` parameter, as described in the [reliable transaction submission best practices](#).

## Best Practices

Applications submitting transactions must take steps to ensure they are submitted reliably, even in the event that a process dies or other failure. Application transaction results must be verified so that applications can act on the final, validated results.

Submission and verification are two separate procedures.

1. **Submission** - The transaction is submitted to the network and a *provisional* result is returned.
2. **Verification** - The authoritative result is determined by examining validated ledgers.

Refer to [Reliable Transaction Submission Best Practices](#) for complete implementation details of transaction submission and verification.

## Additional Resources

[Reliable Transaction Submission Best Practices](#)

Overview of [the Ripple Ledger Consensus Process](#)

[Reaching Consensus in Ripple](#)

Developer Blog Post on [Transaction Format](#)

Developer Blog Post on [Transaction Fees](#)