

# ParseTree fields

Subrules become the fields of ParseTree classes.

You can specify the field name explicitly – if you don't, Nitra will try to automatically form the name from the subrule content.

A subrule name can be explicitly specified in one of the following ways.

- Use the = operator. The rule should be located to the right and the field name – to the left of the operator.
- Specify an alias instead of the subrule using the alias directive which can either reside above the rule it is used in, or in the body of the rule. In the latter case the alias can only effect this very rule, as it is not recognized outside of it.

If the field name is not explicitly specified, Nitra will try to calculate it automatically. Below is the description of the name calculation algorithm:

- If the subrule is a call to another named rule, then its name is used as the field name.
- If the subrule is the ? operator, the subrule name with the added `Opt` suffix is used. For example, a field named `IdentifierOpt` will be created for the `Identifier?` rule.
- If the subrule is a loop, the pluralized subrule name is used.
- If the subrule is a loop with separators, then the rule that describes the separators is ignored, and the name is calculated according to the rule for plain loops.
- If the subrule is a sequence (enclosed by parentheses) that contains only one rule which type is not `void` and which is not a literal token, then this rule's name is used. If there are more than one such rule, then the name should be manually specified, as calculation becomes impossible. In this case the compiler will return an error.
- Names are not calculated for predicates, as the ParseTree nodes are not created for them.

## See also

[ParseTree object](#)