

The Program Driver interface

`ProgramDriver` is a small but important interface which allows the template type-checking machinery to request changes to the current `ts.Program`, and to receive a new `ts.Program` with those changes applied. This is used to add template type-checking code to the current `ts.Program`, eventually allowing for diagnostics to be produced within that code. This operation is abstracted behind this interface because different clients create `ts.Programs` differently. The Language Service, for example, creates `ts.Programs` from the current editor state on request, while the TS compiler API creates them explicitly.

When running using the TS APIs, it's important that each new `ts.Program` be created incrementally from the previous `ts.Program`. Under a normal compilation, this means that programs alternate between template type checking programs and user programs:

- `ts.Program#1` is created from user input (the user's source files).
- `ts.Program#2` is created incrementally on top of #1 for template type-checking, and adds private TCB code.
- `ts.Program#3` is created incrementally on top of #2 when the user makes changes to files on disk (incremental build).
- `ts.Program#4` is created incrementally on top of #3 to adjust template type-checking code according to the user's changes.

The `TsCreateProgramDriver` performs this operation for template type-checking `ts.Programs` built by the command-line compiler or by the CLI. The latest template type-checking program is then exposed via the `NgCompiler's` `getCurrentProgram()` operation, and new user programs are expected to be created incrementally on top of the previous template type-checking program.

Programs and the compiler as a service

Not all clients of the compiler follow the incremental tick-tock scenario above. When the compiler is used as a service, new `ts.Programs` may be generated in response to various queries, either directly to `NgCompiler` or via the `TemplateTypeChecker`. Internally, the compiler will use the current `ProgramDriver` to create these additional `ts.Programs` as needed.

Incremental builds (new user code changes) may also require changing the `ts.Program`, using the compiler's incremental ticket process. If the `TsCreateProgramDriver` is used, the client is responsible for ensuring that any new incremental `ts.Programs` are created on top of the current program from the previous compilation, which can be obtained via `NgCompiler's` `getCurrentProgram()`.