

Shared Files Feature

We reimplemented the way the shared files are generated by the VBUc. By using compilation directives, it is possible to maintain the logic of each project. This allows different lines of code to coexist within the same file, enabling each project to compile only the logic relevant to it, without impacting or interfering with other projects.

```
internal static object PrimitiveSharedTest01(object p1)
{
//UPGRADE_NOTE: (1707) The resulting shared file would differ depending on whether all or so
#if Project01
    return Module01.PrimitiveTest01(ReflectionHelper.GetPrimitiveValue<string>(p1));
#elif Project02
    return Module02.PrimitiveTest01(ReflectionHelper.GetPrimitiveValue<bool>(p1));
#elif Project03
    return Module03.PrimitiveTest01(ReflectionHelper.GetPrimitiveValue<int>(p1));
#elif Project04
    return Module04.PrimitiveTest01(ReflectionHelper.GetPrimitiveValue<double>(p1));
#endif
}
```

This new design significantly reduces the amount of code generated, as shared files can now accommodate multiple project-specific variations within a single source. It promotes a more modular and centralized structure, simplifying updates and improving maintainability across the board. Moreover, it enhances consistency and reduces the risk of errors or functional discrepancies between projects, contributing to a more reliable and scalable migration process.

Bug Fixing and Improvements

Some bugs were fixed, some of them are, but not limited to:

- Discards improvements on performance
- Interpolated string improvements
- True DBGrid improvements
- EditLib improvements