

## Design Moves and Modular Operators

Design Move	Coding Using Modular Operators
1A	INVERT (scattered configuration information) and AUGMENT (configuration scripts) to create configuration engine module
1B	SUBSTITUTE (configuration engine) and AUGMENT (configuration database) to add new logging functionality to the configuration engine
1C	INVERT (configuration database) to SUBSTITUTE (configuration settings) during new implementation
2A	INVERT (scattered function calls) and AUGMENT (tool set) to add arguments to function calls
2B	Allow customers to AUGMENT (customer-written code) to designated exit points to create new functionality
2C	SUBSTITUTE (tool set) to help customers create their own exit points for custom function calls
3A	EXCLUDE (customer-written code and function calls) and PORT (core functional modules) to a component-based architecture
3B	AUGMENT (application composer module) to component architecture
3C	Allow partners to AUGMENT (third-party components) to component repository and SUBSTITUTE (original components) during implementation